IDENTIFICACIÓN DE ESPECTROS RAMAN MEDIANTE UN SISTEMA DE LÓGICA DIFUSA

M.J.Soneira **M.Castanys** **R.Perez-Pueyo** **S.Ruiz-Moreno**
Dpto.Teoría de la Señal y Comunicaciones, ETSETB, Universitat Politécnica de Catalunya,
C/Sor Eulalia de Anzizu, s/n, D5, Campus Nord, 08034 Barcelona.
soneira@tsc.upc.es micatu@hotmail.com rperez@tsc.upc.es sruiz@tsc.upc.es

Resumen

En esta comunicación se presenta un sistema de lógica difusa para la identificación de espectros Raman de pigmentos artísticos. Habitualmente, la identificación de un espectro Raman es realizada por el analista mediante la inspección ocular del espectro comparándolo con un espectro patrón adecuado según su conocimiento y experiencia. Dependiendo de esta comparación el material analizado es identificado. Obviamente, este proceso es lento, impreciso y no automático. El sistema propuesto basado en lógica difusa evita los posibles errores subjetivos que este proceso puede conllevar realizando la identificación de forma automática. Dicho sistema se basa en un conjunto de reglas que proporcionan la estrategia de procesado para reconocer un espectro Raman de entre los de una base de datos. La comparación entre espectros se basa en el valor medio y la varianza del vector diferencia cuadrática entre ellos. En la comunicación, se explica el razonamiento seguido para construir el sistema de lógica difusa y se evalúan las prestaciones del sistema mediante algunos ejemplos prácticos.

Palabras Clave: Espectros, Raman, pigmentos, Identificación.

1 INTRODUCCIÓN

La Espectroscopia Raman reaparece como técnica fotónica de análisis a finales de los años 60, después de un paréntesis forzado desde su descubrimiento debido a problemas tecnológicos, y desde entonces se ha ido consolidando cada vez más como una herramienta de análisis molecular muy potente, con un gran abanico de posibilidades y aplicaciones [1],[2],[3]. Dicha técnica que se basa en el efecto Raman, descubierto en 1928 por K.V.Raman y por el cual le fue otorgado el premio Nóbel, proporciona información molecular de cualquier compuesto químico, sea cual sea su estado físico, permitiendo su identificación [4],[5]. Cuando un haz de luz monocromático incide sobre un material éste puede ser transmitido, absorbido o dispersado. El análisis mediante espectroscopía Raman se basa en el examen de la luz dipersada por un material. La mayor parte de la luz dispersada tiene la misma frecuencia que la luz incidente y el efecto se conoce por dispersión Rayleigh o elástica. Por el contrario, si la dispersión es inelástica, la luz dispersada tiene una frecuencia ligeramente inferior o superior a la de la luz incidente dando lugar al denominado efecto Raman o dispersión Raman. En el caso de efecto Raman, las diferencias de energía entre el fotón incidente y dispersado están cuantizadas y corresponden a diferencias de niveles energéticos rotacionales y vibracionales de la molécula. Por tanto, estos cambios de frecuencia son característicos del material analizado e independientes de la frecuencia de la luz incidente. Un espectro Raman consiste en la representación gráfica de la intensidad de luz dispersada en función del número de onda normalizado o desplazamiento Raman expresado en cm^{-1} relativo a la frecuencia de la luz incidente. Cada pico o banda espectral representada denominada banda Raman corresponde a un desplazamiento Raman desde la energía de la luz incidente. El conjunto de bandas Raman es característico y particular de cada material y permite la identificación unívoca del mismo. Es por ello que el espectro Raman de un material se interpreta como su "huella dactilar". En la figura 1 se representa el espectro Raman del pigmento azul de ultramar.

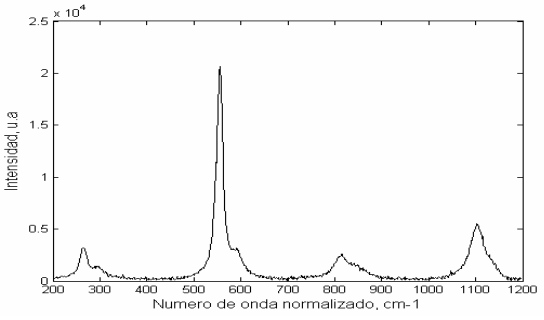


Figura 1: Espectro Raman del pigmento Azul ultramar.

Las aplicaciones de la Espectroscopia Raman son muchas, pero destaca su presencia en el mundo del arte [6],[7],[8] y en concreto en el campo de la investigación de materiales artísticos y en las tareas de restauración y conservación de obras de arte. Es en este contexto donde este estudio toma sentido, ya que se centra en el análisis de materiales artísticos y en particular en la identificación de pigmentos pictóricos. La identificación de un pigmento artístico se consigue mediante la localización de las bandas Raman de su espectro y permite datar, restaurar, conservar y/o autentificar la obra de arte en cuestión. Un equipo de espectroscopía Raman, con el que se miden los espectros, consta principalmente de tres bloques que son, una fuente de luz monocromática (un láser), un monocromador y un sistema receptor. En concreto, el equipo utilizado para obtener los espectros Raman en esta comunicación cuenta con tres fuentes de excitación intercambiables: un láser rojo (632 nm), uno verde (514 nm) y otro infrarrojo (1064 nm) e incorpora tecnología de fibra óptica. Su diagrama de bloques se presenta en la figura 2. A grandes rasgos su funcionamiento es el siguiente: la luz del láser es guiada a través de la fibra óptica de excitación hasta la entrada del cabezal óptico. El objetivo del cabezal es focalizar esta luz en la zona que se desea analizar y recoger la luz dispersada por la fibra óptica de colección. Dicha fibra guía la luz dispersada hasta el monocromador donde es separada espacial y espectralmente. El detector CCD realiza la conversión de la señal óptica en eléctrica y remite la información hasta el ordenador para su visualización y almacenamiento. Además el PC se encarga del control del equipo.

Por otra parte cabe destacar que la aplicación de la Espectroscopia Raman en el mundo del arte, como herramienta para el análisis de materiales artísticos, resulta aún más oportuna ya que al tratarse de un método no destructivo, que no requiere de ninguna preparación de la muestra a analizar, asegura el no deterioro de la obra de arte. Por todas estas ventajas, la espectroscopia Raman es una de las técnicas de análisis más potente para la identificación de pigmentos [9],[10]. El método habitual de identificar un material es la comparación visual por parte del espectroscopísta de un espectro Raman desconocido con otro espectro Raman conocido. Este método visual, aun siendo muy sencillo, presenta claros inconvenientes como son su subjetividad ligada a la experiencia del personal del laboratorio, su no automatización y su lentitud. En esta comunicación se ha buscado una opción automática, objetiva, global y rápida que permita identificar espectros provenientes de muestras desconocidas, mediante su comparación automática con otros espectros Raman conocidos usando un operador matemático como la distancia cuadrática. Este operador matemático permite cuantificar objetivamente la similitud de dos espectros, aunque la identificación no podrá hacerse directamente a partir de sus resultados debido a que diferentes aspectos como son el ruido o los errores tanto en la medida como en el equipo Raman pueden contaminar el espectro medido. La Lógica Difusa [11],[12] es una técnica útil para tratar la incertidumbre de los resultados obtenidos tras la comparación matemática. Hoy en día, se emplea en diversos campos del procesado de señal [13],[14], y en este caso se propone un sistema de lógica difusa para la identificación de espectros Raman.

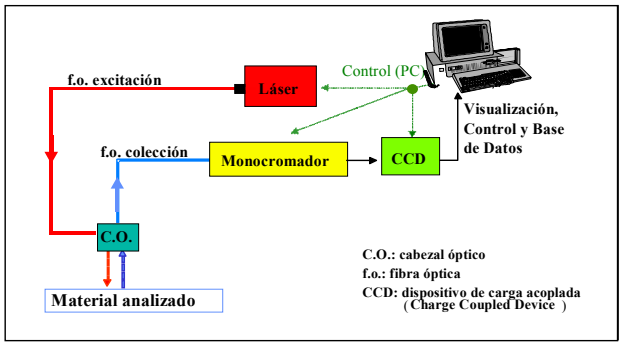


Figura 2: Equipo Raman

2 METODOLOGIA

El método de identificación propuesto se basa en la comparación de dos espectros (uno incógnita y otro conocido). Para ello se dispone de una base de datos formada por espectros Raman de pigmentos puros conocidos y medidos en las condiciones más óptimas. Una vez medido el espectro del pigmento a identificar, se compara con cada uno de los patrones. Un espectro Raman se puede considerar como un vector de N puntos, e(i), donde cada coordenada representa la intensidad Raman para cada número de onda normalizado (i) del espectro. La función de similitud empleada para comparar los espectros es la diferencia cuadrática entre los espectros desconocido e(i) y patrón p(i) definida como $d(i)=(e(i)-p(i))^2$, que es un vector de idéntica dimensión que los espectros comparados. Como la medida de la distancia depende de la intensidad Raman de los espectros, estos se normalizan antes de compararlos de la forma siguiente:

$$en(i) = \frac{e(i) - mín\,(e(i)}{máx\,(e(i)) - mín\,(e(i))} \qquad (1)$$

siendo máx(e(i)) y mín(e(i)) el valor máximo y mínimo respectivamente de la intensidad Raman en el espectro e(i) y en(i) el espectro e(i) normalizado. Con esta normalización el margen de intensidad Raman en el espectro varía entre 0 y 1. Y, el resultado de la distancia cuadrática es independiente de los valores de las intensidades de las bandas Raman que dependen de las condiciones de medida como el tiempo de exposición de la muestra al láser o la potencia, el spot y la frecuencia del láser. Cabe destacar que el vector distancia no informa

sobre el parecido global de los espectros, sino de sus coincidencias o diferencias punto a punto. Para disponer de un índice de similitud global, se calcula el valor medio $đ_{epi}$ y la varianza σ^2_{epi} de dicho vector, siendo estos dos valores numéricos los índices de similitud empleados. Por su propia definición, y teniendo en cuenta que los espectros tienen todos una excursión entre 0 y 1, cuanto más parecidos sean los espectros comparados, más cercanos a cero serán sus valores ($đ_{epi} \rightarrow 0$ y $\sigma^2_{epi} \rightarrow 0$), mientras que cuanto más diferentes sean los espectros, mayores serán los índices calculados. Ahora bien, en la práctica, debido principalmente al ruido, los índices de similitud encontrados pueden presentar ambigüedad. En esta comunicación se propone un sistema de identificación basado en lógica difusa que determine si los valores $đ_{epi}$ y σ^2_{epi} son o no suficientemente próximos a cero para poder decidir si los espectros corresponden o no al mismo pigmento. A continuación se describe dicho sistema.

3 SISTEMA DE IDENTIFICACIÓN

Las variables de entrada al sistema son los índices de similitud $đ_{epi}$ y σ^2_{epi}, cuyo rango de valores es el intervalo [0,1] debido a la normalización de los espectros. Se definen dos conjuntos difusos de entrada para cada variable: "Valor medio grande" y "Valor medio pequeño", "Varianza grande" y "Varianza pequeña". Las funciones características que los definen han sido diseñadas a partir de la función coseno teniendo en cuenta que la zona de mayor ambigüedad del Universo de discurso de las variables de entrada es donde sus valores son bajos. En la figura 3 se representan dichas funciones. En la misma, γ determina cuándo el índice de similitud (valor medio o varianza) es suficientemente pequeño como para considerar que los espectros comparados son parecidos, o cuándo es lo suficientemente grande como para interpretar que los espectros comparados no se parecen. El valor de este umbral se determina en función de la base de datos de pigmentos a identificar y se han definido para cada variable de entrada como el máximo valor medio y la máxima varianza del vector distancia entre todos los espectros patrones considerando que si el valor del índice de similitud encontrado al comparar el espectro desconocido es mayor que el umbral, entonces no se parece a ninguno de los espectros patrones. La variable de salida del sistema es el grado de similitud entre el espectro analizado y el patrón comparado y puede tomar valores en el intervalo [0,10]. Se han definido dos conjuntos difusos para esta variable denominados Identificado y No Identificado representados por dos funciones características lineales en el intervalo [0, 10] tal y como se representa en la figura 4. Un conjunto de reglas define la forma de obtener el valor de salida del sistema a partir de las variables de entrada. En el problema de identificación, el "parecido" entre dos espectros está cuantificado por los valores numéricos de los índices de similitud definidos, valor medio y varianza del vector distancia cuadrática. De manera que las reglas del sistema tienen la siguiente estructura: "Si el valor medio es pequeño y la varianza es pequeña, entonces el espectro es identificado". A continuación se representan las cuatro reglas que modelan el problema:

Si $đ_{epi}$ P y σ^2_{epi} P, entonces el espectro e es I con p_i
Si $đ_{epi}$ G y σ^2_{epi} P, entonces el espectro e es NOI con p_i
Si $đ_{epi}$ P y σ^2_{epi} G, entonces el espectro e es NOI con p_i
Si $đ_{epi}$ G y σ^2_{epi} G, entonces el espectro e es NOI con p_i

Donde P y G significan pequeño y grande respectivamente e I y NOI son identificado y no identificado respectivamente.

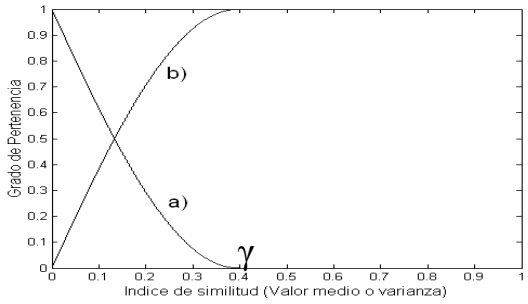


Figura 3: Funciones características asociadas a los conjuntos difusos de entrada para cada una de las variables valor medio y varianza: a) Pequeño, b) Grande

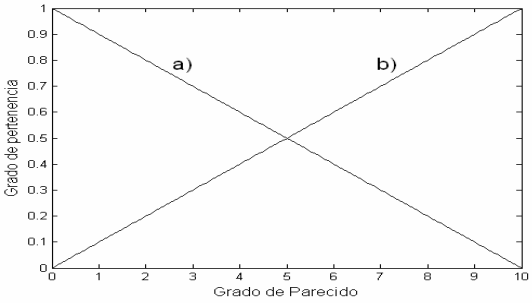


Figura 4: Funciones características asociadas a los conjuntos difusos de salida para la variable grado de similitud: a) No Identificado, b) Identificado

La interpretación de cada una de las reglas proporciona un conjunto difuso. Tras la agregación, implementada por el operador matemático suma, de los cuatro conjuntos resultantes, se obtiene un conjunto difuso de salida a partir del cual, y aplicando el método del centroide, se encuentra un valor para la variable de salida, grado de similitud (g_{epi}, que cuantifica el parecido entre el espectro desconocido e y el patrón p_i). Finalmente, de todos los grados de similitud calculados (tantos como espectros patrones se consideren), se escoge el máximo. Si este

valor supera un umbral previamente definido (z_{umbral}), el sistema proporciona la identificación del pigmento analizado con el patrón correspondiente (p_i). Para la elección de dicho umbral se ha tenido en cuenta que la forma triangular de los conjuntos difusos de salida proporciona valores para el centroide en el intervalo [10/3, 20/3], y el valor escogido ha sido z_{umbral}=5. Finalmente, con el objetivo de valorar la seguridad en el resultado final del sistema se ha definido el grado de seguridad o certeza (gs) que indica con qué seguridad el sistema realiza la identificación de la muestra desconocida, siendo ésta una manera de valorar su fiabilidad. El grado de seguridad se presenta en forma de porcentaje, y se calcula a partir de una relación lineal diseñada para que su valor sea del 50% cuando el grado de similitud normalizado (g_{nepi}, normalizado de la misma forma que los espectros pero en el rango [0,10]) sea igual a 6.0, y que la seguridad en el resultado sea del 100% cuando el valor de salida del sistema sea igual a 10, es decir,

$$g_S\left(g_{nepi}\right) = \left(\frac{1}{8}\cdot g_{nepi} - \frac{1}{4}\right)\cdot 100 \qquad (2)$$

4 RESULTADOS Y DISCUSIÓN

Para caracterizar una obra de arte interesa, por un lado, datarla, identificando, por ejemplo, los pigmentos que la componen, ya sean modernos y/o antiguos, y por otro lado, recaudar el máximo de información sobre la técnica empleada por su autor para descubrir o no la mano de un artista, así, llegar a describir la paleta utilizada en una obra de arte puede facilitar identificar su autor. Por todo ello, y a modo de ejemplo de aplicación, la base de datos con la que se trabaja en este estudio se ha formado con ocho pigmentos correspondientes a los colores mas habituales de la paleta de un pintor que son el blanco, el amarillo, el azul y el rojo, y para cada uno de ellos se han escogido dos tipos de pigmentos, uno existente desde la antigüedad y otro más utilizado a partir de que el anterior quedara en desuso. Como pigmentos blancos se dispone del Blanco de Plomo y de la Anatasa, para los amarillos se encuentran el Amarillo de Nápoles y el Amarillo de Cromo, el Bermellón y el Minio de Plomo son los representantes de los rojos y finalmente para el tono azul se dispone de la Azurita y del Azul Ultramar. El sistema ha sido probado con espectros medidos en el Laboratorio Raman de la Universitat Politècnica de Catalunya, provenientes de obras de arte. A continuación se muestran algunos de los resultados obtenidos tras el análisis de algunas muestras.

El primer ejemplo presentado es el análisis de un pigmento desconocido cuyo espectro Raman se representa en la figura 5. El máximo grado de parecido calculado por el sistema se obtiene con el patrón del Bermellón (ver tabla 1) y por tanto el espectro desconocido es identificado con dicho pigmento. El grado de parecido entre ambos espectros es de gn_{epi}=9.29, con una seguridad en el resultado del 91.12%, un valor muy alto debido a la coincidencia de sus bandas Raman, tal y como se puede apreciar en la figura 5. El segundo caso presentado es el del análisis de un espectro que el sistema ha identificado como el del pigmento Azurita (figura 6). Esta vez el grado de similitud normalizado entre el pigmento desconocido y el patrón que le ha sido atribuido es de 7.31, valor más bajo que en el caso anterior. Esta diferencia se debe principalmente al ruido que contamina el espectro incógnita medido. Otro aspecto que puede hacer bajar el parecido entre espectros de un mismo pigmento es el desplazamiento de sus bandas, provocado por ejemplo por un problema de calibración de los dispositivos del equipo Raman. Como ejemplo de esta situación se presenta el caso mostrado en la figura 7. El espectro incógnita ha sido identificado correctamente con el espectro del patrón del Blanco de Plomo, pero con un grado de similitud igual a 6.66 y un grado de seguridad en el resultado de sólo el 58.35%. Finalmente, se analiza el espectro del pigmento Amarillo de Zinc, pigmento que no forma parte de la base de datos, por lo que ninguno de los grados de similitud obtenidos con los diferentes patrones supera el valor umbral determinado y el sistema no puede reconocer dicho pigmento incógnita y da como salida la no identificación del pigmento analizado.

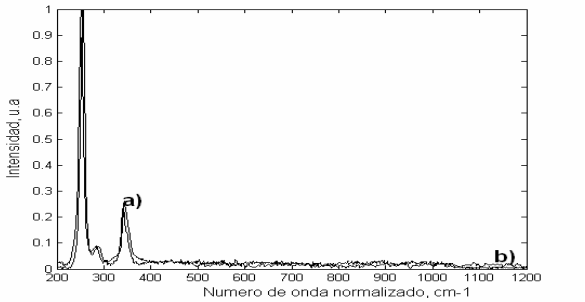


Figura 5: a) Espectro Raman desconocido, b) Espectro Raman del patrón correspondiente al pigmento Bermellón.

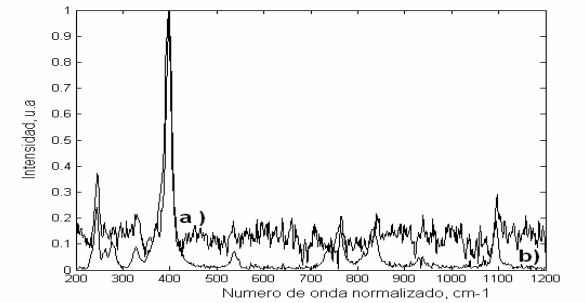


Figura 6: a) Espectro Raman del pigmento analizado, b) espectro Raman del patrón del pigmento Azurita

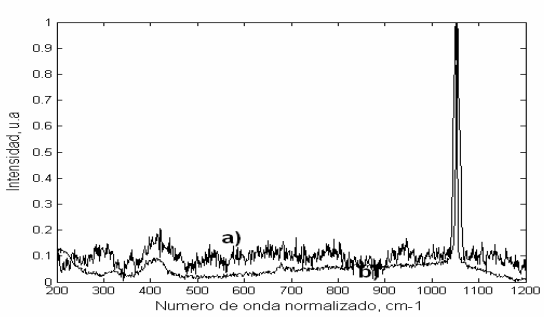


Figura 7: a) Espectro Raman de un pigmento desconocido, b) Espectro Raman patrón correspondiente al pigmento Blanco de plomo

A modo de resumen en la tabla 1 se muestran los resultados obtenidos del grado de similitud normalizado de cada pigmento analizado con los espectros de los pigmentos patrones de la base de datos utilizada detallada en la tabla 2. Donde se han subrayado los valores máximos en cada caso.

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
e_1	1.00	1.92	1.93	2.48	1.66	**9.29**	2.03	0.86
e_2	2.23	1.97	**7.31**	2.43	1.92	2.18	2.07	1.18
e_3	1.97	2.29	2.46	**6.66**	2.20	2.46	2.51	0.92
e_4	0.90	1.70	1.53	2.39	1.91	1.86	1.85	0.11

Tabla 1: grados de similitud normalizados

p_1 : Anatasa	p_5 : Amarillo de Cromo
p_2: Azul Ultramar	p_6 **:** Azul Ultramar
p_3 : Azurita	p_7 : Minio de Plomo
p_4 : Blanco de Plomo	p_8 : Amarillo de Nápoles

Tabla 2: Relación de los espectros patrones p_i y los pigmentos que representan

4 CONCLUSIÓN

El objetivo que impulsó este estudio fue diseñar un sistema capaz de identificar automáticamente pigmentos artísticos a través de sus espectros Raman para poder analizar la pigmentación de obras de arte de manera objetiva, sistemática y sin la intervención del analista. A la vista de los resultados obtenidos, se puede decir que el sistema diseñado permite realizar la identificación automática de pigmentos de una manera rápida y fiable. También, a partir de dichos resultados, se ha puesto en evidencia que el ruido o problemas experimentales, como un error de calibración, pueden dar lugar a la no identificación de los espectros, así, aunque el sistema es robusto, se requiere que las condiciones experimentales de medida de los espectros sean las idóneas. Cabe añadir que el sistema se ha diseñado en MATLAB y tiene un tiempo de procesado muy corto. Un aspecto interesante sería plantear una ampliación de dicho sistema para que fuera capaz de identificar pigmentos mezcla.

Agradecimientos

Este trabajo se enmarca dentro del proyecto CICYT TIC 2002-03040 subvencionado por el gobierno español.

Referencias

[1] G.Turrell, J.Corset. Raman microscopy: Develop. and Applications, Academic Press Inc. 1996.

[2] A.Brookes. Prospects for the future of Raman spectroscopy and barriers to its implementation. Spectroscopy Europe, Vol.11, Number 2, 1999.

[3] E. Ciliberto, G. Spoto. Modern Analytical Methods in Art and Archeology, Chem. Analysis Vol.155, 2000.

[4] Banwell. Fundamentos de Espectroscopia Molecular, Ediciones del Castillo 1977.

[5] R. L. McCreery. Raman Spectroscopy for Chemical Analys*is*, Chemical Analysis Vol.157, 2000.

[6] L. Burgio, R. J. H. Clark, T. Stratoudaki, M. Doulgeridis, and D. Anglos. Pigment Identification in Painted Artworks: A Dual Analytical Approach Employing Laser-Induced Breakdown Spectroscopy and Raman Microscopy. Applied Spectroscopy, vol.54, No.4, 2000.

[7] H.G.M. Edwards, F.Rull, P. Vandenabeele, E.M Newton, L. Monees, J. Medina, and C. Garcia. Mediaeval Pigments in the Monastery of San Baudelio, Spain: A Raman Spectroscopic Analysis. Applied Spectroscopy, Vol.55, No.1, 2001.

[8] R. L. Frost, W. Martens, J. T. Kloprogge, and P. A. Williams. Raman Spectroscopy of the basic copper chloride minerals atacamite and paratacamite: implications for the study of copper, brass and bronze objects of archaeological significance. Journal of Raman Spectroscopy, Vol. 33, 2002.

[9] M. Castillejo, M. Martín et al. Spectroscopy Analysis of Pigments and Binding Media of Polychromes by the Combination of Optical Laser-Based and Vibrational Techniques. Applied Spectroscopy, Vol. 55, No.8, 2001.

[10] P. Vandenabeele, A. Hardy, H. G. M. Edwards, and L. Moens. "Evaluations of a Principal Components-Based Searching Algorithm for Raman Spectroscopic Identification of Organic Pigments in 20th Century Artwork". Applied Spectrsocopy, Vol.55, No.5, 2001.

[11] J.M.Mendel. Fuzzy Logic Systems for Engineering: a Tutorial. IEEE, 1995.

[12] J. Yen, R. Langari. Fuzzy Logic, Intelligence, Control, and Information, 1999.

[13] P. Gader, J. M. Keller and J. Cai. A Fuzzy Logic System for the Detection and Recognition of Handwritten Street Numbers. IEEE Transactions on Fuzzy systems, Vol.3, No.1, 1995.

[14] F. Russo. Fuzzy Systems in Instrumentation: Fuzzy Signal Processing. IEEE, Trans. on instrumentation and measurement, vol.45, No.2, 1996.

APLICACIÓN DE LÓGICA DIFUSA EN SISTEMAS DE CAPTURA DEL MOVIMIENTO

Carlos Gonzalez-Morcillo, Luis Jiménez
E. Superior de Informática
Universidad de Castilla-La Mancha
Paseo de la Universidad 4, Ciudad Real
{Carlos.Gonzalez, Luis.Jimenez}@uclm.es

Juan Moreno-Garcia
E.U.I. Técnica Industrial
Universidad de Castilla-La Mancha
Avda. Carlos III s/n, Toledo
Juan.Moreno@uclm.es

Resumen

Las capacidades descriptivas de los sistemas de reglas basados en Lógica Difusa ofrecen gran facilidad a la hora de modelar el conocimiento de un experto. Dentro del ámbito de los sistemas de captura del movimiento, ampliamente utilizados en la generación de animaciones con personajes digitales, se suelen emplear modelos basados en curvas de interpolación para la corrección de los diferentes tipos de errores con los que cuentan. Por otro lado, la identificación de este tipo de errores suele realizarse manualmente, con el esfuerzo que esto supone. En este trabajo se presenta un método de identificación de errores en sistemas de captura del movimiento, utilizando un modelo cualitativo, con sencilla validación por parte de un experto. Para finalizar, se muestra un ejemplo desarrollado con el método propuesto junto con los resultados obtenidos.

Palabras Clave: Captura Movimiento, Corrección Errores, Sistemas Cualitativos, Razonamiento Aproximado, Variables Difusas.

1 INTRODUCCIÓN

Habitualmente suele emplearse la lógica difusa en aplicaciones de control [7], [8] y sistemas de tipo Mamdani [3]. Sin embargo, con menor frecuencia se emplea por sus propiedades y facilidades descriptivas [11]. En este trabajo se aprovechan estas características de los sistemas basados en lógica difusa [9] para el modelado de un conjunto de sistemas de corrección de errores en captura del movimiento.

En cualquier producción de animaciones 3D con personajes digitales se pueden distinguir dos fases principalmente; por un lado, la construcción y definición de los elementos que intervienen en la escena (personajes, escenario, etc...) y sus propiedades (iluminación, texturas, cámara, etc...), junto con la animación de los mismos.

En animación, se han desarrollado dos alternativas de trabajo de *alto nivel* frente a las técnicas clásicas de animación basadas en fotogramas clave. Mediante **sistemas de huesos**, el animador puede establecer la posición y rotación de cada elemento de la cadena, afectando a los huesos cuyo nivel de jerarquía sea inferior. Aunque se cuente con cálculo automático de cinemática inversa, este tipo de sistemas requieren un esfuerzo importante por parte del animador, que debe especificar, al menos, la posición de los extremos de las cadenas de huesos en cada keyframe. Por esta razón, se utilizan sistemas de **captura del movimiento**, que captan el movimiento de un actor por medio de unos sensores en el mundo real, aplicándolo posteriormente a los personajes virtuales. Los sistemas de captura generan información acerca de la posición tridimensional de las marcas en cada frame.

1.1 SISTEMAS DE CAPTURA DEL MOVIMIENTO

Existen varios tipos de sistemas de captura del movimiento. Entre los más utilizados están los **sistemas de captura magnéticos**, basados en campos magnéticos y antenas como sensores y los **sistemas de captura ópticos** que utilizan varias cámaras para realizar el seguimiento de unas marcas situadas en el cuerpo del actor.

Los sistemas de captura magnéticos presentan distorsiones en presencia de campos magnéticos y trabajan con un número limitado de marcas. Sin embargo, presentan menos problemas a la hora de identificar las marcas y pueden ser utilizados casi sin procesamiento

posterior, por lo que son utilizables en aplicaciones interactivas. Los sistemas de captura ópticos (como el mostrado en la figura 1), no presenta limitación en cuanto al número de marcas y suelen emplearse en la captura de movimientos complejos.

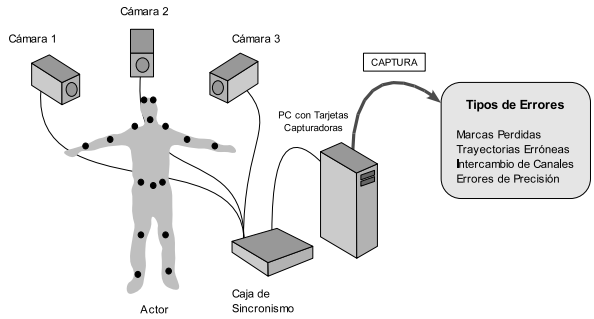


Figura 1: Sistema de Captura Óptico

1.2 PROBLEMÁTICA

La captura del movimiento realizada con sistemas ópticos presenta una serie de problemas (representados en la figura 2). En la figura 2, el eje de abcisas representa el tiempo, y el eje de ordenadas la posición de la marca en el espacio. Un tipo de problema se debe a la **oclusión de marcas** por el propio cuerpo del actor u otros elementos de la captura. Si el actor durante la captura oculta alguna marca en zonas donde al menos dos cámaras[1] puedan verla simultáneamente, el sistema generará errores puntuales en los frames donde la marca no esté visible.

Otro tipo de error muy común es la mala identificación de marcas por parte del sistema de reconstrucción 3D. Partiendo del conjunto de imágenes 2D capturadas, el sistema en algunas ocasiones no asigna las coordenadas 3D reales a una marca, volviendo a tomar de nuevo la trayectoria correcta unos frames más adelante. Esto nos provoca un efecto de **trayectorias erróneas** en ciertos fragmentos de la captura.

Dos marcas que estén próximas en un momento determinado, pueden intercambiar sus valores en un intervalo de tiempo, dándose lugar el tercer tipo de errores de **intercambio de canales**. Estos errores del sistema se deben a una identificación incorrecta de las marcas, y son difíciles de localizar automáticamente de forma general.

Por último, existe un tipo de error por **falta de precisión** que provoca un efecto de *trayectorias temblorosas* en la captura de forma global. Es, en parte, debido a la curvatura de la lente, la mala fijación de

[1]Son necesarias dos cámaras simultáneas para poder realizar visión estereoscópica

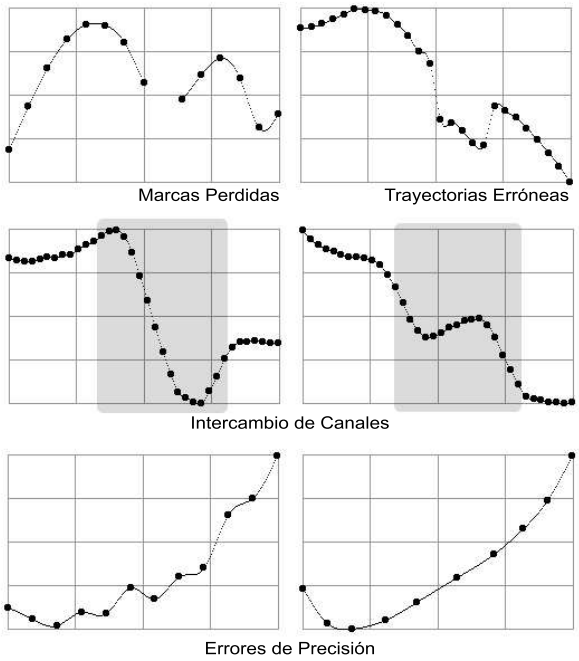


Figura 2: Tipos de errores en Sistemas Ópticos

las marcas reflectantes a las articulaciones del actor y a la pérdida de exactitud en la identificación de puntos 3D.

La detección de estos tipos de errores suele realizarse de forma manual. Un experto reproduce los datos captados y analiza los diferentes canales de entrada, identificando los problemas que han ocurrido en la fase de captura. Hecho esto, se procede a una fase de limpieza de los datos.

La limpieza de los datos suele realizarse bien de forma semiautomática, empleando interpolación mediante curvas [1], o bien manualmente, situando las marcas erróneas en una posición válida para cada frame.

Esta identificación de errores y su posterior limpieza es una labor muy costosa en tiempo si se realiza de forma manual o con procesos semiautomáticos, llegando a ser el verdadero cuello de botella en los sistemas de captura del movimiento.

2 USOS DE LÓGICA DIFUSA

Nuestra principal motivación es el modelado de un sistema capaz de identificar los errores en flujos de captura de movimiento tal y como lo hace un experto. Para ello, emplearemos sistemas de producción conceptuales basados en lógica difusa, que modelen de forma cualitativa el conocimiento experto. Esto nos permitirá una construcción y validación sencilla de

nuestro sistema de corrección y detección de errores.

Como se ha expuesto anteriormente, hay dos problemas principales al trabajar con sistemas de captura ópticos; por un lado la detección de los errores en cada canal (de algún tipo de los vistos en el apartado 1.2), y por otro la corrección de estos errores. Habitualmente estos procesos suelen realizarse de forma manual. A continuación veremos cómo incorporar detección y corrección de errores de forma automática utilizando lógica difusa.

2.1 DETECCIÓN DE ERRORES

Nuestra intención es construir sistemas difusos especialistas en cada uno de los tipos de errores descritos anteriormente. Así, tendremos un sistema de reglas encargado de localizar errores de marcas perdidas, otro para gestionar trayectorias erróneas, etc. Esto nos permitirá detectar automáticamente los errores en la captura y, generando tal y como lo haría un experto, los parámetros necesarios para proceder a una corrección de errores automática empleando técnicas de interpolación.

Sin embargo, para la detección de intercambios de marcas, necesitamos conocer la configuración jerárquica de las articulaciones que forman nuestro modelo. Esto implica la necesidad de poder calcular la cinemática directa e inversa de nuestro sistema. Tal y como se describe en [5], es posible calcular de manera eficiente la cinemática inversa de estructuras complejas mediante el uso de sistemas de reglas basados en lógica difusa y redes neuronales. Este último aspecto queda fuera del ámbito presente trabajo.

2.2 CORRECCIÓN DE ERRORES

La representación de una marca moviéndose a lo largo del tiempo produce curvas que pasan por un conjunto de puntos de control. Estas curvas pueden gestionarse de forma cómoda mediante polinomios cúbicos, que además no presentan discontinuidades ni ruido en los extremos del intervalo[2]. Así, podemos utilizar Splines Cúbicas Naturales [6] como método de interpolación de curvas para reparar gran parte de los errores detectados en el paso anterior.

Sería posible, además, incorporar filtros de corrección que incorporaran cierto grado de libertad a la hora de obtener un conjunto de trayectorias probables de una marca. Las soluciones cinemáticas obtenidas mediante splines dan resultados buenos, pero no son los únicos posibles. Se podrían tener en cuenta otros factores calculados (velocidades, aceleraciones, etc...) para obtener soluciones particulares para una configuración determinada de esos parámetros. Esta solución necesitaría de la construcción de un sistema de reglas que modelara la salida de un parámetro de tensión (empleando otro tipo de curvas de interpolación, como splines de hermite) con el que obtendríamos movimientos que generaran, por ejemplo, una mayor velocidad de movimientos, una menor aceleración, etc...

[2]Este problema se conoce como fenómeno de Runge

3 DESARROLLO DEL SISTEMA

Para la experimentación, validación y prueba de todas las propuestas del presente trabajo, se ha construído una aplicación denominada Pin8 [2], que utiliza un conjunto de sistemas de reglas difusos para la detección de los diferentes tipos de errores existentes en las capturas, donde se han implementado los mecanismos necesarios para la incorporación del conocimiento experto, así como herramientas para la corrección y visualización 3D, como puede verse en la figura 3. Esta aplicación permite la incorporación de nuevos sistemas de reglas difusos mediante la edición de ficheros XML, gestionando en tiempo de ejecución la incorporación de nuevos conjuntos difusos [10] que no estuvieran definidos previamente.

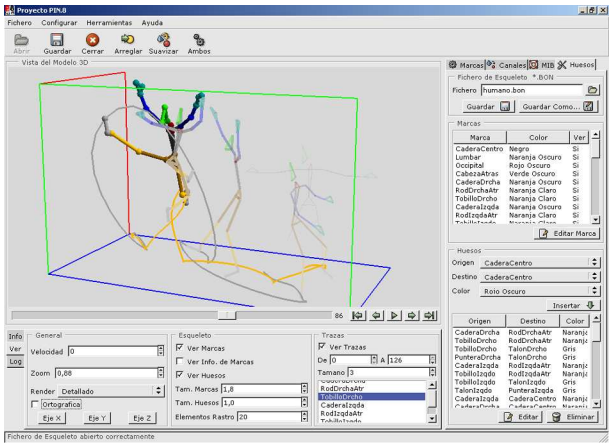


Figura 3: Interfaz de la aplicación

3.1 ANÁLISIS DEL DOMINIO DEL PROBLEMA

En nuestra primera aproximación al problema, nos vamos a centrar en la definición de un conjunto de sistemas que modelen el conocimiento experto en identificación de errores sobre unos canales de entrada. Esto, nos dará un conjunto Ξ de sistemas de reglas. Cada sistema de reglas estará definido sobre su propio conjunto de variables de entrada y sus dominios de definición. Tendremos distintas variables con las que construir nuestros sistemas σ como por ejemplo,

la posición de una marca s en un instante i, que notaremos como $P^{\sigma}_{s,i}$.

En la identificación de errores, nos hace falta conocer la posición de las marcas en el instante actual t, y en una serie de instantes anteriores t_{i-n}, siendo n el número de pasos de histórico que conservaremos. De esta forma, tenemos $P^{\sigma}_{s,t-i} = [x_{t-i}, y_{t-i}, z_{t-i}]\ \forall\sigma \in \Xi\ , i = 1..n$ en cada coordenada cartesiana.

Detectamos cambios en la velocidad de movimiento de las marcas mediante el conjunto de variables $V_{s,t-i}$ definidas como $V^{\sigma}_{s,t-i} = \Delta(P_{s,t-i}) = P^{\sigma}_{s,t-i} - P^{\sigma}_{s,t-i-1}\ \forall\sigma \in \Xi\ , i = 1..n$. De igual forma, nos interesa detectar cambios en la aceleración de las marcas definiendo un conjunto de variables $A_{s,t-i}$ definidas como $A^{\sigma}_{s,t-i} = \Delta(V_{s,t-i}) = V^{\sigma}_{s,t-i} - V^{\sigma}_{s,t-i-1}\ \forall\sigma \in \Xi\ , i = 1..n$.

Una vez definidas las variables y los sistemas de reglas, éstos permanecerán sin cambios siempre que utilicemos el mismo tipo de capturas y en el mismo rango de movimientos. Naturalmente, si necesitamos realizar capturas con otro tipo de actor o en otro tipo de problemas, habrá que cambiar el dominio de definición de cada variable difusa (pero no su distribución).

3.2 DEFINICIÓN DEL CONJUNTO DE REGLAS

A continuación vamos a definir los sistemas de reglas que hemos utilizado en la detección de errores en los canales de captura.

3.2.1 Detección de Marcas Perdidas

El primer sistema identifica los errores de marcas perdidas. Estos errores puntuales, en nuestro sistema de captura, vienen representados por un valor máximo que no se alcanza normalmente. Esto se traduce en un cambio muy grande puntual en la velocidad de la marca. Detectaremos cada intervalo que contenga marcas perdidas teniendo en cuenta estos cambios de velocidad. Así, nuestro primer sistema vendrá definido por V_t y V_{t-1} como variables de entrada y E como variable de salida, representando la posibilidad de error en el intervalo.

De este modo, nuestras variables de velocidad definidas para la marca s en el sistema σ, y el tiempo t (con un grado de histórico i), quedan representadas mediante cinco conjuntos difusos: MP (Muy Pequeña), P (Pequeña), N (Nula), M (Media) y G (Grande). Esta notación de los conjuntos difusos se conservará a lo largo del trabajo, aunque varíe el dominio de definición. De esta forma, tenemos, por ejemplo, la definición de los conjuntos difusos

$$MP = \int_{-10}^{-9} 1\,/x + \int_{-9}^{-8} -8 - x\,/x$$

$$P = \int_{-7}^{-6} x + 7\,/x + \int_{-6}^{-4} 1\,/x + \int_{-4}^{-1} \left(\frac{-1-x}{3}\right)/x$$

$$N = \int_{-2}^{0} \left(\frac{x+2}{2}\right)/x + \int_{-4}^{-1} \left(\frac{2-x}{2}\right)/x$$

$$M = \int_{1}^{4} \left(\frac{x-1}{3}\right)/x + \int_{4}^{6} 1\,/x + \int_{6}^{8} \left(\frac{8-x}{2}\right)/x$$

$$G = \int_{7}^{9} \left(\frac{x-7}{2}\right)/x + \int_{9}^{10} 1\,/x$$

Las variables lingüísticas de entrada se definen sobre el conjunto de etiquetas $\{MP, P, N, G, MG\}$ y la variable de salida sobre $\{P, M, G\}$. Obtenemos el siguiente conjunto de reglas:

R_1 : IF V_t is MG and V_{t-1} is MP THEN E is G
R_2 : IF V_t is MG and V_{t-1} is P THEN E is G
R_3 : IF V_t is MG and V_{t-1} is N THEN E is M
R_4 : IF V_t is MG and V_{t-1} is G THEN E is P
R_5 : IF V_t is G and V_{t-1} is MP THEN E is G
R_6 : IF V_t is G and V_{t-1} is P THEN E is G
R_7 : IF V_t is G and V_{t-1} is N THEN E is M

Sólo nos interesa conocer el comienzo de cada intervalo de error. La implementación del algoritmo de corrección de errores, basado en Splines Cúbicas Naturales, sólo necesita conocer el punto de comienzo de la interpolación. El punto final será calculado automáticamente. Por esta razón, no se han incluido el conjunto de reglas con valores MG para la variable V_{t-1}.

3.2.2 Detección de Trayectorias Erróneas

Un segundo sistema se encarga de la detección de trayectorias erróneas. Para ello, se utilizan variables de entrada de aceleración. Una aceleración *grande* (ya sea positiva o negativa) en la trayectoria de una marca implica una desviación de la trayectoria original. Esta desviación es signo de trayectoria errónea. Mediante el siguiente sistema podemos localizar, además, los límites del intervalo de frames en el que se produce el error. Las variables de entrada (A_t y A_{t-1}) se definen sobre el conjunto de etiquetas $\{MP, P, N, G, MG\}$. La variable de salida D (Desviación), sobre $\{P, M, G\}$.

Como se ha comentado anteriormente, el dominio de definición de cada variable es dependiente de cada tipo de captura. Un ejemplo de disposición de valores puede verse en la figura 4.

De esta forma, el sistema de reglas empleado en la detección de trayectorias erróneas estará formado por el siguiente conjunto de reglas:

R_1 : IF A_t is MG and A_{t-1} is N THEN D is G
R_2 : IF A_t is G and A_{t-1} is N THEN d is M
R_3 : IF A_t is MP and A_{t-1} is N THEN D is G
R_4 : IF A_t is P and A_{t-1} is N THEN D is M

Obviamente, R_1 y R_2 servirán para detectar un extremo del intervalo, mientras que R_3 y R_4 detectarán el otro intervalo. Una vez detectado el intervalo de error, se empleará una variación en el método de interpolación de Splines Cúbicas Naturales del sistema anterior para la corrección de los errores.

3.2.3 Detección de Trayectorias Temblorosas

El último sistema utilizado se encarga de la detección de trayectorias temblorosas. Este tipo de detección nos interesa que se realice a nivel global. No es importante detectar en qué intervalo se produce, ya que es un fenómeno general de la marca. Podría desarrollarse un método de asignación del factor de suavizado de la trayectoria dependiente del valor obtenido como salida en la desborrosificación. No obstante, se ha utilizado un valor fijo que da buenos resultados en todas las ocasiones para las capturas realizadas.

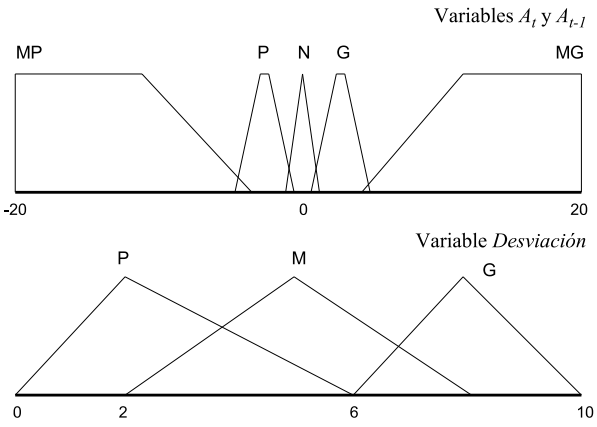


Figura 4: Sistema de detección de Trayectorias Erróneas

En este sistema, las variables de entrada elegidas son V_t, V_{t-1}, V_{t-2}, V_{t-3} definidas sobre las etiquetas $\{P, V\}$ (valores positivo y negativo). La variable de salida S, indicará si el canal necesita suavizado mediante las etiquetas $\{P, M, G\}$. Se necesitará suavizado si en algún punto de la curva se genera una trayectoria con *"dientes de sierra"*.

R_1 : IF V_t is P and V_{t-1} is N and and V_{t-2} is P and V_{t-3} is N THEN S is G

La detección uniforme de **intercambio de canales** presenta algunos problemas a la hora de establecer un sistema genérico. Por un lado, se puede pensar en controlar la distancia entre marcas a lo largo de la captura. En el momento en que la distancia entre cada par de marcas varíe respecto de su evolución histórica, estamos ante un intercambio de canales.

Esto exige la definición de una jerarquía de marcas o esqueleto interno, que nos permita realizar estas operaciones. Mediante la definición de dos sistemas (uno de detección del error y otro de identificación de parejas), se podrían reparar automáticamente este tipo de errores.

3.3 ANÁLISIS DE RESULTADOS

En la figura 5 se muestra el resultado de aplicar los sistemas de detección y corrección de errores sobre un intervalo de 70 frames de una captura real, en la marca correspondiente a la rodilla derecha de una persona tomando impulso para saltar. La trayectoria representada con trazos continuos muestra el resultado después de aplicar la corrección. Cada punto marcado en la gráfica se corresponde con la posición X, Y, Z de los canales asociados a cada marca en un instante de tiempo t. Como puede apreciarse, la trayectoria procesada se ajusta de forma correcta a la trayectoria original, produciendo un movimiento suave.

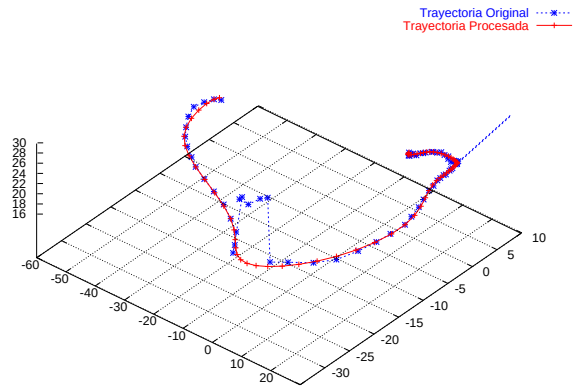


Figura 5: Resultado de aplicar los sistemas de detección y corrección de errores

El sistema de detección de marcas perdidas encuentra en el frame 27 un error puntual, debido a que la

velocidad en el canal respecto del eje X sufre un incremento muy alto. Esto puede observarse en la gráfica de la figura 6. En ese frame, se activa la regla R_2 y se aplica interpolación en el intervalo de frames 24-30.

En el intervalo de frames 47-53 hay una trayectoria errónea. Como puede verse en la figura 6, el sistema detecta el inicio del intervalo correctamente, y el final del mismo un frame después (debido al conjunto de reglas R_1 y R_2 en nuestro ejemplo). Esto no representa ningún problema, ya que la corrección se realiza tomando un intervalo de frames mayor. Debido a esto, la trayectoria reparada suaviza el camino seguido antes y después del intervalo erróneo.

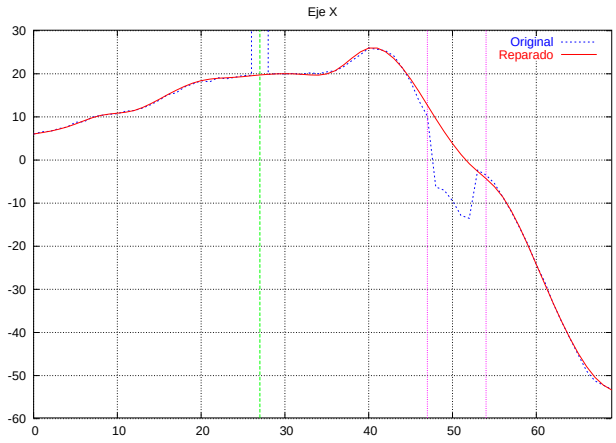


Figura 6: Resultado respecto del eje X

4 CONCLUSIONES

En este trabajo se ha presentado un método para la detección automática de errores en capturas de movimiento obtenidas mediante el uso de sistemas ópticos. La identificación de estos errores se ha realizado empleando conocimiento experto, que ha sido modelado utilizando las propiedades descriptivas de los sistemas de reglas basados en lógica difusa.

En futuros trabajos se avanzará en la incorporación de dependencias fuzzy entre articulaciones para el modelado de acciones complejas, definidas mediante un conjunto de sistemas dinámicos [4]. Esto nos permitirá detectar de forma sencilla las inconsistencias en los movimientos debidos al intercambio de canales, además de la generación cómoda de animaciones para personajes sintéticos y llegar a la definición puramente lingüística de acciones complejas.

Agradecimientos

Este trabajo ha sido financiado por los proyectos PREDACOM PB-03-004 de la Junta de Comunidades de Castilla-La Mancha y DIMOCLUST TIC2003-08807-C02-02 del Ministerio de Ciencia y Tecnología.

Referencias

[1] J. Foley, Computer Graphics: Principles and Practice, *Addison Wesley*, 1995.

[2] C. Gonzalez-Morcillo, L. Jimenez, Mejora de un Sistema de Captura de Movimiento mediante modelos teóricos y algoritmos de Inteligencia Artificial:Proyecto Pin8. http://www.inf-cr.uclm.es/www/cglez/pin8/Pin8Setup.exe

[3] E.H. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, *Int. J. Man-Mach, Studies, 7*, 1-13, 1975.

[4] J. Moreno-Garcia, L. Jimenez, J.J. Castro-Schez, L. Rodriguez, A linguistic modelling aproach for dynamic systems by using Temporal Fuzzy Models, *Subm. to Int. Journal of Approximate Reasoning*, 2004.

[5] S.M. Netto, A. Evsukoff et al. Fuzzy Systems to Solve Inverse Kinematics Problem in Robots Control; Application to an Hexapod Robots' Leg. *Sixth Brazilian Symposium on Neural Networks*, 2000.

[6] D. B. Olfe, Computer Graphics for Design. From Algorithms to AutoCAD. *Prentice Hall*, 1995.

[7] K. Ogata. Ingeniería de control moderna. *Prentice-Hall HispanoAmericana SA*, 1998.

[8] T. Takagi, M. Sugeno. Fuzzy identification of system and its application to modeling and control. *IEEE Transaction on system, Man and Cibernetics*, smc-15:116-132, January 1985.

[9] K. Tanaka. An introduction to fuzzy logic for practical aplications. *Springer*, 1998.

[10] L. A. Zadeh. Fuzzy set. *Information and control*, 338-353, 1965.

[11] L. A. Zadeh. The concept of a linguistic variable and its applications to Approximate reasoning. Part I, II, III. *Information Science*, 1975.

Clasificación difusa de imágenes digitales basada en la coloración de grafos

Daniel Gómez
Dept. Estadística e I.O. III
Universidad Complutense
dagomez@estad.ucm.es

Javier Montero
Dept. Estadística e I.O. I
Universidad Complutense
javier_montero@mat.ucm.es

Javier Yáñez
Dept. Estadística e I.O. I
Universidad Complutense
jayage@mat.ucm.es

1 Introducción

Una de las principales críticas que se hacen a los algoritmos de clasificación de imágenes digitales en el caso no supervisado, es que la mayoría de ellos no tienen en cuenta la información del entorno que rodeaba a un pixel. Por este motivo en este trabajo se presenta una vía para obtener una clasificación difusa que tiene en cuenta esta información.

El objetivo de este trabajo será el de obtener una clasificación difusa no supervisada que tenga en cuenta el entorno y la situación de cada pixel. Para obtener esta clasificación difusa primero se modelizará la imagen digital como un grafo difuso. De la coloración de este grafo difuso se obtendrá una clasificación nítida que nos permitirá determinar las principales regiones homogéneas de la imagen, esto dará lugar a una posterior clasificación difusa de la imagen digital.

Este trabajo está dividido de la siguiente forma: en la sección 2 se introduce el grafo difuso de pixels asociado a la imagen digital. En la sección 3 se presenta el algoritmo estricto de coloración. Este algoritmo tiene algunos problemas computacionales, y por este motivo se presenta en la sección 4 el algoritmo de coloración relajado. Esta coloración dará lugar a una clasificación difusa (sección 5).

2 Grafo difuso de pixels

Una imagen puede entenderse como un mapa de puntos (pixels), cada uno de los cuales está caracterizado por una serie de medidas como pueden ser intensidad del blanco, rojo, azul, altitud, etc.

Matemáticamente, se llamará P al conjunto de pixels

$$P = \{v = (i,j) \ / \ i \in \{1,\ldots,r\} \ j \in \{1,\ldots,s\}\}$$

de una imagen I. Cada pixel es caracterizado por b medidas numéricas. La imagen I puede ser caracterizada por esos valores numéricos, así se tiene que:

$$I = \{\overline{x_{i,j}} = (x^1_{i,j},\ldots x^b_{i,j}) \ / \ (i,j) \in P\}$$

Dada una imagen I, un problema estándar de clasificación nítida es el de la búsqueda de una *buena* partición del conjunto de pixels en regiones. Estas regiones serán consideradas como candidatos para una nueva clase, en el caso en el que la región sea lo suficientemente homogénea. Así pues en primer lugar se tratará de determinar una familia de pixels $\{A_1,\ldots,A_c\}$ de forma que $P = \cup_{k=1}^{c} A_k$ pero $A_l \cap A_m = \emptyset, \forall l \neq m$.

Debido a las propiedades de adyacencia entre pixels en una imagen, se modelizará la imagen I como un grafo planar difuso cuyos nodos son pixels, descritos por medio de sus coordenadas cartesianas $i \in \{1,\ldots,r\}$ y $j \in \{1,\ldots,s\}$.

El grafo será planar en el sentido de que dos pixels $v = (i,j)$ y $v' = (i',j')$ no pueden ser conectados si $|i-i'| + |j-j'| > 1$. Consecuentemente, dos pixels podrían ser adyacentes solamente si ellos comparten una coordenada, siendo la otra contigua.

La primera definición de grafo difuso fue propuesta por Kaufmann en [10], desde las relaciones difusas introducida por Zadeh en [14]. Aunque Rosenfeld introdujo en [13] otra definición elaborada, incluyendo nodos difusos y aristas difusos, en este capítulo se tratará con nodos nítidos y aristas difusos (ver [11]).

Sea $\tilde{G} = (V, \tilde{E})$ un grafo difuso, donde V es el conjunto de nodos, y $\tilde{E}$ el conjunto de aristas difusos caracterizados por la matriz $\mu = (\mu_{v,v'})_{v,v' \in V}$:

$$\mu_{v,v'} = \mu_{\tilde{E}}(\{v,v'\}) \qquad \forall v, v' \in V$$

siendo $\mu_{\tilde{E}} : V \times V \longrightarrow M$ su función de pertenencia.

Cada elemento $\mu_{v,v'} \in M$ representa el nivel de intensidad de una arista $\{v,v'\}$ para cada $v, v' \in V$. En este sentido, un grafo difuso puede ser también denotado como $\tilde{G} = (V, \mu)$.

El conjunto M es linealmente ordenado de forma que dos valores $\mu_{v,v'} \prec \mu_{u,u'}$ verifican que el nivel de intensidad de la arista $\{v, v'\}$ es menor que el nivel de intensidad de la arista $\{u, u'\}$. El conjunto M permite la graduación literal del conjunto de aristas; por ejemplo, si $M = \{n, l, h\}$ las aristas pueden ser graduados como nulo (n), bajo (l) o alto (h).

Un grafo difuso $\tilde{G}$ puede ser considerado como una generalización del grafo G, ya que, tomando $M = \{0, 1\}$, $\tilde{G}$ es un grafo nítido si la matrix μ es definida como

$$\mu_{v,v'} = \begin{cases} 1 & \text{if } \{v, v'\} \in E \\ 0 & \text{en otro caso} \end{cases} \qquad \forall v, v' \in V$$

Definición 2.1 *Dada la imagen I, y una distancia d entre pixels, el* grafo difuso de pixels *es definido como el par $\widetilde{G(I)} = (P, \tilde{E})$ siendo $\tilde{E}$ igual a*

$$\left\{ \{\mu_{v,v'} = d(\overline{x_{i,j}}; \overline{x_{i',j'}})\} \ / \ \vee \begin{bmatrix} i = i' & |j - j'| = 1 \\ j = j' & |i - i'| = 1 \end{bmatrix} \right\}$$

Teniendo en cuenta la topología de $\widetilde{G(I)}$, los pixels del grafo difuso pueden ser también caracterizados por el conjunto P y las dos matrices $r \times s$, D^1 y D^2, donde

$$D^1_{i,j} = d(\overline{x_{i,j}}, \overline{x_{i+1,j}}) \ \ \forall (i,j) \in \{1, \ldots, r-1\} \times \{1, \ldots, s\}$$

y

$$D^2_{i,j} = d(\overline{x_{i,j}}, \overline{x_{i,j+1}}) \ \ \forall (i,j) \in \{1, \ldots, r\} \times \{1, \ldots, s-1\}$$

Así pues el conjunto de aristas que modeliza la imagen queda definido como sigue:

El proceso de coloración propuesto en este capítulo estará basado en esta representación alternativa, de ahora en adelante se identificará el grafo difuso de pixels $\widetilde{G(I)}$ por (r, s, D^1, D^2).

Ejemplo 2.1 *Sea (r, s, D^1, D^2) grafo de pixels difuso con $r = 3$, $s = 4$ y*

$$D^1 = \begin{pmatrix} 1.5 & 2.0 & 1.9 & 3.6 \\ 1.7 & 1.8 & 0.9 & 1.8 \end{pmatrix} \quad D^2 = \begin{pmatrix} 3.1 & 1.9 & 3.5 \\ 1.6 & 4.0 & 1.8 \\ 2.1 & 0.7 & 0.8 \end{pmatrix}$$

EL grafo de pixels difuso puede verse en la figura 1.

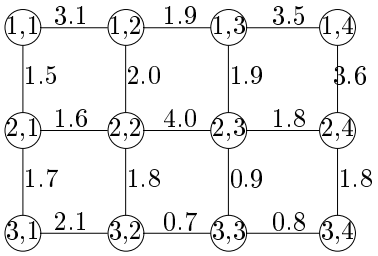


Figura 1: Grafo difuso de pixels del ejemplo 2.1

3 Algoritmo estricto de Coloración

Una c-coloración de un grafo $G = (V, E)$ (ver [12]) es una función

$$C \ : \ V \ \longrightarrow \ \{0, \ldots, c-1\}$$

que verifica que $C(v) \neq C(v')$ si $\{v, v'\} \in E$. Una c-coloración induce una clasificación nítida del conjunto de nodos V, asociando a cada color una clase:

$$V_C(k) = \{v \in V \ / \ C(v) = k\} \qquad k \in \{0, \ldots, c-1\}$$

Una coloración binaria de un grafo $G = (V, E)$ es un caso particular de una 2-coloración

$$col \ : \ V \ \longrightarrow \ \{0, 1\}$$

El objetivo que se persigue es obtener una clasificación del conjunto de pixels por medio de una c-coloración C del grafo difuso de pixels $\widetilde{G(I)}$: el pixel $(i, j) \in P$ estará clasificado como $k \in \{0, \ldots, c-1\}$ si su color es $C(i, j) = k$.

El problema de coloración de un grafo difuso puede verse en [10]. Se define la familia de G_α, a los grafos nítidos definidos mediante α-cortes del conjunto de aristas

$$E_\alpha = \{\{v, v'\} \in P \times P \ / \ \mu_{v,v'} \geq \alpha\}$$

Los valores del parámetro α serán seleccionados de forma que un sucesivo proceso de coloración binario será aplicado a algunos subgrafos parciales difusos de $\widetilde{G(I)}$.

La primera coloración binaria analiza el conjunto de pixels P clasificando cada pixel como 0 ó 1. La segunda coloración binaria es aplicada por separado a cada subgrafo generado por aquellos pixels coloreados como 0 obteniendo así las clases 00 y 01, y para el subgrafo generado por aquellos pixels coloreados como 1, para obtener las clases 10 y 11.

Este proceso jerárquico de coloración binaria es repetido en el proceso de coloración estricto. En este sentido

una c-coloración C será definida en $\widetilde{G(I)}$: Si, por ejemplo, $C(i,j) = 5$, al ser 101 la representación binaria de 5, el pixel (i,j) será coloreado tres veces como 1, 0 y 1 respectivamente; en este caso se necesitan tres procesos binarios de coloración fijando en cada uno de ellos el primer, segundo y tercer dígito de la representación binaria del color.

3.1 Procedimiento de coloración binaria

El procedimiento de coloración binaria clasifica dos pixels adyacentes como 0 y 1 si sólo si la distancia ente ellos es mayor o igual que un valor prefijado α. Esta es una aproximación alternativa al problema de clasificación clásico ya que en la aproximación clásica dos pixels serán clasificados en la misma clase si son similares sin tener en cuenta si son adyacentes o no.

Formalmente, para definir el primer procedimiento de coloración binario, dado un valor α, sea G_α el α-corte del grafo difuso $\widetilde{G(I)}$:

$$G_\alpha = (P, E_\alpha)$$

donde E_α es el conjunto de todos los pares de pixels adyacentes con distancia d menor que α.

Sea $col : P \longrightarrow \{0,1\}$ una coloración binaria de G_α. Teniendo en cuenta la topología del grafo difuso de pixels, la primera coloración binaria puede ser obtenida asignando color arbitrario ("0" o "1") a un pixel concreto, y definiendo un orden en el cuál cada pixel será coloreado; si se supone, por ejemplo, que el primer pixel es el $(1,1)$ en la parte superior izquierda de la imagen, los pixels serán coloreados desde la izquierda hasta la derecha y de arriba a abajo con un valor de α fijo.

Dado un pixel coloreado (i,j), los pixels adyacentes $(i+1,j)$ y $(i,j+1)$ son coloreados; el pixel $(i+1,j+1)$ puede ser coloreado desde el pixel $(i+1,j)$ y desde el pixel $(i,j+1)$. Una restricción natural es que ambos colores deben ser iguales, en otro caso el pixel será denotado como *inconsistente*.

Definición 3.1 *Dado el conjunto de pixels P un* cuadrado *es un subconjunto de cuatro pixels*

$$sq(i,j) \equiv \{(i,j);(i+1,j);(i,j+1);(i+1,j+1)\}$$

siendo $i \in \{1,\ldots,r-1\}$ *y* $j \in \{1,\ldots,s-1\}$.

Denotaremos entonces por PS el conjunto de todos los *cuadrados*.

Definición 3.2 *Dado un grafo difuso de pixels (r,s,D^1,D^2), un cuadrado $sq(i,j) \in PS$ es* consistente *a nivel α si, dado un color arbitrario $col(i,j)$, el anterior procedimiento de coloración binaria asigna el mismo color al pixel $(i+1,j+1)$, tanto si en coloreado desde el pixel $(i,j+1)$ o desde el pixel $(i+1,j)$. En otro caso se dirá que el cuadrado de pixels es* inconsistente.

En el ejemplo 2.1, el pixel $sq(1,1)$ es inconsistente al nivel $\alpha = 3.1$, pero consistente al nivel $\alpha = 3.2$.

Definición 3.3 *Dado un valor de α, el grafo difuso de pixels (r,s,D^1,D^2) es* consistente al nivel α *si todos los cuadrados $sq(i,j) \in PS$ son consistentes al nivel α.*

Existen dos valores extremos que garantizan la consistencia de todos los cuadrados:

- $\overline{\alpha} = \max_{(i,j)\in P}\{d^1_{i,j}; d^2_{i,j}\}$

  Si se fija un valor $\alpha > \overline{\alpha}$, entonces toda la imagen es considerada como una clase única ($col(i,j) = col(1,1)\ \forall (i,j) \in P$).

- $\underline{\alpha} = \min_{(i,j)\in P}\{d^1_{i,j}; d^2_{i,j}\}$

  En el caso $\alpha < \underline{\alpha}$, la imagen parece un tablero de ajedrez, siendo todos los pixels adyacentes clasificados como "0" y "1":

$$col(i,j) = \begin{cases} col(1,1) & si\ (i+j)\ es\ impar \\ 1 - col(1,1) & en\ otro\ caso \end{cases}$$

En este sentido, solamente el intervalo $[\underline{\alpha}, \overline{\alpha}]$ debería ser considerado. Determinar un nivel α intermedio apropiado no es una cuestión trivial.

Dado un nivel α, los cuadrados inconsistentes pueden ser detectados con la función binaria $inconsis_\alpha(i,j,D^1,D^2)$ que devuelve 1 si ese cuadrado es consistente y 0 si no lo es.

Se puede notar que esta función binaria es muy importante, así que se debería escoger un valor apropiado para α. Esta elección dará lugar a la coloración binaria *col*. Sin embargo la función $inconsis_\alpha(i,j,D^1,D^2)$ no depende del color particular del pixel (i,j).

El grafo difuso del ejemplo 2.1 es inconsistente al nivel $\alpha = 3.1$, pero es consistente al nivel $\alpha = 4.1$.

La coloración que se propone en este articulo está basada en sucesivas coloraciones binarias para diferentes niveles consistentes (que van decreciendo), por este motivo nos interesa empezar con el máximo valor (α^*) que hace que el grafo difuso de pixels sea consistente.

Definición 3.4 *Dado el grafo difuso de pixels (r,s,D^1,D^2), el* nivel de consistencia, *denotado como α^*, es el máximo valor $\alpha \in [\underline{\alpha}, \overline{\alpha}]$ para el cual el grafo difuso es consistente.*

La existencia de este nivel de consistencia está siempre asegurada, al menos mientras la imagen contenga un número finito de pixels. Si alguna inconsistencia es detectada para algún valor de α dado, puede ser introducido un procedimiento decreciente para encontrar un valor α^* menor que asegure la consistencia. Este procedimiento será inicializado con $\alpha^* = \overline{\alpha}$.

En el ejemplo 2.1, el cuadrado $sq(1,2)$ es inconsistente al nivel $\alpha = \overline{\alpha} = 4.0$, ya que $col(1,2) = col(1,3) = col(2,2)$ pero $col(2,3) = col(1,3)$ y $col(2,3) = 1 - col(2,2)$. Este cuadrado no puede ser consistente para un nivel $\alpha \in (2.0, 4]$. Para $\alpha = 2.0$, sin embargo, el cuadrado $sq(2,1)$ es inconsistente para cualquier valor de $\alpha \in (1.8, 2.0]$. De hecho, puede ser chequeado que si $\alpha = 1.8$, el grafo difuso de pixels es consistente siendo inconsistente para cualquier valor de $\alpha \in (1.8, 4.0]$. En este caso, $\alpha^* = 1.8$.

Para determinar α^* se busca entre aquellos cuadrados inconsistentes $sq(i,j)$: Si este cuadrado es inconsistente al nivel α, se puede computar el máximo valor α' para el cuál este cuadrado será consistente, por medio de una nueva función *newalpha*, que será evaluada para cada cuadrado.

En el ejemplo 2.1, se puede ver que

$$newalpha_{4.0}(1,2,D^1,D^2) = 2.0$$

$$newalpha_{2.0}(2,1,D^1,D^2) = 1.8$$

En este ejemplo, todos los pixels han sido coloreados al principio, después, se ha buscado el valor α^* que asegura la consistencia. Así pues se tiene a los pixels clasificados en las clases "0" o "1". Se debería ahora proceder a conseguir un color más preciso para ambas clases (la clase "0" debe a su vez ser divida en la clase "00" o "01"). Esto será realizado de forma separada, activando alternativamente solo una de las clases que ya están coloreadas en una etapa anterior.

De la misma forma se procederá en las posteriores etapas, de forma que el anterior proceso de coloración binaria es aplicado a aquellos pixels activados, es decir un subconjunto de pixels $P' \subset P$. Este subconjunto de pixels P' puede ser caracterizado por una matriz *act* tal que $act(i,j) = 1, \forall (i,j) \in P'$ y $act(i,j) = 0 \quad \forall (i,j) \notin P'$.

Se puede computar el intervalo $[\underline{\alpha}, \overline{\alpha}]$ para los pixels activados (proceso de llamaremos en el desarrollo del trabajo $initalpha(r,s,D^1,D^2,act)$). En el caso en que no existan dos pixels adyacentes activados el proceso debería parar.

De nuevo se puede observar que un cuadrado dado puede ser consistente para un valor α pero inconsistente para otro valor $\alpha' < \alpha$. Aquí un procedimiento decreciente repetido para el conjunto de cuadrados de pixels PS encontrará el nuevo valor α^* asegurando que todos los cuadrados son hechos consistentes en la nueva coloración.

La siguiente función, *consislevel*, es el núcleo de nuestro algoritmo: esta función será evaluada iterativamente obteniendo así el nivel de consistencia α^* para la familia de pixels que han sido activados. Los inputs del procedimiento *consislevel* son los pixels del grafo difuso (r,s,D^1,D^2) y la matriz *act*. Esta función evalúa el nivel de consistencia $\alpha^*(act)$, para un subconjunto de pixels activados $P' \subset P$. Hay que observar que inicialmente cuando $act(i,j) = 1 \ \forall (i,j) \in P$, entonces $\alpha^*(act) = \alpha^*$, el nivel de consistencia del grafo difuso de pixels.

Siguiendo un orden estándar en los pixels activados, el nivel $\alpha^*(act)$ asegura un procedimiento de coloración binaria válido *col* que permite determinar el color $col(i+1,j)$ con dos opciones:

- $col(i,j) \ \forall (i,j) \in P$ tal que $act(i,j) * act(i+1,j) = 1$ y $d^1_{i,j} < \alpha^*(act)$
- $1 - col(i,j) \ \forall (i,j) \in P$ tal que $act(i,j) * act(i+1,j) = 1$ y $d^1_{i,j} \geq \alpha^*(act)$

Análogamente, se determina $col(i,j+1)$:

- $col(i,j) \ \forall (i,j) \in P$ tal que $act(i,j) * act(i,j+1) = 1$ y $d^2_{i,j} < \alpha^*(act)$
- $1 - col(i,j) \ \forall (i,j) \in P$ tal que $act(i,j) * act(i,j+1) = 1$ y $d^2_{i,j} \geq \alpha^*(act)$

Todos estos cálculos se agruparán en el proceso que denotamos por $bincol_\alpha(r,s,D^1,D^2,act,col)$. Este proceso calcula la coloración binaria de los pixels activados en el nivel α. Este procedimiento se inicializa $col(i,j) = 0 \quad \forall (i,j) \in P$.

Volviendo de nuevo al ejemplo 2.1, y a la asignación $act(i,j) = 1 \ \forall (i,j) \in P$ y $\alpha = \alpha^* = 1.8$, el procedimiento *bincol* computa la coloración binaria vista en la figura 2.

Después de una primera etapa, los pixels coloreados son clasificados en dos clases: 0 y 1, dependiendo de cuál es la distancia considerada entre pixels adyacentes.

3.2 Algoritmo estricto de coloración

El proceso de clasificación inducido por la coloración binaria puede ser redefinido, aplicado a cada matriz *act*, para clasificar pixels de la clase 0 de la siguiente forma:

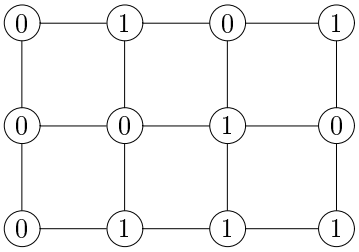


Figura 2: Primera coloración binaria del grafo de pixels difuso del ejemplo 2.1

$$act(i,j) = \begin{cases} 1 & \forall (i,j) \in P \ / \ col(i,j) = 0 \\ 0 & en \ \ otro \ \ caso \end{cases}$$

El procedimiento *bincol* es entonces aplicado con el valor $\alpha = consislevel(r, s, D^1, D^2, act)$. Los pixels activados serán clasificados de nuevo como 0 o 1 dando lugar a las clases "00" y "01".

Análogamente, las clases 10 y 11, son definidas cuando este segundo proceso de coloración es repetido para la clase 1, de forma que cuatro clases son obtenidas: "00", "01", "10" y 11.

Los pixels clasificados pueden ser identificados por la función:

$$C \ : \ P \ \longrightarrow \ \{0,1,2,3\}$$

donde $C(i,j)$ es el número entero asociado con el número binario de la clase.

El mismo procedimiento de coloración puede ser sucesivamente aplicado a cada familia de pixels pertenecientes a la misma clase de color, siempre y cuando existan pixels adyacentes con al menos dos distancias distintas entre ellos. (es decir, siempre que se cumpla que $consislevel(r, s, D^1, D^2, act) > 0$). Así, si este proceso de coloración es sucesivamente aplicado t veces, obtendremos 2^t clases a lo sumo.

En el caso en que se añada un nuevo nivel de clasificación, incrementando el parámetro t, es porque existen al menos dos pixels adyacentes ((i,j) y (i',j')) en la misma clase ($C(i,j) = C(i',j')$) verificando que $d((i,j),(i',j')) \geq \alpha^*(act)$, cuando $act(i,j) = act(i',j') = 1$; este número de pares será denotado por *adp*. En otro caso, el proceso de coloración para. Este proceso de coloración es denotado por $\mathbf{crucol}(r, s, D^1, D^2, C)$.

Este procedimiento de clasificación puede verse como un método jerárquico divisible [7]. Estos métodos de clasificación empiezan con un cluster que contiene a todos los elementos, y en cada paso se dividen las clases obteniendo mayor número de clases en la siguiente iteración. Si el proceso lo dejásemos correr, finalmente se tendrían tantas clases como objetos.

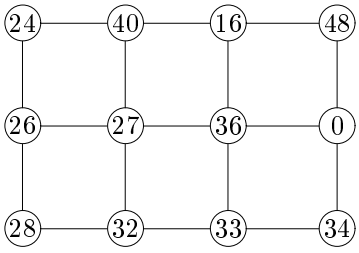


Figura 3: Coloración del grafo difuso de pixels del ejemplo 2.1

4 Algoritmo relajado de coloración

Cuando se trabaja con imágenes digitales de tamaño medio o alto, el número de inconsistencias en relación con el tamaño de la imagen suele ser muy pequeño. Sin embargo puede ocurrir, que cuando se busca en el proceso de decrecimiento el valor $\alpha^*(act)$ este sea el valor 0. Consecuentemente los pixels no son clasificados.

Para paliar este problema, se propone relajar las restricciones de consistencia dando lugar así a un nuevo procedimiento de coloración binaria, permitiendo algunas inconsistencias (por ejemplo un 1 % de cuadrados inconsistentes). La complejidad computacional queda resuelta fijando un pequeño número de iteraciones (4 o 5, por ejemplo).

Denotamos por *incratio* como el porcentaje de inconsistencias del proceso de coloración binario, es decir el número de pixels inconsistentes divididos por $(r-1)(s-1)$. Como se puede observar si permitimos valores grandes de *incratio*, el valor de α que necesitamos no decrecerá mucho y necesitaremos por tanto un número de clases pequeño. Por otro lado, a mayor valor de *incratio* mayor número de pixels serán coloreados erróneamente.

Así pues, se buscará un valor de *compromiso* entre estas dos situaciones. Esta solución de compromiso debe tener en cualquier caso ratios de inconsistencia pequeños (por ejemplo del orden de 0.01, de forma que un porcentaje muy pequeño de pixels serán coloreados de forma equivocada, en el caso de que la imagen sea grande esa clasificación errónea no podrá ser percibida por el ojo humano). Cada pixel inconsistente deberá entonces ser *aislado* de forma que su inconsistencia no induzca más inconsistencias en los pixels adyacentes.

Sea *col* una coloración binaria. Inicialmente, $col(i,j) = 0 \ \forall (i,j) \in P$, y entonces se procederá a asignar nuevos colores a cada pixel. El orden con el

que los pixels serán de nuevo coloreados será el mismo que teníamos en el algoritmo estricto de coloración (de izquierda a derecha y de arriba a abajo).

En el proceso de coloración de la primera fila, desde la izquierda hasta la derecha, no se producirán inconsistencias, así como el proceso de coloración vertical de las columnas desde la primera. Las inconsistencias, si existen, aparecerán cuando exista un cuadrado $sq(i,j)$ tal que el pixel $(i+1, j+1)$ sea coloreado de forma distinta si se hace desde el pixel $(i+1, j)$ que si se hiciese desde el pixel $(i, j+1)$. Denotaremos por *ninc* al número total de pixels inconsistentes.

Para que el proceso de coloración funcione correctamente, dado un pixel activado $(i,j) \in P$, es muy importante distinguir entre aquéllos que todavía no han sido coloreados, $act(i,j) = 2$, y aquéllos que han sido coloreados, $act(i,j) = 1$. Inicialmente $act(i,j) = 2$, para todos los pixels activados (i,j).

Así pues, para cada pixel activado $(i,j) \in P$, se puede asociar el valor $act(i,j) = 2$ si este pixel no ha sido coloreado todavía y $act(i,j) = 1$ en otro caso. Inicialmente, $act(i,j) = 2$ para todos los pixels activados. Una vez que un pixel (i,j) es coloreado, $act(i,j) = 1$ si es consistente (siempre al actual nivel α); en otro caso $act(i,j) = -1$. como se ha mencionado anteriormente, estos pixels inconsistentes deben ser aislados para que no *contaminen* a los pixels adyacentes.

Se distinguirán los pixels activados $act(i,j) \neq 0$ de los no activados $act(i,j) = 0$, en el primer caso caso $act(i,j) = 2$ si no han sido coloreados y $act(i,j) = 1$ cuando lo sean consistentemente y $act(i,j) = -1$ en otro caso. Como se ha mencionado anteriormente, estos pixels inconsistentes deben ser aislados para que no *contaminen* a los pixels adyacentes.

Un pixel inconsistente puede ser arbitrariamente coloreado. Iteraciones posteriores suavizaran los efectos de esta coloración arbitraria.

Para detener el crecimiento exponencial de las iteraciones del algoritmo estricto, el número de iteraciones t debe estar acotado (en este sentido se tiene un control a priori sobre el número de particiones). Se denota por it_M dicha cota.

En cada iteración, el valor de α se escoge teniendo en cuenta los diferentes valores de las distancias entre pixels. Por otra parte el parámetro it_M deber ser elegido teniendo en cuenta muchos factores. Empíricamente, así como para mejorar la visualización de los resultados un valor razonable es $it_M = 6$

Una vez se ha fijado el valor de it_M, diferentes valores de α pueden ser elegidos. Denotamos por α_t, con $t \in \{1, \ldots, it_M\}$, la familia de esos valores. El procedimiento *vecalpha* devuelve eso valores en el vector $(\alpha_1, \ldots, \alpha_{it_M})$.

5 Clasificación difusa

A partir de los algoritmos introducidos anteriormente, se tiene una clasificación nítida. Sean $C_1, \ldots C_{2^{it}}$ las clases de dicha clasificación.

Definición 5.1 *Dada una clase C_k Diremos que región $R \subset P \times P$ es homogénea respecto a dicha clase si $\forall (i,j) \in R$, $(i,j) \in C_k$ y además está rodeada de pixels que no pertenecen a C_k*

De esta forma se obtienen las diferentes regiones homogéneas dentro de la imagen digital. Denotaremos por $NREG$ al número de regiones identificadas por ese proceso de clasificación.

Ejemplo 5.1 *Sea (P, I) una imagen digital, de tamaño 3x7. Una vez hecha la clasificación utilizando cuatro colores, tenemos cuatro clases:* 0, 1, 2, 3. *A continuación se ve el grafo asociado a dicha clasificación.*

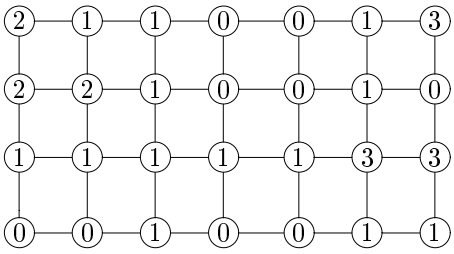


Figura 4: Regiones homogéneas de la red de pixels

Como puede verse la región $R = \{(1,1), (2,1), (2,2)\}$ es una región homogénea respecto a la clase 2. En esta red de pixels se distinguen diez regiones homogéneas.

Las regiones homogéneas aquí obtenidas son fruto de una coloración de un grafo difuso que representa una imagen digital. Sin embargo cuando la red de pixels es grande, la forma de representar los datos es esencial para el posterior desarrollo de la clasificación.

Para representar las diversas regiones homogéneas se usará una nueva imagen digital donde cada región es representada por un color que resuma las principales características espectrales de la región (ver [8]).

Una vez hecho esto cada región quedará representada por un color que presenta la vaguedad de la región definida. Esta representación (difusa) de las regiones homogéneas puede dar lugar (según sea el interés del

estudio) a una clasificación difusa de la imagen digital (sin más que aplicar algoritmos de clasificación difusa de datos como los algoritmos vistos en [5]), o a la obtención de funciones de pertenencia para las distintas regiones homogéneas (mediante funciones de verosimilitud a cada región) que han sido determinadas en la imagen teniendo en cuenta el entorno de cada pixel.

Aplicando este algoritmo a mas de 20 imágenes digitales se observa que con pocos colores se obtiene una clasificación buena de las diversas regiones homogéneas. Si el número de colores es suficientemente grande se tendrán tantas regiones homogéneas como se tienen en la fotografía original. Una vez que se tiene una buena información acerca del número de clases y sus características se está en disposición de realizar una clasificación difusa (ver [2, 3, 4]).

6 Conclusiones Finales

Los principales algoritmos no supervisados que abordan el problema de la clasificación no tienen en cuenta el entorno que rodea a cada pixel, haciendo imposible la clasificación correcta en determinados casos. La clasificación que aquí se propone tiene en cuenta las disimilitudes entre pixels adyacentes para una clasificación nítida inicial. En este trabajo se ha hecho hincapié en la fase de la determinación de regiones homogéneas de la imagen digital, por este motivo esta propuesta, debe ser entendida como una pieza de un algoritmo mucho mas complejo en que se aborden temas que aquí se tratan de pasada como: la determinación de la adecuada función de pertenencia, número de clases difusas que obtienen una buena representación de los datos, procesos de agregación de la información ...etc.

Referencias

[1] A. Amo, J. Montero y G. Biging: *Relevancia y redundancia en sistemas de clasificación difusa.* En: Proceedings Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF 2000 (A. Ollero, S. Sánchez, B. Arrue y I. Baturone, eds., Universidad de Sevilla, Sevilla); pp. 521-525 (2000).

[2] A. Amo, D. Gómez, J. Montero y G. Biging: Relevance and redundancy in fuzzy classification systems. Mathware and Soft Computing 8, 203-216 (2001)

[3] A. Amo, J. Montero, A. Fernández, M. López, J. Tordesillas y G. Biging: Spectral fuzzy classification: an application. IEEE TRANS.SYST.MAN AND CYB. C 32, 42-48 (2002)

[4] A.Amo, D. Gómez, J. Montero: Spectral fuzzy classification: a supervised approach. Mathware and Soft Computing 10:141-154 (2003)

[5] A. Amo, J. Montero, G. Biging y V. Cutello: Fuzzy classification systems. EUR. JOUR. OPER. RES. (to appear)

[6] A. Amo, J. Montero y V. Cutello: *On the principles of fuzzy classification.* En: Proceedings North American Fuzzy Information Processing Society Conference; 675–679 (1999).

[7] B. S. Everitt *Cluster analysis.* 1993.

[8] D. Gómez. *Algunas aportaciones sobre relaciones de preferencia.* Ph.D. Thesis, Complutense University of Madrid, Spain (in Spanish).

[9] J. R. Jensen: *Introductory digital image processing.* 1986 Prentice-Hall.

[10] A. Kaufmann: *Introduction à la théorie des sous-ensembles flous, 1. Eléments théoriques de base.* Masson et Cie, Paris, France, (1976).

[11] L. Kóczy *Fuzzy graphs in the evaluation and optimization of netwoks.* Fuzzy sets and systems; 46, (1992) 307-319.

[12] P. M. Pardalos, T. Mavridou, J. y Xue The Graph Coloring Problem: A Bibliographic Survey. In: Du, D.-Z., and Pardalos, P.M. (Eds.). Handbook of Combinatorial Optimization, vol. 2 (Kluwer Academic Publishers, Boston, 1998), pp. 331-395.

[13] A. Rosenfeld *Fuzzy graphs.* In: Zadeh, L.A. Fu, K.S., y Shimura,M., eds, Fuzzy sets and their applications to cognitive and decision processes, Academic Press New York, (1975) 77-95.

[14] L.A. Zadeh: *Similarity relations and fuzzy orderings.* Information Science 3:177–200 (1971).

ANÁLISIS DE UN MECANISMO DE AHORRO DE ENERGÍA MEDIANTE LÓGICA FUZZY PARA LA MEJORA DE SISTEMAS WLAN 802.11

Begoña Otal Ruiz

Philips Research Laboratories
Weisshaustrasse 2, 52066 Aachen, Alemania

begonya.otal@philips.com

Luis Alonso Zárate

Dept. de Teoría del Señal y Comunicaciones
Universitat Politècnica de Catalunya (UPC)
Av. Canal Olímpic s/n EPSC. D-210.
08860 Castelldefels, Barcelona
luisg@tsc.upc.es

Resumen

La mayoría de los protocolos de acceso al medio, los llamados protocolos MAC (Medium Access Control), para sistemas WLAN (Wireless Local Area Network), fijan una potencia de transmisión constante sin considerar técnicas de ahorro de energía centradas en modelos de medida o estimación del estado del canal. Con la finalidad de prolongar el tiempo de vida de las baterías de los aparatos inalámbricos y de mejorar el rendimiento en conjunto de la capacidad de procesamiento de dichos sistemas, en esta contribución se propone un mecanismo de ahorro de energía basado en un protocolo MAC de colas distribuidas que incorpora lógica *fuzzy*. Se han realizado simulaciones por ordenador para obtener resultados, bajo diferentes escenarios, del máximo rendimiento de la capacidad de transmisión y del ahorro en la energía consumida. Los resultados hallados muestran que la capacidad obtenida supera la del estándar IEEE 802.11 y además se obtiene un ahorro de energía notable.

Palabras Clave*:* Lógica Fuzzy, IEEE 802.11, MAC, Cross-layer**.**

1. INTRODUCCIÓN

En el modo infraestructura de IEEE 802.11, todas las estaciones móviles se comunican con el punto de acceso, AP (*Access Point*) compartiendo el canal radio. El AP facilita tanto la conexión a la LAN cableada, si existe, como las funciones en relación local. En los estándares actuales no existe ningún esquema de gestión de la potencia utilizada mientras se están transmitiendo los datos teniendo en cuenta el estado del canal radio. En este documento se propone una técnica de ahorro de energía mediante *cross-layer* (PHY-MAC) utilizando técnicas de lógica *fuzzy*. Se utilizará un protocolo MAC distribuido, llamado DQCA (*Distributed Queuing Collision Avoidance*) 0-[2], que está basado en un esquema de colas distribuidas y cuya estructura está inherentemente preparada para incluir este tipo de técnicas.

2. PROTOCOLO MAC

DQCA es un protocolo MAC estable y de alto rendimiento que se comporta como un mecanismo de acceso aleatorio (e.j.: ALOHA) para bajas cargas de tráfico, mientras que cuando la carga aumenta cambia gradualmente y de forma automática a un protocolo distribuido de reserva. Este protocolo MAC está basado en dos colas distribuidas: i) la cola de transmisión de datos, DTQ (*Data Transmission Queue*), dedicada al reparto de los paquetes de datos a transmitir, y ii) la cola de resolución de colisiones, CRQ (*Collision Resolution Queue*), dedicada al algoritmo de resolución de colisiones. Ambas colas se gestionan simultáneamente usando un conjunto de reglas que permiten la eliminación completa de colisiones en el período de las transmisiones de paquetes de datos. Estas dos colas se representan de forma simple mediante cuatro números enteros que cada nodo debe mantener y actualizar en cada trama, basándose en la información de control que manda el AP de forma *broadcast* a través de un paquete CTS especial.

Estos cuatro números enteros se denominan TQ, RQ, pTQ y pRQ. TQ representa el número de mensajes que están esperando a transmitir en la cola DTQ. RQ es el número de colisiones a la espera de ser resueltas en la cola CRQ. pTQ es la posición del nodo en la cola DTQ. pRQ es la posición del nodo en la cola CRQ. Se reserva un pequeño intervalo de tiempo en cada trama para intentos de acceso. Las transmisiones de petición de acceso se realizan con un paquete RTS especial (ver Figura 1). La idea básica del protocolo MAC es concentrar las peticiones de acceso de los usuarios y las colisiones en este intervalo de control reservado, mientras que el resto de la trama está dedicado a la transmisión de datos libre de colisiones. Por cuestiones de espacio no es posible describir en esta

contribución todos los detalles del funcionamiento del protocolo MAC. Los lectores interesados en ello deben referirse a [2].

3. DIÁLOGO DE CROSS-LAYER

Existen diversas estrategias en la literatura donde se realiza una gestión apropiada de la potencia consumida de las transmisiones radio, con el propósito de mitigar interferencias y ahorrar energía. Todas ellas se basan en modificar diferentes capas de los niveles del sistema de comunicaciones: algunos resultados muestran que aplazar las transmisiones cuando el canal presenta desvanecimientos mejora significativamente la efectividad del enlace [3]-[4]. Otras propuestas incluyen la adaptación de los parámetros de las transmisiones radio (longitud de trama, control de error, ecualización, etc.) para minimizar el consumo de energía de las baterías según las condiciones del canal [5].

Aquí se presenta un diálogo cross-layer (PHY-MAC) con el propósito de mejorar de forma global el rendimiento de la capacidad de transmisión en los sistemas WLAN al mismo tiempo que se reduce la energía consumida. Los paquetes de datos a ser transmitidos se organizarán según una lógica fuzzy.

En 0 se puede observar que el protocolo DQCA incluye las transmisiones requeridas de peticiones de acceso, las cuales se pueden usar para estimar de manera efectiva el estado del canal (capa física). DQCA intercambia en cada trama un paquete en el enlace de bajada que resulta de la unión de un paquete ACK y un CTS de 802.11. Este paquete puede contener información de cross-layer. De hecho, juntamente con el estado de peticiones de acceso, el paquete CTS puede incluir la selección apropiada del modo de transmisión de la capa física con tal de maximizar la capacidad de transmisión absoluta de la red. Esta selección se realiza basándose en el estado del canal detectado por la transmisión del paquete RTS. Se han definido un conjunto de umbrales en función de la relación Señal a Ruido, SNR (*Signal-to-Noise Ratio*), para seleccionar la velocidad de transmisión adecuada en la capa física. Estos umbrales han sido seleccionados basados en los resultados presentados en [7] y la extensión 802.11b del estándar WLAN. Los valores seleccionados se muestran en la Tabla 1.

Tabla 1. Umbrales según la SNR

Rate	1 Mbps	2 Mbps	5.5 Mbps	11 Mbps
SNR	<4 dB	4-7.5 dB	7.5-11 dB	>11 dB

4. MECANISMO DE AHORRO DE ENERGÍA BASADO EN LÓGICA FUZZY

La idea aquí introducida se basa en la predicción del futuro estado del canal entre todos los nodos y en aplicar técnicas de decisión de lógica difusa con la finalidad de seleccionar el nodo adecuado para transmitir y la velocidad de transmisión que más se adapte al estado del canal. Existen en la literatura esquemas de predicción de canal [5]-[6]. En particular, se ha utilizado el esquema en [6] por su simplicidad y precisión en el entorno estudiado.

El comportamiento de la red se puede observar como un sistema complejo no lineal. El siguiente mecanismo puede ser interpretado como un controlador lógico fuzzy (enfoque de *Control Fuzzy* [9]), porque las variables de entrada de las reglas del mecanismo son extraídas de un proceso dinámico. La variables de salida de las reglas permitirán la organización de los paquetes en las colas distribuidas de una manera más eficiente en cuanto al ahorro de energía y la capacidad de procesado teniendo en cuenta predicciones del canal. Estas variables de salida representan acciones de control en el sistema dinámico de control.

La selección de las variables de entrada, como también la formulación de las reglas depende en gran parte de la aplicación específica donde se trabaja.

En nuestro mecanismo basado en conceptos de cross-layer, cada estación hará uso de las siguientes variables de entrada, representadas gráficamente en la Figura 3 y la Figura 4:

- pTQ (posición de la estación en la cola DTQ)
- SNR predicho para la trama actual.
- SNR predicho para la siguiente trama (SNR_{fut})

Hay que resaltar que el indicador pTQ representa indirectamente el tiempo una estación lleva esperando en la Cola de Transmisiones de Datos, DTQ. Por lo tanto, cuanto más pequeño sea el valor de pTQ, más tiempo hará que esta estación lleva esperando en la cola de transmisiones.

La Figura 3 muestra una representación gráfica mediante lógica fuzzy de los valores reales de pTQ y sus correspondientes variables lingüísticas.

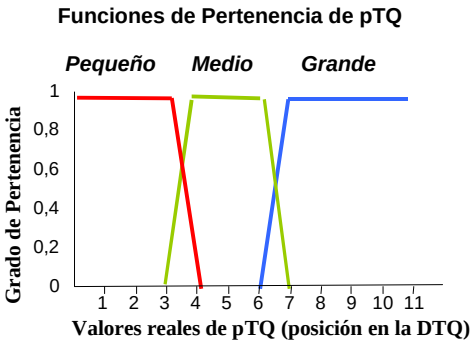


Figura 3.

Como ya se ha mencionado, el protocolo DQCA incluye la información necesaria para estimar el estado del canal de forma efectiva. Este concepto combinado con el algoritmo de predicción de canal en [6], permite el dialogo de cross-layer y por lo tanto la estimación de los valores de SNR. Para el algoritmo en cuestión aquí presentado, se usarán los valores predichos de SNR para la trama actual y para la siguiente trama. La Tabla 2

muestra la relación entre el conjunto de umbrales establecido para SNR basados en [7] y sus consiguientes variables lingüísticas.

Tabla 2. Términos lingüísticos para los umbrales de SNR

SNR (dB)	<4	4-7.5	7.5-11	>11
Términos Lingüísticos	Muy Deficiente	Deficiente	Eficiente	Muy Eficiente

La representación gráfica correspondiente se muestra en la Figura 4.

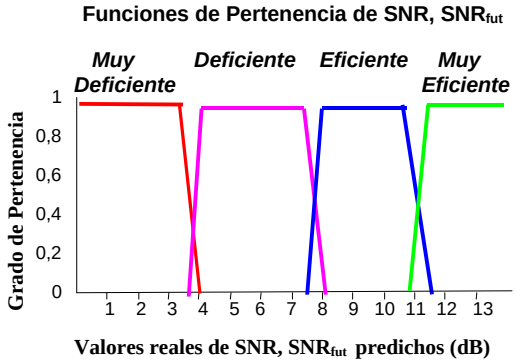


Figura 4.

Definimos la variable de salida pPQ como una variable lingüística, la cual puede tomar los valores lingüísticos aquí denominados "Primero", "Segundo" o "En cola" dependiendo de la posición actualizada en la nueva cola creada a partir del algoritmo de predicción de canal y la técnica cross-layer. La nueva cola se denomina Cola de Transmisiones por Prioridades, PTQ (Priority Transmission Queue, ver III.C). La representación en forma gráfica se muestra en la Figura 4.

Pasamos ahora a la descripción de las reglas fuzzy del mecanismo de cross-layer aquí estudiado, las cuales quedan resumidas en el tabla siguiente y extendidas mediante el diagrama de flujo de la Figura 5.

pTQ	SNR	SNR_{fu}	pPQ
Pequeño	ME o E	D o MD	**Primero**
Pequeño	D o MD	ME o E	**Segundo**
Mediano	ME	MD	**Primero**
Mediano	No ME	ME	**Segundo**
Grande	ME	MD	**Primero**
Grande	No ME	ME	**Segundo**

Donde E quiere decir Eficiente, ME Muy Eficiente, D es Deficiente y MD Muy Deficiente. Todos los otros resultados aquí no comentados tendrán como variable de salida pPQ="en Cola". A continuación, en el proceso de defuzzyficación, conjuntamente con la aplicación de las reglas DQCA adaptadas (ver sección C) se decidirá el valor real de pPQ. En concreto aquí se detallan las reglas más significativas.

1) SI pTQ="" **Y** SNR=("Muy Eficiente" **O** "Eficiente") **Y** SNR_{fut}=("Deficiente" **O** "Muy Deficiente")
ENTONCES pPQ="Primero".
Esta regla da la posibilidad de transmitir en primer lugar a uno de los tres primeros candidatos de la cola DTQ si la SNR predicha para la trama actual supera los 7.5 dB (ver Tabla 2) y el valor para el SNR de la siguiente trama se considera deficiente (menor que 7.5 dB). De esta manera, se garantiza un ahorro de energía en esta trama y en la próxima.

2) SI pTQ="Pequeño" **Y** SNR=("Deficiente" **O** "Muy Deficiente") **Y** SNR_{fut}=("Muy Eficiente" **O** "Eficiente")
ENTONCES pPQ="Segundo".
Ésta es la regla complementaria a la regla anterior. De los tres primeros candidatos de la cola DTQ, el que disponga del mejor valor predicho de la SNR para la siguiente trama (SNR_{fut}), y al mismo tiempo el SNR actual es deficiente, adquirirá la posición segunda en la nueva cola de transmisiones PTQ. De esta manera retrasamos una transmisión con un canal deficiente en la trama actual.

3) SI pTQ="Medio" **Y** SNR="Muy Eficiente" **Y** SNR_{fut}=("Muy Deficiente")
ENTONCES pPQ="Primero".
Esta regla implica que si no existe ninguna otra estación con un menor valor de pTQ que haya adquirido la primera posición en PTQ (su SNR no es suficientemente eficiente o su SNR_{fut} es suficientemente eficiente), esta estación ocupara dicha posición primera.

4) SI pTQ="Medio" **Y** SNR **NO**="Muy Eficiente" **Y** SNR_{fut}="Muy Eficiente"
ENTONCES pPQ="Segundo".
Esta regla es la complementaria a la regla anterior.

5) SI pTQ="Grande" **Y** SNR="Muy Eficiente"
Y SNR_{fut}=("Muy Deficiente")
ENTONCES pPQ="Primero".
Esta regla es prácticamente la misma regla que la regla 3). Se cumplirá si no existe ninguna otra estación con un valor menor de pTQ que satisfaga la regla 3.

6) SI pTQ="Grande" **Y** SNR **NO**="Muy Eficiente" **Y** SNR_{fut}="Muy Eficiente" **ENTONCES** pPQ="Segundo".
Esta regla es la complementaria la regla anterior.

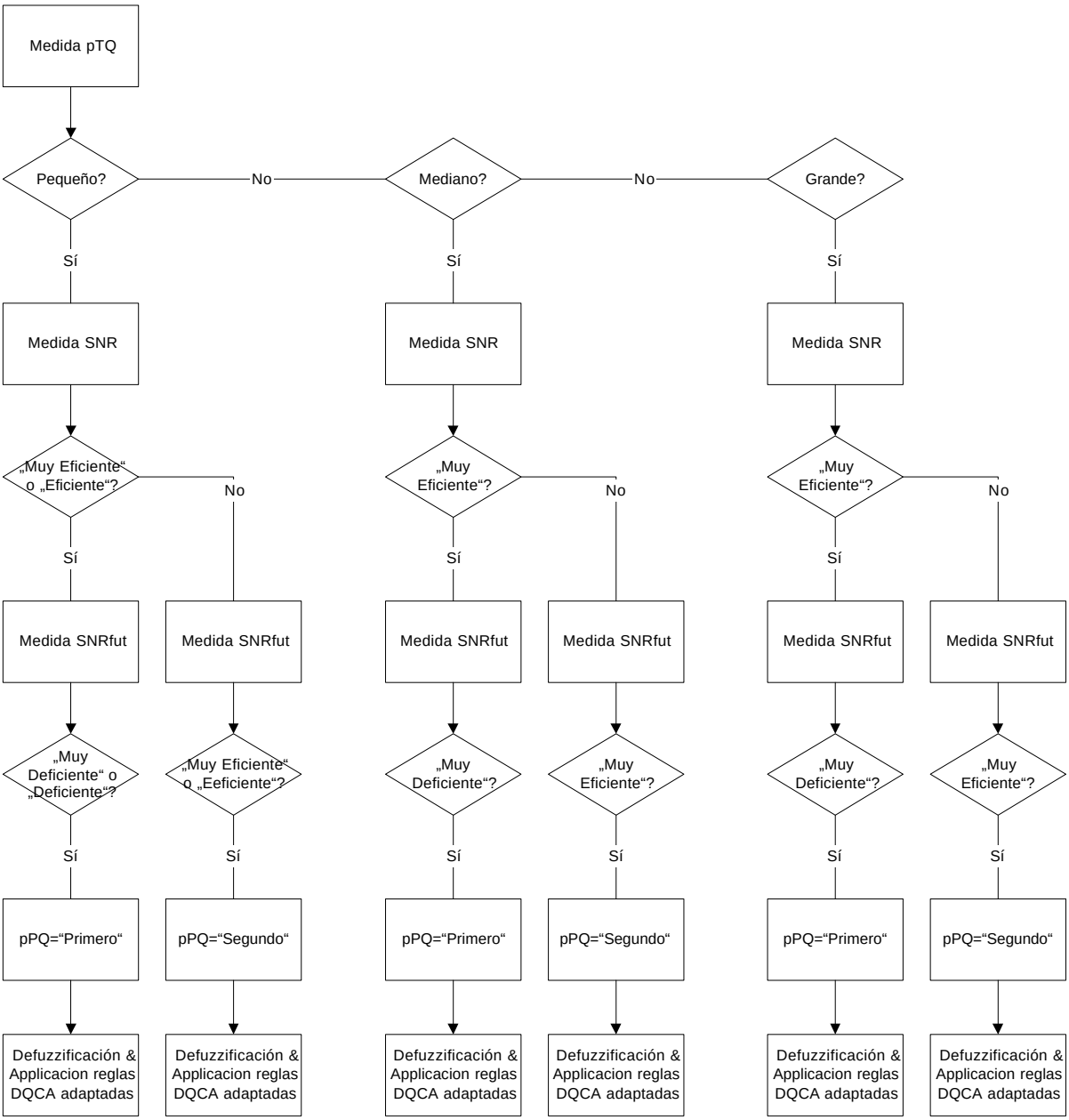


Figura 5. Diagrama de reglas fuzzy

Una vez analizado el mecanismo mediante reglas de lógica fuzzy, se presentan a continuación las nuevas reglas que el protocolo DQCA requiere para completar y adaptarse al concepto de cross-layer previamente descrito, conjuntamente con el proceso de defuzzyficación para obtener el valor real de pPQ.

5. REGLAS DE ADAPTACIÓN PARA DQCA

La principal incorporación al protocolo DQCA, tal y como se describe en 0 es la creación de la nueva cola distribuida denominada Cola de Transmisión por Prioridades, PTQ (Priority Transmisión Queue). Esta cola tiene el mismo número de elementos que la cola DTQ, pero en una ordenación diferente. De hecho, cada elemento de la cola PTQ tendrá el valor de la variable de salida de lógica fuzzy pPQ.

En cada trama, después de ejecutar las reglas de gestión de las colas (ver 0), cada nodo de la red debe ejecutar las reglas fuzzy y actualizar el valor de pPQ para cada uno de sus elementos. Entonces, la transmisión en la siguiente trama inmediata se realizará teniendo en cuenta el valor de pPQ de los paquetes y no el de pTQ como en la anterior versión del protocolo. En el caso de que más de un paquete adquiera el valor “Primero” en su variable pPQ, entonces para resolver este problema

transmitirá el paquete que disponga de pTQ menor. De la misma manera se tratará el semejante problema con el valor "Segundo" de pPQ.

El resto de las reglas de DQCA permanecen intactas, manteniendo la baja carga computacional del protocolo MAC y permitiendo una operación distribuida entre las estaciones a transmitir.

6. RESULTADOS DE LAS SIMULACIONES

Las simulaciones por ordenador verifican el buen comportamiento del mecanismo de cross-layer mediante reglas fuzzy. Se ha seleccionado un escenario con N estaciones activas permanentemente. Para la variación del canal radio se ha escogido un modelo de desvanecimientos de Rayleigh con la finalidad de evaluar la SNR mediante el esquema de predicción de canal escogido [6]. Para la el modo de operación MAC de 802.11 se supone detección de la SNR y adaptación del enlace ideal. La Figura 6 muestra la comparación obtenida de la capacidad de procesado versus la longitud de los paquetes para N=20 con dos diferentes técnicas; 802.11b MAC y el mecanismo aquí propuestos basado en reglas de lógica fuzzy.

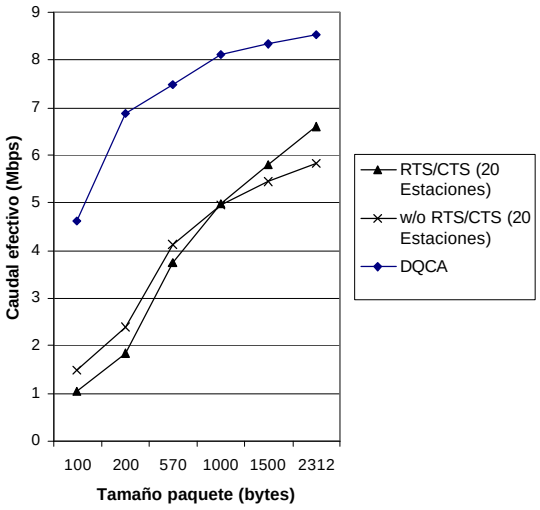


Figura 6. Caudal efectivo

Para los resultados aquí presentados se supone que el buffer del transmisor dispone siempre de paquetes a transmitir, es decir, se encuentra en saturación. En la figura se puede observar que la ganancia de la capacidad de procesado supera el 25% con respecto al 802.11b MAC.

La Figura 7 muestra la energía media consumida en la transmisión de cada paquete de datos en el mismo escenario. El mecanismo de cross-layer basado en reglas de lógica fuzzy adaptadas al protocolo DQCA organiza los paquetes a transmitir de manera inteligente con el propósito de ahorrar en el consumo de energía de una manera remarcable. Esta reducción alcanza valores de más de un 50% en ahorro de energía por paquete.

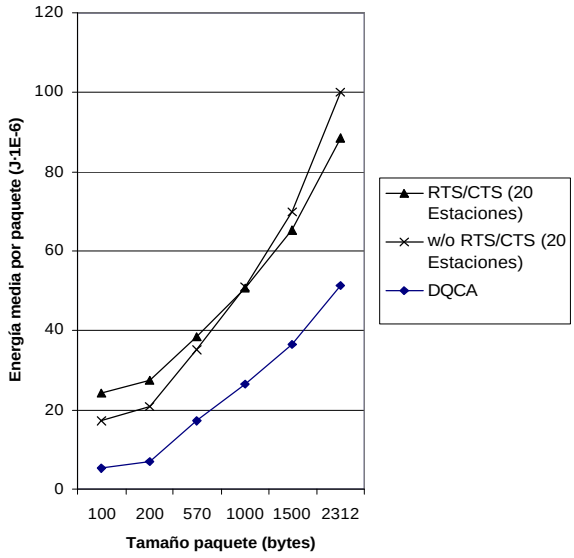


Figura 7. Energía Media Transmitida por Paquete

7. BENEFICIOS DEL MECANISMO BASADO EN LÓGICA FUZZY

Las ventajas de usar un mecanismo basado en lógica fuzzy son las siguientes:

1.La información de control *cross-layer* necesaria para aplicar las reglas fuzzy queda simplificada respecto a la que hacía falta en el caso del funcionamiento clásico del protocolo.

2. Se reduce de forma remarcable la carga computacional necesaria en cada una de las estaciones de transmisión para realizar el control distribuido del protocolo de acceso.

Bien es cierto que el algoritmo también presenta algún inconveniente como por ejemplo que la descentralización del protocolo MAC DQCA se ve parcialmente reducida al aplicar la reglas fuzzy debido a que se hace necesario el conocimiento por el AP de todos los estados de todas las estaciones activas en el sistema.

Sin embargo, se comprueba que se pueden alcanzar mejoras de rendimiento y de ahorro de energía muy significativas.

8. CONCLUSIONES

En esta contribución se ha presentado un nuevo mecanismo basado en técnicas cross-layer y en lógica fuzzy, que mejora de la eficiencia y obtiene un ahorro de energía para sistemas WLAN. Se utiliza un MAC estable que funciona a base de colas distribuidas. Se han realizado simulaciones por ordenador que han permitido verificar el correcto funcionamiento de la propuesta y los beneficios obtenidos. Los valores del

caudal (*throughput*) neto del sistema superan los obtenidos con el estándar IEEE 802.11 al mismo tiempo que se consigue una reducción del consumo de energía.

En concreto, en el escenario estudiado, el uso de la lógica fuzzy y el mecanismo de *cross-layer* mejora en un 25% el *throughput* obtenido y se reduce en un 50% la energía consumida respecto al uso de 802.11b.

Referencias

[1] L. Alonso, R. Ferrús, R. Agusti, "MAC-PHY Enhancement for 802.11b WLAN Systems via Cross-layering" *IEEE VTC'03 Fall, Orlando, October 2003.*

[2] L. Alonso, R. Agusti, O. Sallent, "A near-optimum MAC protocol based on the distributed queueing random access protocol (DQRAP) for a CDMA mobile communication system", *IEEE Journal on Selected Areas in Communications, Vol. 18, Issue 9, pp. 1701 –1718. Septiembre 2000.*

[3] B.R. Badrinath, P. Sudame, "To send or not to send: Implementing deferred transmissions in a mobile host", *Proc. 16th Int. Conf. Distributed Computing Systems, pp. 327-333, Hong-Kong, May 1996.*

[4] M. Zorzi, R.R. Rao, "Error Control and Energy Consumption in Communications for Nomadic Computing", *IEEE Transactions on Computing, Vol. 46, pp. 279-289, March 1997.*

[5] A. Arredondo, K.R Dandekar, Xu Guanghan, "Vector channel modeling and prediction for the improvement of downlink received power", *IEEE Transactions on Communications, Vol. 50, Iss. 7, pp. 1121-1129, July 2002.*

[6] T. Ekman, M. Sternad, M., A. Ahlen, "Unbiased power prediction of Rayleigh fading channels". *Proceedings. VTC 2002-Fall. 2002. Vol. 1, pp. 280-284, 24th-28th Sept. 2002.*

[7] J. Del Prado, S. Choi, "Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement", *ICC'03, pp. 1108-1113, Anchorage, Alaska, May 2003.*

[8] L. A. Zadeh, "Fuzzy sets," *Inform. Control, vol. 8, pp. 338-353, 1965.*

[9] E.H. Mandani and S.Assilian, "An experiment in linguistic synthesis with fuzzy logic controller," *Int. J. Man-Machine Sudies, no. 7, pp. 1-13, 1975.*

GARANTÍA DE QoS EN SISTEMAS DE COMUNICACIONES MÓVILES MEDIANTE AUTÓMATAS FUZZY

J.J. Astrain
Dpt. Matemática e Informática
Universidad Pública de Navarra
Campus de Arrosadía
31006 Pamplona (Spain)
josej.astrain@unavarra.es

J. Villadangos
Dpt. Automática y Computación
Universidad Pública de Navarra
Campus de Arrosadía
31006 Pamplona (Spain)
jesusv@unavarra.es

M. Castillo
Dpt. Matemática e Informática
Universidad Pública de Navarra
Campus de Arrosadía
31006 Pamplona (Spain)
maria.castillo@unavarra.es

Resumen

Se presenta una solución basada en el uso de autómatas fuzzy para garantizar calidad de servicio (QoS) en sistemas de comunicaciones móviles. La información de localización de un terminal móvil, obtenida mediante triangulación de potencias a lo largo de su desplazamiento dentro de una celda, se codifica en términos de símbolos fuzzy que tienen en cuenta los problemas de atenuación y propagación multitrayecto propios de estos sistemas. Con la ayuda de autómatas fuzzy se computa la similaridad de la trayectoria seguida por el terminal con los desplazamientos más frecuentes para esa celda que se encuentran contenidos en un diccionario.

Palabras Clave: Autómata fuzzy, similaridad fuzzy, QoS, comunicaciones móviles.

1. INTRODUCCIÓN

El crecimiento exponencial experimentado por los sistemas de comunicaciones móviles en cuanto a tamaño y número de usuarios ha motivado el desarrollo de tecnologías como GSM, GPRS y más recientemente UMTS, también conocida como telefonía de tercera generación (3G). Estos sistemas precisan conocer la localización del terminal móvil (el teléfono móvil del abonado), y para ello se emplea tiempo, potencia y recursos para localizar al usuario (*paging*), lo que en ciertos casos supone un coste importante para el sistema.

Una de las principales dificultades que surgen en los sistemas de comunicaciones móviles actuales es la garantía de la calidad de servicio (*Quality of Service*, QoS) ofrecida por dichos sistemas. Esto se debe a la naturaleza móvil de los terminales de dichos sistemas de comunicación. Su idiosincrasia hace que sea realmente complejo controlar la localización de todos los terminales que forman parte de dicho sistema de comunicaciones. Sin conocer la localización de éstos, es realmente complicado poder garantizar QoS, lo cual implica una degradación de la calidad del servicio. Esta problemática se manifiesta en los sistemas de telefonía móvil en el instante en el que un teléfono móvil con una llamada en curso realiza la transición de una unidad celular a otra adyacente. Si en la celda vecina de destino se había realizado un reserva previa, la llamada en curso no se verá afectada por el desplazamiento del terminal, pero si la reserva previa no se había efectuado y la celda de destino estuviese saturada (no se dispone de canales de comunicación libres), se produciría una interrupción de la comunicación por no disponerse de los recursos necesarios en la celda de destino. Una transmisión no puede interrumpirse por la falta de previsión a la hora de reservar ranuras de tiempo (o *time slots*) en las estaciones base al producirse un cambio de celda durante el desplazamiento del usuario. Es en este punto donde la localización del terminal se convierte en elemento fundamental para poder garantizar QoS en entornos móviles distribuidos.

Con el fin de evitar estas situaciones, se propone emplear un sistema de localización de un terminal móvil en cualquier instante, de modo que sea posible efectuar una reserva de recursos en la celda de destino con la suficiente antelación para garantizar la correcta conclusión de la comunicación sin que se produzcan interrupciones no deseadas del servicio. Para ello, se propone el empleo de autómatas fuzzy, mediante los cuales se logra minimizar el flujo de mensajes de localización entre el terminal móvil del usuario (TM) y la estación base (EB), lo que permite minimizar el consumo de batería por parte del TM, ahorrar mensajes de señalización y realizar una pre-reserva de recursos que garantice la calidad de servicio requerida por los usuarios de estos sistemas de comunicación.

En el estándar GSM, al proceso de negociación que se

produce entre el TM y la EB cuando el terminal se dispone a cruzar la frontera entre dos celdas vecinas se le denomina *hand-off*. En este proceso, se define una banda o franja en el borde de la celda de modo que cuando un terminal se adentra en dicha franja se realiza una pre-reserva de los recursos que garanticen la continuación de la llamada en curso en el momento en el que se produzca el cambio de celda.

Hasta la fecha, diversos autores han tratado de alcanzar este objetivo de localización, planteando un modelo de movilidad del TM que permita predecir su desplazamiento [1, 3, 12, 2], eligiendo distintas estrategias de ahorro de recursos [6, 9] o buscando patrones de movilidad [5, 10]. Aunque en todos ellos se proponen estrategias para minimizar el consumo de recursos y la ocupación del canal de señalización, la forma de abordar este problema es radicalmente opuesta en este trabajo, puesto que aquí se propone el empleo de técnicas de computación de medidas de similaridad mediante autómatas *fuzzy*. Se trata de un sistema de predicción de trayectorias que permite realizar la pre-reserva de recursos de la forma más eficiente posible, cuyo objetivo último es conseguir una mayor QoS. Tan pronto como el autómata fuzzy determina el trayecto seguido por el TM, la EB puede iniciar las tareas de reserva de recursos en la celda de destino. De esta forma, se puede garantizar la QoS requerida por la comunicación en curso.

La mayor parte de trabajos relacionados con la gestión de QoS en sistemas de comunicaciones móviles proponen una gestión de recursos de tipo probabilístico, modelando el movimiento de los terminales mediante distribuciones probabilísticas y simulando el sistema celular y procedimientos de *hand-off* mediante cadenas de Markov. La aportación de la utilización de patrones para la predicción de las trayectorias de los terminales móviles se va a estudiar sobre la optimización del mecanismo de reserva variable.

El resto del trabajo se organiza del siguiente modo: la sección 2 presenta la arquitectura del sistema propuesto, explicando la codificación de movimientos, el autómata fuzzy que constituye el módulo de predicción de desplazamiento y el funcionamiento del sistema. La sección 3 aborda el modo en el que se puede garantizar calidad de servicio en sistemas de comunicaciones móviles con la ayuda de la técnica que se propone. Finalmente, las conclusiones y referencias bibliográficas ponen el punto final a este trabajo.

2. ARQUITECTURA DEL SISTEMA

Cada EB del sistema de comunicación celular construye y almacena su propio diccionario de trayectos (diccionario de unidad celular), que incluye las trayectorias más frecuentemente seguidas por los usuarios que atraviesan dicha celda. Estos diccionarios pueden construirse mediante una fase de seguimiento de los TMs que atraviesan cada celda, y actualizarse tantas veces como sea oportuno. Pero además, cada usuario puede poseer su propio diccionario de abonado con sus trayectos más frecuentes. Como es lógico, un usuario sólo dispondrá de diccionario de abonado para aquellas celdas que suele atravesar o habitar con frecuencia, siendo esta cualidad de frecuencia fijada por el sistema en la etapa de seguimiento.

Los diccionarios de abonado se almacenan en la tarjeta SIM del usuario o en su TM si la capacidad de la SIM no es suficiente. El tamaño medio de un diccionario de abonado es aproximadamente 500 bytes, por lo que las tarjetas actuales de 128K tienen capacidad para albergar un número importante de estos diccionarios. Para cada celda, el TM construye un diccionario híbrido que resulta de unir el diccionario de unidad celular y el diccionario de usuario para una determinada celda, pero sin duplicar aquellas rutas que pudieran pertenecer a la intersección de ambos diccionarios.

El TM es el encargado de procesar la información de posicionamiento (el patrón de movimiento seguido) y compararla con el diccionario correspondiente. Si el TM está apagado, será en el proceso de inicialización del terminal cuando la EB le enviará el diccionario de la unidad celular en la que se encuentre. Si por el contrario el TM está ya conectado al sistema, tanto en estado pasivo como con una conversación en curso, la EB enviará el diccionario inmediatamente después del proceso de *hand-off*.

Tan pronto como el TM sea capaz de determinar el patrón de desplazamiento que está siguiendo con un cierto grado de certeza (que puede definirse como un parámetro del sistema), lo notificará a su EB para negociar las condiciones de traspaso con la *EB de destino* y realizar la reserva de recursos que sea necesaria en cada momento, así como gestionar y negociar los recursos en el proceso de *hand-off* en el momento adecuado. Si el TM no consigue determinar la trayectoria del terminal, el *hand-off* se realizará de igual forma que se llevaba a cabo en ausencia de mecanismos de predicción de trayectoria. La figura 1 ilustra el modo de operación del sistema propuesto.

La figura 2 muestra la arquitectura del sistema. Como se puede apreciar, la información de localización es obtenida por el TM por triangulación de potencias de las portadoras de las distintas EB que rodean al TM (ver figura 3). Con esta información de localización el TM construye cadenas de símbolos fuzzy que son procesadas por un subsistema de gestión del movimiento que es el encargado de predecir la trayectoria seguida

por el TM, que posteriormente será notificada a la EB correspondiente para tomar las decisiones de reserva y encaminamiento que procedan.

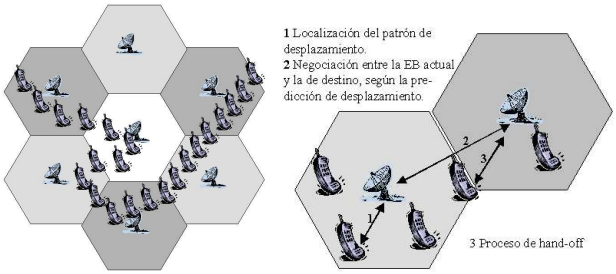


Figura 1: Modo de funcionamiento.

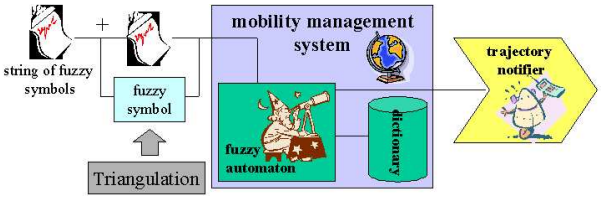


Figura 2: Arquitectura del sistema.

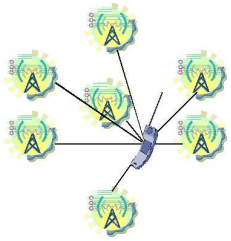


Figura 3: Triangulación de potencia en el TM.

2.1. CODIFICACIÓN DE MOVIMIENTOS

El movimiento del TM se modela siguiendo tres esquemas de codificación diferentes, en los que se emplean caracteres alfabéticos para representar la localización y el movimiento seguido por el TM. Para evaluar de qué modo determina la codificación de la información de localización los resultados finales obtenidos, se consideran tres esquemas diferentes para describir los trayectos seguidos por el TM. El primero de ellos considera la división de la celda en zonas rectangulares no solapadas entre sí, cada una de ellas identificadas por un símbolo alfabético diferente. El segundo esquema considera la división sectorial de la celda sin solapamientos entre zonas. El tercer esquema considera la k-conectividad, empleándose ocho caracteres ($k = 8$) para representar las distintas trayectorias que pueden ser descritas por un TM en su trayectoria de desplazamiento a través de la celda. La figura 4 ilustra los tres esquemas de codificación descritos: cuadricular (cod 1), sectorial (cod 2) y 8-conectividad (cod 3).

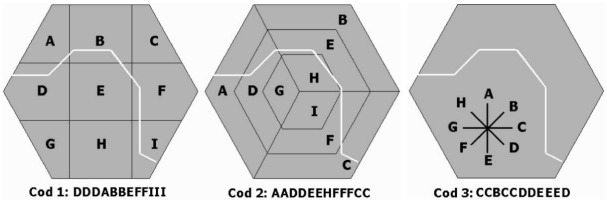


Figura 4: Esquemas de codificación de las trayectorias.

A continuación se estudia el comportamiento de los tres esquemas de codificación cuando existe un conjunto de caminos que tienen un número reducido de posiciones en común. Los caminos pueden poseer distintas longitudes, pero el número de divisiones celulares es el mismo. Se compara la codificación midiendo el tiempo medio que se precisa para determinar el camino seguido por el TM sometido a estudio. Los resultados obtenidos se presentan en la figura 5 (izda).

La figura 5 (izda) muestra la gran influencia, en términos de discriminación de caminos, de la codificación seleccionada. De hecho, la codificación 1 precisa en este caso el menor tiempo medio para determinar el camino seguido. Existe un efecto muy interesante cuando se emplean las divisiones celulares para definir las posiciones de los TMs (este es el caso de las codificaciones 1 y 2). La figura 5 (dcha) presenta los resultados obtenidos al incrementar el número de zonas de localización. En este caso, se divide la celda entre 1 y 25 zonas, observándose que conforme aumenta el número de zonas, disminuye el tiempo medio necesario para determinar el camino seguido por un TM dentro de una celda. Esta figura puede interpretarse como el tiempo estimado que precisa un TM para determinar qué camino está siguiendo, procediendo entonces a notificar a su estación base (BS) cuál es la trayectoria seguida y realizar una reserva de recursos en la celda de destino. Además, basándose en la velocidad seguida por el TM, se puede estimar el instante en el que se producirá el *hand-off*. El caso en el que se tiene una única zona por celda corresponde al tratado en la mayoría de los trabajos de la materia, en los que, el movimiento del TM se modela con representaciones lineales.

2.2. AUTÓMATA FUZZY

La herramienta fuzzy empleada en este trabajo es un autómata fuzzy que es capaz de recuperar errores de edición, de tal modo que mide la distancia entre una cadena observada de símbolos y una cadena patrón.

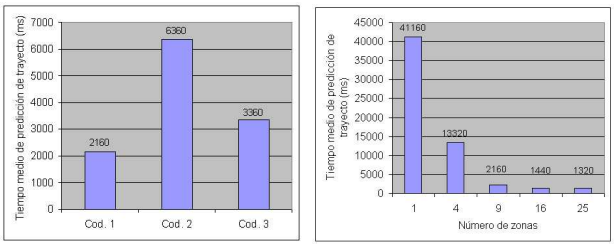


Figura 5: Tiempos medios de detección de trayectoria para diferentes tipos de codificación con división en 9 zonas para los tipos cuadricular y sectorial (izda) y para diferentes longitudes de código (zonas en celda) en una codificación cuadricular (dcha).

Estas cadenas patrón están contenidas en un diccionario. Para modelar las citadas operaciones de edición (inserción, borrado y cambio) es preciso trabajar con la cadena vacía, que se denotará ε. Este elemento permite realizar transiciones entre dos estados del autómata sin consumir símbolo en la entrada. Así, se puede entender el borrado de un símbolo como la sustitución de dicho símbolo por la cadena vacía, es decir, el nuevo autómata consume un símbolo de entrada sin cambiar de estado. Del mismo modo, para la operación de inserción se produce un cambio en el estado del autómata sin consumir símbolo alguno de la cadena de entrada. Esto equivale a sustituir la cadena vacía por un símbolo del alfabeto.

Este autómata, que será capaz de tomar como entrada una cadena imperfecta de símbolos, se denomina $MF(\omega)$. El autómata *fuzzy* $MF(\omega) = (Q, \Sigma, \mu, \sigma, \eta)$ [7, 11, 4], cuyos estados son de naturaleza *fuzzy* y las operaciones realizadas sobre éstos son también operaciones de tipo *fuzzy*, viene definido como:

- $Q = \{q_0, q_1, \ldots, q_n\}$ es el conjunto de estados alcanzables por el autómata, siendo n la longitud de la cadena patrón ω.
- q_0 y q_n son los estados inicial y final del autómata respectivamente.
- Σ es el alfabeto.
- La función de transición $\delta : Q \times Q \times \Sigma \longrightarrow \{0, 1\}$ se define como sigue:
  a) $\delta(q_{k-1}, q_k, a_k) = 1$ $(1 \leq k \leq n)$ donde a_k es el k-ésimo símbolo de la cadena ω;
  b) El resto de valores $\delta(\cdot)$ tienen un valor nulo.
- $\sigma : Q \longrightarrow [0, 1]$ es el estado *fuzzy* inicial, siendo $\sigma(q_0) = 1$ y $\sigma(q_k) = 0$ $(0 < k \leq n)$.
- $\eta : Q \longrightarrow [0, 1]$ es el estado *fuzzy* final, siendo $\eta(q_n) = 1$ y $\eta(q_k) = 0$ $(0 \leq k < n)$.
- La función de transición *fuzzy* $\mu : Q \times Q \times (\Sigma \cup \{\varepsilon\}) \longrightarrow [0, 1]$ se define del siguiente modo:
  (i) Si $\delta(q, p, a) = 1$, $q, p \in Q$, $a \in \Sigma$, entonces
  (i.a) $\mu(q, p, a) = 1$ (la transición original).
  (i.b) $\mu(q, p, \varepsilon) \in [0, 1]$ (inserción del símbolo a).
  (i.c) $\forall\, x \in \Sigma : \mu(q, p, x) \in [0, 1]$ (sustitución del símbolo x por a).
  (ii) $\forall\, q \in Q$, $\forall\, x \in \Sigma : \mu(q, q, x) \in [0, 1]$ (borrado del símbolo x).
  (iii) El resto de valores para las transiciones *fuzzy* es 0.

Sea $\omega = a_1 a_2 \ldots a_n$ la cadena patrón y $\alpha = x_1 \ldots x_{k-1}\, x\, x_{k+1} \ldots x_m$ la cadena observada. La transición *fuzzy* definida $\mu(q, p, x)$ puede interpretarse del siguiente modo:

• Si $x = a_k$, no se ha producido ningún error; el símbolo de entrada es el esperado, luego el valor asociado a la transición es 1 ($\mu(q, p, a_k) = 1$).
• Si $x \neq a_k$, la transición $\mu(q, p, x)$ simula la sustitución del símbolo x por a_k con vistas a recuperar la sustitución de a_k por x en la cadena observada α. El valor asociado a la transición es $\mu_{c_{xa}^{q_{k-1}q_k}}$ que representa el coste asociado a la operación de sustitución del símbolo x por el símbolo a en la k-ésima posición de la cadena observada.
• Si $x = \varepsilon$, la transición $\mu(q, p, \varepsilon)$ simula la inserción del símbolo a_k con vistas a recuperar un posible borrado de este símbolo en la cadena α. El símbolo a_k está unívocamente definido por el par de estados q, p, tal y como se deriva de la definición de $M(\omega)$. El valor asociado a la transición es $\mu_{i_a^{q_{k-1}q_k}}$ que representa el coste asociado a la operación de inserción del símbolo a en la k-ésima posición de la cadena observada.
• La transición $\mu(q, q, x)$ simula el borrado del símbolo x con vistas a recuperar la inserción de x en la cadena observada α y el valor asociado a la transición es $\mu_{d_x^{q_{k-1}}}$, que representa el coste asociado a la operación de borrado del símbolo x en la k-ésima posición de la cadena observada.

Como puede apreciarse en la definición de la función de transición μ, no se permite la sustitución de la cadena vacía por sí misma. En el autómata $MF(\omega)$, los valores $\mu_{i_a^{qp}}, \mu_{c_{xa}^{qp}}, \mu_{d_x^q} \in [0, 1]$, $\forall\, x, a \in \Sigma$, $\forall q, p \in Q$, determinan el valor de similaridad que $MF(\omega)$ puede alcanzar para la cadena observada α. Como se puede observar, el uso de la cadena vacía permite modelar la inserción del símbolo a en la k-ésima posición de la cadena α realizando una transición en el autómata sin consumir símbolo de entrada lo que facilita modelar la operación de inserción (i_a^k): transición $\mu(q, p, \varepsilon)$ con valor de pertenencia $\mu_{i_a^{qp}}$.

La función $\mu^* : \mathcal{F}(Q) \times \Sigma^* \longrightarrow \mathcal{F}(Q)$ que determina el comportamiento del autómata *fuzzy* para una cadena observada (α) se define como sigue:

i) $\mu^*(\tilde{P}, \varepsilon) = \mu^{\varepsilon}(\tilde{P})$, $\tilde{P} \in \mathcal{F}(Q)$

ii) $\mu^*(\tilde{P}, \alpha x) = \mu^{\varepsilon}(\hat{\mu}(\mu^*(\tilde{P}, \alpha), x))$, $\tilde{P} \in \mathcal{F}(Q)$, $x \in \Sigma$, $\alpha \in \Sigma^*$.

Donde $\mathcal{F}(Q)$ es el conjunto de posibles estados *fuzzy* definidos sobre el universo Q y la función $\hat{\mu} : \mathcal{F}(Q) \times \Sigma \longrightarrow \mathcal{F}(Q)$ se define como: $\hat{\mu}(\tilde{P}, x) = \{(p, \mu) \mid \mu = \oplus_{q \in Q}(\mu(q, p, x) \otimes \mu_{\tilde{P}}(q)), p \in Q\}$, con $\tilde{P} \in \mathcal{F}(Q), x \in \Sigma$ siendo los operadores $\oplus$ y $\otimes$ la t-conorma y la t-norma respectivamente. Y $\mu^{\varepsilon}(\tilde{P}) = \{(p, \mu) \mid \mu = \mu_{\tilde{P}}(p) \oplus (\oplus_{q \in Q}(\mu_{\tilde{E}(q)}(p) \otimes \mu_{\tilde{P}}(q)))\ p \in Q\}$, con $\tilde{P} \in \mathcal{F}(Q)$ y siendo $\tilde{E}(q)$ el conjunto *fuzzy* de $\mathcal{F}(Q)$ de estados alcanzables aplicando repetidamente transiciones por cadena vacía desde el estado q (más información en [4]).

3. GESTIÓN DE QoS

Se considera que el autómata *fuzzy* detecta un trayecto cuando la distancia existente entre el trayecto con el valor de pertenencia más alto y los valores de pertenencia del resto de candidatos es superior a 10^5 (es decir, es mayor en cinco órdenes de magnitud) y el valor de pertenencia decrece lentamente.

Tan pronto como el autómata determina correctamente el trayecto seguido por el terminal, la estación base puede iniciar las tareas de reserva de recursos en la celda de destino. De esta forma se puede garantizar la QoS requerida por la comunicación en curso.

El mecanismo de reserva variable está directamente ligado a la distancia óptima de reserva, que delimita la zona en la que puede solicitarse la reserva de recursos (ver figura 6).

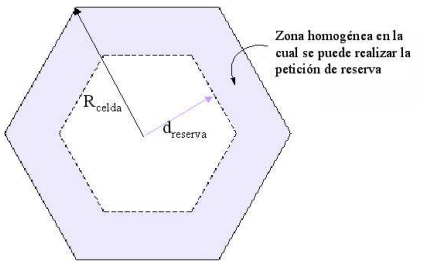


Figura 6: Mecanismo de reserva variable, distancia de reserva.

Los resultados a este respecto obtenidos en [8] muestran la evolución de las prestaciones del sistema, expresadas en términos de probabilidad de bloqueo, probabilidad de interrupción y Grado de Servicio (donde $GoS = P_{bloqueo} + a \cdot P_{interrupcion}$) en función de la distancia de reservas (d). La probabilidad de bloqueo decrece a medida que d disminuye. Esto se debe a que cada vez se reservan canales durante un espacio más corto de tiempo, pudiéndose utilizar para cursar llamadas generadas en la propia celda. Por otra parte, se tiene que la probabilidad de interrupción, también para distancias decrecientes, es cada vez más importante. La QoS garantizada dependerá de varios factores: la velocidad media de los usuarios, el factor de penalización para las llamadas interrumpidas y el tráfico ofrecido.

La utilización del autómata *fuzzy* en la predicción de trayectoria permite, en la mayoría de las situaciones, fijar la distancia de reserva en su valor óptimo, ya que al conocer con anterioridad el trayecto que sigue el terminal y por tanto la celda destino del mismo, se puede solicitar la reserva de recursos en el momento más conveniente, adaptándose a la situación del sistema en ese instante (tráfico ofrecido, velocidad del terminal...).

A medida que aumenta la velocidad del TM, es más eficiente realizar la reserva de canal con anterioridad. Por tanto, el hecho de conocer *a priori* la velocidad media de los usuarios en la celda y el tráfico ofrecido es básico de cara a fijar un umbral d lo más cercano posible al óptimo. Es en este punto en el que la información suministrada por el autómata puede optimizar el mecanismo de reserva variable.

En cuanto a las prestaciones obtenidas al variar el tamaño de la celda, al reducir el tamaño de la celda para un mismo valor de velocidad, la tasa media de traspasos se incrementa de forma notable. Ello produce que la QoS obtenida para una celda de 1000 m de longitud no sea tan buena como la obtenido con tamaños de celda más grandes. Existe una dependencia entre la relación *velocidad/longitud* de la celda y la distancia óptima de reservas. Como el proceso de localización ofrece la situación geográfica en la que se encuentra el terminal y puesto que dicho proceso se realiza en intervalos conocidos, se puede determinar fácilmente la velocidad del terminal. Por lo tanto, la utilización de un mecanismo de predicción de trayectoria permite comunicar al sistema la próxima celda destino y el momento en el que el TM solicitará los recursos a la celda vecina, permitiendo al sistema determinar en qué momento deberá solicitar una reserva de recursos en la celda destino.

Además, en aquellos sistemas en los que el TM sea el encargado de realizar el proceso de actualización de localización y la posterior predicción de trayectoria, se pueden implementar procedimientos de traspaso en los que la decisión pueda ser tomada por el propio termi-

nal móvil, el cual puede negociar directamente con la nueva estación base la realización del procedimiento. De esta manera, puede ser el propio TM el encargado de solicitar y anular la reserva de canales en función no sólo de la calidad del enlace radio, sino de los resultados obtenidos en la predicción de trayectoria. A su vez, la posibilidad de predecir la movilidad de los usuarios posibilita una implementación más eficiente del mecanismo de reserva variable al aumentar la proporción de reservas de recursos que finalmente son utilizadas y minimizar los efectos sobre la probabilidad de bloqueo.

4. CONCLUSIONES

El uso de técnicas *fuzzy* permite mejorar la QoS en sistemas de comunicaciones móviles considerando que se pueden emplear autómatas *fuzzy* para predecir los trayectos seguidos por los TMs durante su desplazamiento a través de las celdas que constituyen el sistema celular.

El desplazamiento del terminal móvil se plasma en una serie de símbolos que constituyen el alfabeto sobre el que trabajan los autómatas *fuzzy*. Los distintos trayectos seguidos por los terminales, caracterizados como cadenas de símbolos, son comparados con los trayectos más frecuentemente seguidos por el terminal móvil en cuestión o por el resto de terminales en la celda objeto de estudio, de tal modo que los autómatas *fuzzy* permiten realizar una discriminación de trayectos posibilitando así la mejora de la QoS ofrecida por el sistema de comunicación al poder realizarse una reserva de recursos en la celda de destino del TM antes de que éste llegue a la frontera con la celda de destino.

Se han propuesto tres esquemas diferentes de codificación de los desplazamientos: división lógica de la celda en distintas zonas o sectores (cuadricular y sectorial) y 8-conectividad. Se ha estudiado el comportamiento del autómata *fuzzy* propuesto con estos esquemas de codificación, analizando el tiempo medio requerido para estimar correctamente el desplazamiento seguido por el terminal, observándose que el tiempo invertido para la estimación decrece conforme aumenta el número de divisiones o sectores, y que la codificación cuadricular es la que obtiene un mejor comportamiento.

El autómata *fuzzy* puede trabajar, mediante cadenas imperfectas de símbolos, con la vaguedad introducida en la localización del terminal móvil. Esta vaguedad es característica de los sistemas de comunicaciones celulares, ya que las atenuaciones en la propagación de la señal y la propagación multitrayecto, ambos fenómenos estrechamente ligados a estos sistemas de comunicación, complican en gran manera el proceso de localización de un terminal móvil mediante la triangulación de potencias.

Referencias

[1] I.F. Akyldiz y J.S.M. Ho. Dynamic mobile user location update for wireless PCS networks. *Wireless networks*, 1:187–196, 1995.

[2] A. Bhattacharya y S.K. Das. LeZi-Update: And Information-Theoretic Approach to Track Mobile Users in PCS Networks. *Proceedings Mobicom '99*, Seattle Washington USA, 2000.

[3] Y. Birk y Yaron Nachman. Using direction and elapsed-time information to reduce the wireless cost of locating mobile units in cellular networks. *Wireless networks*, 1(4):403–412, 1995.

[4] J.R. Garitagoitia, J.R. González de Mendívil, J. Echanobe, J.J. Astrain y F. Fariña. Deformed Fuzzy Automata for Correcting Imperfect Strings of Fuzzy Symbols. *IEEE Transactions on Fuzzy Systems*, 11(3):299–310, 2003.

[5] C.R. Lin y M. Gerla. Adaptative Clustering for Mobile Wireless Networks. *Journal on Selected Areas in Communications*, 15(7), 1997.

[6] U. Madhow, M.L. Honig y K. Steiglitz. Optimization of Wireless Resources for Personal Communications Mobility Tracking. *IEEE/ACM Transactions on Networking*, 3(6), 1995.

[7] M. Mizumoto, J. Toyoda y K. Tanaka. Fuzzy languages. *System Computers Control*, 1(3):36–43, 1970.

[8] M. Oliver i Riera. *Mecanismos de Gestión de Recursos basados en criterios de Calidad de Servicio para una Red de Comunicaciones Móviles de Tercera Generación.* Tesis doctoral, Universitat Politécnica de Catalunya, Barcelona, España, Diciembre 1998.

[9] R. Rezaiifar y A.M. Makowski. From Optimal Search Theory to Sequential Paging in Cellular Networks. *Journal on selected areas in communications*, 15(7), 1997.

[10] X. Shen, J.W. Mark y J. Ye. User mobility profile prediction: An adaptative fuzzy inference approach. *Wireless networks*, 6:363–374, 2000.

[11] M.G. Thomason. Finite fuzzy automata, regular fuzzy languages and pattern recognition. *Pattern Recognition*, 5:383–390, 1973.

[12] M.M. Zonoozi y P. Dassanayake. User Mobility modeling and Characterization of Mobility Patterns. *IEEE Journal on Selected Areas in Communications*, 15(7), 1997.

SISTEMAS DIFUSOS PARA CONTROL DE CONGESTIÓN Y CALIDAD DE SERVICIO EN INTERNET

Federico Montesino[1] **Diego López**[2] **Ángel Barriga**[1] **Santiago Sánchez-Solano**[1]

[1]Instituto de Microelectrónica de Sevilla. Centro Nacional de Microelectrónica – CSIC
Avda. Reina Mercedes, s/n. Edif. CICA. E–41012, Sevilla
{Federico.Montesino,Angel.Barriga,Santiago.Sanchez}@imse.cnm.es

[2]RedIRIS, Red Española de I+D
Avda. Reina Mercedes, s/n. Edif. CICA. E–41012, Sevilla
Diego.Lopez@rediris.es

Resumen

Los nuevos servicios, aplicaciones y tecnologías surgidos en Internet plantean serios problemas de gestión y optimización del uso del ancho de banda y otros recursos de la red. Se propone el uso de sistemas difusos como nuevo enfoque de análisis de los principales problemas y retos actuales en el diseño de mecanismos de control de congestión y calidad de servicio en Internet. En esta presentación se resumen tanto casos específicos de aplicación de este nuevo enfoque como futuros campos de aplicación, considerando el modelado de la red, los sistemas de medida y evaluación de prestaciones, y los sistemas de control de congestión y calidad de servicio.

Palabras Clave: Control de congestión, Lógica difusa, Calidad de servicio, Internet, TCP/IP

1 INTRODUCCIÓN

Los nuevos servicios y tecnologías surgidos durante los últimos años en Internet plantean serios problemas para los mecanismos de control de congestión y gestión del ancho de banda y otros recursos de la red desarrollados hasta la fecha [5]. Es más, el escaso conocimiento que se tiene sobre la dinámica del tráfico de Internet, así como la falta de enfoques de análisis y técnicas de ingeniería del tráfico en redes IP, suponen una importante limitación para la estabilidad y seguridad global de la red, el despliegue de nuevos servicios y el uso eficiente de nuevas tecnologías de comunicaciones.

Actualmente, existe un convencimiento generalizado acerca de la necesidad de modelos y herramientas con mayor base teórica y experimental que las disponibles para el estudio de la dinámica del tráfico en Internet [1, 8], así como la importancia de desarrollar e incorporar esquemas activos de gestión de colas en los nodos intermedios de la red [6, 2]. Igual relevancia tiene el perfeccionamiento de los mecanismos de control de congestión desde los nodos extremos.

Asimismo, los requisitos de calidad de servicio y patrones de tráfico asociados a tipos de servicios y aplicaciones emergentes (como los nuevos protocolos de transporte y señalización, las aplicaciones multimedia de tiempo real o streaming, o las peer-to-peer), implican cambios profundos en los esquemas actuales de control de congestión. Los nuevos esquemas desarrollados han de ser adaptables además a condicionantes tecnológicos tales como el alto incremento del ancho de banda disponible y las características de diversos tipos de enlaces físicos, como los inalámbricos.

Dadas las características de los problemas que plantean estos sistemas (complejidad, no linealidad, utilidad de la aplicación de varios criterios simultáneamente, y disponibilidad de información incompleta e imprecisa), una de las alternativas para su estudio es el empleo de sistemas inteligentes basados en lógica difusa, sin excluir otras técnicas de soft computing.

Son ya conocidas y existen estudios amplios sobre las aportaciones de los sistemas basados en soft computing en áreas de investigación en Internet como la web semántica o esquemas de routing, pero en los niveles de red y transporte, y en particular en el estudio de la dinámica del tráfico de paquetes, las propuestas y resultados existentes son muy puntuales y han sido validadas sólo parcialmente mediante simulación.

Los sistemas difusos son especialmente adecuados para describir y analizar problemas complejos, y destacan por su robustez y capacidad de adaptación y aprendizaje, factores clave en los problemas que tratamos, para los cuales no parece factible una solución analítica. Asimismo, hacen posible el empleo de técnicas de ajuste mediante métodos formales, bien sean específicos, o bien basados en redes neuronales

o algoritmos genéticos. Por último, las implementaciones hardware de los sistemas basados en lógica difusa presentan la rapidez de inferencia necesaria para operar a altas velocidades.

En las siguientes secciones se irá detallando el conjunto de areas de investigación en aplicaciones de la lógica difusa al desarrollo de modelos, estrategias y sistemas de control de congestión en Internet. Presentaremos casos concretos de aplicación desde un punto de vista más general con el que pretendemos proponer un nuevo enfoque que aproveche, entre otras ventajas, los procedimientos formales y automáticos de diseño, aprendizaje y optimización conocidos en lógica difusa y redes neuronales.

2 MODELOS DE INTERNET Y LÓGICA DIFUSA

La arquitectura de Internet está basada en el protocolo de nivel de red IP, protocolo sin conexión de extremo a extremo que proporciona un servicio no fiable de transmisión de paquetes nodo a nodo (no garantiza ni la entrega final ni el orden de los paquetes). El protocolo de nivel de transporte utilizado por las aplicaciones que generan la práctica totalidad del tráfico existente actualmente en Internet es TCP. El protocolo TCP realiza funciones de transporte con conexión de datos fiable de extremo a extremo y entrega ordenada, así como control de flujo y de congestión.

En este contexto, es un problema reconocido la carencia de modelos formales del tráfico de paquetes [8, 1], dado que la mayoría de los desarrollos disponibles hasta la fecha se basan en descripciones poco alejadas del lenguaje natural, salvo excepción de algunos modelos parciales e incompletos. Por ello, se buscan nuevos enfoques de análisis y diseño que fundamenten sistemas de medida y evaluación sistemática de prestaciones, así como modelos que sirvan de base a esquemas de control de congestión y garantía y clases de calidad de servicio.

Entre los modelos existentes válidos para redes de conmutación de paquetes destacan los modelos matemáticos basados en sistemas caóticos y autosimilitud [17]. Por otra parte, el estudio de muchas de las técnicas empleadas para el control de congestión en Internet está basado en la teoría clásica de colas, como en los casos del esquema de control de congestión de TCP, o el sistema de gestión activa de colas RED, ambos tratados con mayor detalle en secciones posteriores.

Con los nuevos servicios, tecnologías y aplicaciones se hace cada vez más patente la necesidad de mejoras en estos sistemas de control de congestión. También se requiere el diseño de esquemas de cooperación entre los nodos extremos y los nodos intermedios o routers (esquema mixto entre un esquema distribuido y centralizado). En este sentido destacamos como especialmente importante el desarrollo de una teoría básica de colas difusas [11], que puede sentar las bases para entender la dinámica del tráfico de Internet desde la perspectiva de los sistemas difusos.

Estos avances y nuevos análisis constituyen una de las alternativas más firmes para formar un novedoso enfoque de estudio de la dinámica del trafico en Internet, proporcionando modelos para fundamentar sistemas de análisis, medida y evaluación de prestaciones, así como esquemas de control de congestión de naturaleza cooperativa y distribuida en los niveles de red y transporte.

A continuación presentamos propuestas y líneas de investigación aún poco desarrolladas así como otras más firmes que ya han dado lugar a resultados preliminares. Describiremos aportaciones a varios niveles: desde procesado estadístico y heurístico para filtrar resultados empíricos hasta métricas definidas a partir de sistemas de inferencia capaces de proporcionar información de alto nivel a partir de datos imprecisos e incompletos.

3 MEDIDA Y EVALUACIÓN DE PRESTACIONES

Diversos factores han llevado a un auge de la investigación en sistemas de medida de Internet, punto de partida para una mejor comprensión de la dinámica del tráfico y el desarrollo de sistemas de control de congestión y calidad de servicio. Especialmente el hecho de que en la Internet actual el soporte que los equipos y protocolos prestan para análisis y medida de prestaciones es mínimo, la necesidad de adaptación dinámica a las condiciones de la red, o de mejores métodos de análisis de prestaciones y diagnóstico de problemas,

Los modelos y sistemas de medida actuales [10],se suelen clasificar en dos grupos: activos y pasivos. Los primeros son de naturaleza distribuida y accesibles a todos los usuarios y aplicaciones, mientras que los segundos son centralizados y suelen ser accesibles sólo a los operadores e ingenieros de red. En ambos casos, es posible disponer de grandes volúmenes de datos y trazas de tráfico real, por lo que consideramos que son un futuro campo de aplicación de técnicas de análisis basado en soft computing desarrolladas para bases de datos difusas y aplicaciones de minería de datos.

Vamos a analizar por separado las áreas de aplicación de la lógica difusa a sistemas de medida activos y pasivos, sobre los que cabe destacar como características

comunes la disponibilidad de información incompleta e imprecisa, problema agravado en el caso de los sistemas activos por imprecisiones y errores de medida e incertidumbre derivada de las limitaciones de los modelos disponibles.

3.1 SISTEMAS PASIVOS

Los sistemas pasivos de medida de Internet constituyen un campo de aplicación de técnicas de análisis e interpretación de grandes volúmenes de datos frecuentemente imprecisos e incompletos. Estos sistemas operan en los routers recogiendo datos (usualmente muestreados de grandes volúmenes de datos en tiempo real).

De entre las estadísticas globales sobre el estado de los nodos, los datos más utilizados habitualmente son los que conforman la matriz de tráfico [15], que especifica, para una subred, la cantidad de tráfico transmitida entre cada par de enlaces de entrada y salida, y que es clave para las funciones de operación de la red.

Muy pocos equipos cuentan con mecanismos de medida que proporcionen toda la información necesaria para completar la matriz de tráfico, dado que el hardware necesario es muy complejo y costoso de implementar y difícil de operar y de mantener. La solución que por no requerir cambios en las infraestructuras se utiliza actualmente, conocida como tomografía de Internet, se basa en la inferencia mediante procedimientos estadísticos de la matriz, pero los procedimientos desarrollados hasta la fecha presentan errores elevados.

Tanto en el caso de la matriz de tráfico como en el de otras medidas de prestaciones, creemos que el empleo de sistemas de inferencia difusa puede aportar mejoras significativas en dos aspectos. Por una parte, inferencia de información a partir de conjuntos de datos incompletos. Por otra parte, proporcionan un marco teórico alternativo para la comprensión de conceptos de definición ambigua, especialmente en el análisis de la dinámica de agregados de flujos de datos, y de métricas para conceptos como nivel de congestión o desorden de paquetes, aspectos en los que incidiremos al tratar los sistemas activos de medida. No obstante, no tenemos constancia de ningún trabajo publicado sobre esta línea de investigación.

3.2 SISTEMAS ACTIVOS

Las herramientas activas de medida en Internet [12] inyectan paquetes sonda en la red para estudiar la respuesta, y, siguiendo algún modelo de red, estimar características como el retardo de transmisión o diversas métricas relacionadas con el ancho de banda.

En el diseño de estos sistemas de medida interviene de manera decisiva el modelo de red seguido. Por ello, el desarrollo de herramientas activas de medida, sobre todo de ancho de banda, constituye actualmente una de las líneas de investigación que más contribuye al estudio de la dinámica del tráfico, especialmente de los flujos agregados en las colas de los routers

No obstante, estas herramientas se enfrentan a la incertidumbre derivada de basarse en modelos complejos para extraer información acerca del estado de la red a partir de información incompleta. Otro factor condicionante es la precisión de los mecanismos de medida, en particular de medida temporal.

No tenemos conocimiento de ningún trabajo hecho público sobre la aplicación de sistemas difusos al diseño y mejora de estas herramientas, no obstante, hemos podido obtener resultados preliminares prometedores en dos aspectos:

- Mejora del procesamiento estadístico y heurístico de las medidas realizadas sobre los paquetes sonda, de modo que se puedan superar algunas de las limitaciones encontradas actualmente debido a los altos requisitos de precisión en la medida de tiempos.
- Utilización de sistemas difusos para inferir el grado de fiabilidad de las estimaciones de ancho de banda.

Por último, esperamos que un mayor desarrollo o reinterpretación de modelos de red bajo la perspectiva de la lógica difusa, haga factible la realización de herramientas activas de medida basadas en esta técnica.

4 CONTROL DE CONGESTIÓN Y CALIDAD DE SERVICIO

Tanto por experiencias pasadas con casos de congestión que han afectado de manera global a la red [9, 5], como por la importancia de tratar de forma adecuada diferentes tipos de tráfico, la función de control de congestión se ha identificado como una de las funciones críticas para la evolución y crecimiento de Internet.

Avanzar en el desarrollo de esquemas de control de congestión que permitan prevenir o corregir estados de saturación e incluso colapso de la red, requiere estudios formales de mayor amplitud que los modelos y esquemas disponibles actualmente, que tratan la simple prevención de estados de saturación de algunos nodos en topologías sencillas.

Una situación de congestión se entiende como la consecuencia de una degradación de las prestaciones de

la red que lleva a una situación de colapso provocado por el incremento progresivo de tráfico de retransmisiones de paquetes perdidos [6, 9]. Este hecho puede llevar a que el porcentaje de paquetes que llegan a su destino disminuya dramáticamente conforme crece el tráfico generado desde los nodos extremos de la red.

Parámetros como el porcentaje de paquetes perdidos y el retardo de transmisión son indicadores de congestión, si bien no existe una definición concreta cuantificada con exactitud (ni en cuanto a los valores ni a la escala a que se debe producir), y depende, entre otros factores, de los requisitos y patrones del tráfico.

Se trata, por tanto, de una situación intermedia entre un aprovechamiento eficiente de los recursos de la red y una situación de sobrecarga. Esto es, simplemente la identificación de un estado de congestión presenta la complejidad y ambigüedad que se ha podido tratar con éxito mediante sistemas basados en lógica difusa en muchos otros campos tecnológicos.

Los esquemas de control de congestión utilizados actualmente así como las alternativas propuestas siguen alguno de los dos enfoques siguientes:

- Distribuido entre los nodos extremos de la red e implementado por los protocolos de transporte empleados por las aplicaciones.
- Centralizado, implementado en los routers, que aun así deben operar de forma cooperativa y por ello distribuida.

Ambos esquemas se describen por separado en las siguientes secciones, tratando el problema de la garantía de calidad de servicio, función estrechamente vinculada al control de congestión en routers, y de escaso despliegue en la Internet actual.

4.1 CONTROL DE CONGESTIÓN MEDIANTE PROTOCOLOS DE TRANSPORTE

El esquema de control de congestión implantado en la Internet actual está basado en los mecanismos de control de congestión de TCP, utilizados por cada flujo de datos transportado mediante conexiones TCP. De este modo, los extremos transmisores y receptores de flujos TCP cooperan para realizar un control de congestión distribuido entre todos los nodos que tienen conexiones TCP establecidas.

Durante los últimos años se han desarrollado protocolos de transporte alternativos adaptados a las necesidades de las nuevas aplicaciones, pero todos con el objetivo de implementar mecanismos de control de congestión compatibles con los de TCP [16], de modo que sea posible su coexistencia y se garantice un reparto equitativo del ancho de banda y otros recursos de la red.

Desde la propuesta del primer esquema de control de congestión para TCP [9], a lo largo de más de 15 años, se han ido sucediendo mejoras, la mayoría de ellas mediante cambios incrementales, que han permitido superar las limitaciones iniciales de TCP. Sin embargo, el incremento en el producto retardo de propagación–ancho de banda, así como la proliferación de tecnologías de acceso al medio con características no consideradas en el diseño del esquema de control de congestión de TCP, plantean problemas para los cuales no existe una clara solución.

A grandes rasgos, TCP basa sus funciones de control de flujo y congestión en un mecanismo clásico de ventana deslizante de tamaño variable [9]. El tamaño de la ventana de congestión limita el número de paquetes que el extremo origen de datos de la conexión puede transmitir sin recibir notificaciones de acuse de recibo procedentes del receptor. Este valor se incrementa de forma lineal y se reduce de forma exponencial, en función de si se detecta pérdida de paquetes o modificaciones en el tiempo de retorno de paquetes entre ambos extremos de las conexiones.

La evolución de la ventana de congestión se regula mediante transiciones en una máquina de estados que refleja el estado de congestión inferido a partir de los parámetros de la conexión. Este esquema sugiere como solución apropiada el uso de sistemas de inferencia difusos para identificar el estado de la conexión.

El único trabajo en torno a la aplicación de sistemas difusos a TCP del que tenemos conocimiento es un estudio preliminar sobre una versión difusa de TCP [3], que sigue la idea descrita para identificar algunos de los estados de congestión considerados en las versiones actuales de TCP. Las pruebas realizadas mediante simulación muestran que el enfoque es factible y da resultados que pueden mejorar el esquema clásico de TCP, aunque se limitan a estudiar escenarios simples.

Siguiendo esta línea de desarrollo, queda como trabajo futuro generalizar el estudio para obtener un modelo difuso completo del esquema de control de congestión de TCP, considerando todos los posibles estados y ampliaciones, al tiempo que se analice la adaptación de este esquema de control de congestión a diferentes tipos de aplicaciones y clases de servicios desde la perspectiva de una base de reglas adaptable.

4.2 CONTROL DE CONGESTIÓN Y CALIDAD DE SERVICIO MEDIANTE GESTIÓN ACTIVA DE COLAS

Actualmente, salvo en entornos minoritarios y experimentales, los routers de Internet no emplean ningún mecanismo de gestión activa de colas de paquetes. El esquema de servicio dominante es el de colas FIFO pasivas sin distinción de clases de servicio (conocido como drop-tail), descartando paquetes únicamente cuando el espacio de almacenamiento se agota.

No obstante, se reconoce que los mecanismos de control de congestión distribuidos entre los nodos extremos no son suficientes para garantizar el uso eficiente de los recursos y la estabilidad de la red [2, 5].

Asimismo, sin esquemas de gestión activa de colas no es posible implementar sistemas de garantía de calidad de servicio ni de provisión de servicios diferenciados, ni tampoco es posible implantar mecanismos de regulación de los recursos utilizados por flujos de transporte de datos y penalización de aquellos que infrinjan las normas básicas de control de congestión compatibles con TCP [6].

Los equipos de los fabricantes más importantes incluyen al menos el esquema de gestión activa de colas más aceptado, RED [7], en alguna de sus numerosas variantes. RED establece una estrategia preventiva, al descartar paquetes de forma anticipada, antes de que el espacio de almacenamiento se agote, de modo que se induzca respuesta en los extremos antes de llegar a situaciones de saturación de las colas y congestión real. Sin embargo, su implantación es aún minoritaria.

Con la aparición de la tecnología de conmutación de redes ATM, surgieron las primeras propuestas de sistemas de gestión activa de colas de paquetes basados en lógica difusa. Dentro del modelo de gestión de recursos de las redes ATM, más cercano a los sistemas de telecomunicaciones tradicionales basados en conmutación de circuitos, se han desarrollado propuestas de sistemas difusos [14] para control de admisión así como control de flujo, congestión y tasa de transferencia.

Siguiendo un enfoque similar, se han desarrollado y analizado mediante simulación controladores difusos para la gestión activa de colas en los routers de Internet. En particular, se ha propuesto una variante difusa del esquema RED [4, 13], así como controladores difusos para la arquitectura estandarizada por el Internet Engineering Task Force para provisión de servicios diferenciados [18], en la que se distingue tráfico con restricciones temporales y tráfico con requisitos de fiabilidad en la entrega.

Estos controladores difusos son similares a los reguladores en tiempo real empleados en numerosos sistemas de control. En el esquema más básico, las entradas o variables lingüísticas de los antecedentes de la base de reglas son el tamaño actual de la cola así como su variación, mientras que la variable de salida se toma como un valor de probabilidad o referencia para decidir si se ha de descartar el último paquete recibido. Así, la variante difusa de RED propuesta en [13] sigue el esquema de un controlador PI (proporcional–integral) difuso.

Mediante reglas adicionales y dividiendo en módulos la base de reglas, se puede añadir la posibilidad de establecer diferentes prioridades de servicio para paquetes con diferentes restricciones, combinándose así funciones de control de congestión y de tratamiento diferenciado y garantía de calidad de servicio, con mayor eficacia y facilidad de análisis que en los esquemas no basados en lógica difusa [18].

A partir de estas propuestas, que proporcionan un estudio preliminar, mediante simulación, del comportamiento de algunos controladores difusos para la gestión activa de colas en Internet, queda por desarrollar un estudio en un rango más amplio de escenarios y condiciones, tanto mediante simulación como mediante prototipos reales, prestando atención a las posibilidades de adaptación a los diferentes patrones de tráfico de Internet, así como a problemas prácticos en su implantación, como la estabilidad y posibilidad de coexistir con otros esquemas.

5 CONCLUSIONES

Se han analizado las funciones de control de congestión y calidad de servicio en Internet desde la óptica de los sistemas difusos, presentando tanto las aportaciones al modelado y análisis de la red, como a los sistemas de medida y evaluación de prestaciones, así como al control de congestión en routers y nodos extremos, tanto centralizado como distribuido, combinado con funciones de calidad de servicio. Así, se han resumido las áreas en que ya se empieza a aplicar este nuevo enfoque de diseño, y se han propuesto nuevas áreas de aplicación.

Aunque se han resaltado ya las áreas concretas de aplicación de técnicas basadas en lógica difusa y soft computing, así como las líneas de investigación abiertas, más allá de técnicas para problemas específicos queremos destacar las expectativas a medio plazo que despierta el estudio general de la dinámica del tráfico de paquetes en Internet desde el punto de vista novedoso y estimulante de los sistemas difusos.

Referencias

[1] Bob Aiken, Victor Bahl, Bob Bhattacharjee, Bob Braden, et al. Report of NSF Workshop on Network Research Testbeds. Technical report, National Science Foundation, Directorate for Computer and Information Science and Engineering, Advanced Networking Infraestructure & Research Division, November 2002. http://gaia.cs.umass.edu/testbed_workshop.

[2] Bob Braden, David D. Clark, Jon Crowcroft, Bruce Davie, et al. Recommendations on Queue Management and Congestion Avoidance in the Internet. RFC 2309, Network Working Group, Informational., April 1998.

[3] Pablo Carbonell, Zhong-Ping Jiang, and Shivendra Panwar. Fuzzy TCP: A Preliminary Study. In *15th IFAC World Congress*, Barcelona, Spain, July 2002. http://ttt.alc.upv.es/~pjcarbon/.

[4] C. Chrysostomou, A. Pitsillides, L. Rossides, and A. Sekercioglu. Fuzy Logic Controlled RED: Congestion
Control in TCP/IP Differentiated Services Networks. *Soft Computing Journal*, 8(20), November 2003. http://www.cs.ucy.ac.cy/networksgroup/publications.htm.

[5] Sally Floyd. Congestion Control Principles. RFC 2914, Network Working Group, Best Current Practice., September 2000.

[6] Sally Floyd and Kevin Fall. Promoting the Use of End-to-End Congestion Control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4):458–472, August 1999. http://www.icir.org/floyd/papers.html.

[7] Sally Floyd and Van Jacobson. Random Early Detection Gateways for Congestion Avoidance. In *IEEE/ACM Transactions on Networking*, volume 1:4, pages 397–413, August 1993. http://citeseer.nj.nec.com/floyd93random.html.

[8] Sally Floyd and Eddie Kohler. Internet Research Needs Better Models. In *ACM SIGCOMM First Workshop on Hot Topics in Networks (HotNets-I)*, Princeton, New Jersey, USA, October 2002. http://www.icir.org/floyd/papers.html.

[9] Van Jacobson. Congestion Avoidance and Control. *ACM Computer Communication Review SIGCOMM '88 Symposium: Communications Architectures and Protocols*, 18(4):314–329, August 1988. http://citeseer.ist.psu.edu/jacobson88congestion.html.

[10] Vern Paxson. End-to-End Internet Packet Dynamics. *IEEE/ACM Transactions on Networking*, 7(3):277–292, June 1999. http://www.icir.org/vern/papers.html.

[11] Y. A. Phillis and R. Zhang. Fuzzy service rate control of queueing systems. *IEEE Trans. Systems, Man, and Cybernetics*, 29(4):503–517, August 1999. http://www.cs.ucy.ac.cy/networksgroup/publications.htm.

[12] R. S. Prasad, M. Murray, C. Dovrolis, and K. Claffy. Bandwidth estimation: metrics, measurement techniques, and tools. *IEEE Network Magazine*, June 2003. http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/publications.html.

[13] L. Rossides, C. Chrysostomou, A. Pitsillides, and A. Sekercioglu. Overview of Fuzzy-RED in Diff-Serv Networks. In *International Conference on Computing in an Imperfect World (Soft-Ware)*, pages 1–13, Belfast, Northern Ireland, April 2002. http://www.cs.ucy.ac.cy/networksgroup/publications.htm.

[14] A. Sekercioglu, A. Pitsillides, and A. Vasilakos. Computational Intelligence in Management of ATM Networks: A Survey of Current State of Research. *Soft Computing Journal*, 4(5):257–263, 2001. http://www.cs.ucy.ac.cy/networksgroup/publications.htm.

[15] George Varghese and Cristian Estan. The Measurement Manifesto. *ACM Computer Computer Communications Review*, 34(1), January 2004. http://nms.lcs.mit.edu/HotNets-II/accepted.html.

[16] Jörg Widmer, Robert Denda, and Martin Mauve. A Survey on TCP-Friendly Congestion Control. *Special Issue of the IEEE Network Magazine "Control of Best Effort Traffic"*, May 2001. http://icapeople.epfl.ch/widmer/publications.html.

[17] W. Willinger and V. Paxson. Where mathematics meets the internet. *Notices of the American Mathematical Society*, 45(8):961–970, August 1998. http://www.icir.org/vern/papers.html.

[18] Runtong Zhang and Jian Ma. Fuzzy QoS Management in Diff-Serv Networks. In *IEEE International Conference on Systems, Man and Cybernetics (IEEE/SMC 2000)*, volume 5, pages 3752–3757, Nashville, TN, USA, October 2000. http://www.sics.se/~runtong/.

CUANTIFICACIÓN BORROSA DE LA DIVERGENCIA ENTRE CONJUNTOS

Inés Couso
Dpto. Estadística e I.O. y D.M.
Universidad de Oviedo, España
couso@pinon.ccu.uniovi.es

Vladimir Janiš
Dpto. Matemáticas
Universidad Matej Bel, Eslovaquia
janis@fpv.umb.sk

Susana Montes
Dpto. Estadística e I.O. y D.M.
Universidad de Oviedo, España
montes@uniovi.es

Resumen

El propósito de este trabajo es introducir una medida de la diferencia entre dos conjuntos borrosos que tome valores en el conjunto de partes borrosas de los números reales positivos. De esta forma, la información que se obtiene al trabajar con medidas que sean números reales puede ser incrementada, al ponderar de distinta manera si la diferencia se ha producido en puntos con alto grado de pertenencia o no. Definiremos este concepto de forma axiomática, considerando tanto los axiomas obtenidos para las medidas "nítidas" ya existentes, como los nuevos que surgirán al trabajar en un conjunto que contiene al de los reales positivos. Finalizaremos este trabajo presentando un procedimiento que nos permita obtener para cualquier medida "nítida", su correspondiente generalización.

Palabras Clave: Conjuntos reales borrosos; pseudoinversa; medida de divergencia.

1 INTRODUCCIÓN

La capacidad de comparar dos conjuntos borrosos es fundamental en muchos campos como, por ejemplo, en la clasificación de los individuos como pertenecientes o no a un determinado grupo. Así pues, esta herramienta puede ser necesaria en el reconocimiento de imágenes, la teoría de cuestionarios, etc.

Este hecho ha motivado que en los últimos años, numerosos autores hayan investigado la forma de realizar dicha comparación. La mayoría de ellos han utilizado el concepto de medida de similaridad y con ella han obtenido el grado de igualdad o similaridad entre dos conjuntos (ver, por ejemplo, [4], [13], [16], ...).

Evidentemente, dos conjuntos pueden ser comparados cuantificando sus similitudes o sus diferencias. Esto ha hecho que, aunque en menor cuantía, también hayan surgido trabajos en los que se comparan dos conjuntos mediante el grado de diferencia entre los mismos. Así, Bouchon-Meunier et al. ([2]) introdujeron el concepto de medida de disimilaridad y en base a este trabajo, Montes et al. ([11], [12]) introdujeron el concepto de medida de divergencia, generalizando el de medida de disimilaridad, excepto por la propiedad simétrica, cuya exclusión puede tener sentido en algunas aplicaciones específicas.

Las medidas de divergencia tienen importantes aplicaciones en la teoría de los cuestionarios con respuestas borrosas ([1]), habiendo sido además utilizadas por los autores, entre otras cosas, para medir la borrosidad de un conjunto, al compararlo con el equilibrio o con el nítido más próximo ([9]).

Hasta el momento, todas las medidas de igualdad-diferencia entre conjuntos borrosos conocidas por los autores, toman valores en el conjuntos de los números reales. Esta restricción puede acarrear, en algunas ocasiones, una importante pérdida de información que pretendemos subsanar al considerar en lugar de números, funciones de los reales en el intervalo $[0, 1]$, es decir, conjuntos borrosos. De esta forma podríamos cuantificar de distinta forma la diferencia si ésta se presenta en puntos con un grado de pertenencia a los conjuntos alto o bajo. La idea sería la siguiente: cuando queremos comparar dos conjuntos, cuanto más de cerca los miremos, más diferencias veremos entre ellos, de forma que, si nos alejamos lo suficiente, dos conjuntos claramente distintos podrían parecernos muy similares . Esta cercanía al ver los conjuntos, puede expresarse mediante los alfa-cortes. Con esta idea de base, podemos generalizar las medidas de divergencia "nítidas", si permitimos que dichas medidas tomen valores que sean conjuntos borrosos definidos sobre los reales en lugar de números, es decir, que su imagen sea un conjunto borroso.

Una vez introducidos nuestros objetivos, presentaremos los conceptos previos necesarios en la Sección 2, centrándonos principalmente en aquellos relativos a las medidas de divergencia nítidas propuestas en [12]. A partir de estas ideas, comenzaremos la Sección 3 introduciendo el concepto de medidas de divergencia borrosas. La definición axiomática de estas medidas, se complementa con algunas interesantes propiedades de las mismas, así como un ejemplo en el que se pueden observar los beneficios, en cuanto a incremento en la información, del uso de estas medidas en lugar de las nítidas. En la Sección 4 se muestra un sencillo método de construcción de medidas borrosas a partir de las ya estudiadas medidas nítidas. Para terminar, en la Sección 5 se presentan algunos comentarios finales y conclusiones, así como posibles trabajos futuros.

2 CONCEPTOS PREVIOS

En esta sección vamos a introducir brevemente aquellos conceptos y notaciones que iremos utilizando a lo largo de todo el trabajo.

El universo de discurso será un conjunto clásico cualquiera no vacío, que denotaremos por Ω. Al conjunto de las partes nítidas de Ω lo denotaremos por $\mathcal{P}(\Omega)$ y al de las partes borrosas por $\mathcal{F}(\Omega)$. Para todo elemento $\widetilde{A}$ de $\mathcal{F}(\Omega)$ y para todo α en el intervalo $(0,1]$, $\widetilde{A}_\alpha$ denotará el alfa-corte de $\widetilde{A}$, es decir, el conjunto nítido formado por todos los puntos de Ω que pertenecen a $\widetilde{A}$ en grado mayor o igual que α. Así pues,

$$\widetilde{A}_\alpha = \{x \in \Omega | \widetilde{A}(x) \geq \alpha\}, \forall \alpha \in (0,1].$$

En el caso $\alpha = 0$, $\widetilde{A}_0$ representa el soporte de $\widetilde{A}$.

El objetivo de este trabajo es cuantificar el grado de diferencia entre cada par de elementos de $\mathcal{F}(\Omega)$, sea cual sea el universo de discurso Ω. Dicha cuantificación se va a hacer mediante un conjunto borroso definido sobre $\mathbb{R}$, con lo que el espacio de llegada de la medida de divergencia va a ser $\mathcal{F}(\mathbb{R})$. En dicho espacio trabajaremos con una relación de orden parcial definida, para cualesquiera $\widetilde{A}$ y B en $\mathcal{F}(\mathbb{R})$ por:

$$\widetilde{A} \leq \widetilde{B} \Leftrightarrow \widetilde{\min}\{\widetilde{A}, \widetilde{B}\} = \widetilde{A},$$

donde $\widetilde{\min}$ representa a la función "mínimo" calculada aplicando el Principio de Extensión de Zadeh (ver, por ejemplo, [7]).

Se puede demostrar que dos conjuntos reales borrosos $\widetilde{A}$ y $\widetilde{B}$ con funciones de pertenencia no crecientes verifican que $\widetilde{A}$ es menor o igual que $\widetilde{B}$ si y sólo si $\widetilde{A}$ está contenido en $\widetilde{B}$, es decir,

$$\widetilde{A} \leq \widetilde{B} \Leftrightarrow \widetilde{A} \subseteq \widetilde{B},$$

donde "$\subseteq$" representa la relación de inclusión para la t-norma del mínimo.

Otro concepto importante en la teoría de los conjuntos difusos es el de pseudo-inversa. En nuestro contexto, la pseudo-inversa de una función no creciente f, definida sobre un intervalo $I \subseteq \overline{\mathbb{R}}$ y tal que su imagen verifique que $[\inf_{x\in I} f(x), \sup_{x\in I} f(x)] \subseteq \overline{\mathbb{R}}$, es otra función que denotaremos por $f^{(-1)} : J \to \overline{\mathbb{R}}$ tal que

$$\max\{a, \sup\{x \in I \mid f(x) > y\}\} \leq f^{(-1)}(y)$$

$$f^{(-1)}(y) \leq \max\{a, \sup\{x \in I \mid f(x) \geq y\}\},$$

donde a representa el ínfimo del intervalo I.

Si f es una función definida en todo $\mathbb{R}$, entonces la función $f^{(-1)} : [\inf_{x\in I} f(x), \sup_{x\in I} f(x)] \to \overline{\mathbb{R}}$ definida por

$$f^{(-1)}(y) = \sup\{x \in \mathbb{R} \mid f(x) > y\}$$

es una pseudo-inversa de f y es también no creciente.

Una vez introducidos los conceptos de conjunto real borroso y de pseudo-inversa, sólo nos queda para terminar con la recopilación de los conceptos previos, comentar brevemente la idea y definición de la medida de divergencia. Estas medidas fueron introducidas y estudiadas en varios trabajos anteriores ([10], [11], [12]), con el propósito de cuantificar el grado de diferencia entre dos conjuntos borrosos de cualquier universo de referencia Ω. Se demostró, además, que las usuales distancias entre conjuntos borrosos ([7]) son casos particulares de estas medidas.

La definición de medida de divergencia fue dada de forma axiomática en base a las siguientes propiedades naturales:

- Debe ser una función no negativa y simétrica. La simetría puede perder el sentido en algunos contextos en los que se compara, por ejemplo, una observación con un patrón; en tales situaciones el axioma relativo a la simetría debería ser eliminado. Abundantes ejemplos de este tipo de divergencias no simétricas pueden encontrarse en la Teoría clásica de la Información. Entre éstos, destacamos por su importancia práctica, por ejemplo, la divergencia de Kullback y Leibler ([8]), la de Burbea y Rao ([3]), la de Salicru et al. ([14]), la de Taneja et al. ([15]).

- La diferencia entre dos conjuntos iguales es nula.

- Si un conjunto es "más similar" a otro que un tercero, la divergencia entre los dos primeros es menor.

La formulación analítica de estas propiedades es inmediata para las dos primeras, mientras que la tercera depende de la formalización de la idea de "ser más similar". En nuestro caso hemos considerado que si un conjunto $\widetilde{C}$ es añadido (en el sentido de la unión) a dos conjuntos borrosos $\widetilde{A}$ y $\widetilde{B}$, los dos conjuntos resultantes son igual o más parecidos que los iniciales y que lo mismo ocurre con la intersección.

Teniendo en cuenta todo lo anterior, se define una medida de divergencia como una función $D : \mathcal{F}(\Omega) \times \mathcal{F}(\Omega) \longrightarrow I\!R$ tal que, para cualesquiera $\widetilde{A}, \widetilde{B}, \widetilde{C} \in \mathcal{F}(\Omega)$, verifique los siguientes axiomas:

1. $D(\widetilde{A}, \widetilde{B}) = D(\widetilde{B}, \widetilde{A})$;
2. $D(\widetilde{A}, \widetilde{A}) = 0$;
3. $\max\{D(\widetilde{A} \cup \widetilde{C}, \widetilde{B} \cup \widetilde{C}), D(\widetilde{A} \cap \widetilde{C}, \widetilde{B} \cap \widetilde{C})\} \leq D(\widetilde{A}, \widetilde{B})$;

en cuya definición se ha utilizado el tercer axioma para expresar que si son más similares, la divergencia es menor. Dicho axioma es en realidad el resultado de unir dos condiciones que, aunque en principio podrían parecer equivalentes, no lo son.

Nótese también que no se ha impuesto en los axiomas la no negatividad de D. Esto es debido a que dicha propiedad se deduce del segundo y el tercer axioma, si se considera $\widetilde{C} = \emptyset$. Así pues, en realidad D es una función de $\mathcal{F}(\Omega) \times \mathcal{F}(\Omega)$ en $I\!R^+$.

Como ejemplo de medida de divergencia en aquellos casos en los que trabajemos en un espacio finito Ω o sólo con subconjuntos borrosos finitos de Ω, podríamos considerar la distancia de Hamming definida por

$$D(\widetilde{A}, \widetilde{B}) = \sum_{x \in \Omega} |\widetilde{A}(x) - \widetilde{B}(x)|.$$

Si (Ω, μ) es un espacio medible, otro ejemplo de medida de divergencia para conjuntos borrosos integrables sería:

$$D(\widetilde{A}, \widetilde{B}) = \int_{\Omega} |\widetilde{A} - \widetilde{B}| d\mu.$$

3 AXIOMAS DE LA MEDIDA DE DIVERGENCIA EN $\mathcal{F}(\Omega)$

Si consideramos dos patrones distintos cualesquiera (es decir, dos conjuntos borrosos) desde muy lejos, ambos conjuntos serán siempre iguales. Ahora bien, a medida que nos vamos acercando vamos observando mayores diferencias entre ellos. Así pues, cuanto "más cerca" estamos de los conjuntos, más información tendremos de las diferencias existentes entre ellos. Dicha proximidad se puede cuantificar mediante los alfa-cortes, con lo que, lo lógico es pensar que dos conjuntos serán más próximos cuantos más alfa-cortes en común tengan o más parecidos sean éstos.

Las medidas de divergencia de la sección anterior no recogen toda la información que acabamos de exponer y para conseguirlo es necesario trabajar con medidas de divergencia que tomen valores en $\mathcal{F}(I\!R^+)$, en lugar de en $I\!R^+$, es decir, que la medida de la divergencia entre dos conjuntos sea un conjunto borroso real en lugar de un simple número real. Dicho conjunto borroso sería evaluado sobre cada valor de $I\!R^+$ de forma que $\widetilde{D}(\widetilde{A}, \widetilde{B})(d)$ podría ser interpretado como la máxima distancia desde la que se puede observar una divergencia mayor o igual de d entre los conjuntos $\widetilde{A}$ y $\widetilde{B}$.

Con estas premisas, se puede obtener una definición axiomática de medida de divergencia borrosa como sigue, teniendo en cuenta que este concepto debe generalizar al ya existente, con lo que debe recoger las ideas planteadas en los axiomas de la sección anterior.

Definición 3.1 *Dada una función* $\widetilde{D} : \mathcal{F}(\Omega) \times \mathcal{F}(\Omega) \longrightarrow \mathcal{F}(I\!R^+)$ *se dice que* $\widetilde{D}$ *es una medida de divergencia borrosa si verifica los siguientes axiomas:*

1. $\widetilde{D}(\widetilde{A}, \widetilde{B}) = \widetilde{D}(\widetilde{B}, \widetilde{A}), \forall \widetilde{A}, \widetilde{B} \in \mathcal{F}(\Omega)$.
2. $\widetilde{D}(\widetilde{A}, \widetilde{A}) = \chi_{\{0\}}, \forall \widetilde{A} \in \mathcal{F}(\Omega)$.
3. $\widetilde{D}(\widetilde{A}, \widetilde{B})$ *es una función no creciente,* $\forall \widetilde{A}, \widetilde{B} \in \mathcal{F}(\Omega)$.
4. *La función* $\widetilde{D}_\alpha : \mathcal{F}(\Omega) \times \mathcal{F}(\Omega) \longrightarrow I\!R^+$ *definida por* $\widetilde{D}_\alpha(\widetilde{A}, \widetilde{B}) = \sup\{x \in I\!R^+ | x \in (\widetilde{D}(\widetilde{A}, \widetilde{B}))_\alpha\}, \forall \widetilde{A}, \widetilde{B} \in \mathcal{F}(\Omega)$ *es una medida de divergencia nítida.*
5. *La medida de divergencia nítida* $\widetilde{D}_\alpha(\widetilde{A}, \widetilde{B})$ *se puede expresar como*

$$\sup_{\beta \geq \alpha}\{\sup\{x \in I\!R^+ \mid x \in (\widetilde{D}(\widetilde{A}_\beta, \widetilde{B}_\beta))_1\}\}.$$

Los cinco axiomas anteriores surgen de forma natural y tienen una sencilla interpretación. Los dos primeros axiomas son simplemente la generalización de los considerados en el caso de medidas nítidas. El tercer axioma puede interpretarse como que si a distancia $\widetilde{D}(\widetilde{A}, \widetilde{B})(d)$ podemos ver una divergencia mayor o igual que d, entonces también podemos ver una divergencia mayor o igual que d', para todo $d' < d$, con lo cual $\widetilde{D}(\widetilde{A}, \widetilde{B})(d') \geq \widetilde{D}(\widetilde{A}, \widetilde{B})(d)$. El cuarto axioma nos dice que la función $\widetilde{D}_\alpha$, que mide la máxima divergencia que hay entre dos conjuntos a una distancia α de

ellos, es una medida de divergencia nítida, para cada valor de α. El quinto axioma surge de exigir que la divergencia entre $\widetilde{A}$ y $\widetilde{B}$ que podemos observar a una distancia α coincida con el supremo de las divergencias entre los conjuntos nítidos $\widetilde{A}_\beta$ y $\widetilde{B}_\beta$ para todo $\beta \geq \alpha$, teniendo en cuenta que a distancia α estos conjuntos serán vistos con total precisión.

En los axiomas de medida de divergencia borrosa en la Definición 3.1, sólo aparecen los dos primeros de los tres considerados en la definición nítida. El tercero ha sido excluido por ser redundante, como puede verse en la siguiente proposición.

Proposición 3.2 *Si $\widetilde{D}$ es una medida de divergencia borrosa se tiene que, para cualesquiera $\widetilde{A}, \widetilde{B}$ y $\widetilde{C}$ en $\mathcal{F}(\Omega)$, se verifica que*

$$\max\{\widetilde{D}(\widetilde{A}\cup\widetilde{C},\widetilde{B}\cup\widetilde{C}),\widetilde{D}(\widetilde{A}\cap\widetilde{C},\widetilde{B}\cap\widetilde{C})\}\leq\widetilde{D}(\widetilde{A},\widetilde{B}),$$

donde " $\leq''$ representa el orden parcial sobre $\mathcal{F}(\Omega)\times\mathcal{F}(\Omega)$ introducido en la Sección 2.

Como consecuencia de esto, se puede ver como realmente la nueva definición recoge la idea de que si dos conjuntos están más próximos, su divergencia es menor.

Proposición 3.3 *Si $\widetilde{D}$ es una medida de divergencia borrosa se tiene que, para cualesquiera $\widetilde{A}_1, \widetilde{A}_2, \widetilde{A}_3$ y $\widetilde{A}_4$ en $\mathcal{F}(\Omega)$ tales que $\widetilde{A}_1\subseteq\widetilde{A}_2\subseteq\widetilde{A}_3\subseteq\widetilde{A}_4$, se verifica que*

$$\widetilde{D}(\widetilde{A}_1,\widetilde{A}_4)\geq\widetilde{D}(\widetilde{A}_2,\widetilde{A}_3),$$

considerando el orden sobre $\mathcal{F}(\Omega)$ definido en la Sección 2.

Ejemplo 3.4 *Dado un referencial finito Ω, la función $\widetilde{D}:\mathcal{F}(\Omega)\times\mathcal{F}(\Omega)\longrightarrow\mathcal{F}(\mathbb{R}^+)$ que a cada par de subconjuntos borrosos $\widetilde{A}$ y $\widetilde{B}$ de Ω le asocia el subconjunto borroso de $\mathbb{R}^+$ con función de pertenencia*

$$\widetilde{D}(\widetilde{A},\widetilde{B})(d)=$$

$$\sup\{\alpha\in[0,1]\mid\sup_{\beta\geq\alpha}\sum_{x\in\Omega}|\widetilde{A}_\beta(x)-\widetilde{B}_\beta(x)|\geq d\}$$

es una medida de divergencia borrosa.

Si $\Omega=\{x_1,x_2,x_3,x_4\}$ y consideramos $\widetilde{A}_1$ y $\widetilde{B}_1$ definidos por:

Ω	x_1	x_2	x_3	x_4
$\widetilde{A}_1$	0′1	1	1	0′2
$\widetilde{B}_1$	0′1	0′7	0′7	0′2

la divergencia $\widetilde{D}$ entre $\widetilde{A}_1$ y $\widetilde{B}_1$ es el subconjunto borroso de $\mathbb{R}^+$ definido por

$$\widetilde{D}(\widetilde{A}_1,\widetilde{B}_1)(d)=\begin{cases}1 & si\ d<2,\\ 0 & si\ d\geq 2,\end{cases}$$

cuya representación gráfica puede verse en la Figura 1.

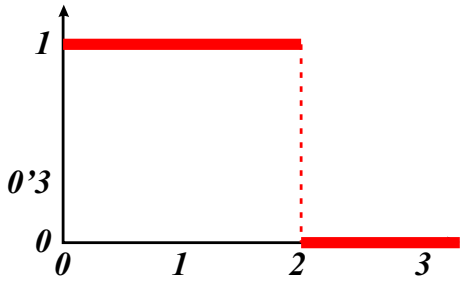


Figura 1: Función de pertenencia de $\widetilde{D}(\widetilde{A}_1,\widetilde{B}_1)$.

Ahora bien, si consideramos los conjuntos:

Ω	x_1	x_2	x_3	x_4
$\widetilde{A}_2$	0′3	1	1	0′3
$\widetilde{B}_2$	0	1	1	0

la divergencia $\widetilde{D}$ pasa a ser el subconjunto borroso de $\mathbb{R}^+$ definido por

$$\widetilde{D}(\widetilde{A}_2,\widetilde{B}_2)(d)=\begin{cases}0'3 & si\ d<2,\\ 0 & si\ d\geq 2,\end{cases}$$

cuya representación gráfica puede verse ahora en la Figura 2.

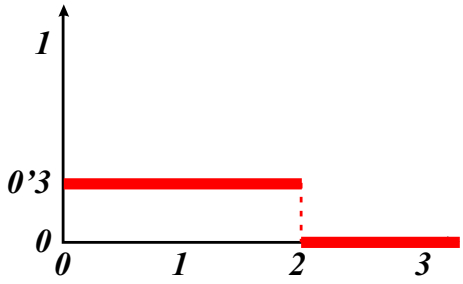


Figura 2: Función de pertenencia de $\widetilde{D}(\widetilde{A}_2,\widetilde{B}_2)$.

De las Figuras 1 y 2 se deduce inmediatamente que la medida de divergencia borrosa obtenida a partir de la distancia de Hamming permite distinguir entre el grado de diferencia de $\widetilde{A}_1$ y $\widetilde{B}_1$ y el de $\widetilde{A}_2$ y $\widetilde{B}_2$. Sin embargo, si se considera la divergencia nítida,

$$\sum_{i=1}^{4}|\widetilde{A}_1(x_i)-\widetilde{B}_1(x_i)|=0'6=\sum_{i=1}^{4}|\widetilde{A}_2(x_i)-\widetilde{B}_2(x_i)|,$$

con lo que, en ambos casos se obtiene que el valor de la diferencia es de 0′6.

4 CONSTRUCCIÓN DE MEDIDAS DE DIVERGENCIA BORROSAS

Para terminar vamos a relacionar las medidas de divergencia nítidas con las borrosas, principalmente con el objetivo de obtener un procedimiento para construir medidas de divergencia borrosas a partir de las medidas de divergencia nítidas ya conocidas y estudiadas en anteriores trabajos ([12]).

Teorema 4.1 *Si D es una medida de divergencia nítida en $\mathcal{P}(\Omega) \times \mathcal{P}(\Omega)$, entonces la función $\widetilde{D}$ definida por*

$$\begin{array}{rcl} \widetilde{D} : \mathcal{F}(\Omega) \times \mathcal{F}(\Omega) & \longrightarrow & \mathcal{F}(I\!R^+) \\ (\widetilde{A}, \widetilde{B}) & \longrightarrow & \phi^{(-1)}_{(\widetilde{A},\widetilde{B})} \end{array}$$

es una medida de divergencia borrosa, donde la función $\phi_{(\widetilde{A},\widetilde{B})} : [0,1] \to I\!R^+$ viene definida, para todo $\widetilde{A}, \widetilde{B}$ en $\mathcal{F}(\Omega)$, por

$$\phi_{(\widetilde{A},\widetilde{B})}(\alpha) = \sup\{D(\widetilde{A}_\beta, \widetilde{B}_\beta) | \beta \geq \alpha\}.$$

Como aplicación de este resultado, es inmediato que la función $\widetilde{D}$ utilizada en el Ejemplo 3.4 es una medida de divergencia borrosa, puesto que no es más que la transformación mediante el teorema anterior de la divergencia nítida D obtenida a partir de la distancia de Hamming.

En el otro sentido, a partir de una medida de divergencia borrosa, podemos obtener una medida de divergencia nítida sobre $\mathcal{P}(\Omega)$, como veremos a partir del siguiente teorema.

Teorema 4.2 *Si $\widetilde{D}$ es una medida de divergencia borrosa en $\mathcal{F}(\Omega) \times \mathcal{F}(\Omega)$, entonces la función D definida por*

$$\begin{array}{rcl} \widetilde{D} : \mathcal{P}(\Omega) \times \mathcal{P}(\Omega) & \longrightarrow & I\!R^+ \\ (A, B) & \longrightarrow & \sup_{\{x \in I\!R^+ | x \in (\widetilde{D}(A,B))_1\}} \{x\} \end{array}$$

es una medida de divergencia nítida.

Así pues, el Teorema 4.1 nos permite concluir que, dada una medida de divergencia nítida, podemos construir una medida de divergencia borrosa a partir de ella. Evidentemente, esta medida proporcionará más información que la nítida inicial, como vimos en el Ejemplo 3.4. En el Teorema 4.2 vimos como obtener una medida de divergencia nítida entre conjuntos nítidos a partir de una borrosa. Pues bien, estos procesos están muy relacionados puesto que, si comenzamos con una medida nítida y construimos a partir de ella la correspondiente medida borrosa, y a partir de ésta volvemos a obtener una medida nítida, recuperaríamos la divergencia inicial. Inversamente, si comenzamos por la borrosa, volveríamos a la borrosa. Así pues, tenemos una correspondencia biunívoca entre la familia de las medias de divergencia nítidas en $\mathcal{P}(\Omega)$ (¡no en $\mathcal{F}(\Omega)$!) y la familia de divergencias borrosas en $\mathcal{F}(\Omega)$, como establece el siguiente teorema.

Teorema 4.3 *Sean $\mathcal{D}$ la familia formada por todas las divergencia nítidas en $\mathcal{P}(\Omega)$ y sea $\widetilde{\mathcal{D}}$ la familia formada por todas las divergencia borrosas en $\mathcal{F}(\Omega)$. Existe una correspondencia biunívoca entre estas dos familias de funciones.*

A la vista del teorema anterior, cabe comentar que esta correspondencia biunívoca (que denotaremos por φ, para abreviar) es en los espacios de divergencias, no sobre cada una de ellas. Es decir, esto no implica que dada una divergencia borrosa $\widetilde{D}$, su valor sobre dos conjuntos cualesquiera $(\widetilde{A}, \widetilde{B})$ se pueda obtener como $\varphi \circ D(\widetilde{A}, \widetilde{B})$, para una divergencia nítida D, puesto que ni siquiera tiene sentido componer una función $D : \mathcal{P}(\Omega) \times \mathcal{P}(\Omega) \longrightarrow I\!R^+$ con una función $\varphi : \mathcal{D} \longrightarrow \widetilde{\mathcal{D}}$. Lo que establece este resultado es, por el contrario, que $\varphi(D) = \widetilde{D}$. Para aclarar mejor este punto, vamos a considerar un ejemplo sencillo, no relacionado con las divergencias.

Dado c un número real positivo, sea $\mathcal{F} = \{f_c : [0,1] \to I\!R | f_c(x) = c \cdot x\}$ y sea $\mathcal{G} = \{g_c : [0,1] \to I\!R | g_c(1) = c, g_c(x) = 0 \text{ en el resto}\}$. Es evidente que la aplicación

$$\begin{array}{rcccl} \varphi : & \mathcal{F} & \longrightarrow & & \mathcal{G} \\ & f_c & \longrightarrow & \left\{ \begin{array}{ll} f_c(x) & \text{si } x = 1 \\ 0 & \text{en el resto} \end{array} \right. & \end{array}$$

es una biyección y sin embargo, $g_c(x) \neq \varphi(f_c(x))$. Lo que sí se verifica es que $g_c = \varphi(f_c)$, al considerarlas como elementos de $\mathcal{F}$ y $\mathcal{G}$, respectivamente.

Además, utilizando el símil del párrafo anterior, también se pone de manifiesto que la información que dan ambas medidas de divergencia (nítida y borrosa) puede ser totalmente distinta. Obsérvese que, mientras el conocimiento de $f_c(x)$ nos permite automáticamente determinar el valor de x, el conocimiento de $g_c(x)$ no da más información sobre x de si es o no el valor 1. En este ejemplo g_c haría el papel de la medida de divergencia nítida en $\mathcal{P}(\Omega)$ y f_c el de la medida de divergencia borrosa en $\mathcal{F}(\Omega)$.

5 CONCLUSIONES

En este trabajo hemos presentado una nueva forma de medir la diferencia entre dos conjuntos borrosos utilizando una función que toma, a su vez, valores borrosos.

En este contexto, hemos tenido en cuenta los estudios realizados con anterioridad para medidas de divergencia nítidas, proporcionando incluso un procedimiento para obtener las medidas borrosas a partir de una medida de las ya conocidas con anterioridad. Aunque la información proporcionada por una medida de divergencia nítida puede ser suficiente en algunos casos, las medidas de divergencia borrosas deberían ser utilizadas cuando se quiere recoger la divergencia existente con mayor precisión.

No obstante, esta mayor precisión suele venir acompañada de un mayor esfuerzo en la obtención de la misma. Así pues, como ocurre habitualmente al trabajar con conjuntos borrosos en lugar de con conjuntos nítidos, las operaciones pueden complicarse. Sin embargo, en aquellos casos como los considerados en el ejemplo, de medidas de divergencia borrosas obtenidas a partir de medidas nítidas, la implementación de los cálculos es relativamente sencilla y puede realizarse sin ninguna dificultad.

En relación con este trabajo, tenemos varios puntos abiertos. Entre éstos destacaremos, por ser uno de los más inmediatos, uno de ellos: la generalización del concepto de medida de divergencia local al caso de medidas que tomen valores borrosos, puesto que éstas son las divergencias más manejables en las situaciones prácticas.

Agradecimientos

La investigación de este trabajo está subvencionada en parte a través del Proyecto BFM2001-3515 y la Acción Especial TIC2002-11942-E. Los autores quieren agradecer al Ministerio de Ciencia y Tecnología dicha ayuda económica.

Referencias

[1] C. Bertoluzza y A. Salas. Pseudo-questionnaires in a deterministic enviroment. *Proceedings of ESIT'99 Conference*, 1999.

[2] B. Bouchon-Meunier, M. Rifqi y S. Bothorel. Towards general measures of comparison of objets. *Fuzzy Sets and Systems* **84**, Pág.143-153, 1996.

[3] J. Burbea y C.R. Rao. Entropy differential metric, distance and divergence measures in probabilistic spaces: a unified approach. *T. Multiv. Anal.* **12**, Pág.575-596, 1982.

[4] S.M. Chen, M.S. Yeh y P.Y. Hsiao. A comparison of similarity measures of fuzzy values. *Fuzzy Sets and Systems* **72**, Pág.79-89, 1995.

[5] I. Couso y P. Gil. Measures of fuzziness of type 2 fuzzy sets. *Proceedings of IPMU'96 Conference*, 1996.

[6] D. Denneberg. Non-additive measure and integral. *Kluwer Academic Publishers (The Netherlands)*, 1994.

[7] G.J. Klir y T.A. Folger. Fuzzy Sets, Uncertainty, and Information. *Prentice-Hall International (UK)*, 1988.

[8] S. Kullback y A. Leibler. On the Information and Sufficiency. *Ann. Math. Statist.* **22**, Pág.79-86, 1951.

[9] S. Montes, I. Couso y P. Gil. Relationships Between Uncertainty Measures, Fuzziness Measures And Divergence Measure. *Proceedings of EUSFLAT-ESTYLF Joint Conference*, 1999.

[10] S. Montes y P. Gil. Some classes of divergence measures between fuzzy subsets and fuzzy partitions. *Mathware and Soft Computing* **5**, Pág.253-265, 1999.

[11] S. Montes, P. Gil y C. Bertoluzza. Divergence between fuzzy sets and fuzziness. *Proceedings of IPMU'98 Conference*, 1998.

[12] S. Montes, I. Couso, P. Gil y C. Bertoluzza. Divergence measure between fuzzy sets. *International Journal of Approximate Reasoning* **30/2**, Pág.91-105, 2002.

[13] C.P. Pappis y N.I. Karacapilidis. A comparative assessment of measures of similarity of fuzzy values. *Fuzzy Sets and Systems* **56**, Pág.171-174, 1993.

[14] M. Salicru, M.L. Menéndez, D. Morales y L. Pardo. A Test of Independence based on the (r,s)-Directed Divergence. *Tamkang J. Math.* **23**, Pág.95-107, 1992.

[15] I.J. Taneja, L. Pardo, D. Morales y M.L. Menéndez. On Generalized Information and Divergence Measures and their Applications: A Brief review. *Qestiió.* **13**, Pág.47-75, 1989.

[16] X. Wang, B. De Baets y E. Kerre. A comparative study of similarity measures. *Fuzzy Sets and Systems* **73**, Pág.259-268, 1995.

ESTUDIO COMPARADO MEDIANTE DISTANCIAS NORMALIZADAS ENTRE EL CIERRE TRANSITIVO Y OTROS ALGORITMOS DE TRANSITIVIZACIÓN.

Luis Garmendia
Universidad Complutense de Madrid.
Dpto. de Lenguajes y Sistemas Informáticos.
Facultad de Informática.
lgarmend@fdi.ucm.es

Adela Salvador
Universidad Politécnica de Madrid..
Dpto. de Matemática Aplicada
E.T.S.I. Caminos.
ma09@caminos.upm.es

Resumen

Se ha confeccionado un programa en C++ que genera de forma aleatoria matrices borrosas de dimensión predeterminada, obtiene sus cierres Min-transitivos, Prod-transitivos y W-transitivos y las matrices Min, Prod y W-transitivizadas contenidas en las matrices de origen. Se comparan utilizando distancias normalizadas los cierres T-transitivos con las relaciones T-transitivizadas obteniéndose que la distancia normalizada entre la relación borrosa original y la relación transitivizada es menor que dicha distancia obtenida con el cierre transitivo.

Palabras Clave: Cierre T-transitivo, aproximación T-transitiva, relación borrosa.

1 INTRODUCCIÓN

Un nuevo método de T-transitivización de relaciones borrosas [Garmendia y Salvador; 2000] puede usarse para dar una medida de T-transitividad de relaciones borrosas, y para construir relaciones borrosas T-transitivas próximas a una relación borrosa de partida, lo que también sirve para construir T-preórdenes, distintos del cierre T-transitivo, a partir de relaciones borrosas reflexivas.
Se implementa un programa [Garmendia y Salvador, 2004] que obtiene relaciones borrosas, de forma aleatoria, de una dimensión dada. Construye sus cierres Min-transitivo, Prod- transitivo y W-transitivo y las relaciones Min-transitivizada, Prod-transitivizada y W-transitivizada.
Se calculan medidas de T-transitividad de las relaciones borrosas originales midiendo la diferencia entre dicha relación borrosa y su relación T-transitivizada mediante la distancia normalizada del valor absoluto, la distancia normalizada euclídea. Se calculas esas mismas distancias entre la relación borrosa original y sus cierres T-transitivos. Se comprueba, como las distancias entre el cierre T-transitivo con la original es mayor que esa misma distancia con la relación T-transitivizada.

2 PRELIMINARES

2.1. CIERRE TRANSITIVO

El cierre T-transitivo R^T de una relación borrosa R, se define como la menor relación que contiene a R y que es T-transitiva. El algoritmo para obtenerlo es el siguiente:
1) $R' = R \quad_{Max}(Ro_{Sup\text{-}T}R)$
2) Si R' R entonces R := R' y volver a 1), y en otro caso terminar, y $R^T := R'$.

2.2. ALGORITMO DE TRANSITIVIZACIÓN

En *On a new method to T-transitivize fuzzy relations* [Garmendia y Salvador; 2000] se presentó un nuevo método algorítmico para T-transitivizar relaciones borrosas, es decir, partiendo de una relación borrosa se proporcionaba un método para obtener una relación borrosa lo más parecida posible a la dada, contenida en la original y que sea T-transitiva. Este método permite también conocer si la relación dada es ya T-transitiva.

2.3. CONCEPTOS PREVIOS

Sea $E = \{a_1, ..., a_n\}$ un conjunto finito. Sea T una t-norma triangular. Una relación borrosa R: E E [0, 1] es **T-transitiva** si T(R(a,b), R(b,c)) R(a,c) para todo a, b, c de E.
El algoritmo [Garmendia y Salvador, 2000] que se propone consiste en transformar una relación borrosa R^0 en otra relación T-transitiva R_T contenida en R^0 mediante n^2-1 pasos. Para poder asegurar que los elementos T-transitivizados sigan siendo elementos T-transitivos de todas las relaciones posteriores, se debe elegir en cada paso el elemento a T-transitivizar de la siguiente manera: Se elige el elemento no T-transitivo de la relación salida del paso anterior que sea el de menor grado de relación

entre todos los que todavía no han sido T-transitivizados. Para T-transitivizar dicho elemento $a_{i,j}^m$ de R^m se reducirá el grado de relación de otros elementos no T-transitivizados de R^m, y se logrará que $a_{i,j}^m = a_{i,j}^r$ para todo r m, pues al haberse elegido el menor elemento no T-transitivizado, este elemento no será causante de no T-transitividad de otros elementos no T-transitivizados (pues son mayores) por lo que no será recesario reducir su grado de relación en los siguientes pasos. Sea un conjunto de pares (i, j) donde i, j toman valores desde 1 hasta n.

Definición: ^{m} es un subconjunto de definido inductivamente de la siguiente forma:
1) $^0=$
2) $^{m+1} = {}^m$ (i, j) si $a_{i,j}^m$ es el elemento de R^m elegido para ser T-transitivizado durante el paso m+1-ésimo.
Es decir, ^{m} es el conjunto de los m pares (i, j) que corresponden a elementos T-transitivizados de R^m y (^{m})' es el conjunto de los n^2-m pares (i, j) que corresponden a elementos no T-transitivizados candidatos a serlo en los siguientes pasos.

Construcción de R^{m+1} a partir de R^m: Sea $a_{i,j}^m$ el elemento de R^m no transitivizado de mínimo valor que ha sido elegido para ser T-transitivizado en el paso m+1 ($a_{i,j}^m$ = Min{$a_{v,w}^m$ tal que (v, w) (^{m})'}).
Se define $a_{r,s}^{m+1}$ como

$$J^T(a_{s,j}^m, a_{i,j}^m) \quad \text{si } r = i, T(a_{r,s}^m, a_{s,j}^m) > a_{i,j}^m \text{ y } a_{i,s}^m \quad a_{s,j}^m$$
$$J^T(a_{i,r}^m, a_{i,j}^m) \quad \text{si } s = j, T(a_{i,r}^m, a_{r,s}^m) > a_{i,j}^m \text{ y } a_{i,r}^m \quad a_{r,s}^m$$
$$a_{r,s}^m \quad \text{en otro caso}$$

En el caso de que $T(a_{i,k}^m, a_{k,j}^m) > a_{ij}^m$, se elige entre los elementos $a_{i,k}^m$ y $a_{k,j}^m$, para decidir cual de los dos verá reducido su grado en el siguiente paso para que $T(a_{i,k}^{m+1}, a_{k,j}^{m+1})$ $a_{ij}^{m+1} = a_{ij}^m$.

3 EL PROGRAMA

3.1 DESCRIPCIÓN DEL PROGRAMA

El programa [Garmendia y Salvador], escrito en C++ Buider, escribe de forma aleatoria la matriz de una relación borrosa y computa sus cierres Min-transitivo, Prod-transitivo y W-transitivo calculando la distancia del valor absoluto y la distancia euclídea a la matriz de origen. Computa mediante el algoritmo mencionado las matrices Min-transitivizadas, Prod-transitivizadas y W-transitivizadas, calculando también sus distancias a la matriz de partida.
En especial se ha hecho que el programa calcule 100 matrices de cada dimensión, desde la dimensión 2 a la dimensión 100 y anote en una tabla las medias de las distancias obtenidas.
Veamos un ejemplo del aspecto del programa en la Fig. 1:

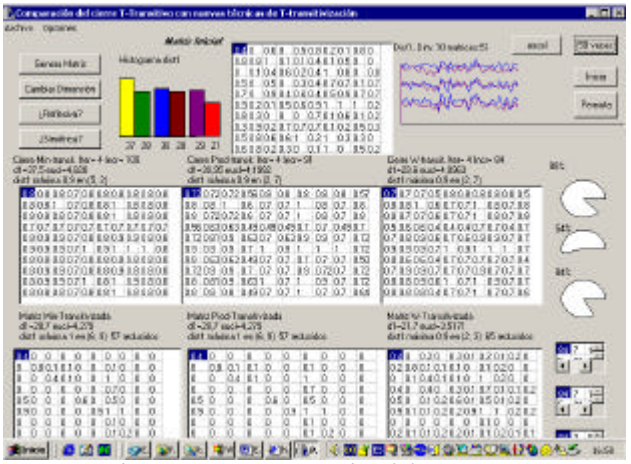


Figura 1: Aspecto exterior del programa

4 RESULTADOS

Al ejecutar el programa para cien relaciones aleatorias de cada dimensión, desde 2 hasta cien, se comprueba que las medias de las distancias obtenidas con el algoritmo son siempre menores que las obtenidas mediante el cierre transitivo [Garmendia y Salvador, 2004]. Pero estos valores son mucho menores cuando se utiliza la distancia euclídea que los que se obtienen con la distancia del valor absoluto.
Para anular las diferencias entre las distancias del valor absoluto y euclídeas se calculan las distancias normalizadas, dividiendo para ello la distancia del valor absoluto entre el cuadrado de la dimensión de la matriz, y dividiendo la distancia euclídea por la dimensión de la matriz. Se observa que las gráficas se ajustan en todos los casos mediante una función con una asíntota horizontal que oscila entre los valores de la tabla:

Tabla 1: Asíntota horizontal a las que se ajusta las distancias normalizadas del valor absoluto y las distancias normalizadas euclídeas, de los promedios de las distancias entre las relaciones borrosas generadas y las relaciones transitivas obtenidas mediante el cierre T-transitivo y mediante el algoritmo de T-transitivización.

Distancia normalizada del valor absoluto	**Min**	**Prod**	**W**
Cierre transitivo	y=0,499	y=0,499	y=0,499
Algoritmo de transitivización	y=0,43	y=0,43	y=0,42
Distancia normalizada euclídea	**Min**	**Prod**	**W**
Cierre transitivo	y=0,59	y=0,59	y=0,59
Algoritmo de transitivización	y=0,54	y=0,54	y=0,53

Las gráficas obtenidas tienen ahora una forma similar a la indicada a continuación, sea cual sea la t-norma utilizada, Min, Prod o W:

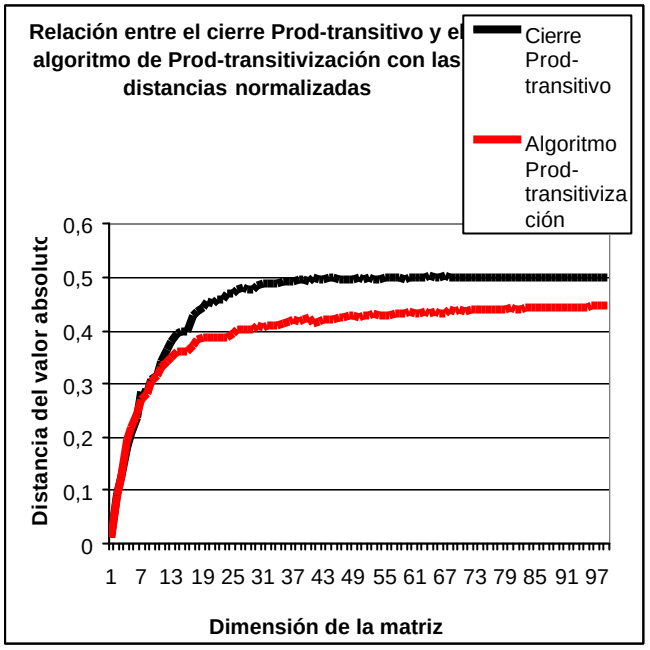


Figura 2: Promedios de distancias normalizadas del valor absoluto, para cada dimensión, de las relaciones borrosas generadas y las relaciones transitivas obtenidas mediante el cierre Prod-transitivo y el algoritmo de Prod-transitivización

Para analizar mejor la forma de la curva se toman separadamente las dimensiones de las matrices, de 12 a 26, de 26 a 52 y de 52 a 100. En la primera gráfica únicamente las matrices de dimensiones desde 2 a 26:

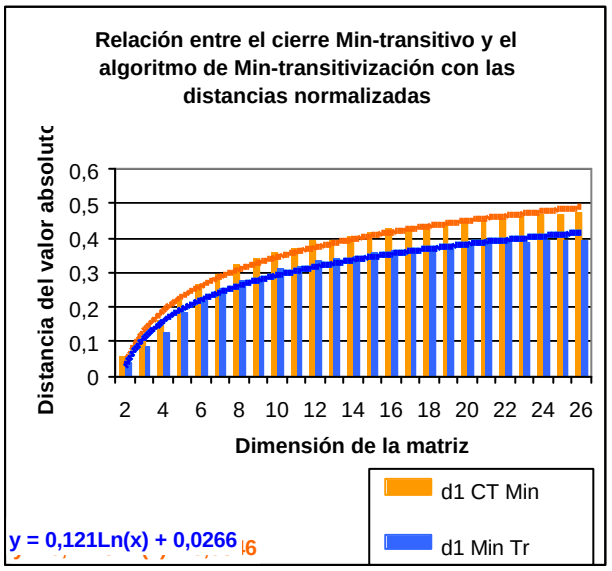


Figura 3: Promedios de distancias normalizadas del valor absoluto, para cada dimensión desde 2 a 26, de las relaciones borrosas generadas y las relaciones transitivas obtenidas mediante el cierre Min-transitivo y el algoritmo de Min-transitivización

Se observa que las distancias crecen rápidamente siendo siempre mayor la del cierre T-transitivo, y que pueden ajustarse bien mediante una función logarítmica, para el cierre transitivo:

Distancia = 0,14 ln(dimensión) + 0,035

y para el algoritmo de transitivización:

Distancia = 0,12 ln(dimensión) + 0,026.

A continuación se representan gráficamente las distancias de las matrices desde la dimensión 26 a la 52, y se observa como las diferencias tienden a estabilizarse, también para cualquiera de las t-normas utilizadas. En la gráfica puede verse la relación entre las distancias obtenidas, para cada dimensión, entre la relación de partida y su cierre transitivo cuando la t-norma utilizada es la de Lukasiewicz, y entre dicha relación de partida y la relación obtenida mediante el algoritmo de W-transitivización. Se observa que se ajustan bien a rectas de pendientes muy bajas, siendo menores las distancias obtenidas con el algoritmo de W-transitivización.:

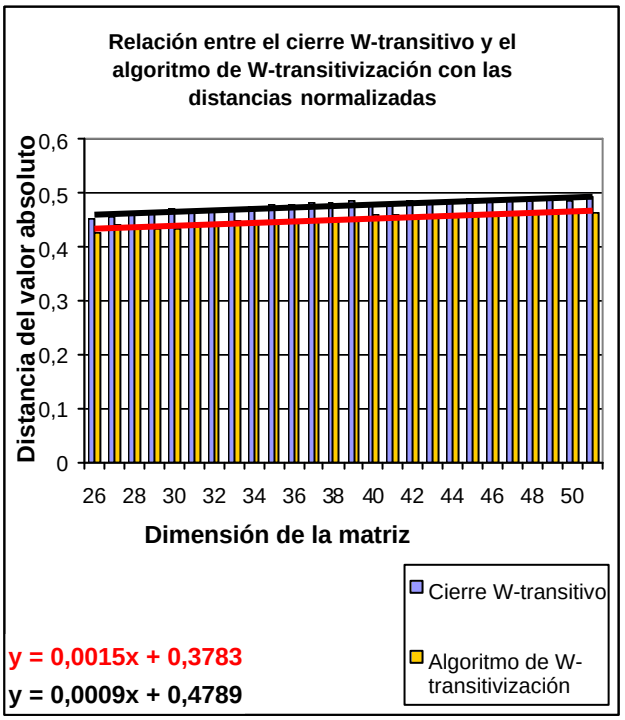


Figura 4: Promedios de distancias normalizadas del valor absoluto, para cada dimensión desde 26 a 52, de las relaciones borrosas generadas y las relaciones transitivas obtenidas mediante el cierre W-transitivo y el algoritmo de W-transitivización

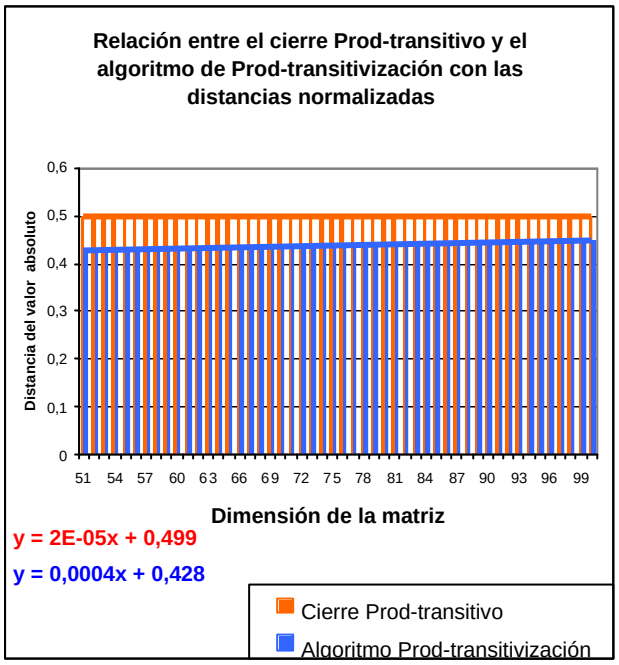


Figura 5: Promedios de distancias normalizadas del valor absoluto, para cada dimensión desde 50 a 100, de las relaciones borrosas generadas y las relaciones transitivas obtenidas mediante el cierre Prod-transitivo y el algoritmo de Prod-transitivización.

Al representar las gráficas de los promedios de las distancias del valor absoluto para las dimensiones desde 50 hasta a 100 se observa como tienden a una asíntota horizontal y cual es el valor de dicha asíntota.

Parece interesante conocer el valor entre las diferencias entre dichas distancias. Se representan las gráficas de las diferencias entre las distancias normalizadas del valor absoluto, siendo en los tres casos: Min, Prod, y W de la siguiente forma:

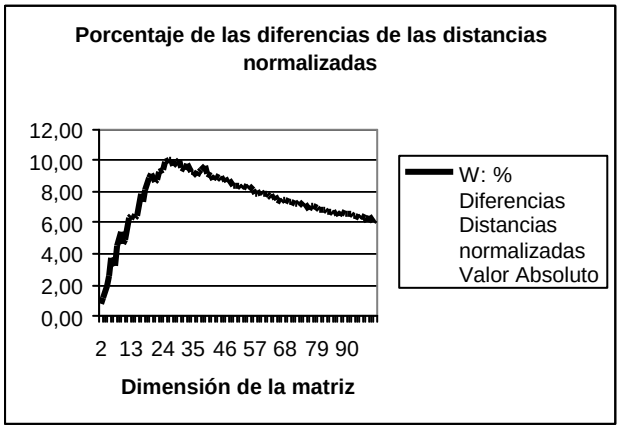


Figura 6: Diferencias, en porcentaje, entre los promedios de distancias normalizadas del valor absoluto, para cada dimensión desde 2 a 100, de las relaciones borrosas generadas y las relaciones transitivas obtenidas mediante el cierre W-transitivo y el algoritmo de W-transitivización.

Se observa que esa diferencia crece hasta la dimensión 30 y luego decrece, alcanzando un valor máximo de un 10 % de diferencia.
Se representan las gráficas de las diferencias de distancias normalizadas euclídeas, siendo ahora de una forma muy parecida a la anterior:

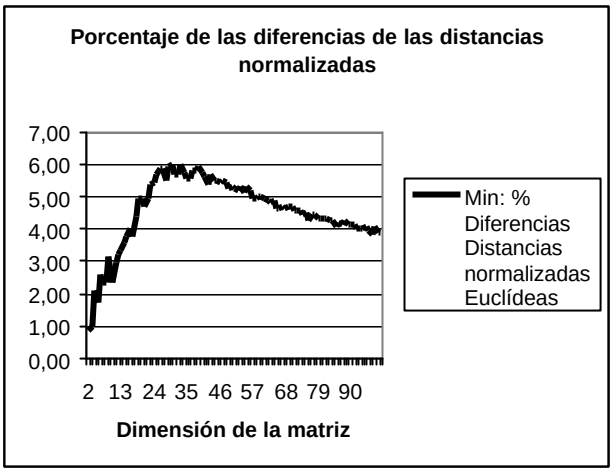


Figura 7: Diferencias, en porcentaje, entre los promedios de distancias normalizadas euclídeas, para cada dimensión desde 2 a 100, de las relaciones borrosas generadas y las relaciones transitivas obtenidas mediante el cierre Min-transitivo y el algoritmo de Min-transitivización.

5 CONCLUSIONES

Nuestro algoritmo de transitivización de relaciones borrosas resulta mucho mejor que el del cierre transitivo pues, para cualquier dimensión, reduce las distancias entre la matriz T-transitiva obtenida y la original. Se está trabajando en otras formas de mejorarlo. El resultado obtenido por el cierre transitivo es único. Sin embargo en el algoritmo propuesto se elige en ocasiones el elemento a transitivizar (el menor por filas, por columnas...). Se pretende utilizar el programa con modificaciones para encontrar, bajo algún criterio, la mejor matriz, contenida en la de partida, que sea T-transitiva.

Referencias

[1] D. Boixader. On the relationship between T-transitivity and approximate equality. *Fuzzy Sets and Systems* 33, 6 – 69, 2003

[2] C. Campo; S. Cubillo; L. Garmendia; A. Salvador. A Method to Make Some Fuzzy Relations T-Transitive. *International Journal of Intelligence Systems*. Vol. 14, Nº 9, 873 – 882, 1999.

[3] L. Garmendia; A. Salvador, On a new method to T-transitivize fuzzy relations, *Information Processing and*

Management of Uncertainty in Knowledge - based Systems, IPMU 2000. 864 – 869, 2000.

[4] L. Garmendia; A.. Salvador, On a new method to T-transitivize fuzzy relations, in *Technologies for Constructing Intelligent Systems* 2, Springer. Editado por Bouchon-Meunier, B., Gutierrez-Rios, J., Magdalena, L., Yager, R. R, 251 – 260, 2000.

[5] J. Jacas, J. Recasens. Fuzzy T-transitive relations: eigenvectors and generators, *Fuzzy Sets and Systems* 72 , 147–154, 1995.

[6] J. Jacas, J. Recasens. Decomposable indistinguishability operators, *Proceedings of the Sixth IFSA Congress*, Sao Paulo, 1995.

[7] G. J. Klir; B. Yuan, *Fuzzy Sets and Fuzzy Logic. Theory and Applications*, Prentice Hall, New Jersey, 1995.

[8] H. Hashimoto, Transitivity of generalized fuzzy matrices, *Fuzzy Sets and Systems*. Vol. 17, no. 1, 83-90,1985.

[9] H. S. Lee. An optimal algorithm for computing the max–min transitive closure of a fuzzy similarity matrix , *Fuzzy Sets and Systems* 123, 129–136, 2001.

[10] F. Montero, J. Tejada, On fuzzy transitivity, Instituto Nacional de Estadística, 111, 49-58, 1986.

[11] S. Ovchinnikov, Representations of Transitive Fuzzy Relations, in Aspects of Vagueness, H. J. Skala, S. Termini y E. Trillas (Eds.), Reidel Pubs., 105-118, 1984.

[12] H. B. Potoczny. On similarity relations in fuzzy relational databases, *Fuzzy Sets and Systems* 12 (3) 231–235, 1984.

[13] B. Schweizer; A. Sklar, Probabilistic Metric Spaces, North-Holland, New York, 1983.

[14] E. Trillas; C. Alsina; J. M. Terricabras, Introducción a la lógica borrosa, Ariel Matemática, 1995.

[15] L. Valverde. On the structure of F-indistinguishability operators, *Fuzzy Sets and Systems* 17, 313–328, 1985.

[16]M. Wagenknecht. On transitive solutions of fuzzy equations, inequalities and lower approximation of fuzzy relations. Fuzzy Sets and Systems 75, 229-240, 1995.

[17] Xian Xiao, An algorithm for calculating fuzzy transitive closure, *Fuzzy Math*. 5 (4), 71–73, 1985.

[18] L.A. Zadeh. Fuzzy sets. *Inform. and Control* 8, 338–353, 1965.

[19] L. A. Zadeh. Similarity relations and fuzzy orderings, Inform. Sci. 3, 177–200, 1971.

[20] Zhen Zhao. An improved algorithm for fuzzy classification analysis, J. Lanzhou Univ. 19 (3) (1983) 160–163.

Construcción de medidas fuzzy de inclusión fuerte.

H. Bustince, V. Mohedano, F. Soria

Dpt. Automática y computación, Universidad Pública de Navarra,

Campus Arrosadía s/n, 31006 Pamplona. Spain, e-mail: bustince@unavarra.es

Resumen

En este artículo presentamos la definición de medida de inclusión fuerte difusa. Estudiamos como obtener nuevas medidas de inclusión fuerte a partir de otras y presentamos una forma de construir estas nuevas medidas de inclusión a partir de operadores de agregación y funciones reales.

Palabras clave: Medidas de Inclusión; Operadores de Agregación.

1. Introducción

En todo el trabajo representaremos con 1 al siguiente conjunto $\{< x, \mu(x) = 1 > | x \in X\}$. Denotaremos con e al punto de equilibrio de la negación fuerte considerada.

Es conocido que las dos formas más importantes de definir la inclusión entre conjuntos fuzzy son las siguientes:

A) La más usual, propuesta por Zadeh [10]:

$A \leq B$ si y sólo si $\mu_A(x_i) \leq \mu_B(x_i)$ para todo $x_i \in X$.

B) La inclusión fuerte dada por:

$$A \subseteq_S B \text{ si y sólo si } A_c \vee B = 1;$$

es decir,

$$\mu_A(x_i) = 0 \text{ o } \mu_B(x_i) = 1 \text{ para todo } x_i \in X.$$

Esta definición fue dada por D. Dubois y H. Prade en 1980 [2].

La definición de Dubois y Prade nos lleva a considerar que conjuntos genuinamente fuzzy nunca se pueden considerar perfectamente iguales, realmente lo que ocurre es lo siguiente:

$$A \subseteq_S B \text{ y } B \subseteq_S A \text{ si y sólo si}$$

para todo $x_i \in X$, $\mu_A(x_i) = \mu_B(x_i) \in \{0, 1\}$.

Estamos interesados en construir a partir de agregaciones, medidas de este último tipo, que van a ser a la vez medidas débiles de V.Young y de J. Fan [7].

1.1. Preliminares

Entre todas las definiciones de medidas de inclusión que se han dado en la Literatura difusa, una de las más utilizadas es la dada por Sinha y Dougherty.

Estos autores [6] presentan nueve axiomas para las medidas de inclusión. También presentan tres axiomas opcionales además de esos nueve. Estos tres últimos se ponen como opcionales, sin embargo ellos mantienen que para ciertas aplicaciones resultan más apropiados que alguno de los nueve anteriores.

Axioma 1. $\sigma(A, B) = 1$ si y sólo si $A \leq B$ en el sentido de Zadeh.

Axioma 2. $\sigma(A, B) = 0$ si y sólo si $\text{Ker}(A) \cap (\text{supp}(B))_c \neq \emptyset$, donde $\text{Ker}(A) = \{x \in X | \mu_A(x) = 1\}$ y $\text{supp}(B) = \{x \in X | \mu_B(x) > 0\}$; o sea, $\sigma(A, B) = 0$ si y sólo si existe $x \in X$ tal que $\mu_A(x) = 1$ y $\mu_B(x) = 0$.

Axioma 3. Si $B \leq C$, entonces $\sigma(A,B) \leq \sigma(A,C)$. En otras palabras, el indicador es una función no decreciente en la segunda variable.

Axioma 4. Si $B \leq C$, entonces $\sigma(C,A) \leq \sigma(B,A)$. En otras palabras, el indicador es una función no creciente en la primera variable.

Axioma 5. $\sigma(A,B) = \sigma[S(A),S(B)]$. si consideramos cualquier aplicación biyectiva $s : X \to X$. La operación de cambio unaria está inducida por $s : S : [0,1]^X \to [0,1]^X$ tal que $\mu_{S(A)}(x) = \mu_A[s(x)]$.

Axioma 6. $\sigma(A,B) = \sigma(B_c, A_c)$.

Axioma 7. $\sigma(B \vee C, A) = \wedge\{\sigma(B,A), \sigma(C,A)\}$ para todo $A, B, C \in \mathcal{F}(X)$.

Axioma 8. $\sigma(A, B \wedge C) = \wedge\{\sigma(A,B), \sigma(A,C)\}$ para todo $A, B, C \in \mathcal{F}(X)$.

Axiom 9. $\sigma(A, B \vee C) \geq \vee\{\sigma(A,B), \sigma(A,C)\}$ para todo $A, B, C \in \mathcal{F}(X)$.

Axiomas adicionales

Axioma 10. $\sigma(A,B) + \sigma(A,B_c) \geq 1$.

Axioma 11. Si A es un refinamiento de B (es decir, si para todo $x \in X$ se cumple: si $\mu_B(x) \geq e$, entonces $\mu_A(x) \geq \mu_B(x)$; en otro caso $\mu_A(x) \leq \mu_B(x)$), entonces $\sigma(A \vee A_c, A \wedge A_c) \leq \sigma(B \vee B_c, B \wedge B_c)$.

Axioma 12. Si A está débilmente incluido en B,(es decir, si para todo x ó $\mu_A(x) \leq e$ ó $\mu_B(x) > e$), entonces $\sigma(A,B) \geq \frac{1}{2}$.

Un estudio de esta definición de medida de inclusión y de otras como la de Kitainik, V. Young y J. Fan ha sido realizado por F. Soria en [7]. Por su utilización en este artículo destacamos las definiciones de medida de inclusión débil debidas a V. Young [9] y J. Fan [3]:

Definición 1. ([9])*Si* $\sigma_W : \mathcal{F}(X) \times \mathcal{F}(X) \to [0,1]$ *cumple:*

(a) σ_W *no satisface la siguiente condición: Si* $A \leq B$, *entonces* $\sigma_W(A,B) = 1$;

(b) *Si* $A \geq e$, *entonces* $\sigma_W(A, A_c) = 0$ *si y sólo si* $A = \{< x_i, \mu_A(x_i) = 1 > | x_i \in X\}$;

(c) *Si* $A \leq B \leq C$, *entonces* $\sigma_W(C,A) \leq \sigma_W(B,A)$ *y si* $A \leq B$, *entonces* $\sigma_W(C,A) \leq \sigma_W(C,B)$

entonces decimos que σ_W *es una medida fuzzy de inclusión débil sobre* X *en el sentido de V. Young.*

Definición 2. *(J. Fan, W. Xie, J. Pei* [3]*) Una función* $\sigma_F : \mathcal{F}(X) \times \mathcal{F}(X) \to [0,1]$ *se llama medida fuzzy de inclusión débil (en el sentido de Fan) sobre* X *, si* σ_F *satiface las siguientes propiedades:*

(1) $\sigma_F(0,0) = \sigma_F(0,1) = \sigma_F(1,1) = 1$;

(2) $\sigma_F(1,0) = 0$;

(3) *Si* $A \leq B \leq C$, *entonces* $\sigma_F(C,A) \leq \sigma_F(B,A)$ *y* $\sigma_F(C,A) \leq \sigma_F(C,B)$.

Además, la construcción que vamos a proponer, esta realizada a partir de operadores de agregación con propiedades especiales, es decir a dichos operadores les vamos a pedir alguna restricción más fuerte que las demandadas usualmene. Para nosotros, una agregación va a ser una función

$$M : [0,1]^n \to [0,1]$$

para algún $n \geq 2$, que satisface al menos las tres condiciones siguientes:

$A1$. $M(x_1, \cdots, x_n) = 0$ si y sólo si $x_i = 0$ para todo $i \in \{1, \cdots, n\}$.

$A2$. $M(x_1, \cdots, x_n) = 1$ si y sólo si $x_i = 1$ para todo $i \in \{1, \cdots, n\}$.

$A3$. Para cualquier par $(x_1, \cdots, x_n)$ e $(y_1, \cdots, y_n)$ de n-tuplas tal que $x_i, y_i \in [0,1]$ para todo $i \in \{1, \cdots, n\}$, si $x_i \leq y_i$ para todo $i \in \{1, \cdots, n\}$, entonces

$$M(x_1, \cdots, x_n) \leq M(y_1, \cdots, y_n);$$

es decir, M es monótona creciente en todos sus argumentos.

A este axioma lo denotaremos con $A3S$ cuando exijamos que el operador sea estríctamente creciente. Es decir: M es estríctamente creciente si y sólo si $x_i < y_i$ implica que

$$M(x_1, \cdots, x_i, \cdots, x_n) < M(x_1, \cdots, y_i, \cdots, x_n),$$

$$i = 1, \cdots, n.$$

2. Medidas fuzzy de inclusión fuerte sobre X

D. Dubois, W. Ostasiewicz y H. Prade [1] proponen que cuando la inclusión considerada es la fuerte, los axiomas que debe cumplir el índice de inclusión son el 2, el 3, el 4 y reemplazar el Axioma 1 por la siguiente condición:

$$\sigma(A, B) = 1 \text{ si y sólo si } A \subseteq_S B.$$

Esta propuesta nos ha sugerido estudiar las medidas fuzzy de inclusión cuando sustituimos el Axioma 1 por la condición de Dubois y Prade.

Por otra parte, teniendo en cuenta los trabajos de V. Young [9], J. Fan [3], J. A. Goguen [4] y B. Kosko [5], consideramos que debemos estudiar las medidas de inclusión fuerte difusas cuando sustituimos el Axioma 2 de Sinha y Dougherty por la condición:

$$\sigma(A, B) = 0 \text{ si y sólo si } A = 1 \text{ y } B = 0.$$

Estas consideraciones nos llevan a presentar la siguiente definición de medida fuzzy de inclusión fuerte sobre X [7].

Definición 3. *Si* $\sigma_S : \mathcal{F}(X) \times \mathcal{F}(X) \to [0, 1]$ *cumple:*

(a) $\sigma_S(A, B) = 1$ *si y sólo si* $\mu_A(x_i) = 0$ *o* $\mu_B(x_i) = 1$ *para todo* $i \in \{1, \cdots, n\}$*; es decir,* $A \subseteq_S B$*;*

(b) $\sigma_S(A, B) = 0$ *si y sólo si* $A = \{< x_i, \mu_A(x_i) = 1 > |x_i \in X\}$ *y* $B = \{< x_i, \mu_B(x_i) = 0 > |x_i \in X\}$*;*

(c) *Si* $A \leq B$*, entonces* $\sigma_S(A, C) \geq \sigma_S(B, C)$ *y* $\sigma_S(C, A) \leq \sigma_S(C, B)$*,*

entonces decimos que σ_S *es una medida fuzzy de inclusión fuerte sobre* X*.*

Ejemplo 1.

La inclusión débil propuesta por Dubois y Prade [2]:

$$\frac{1}{n}\sum_{i=1}^{n} \vee(1 - \mu_A(x_i), \mu_B(x_i))$$

La inclusión a partir de implicación [8, 9]:

$$\frac{1}{n}\sum_{i=1}^{n}(1 - \mu_A(x_i) + \mu_A(x_i) \cdot \mu_B(x_i))$$

Teorema 1. *Los siguientes items se cumplen:*

i) Toda medida de inclusión fuerte es una medida de inclusión débil en el sentido de V. Young;

ii) Toda medida de inclusión fuerte es una medida de inclusión débil en el sentido de J. Fan.

El recíproco de los dos items del Teorema anterior no es cierto. Por Ejemplo, la siguiente expresión es una medida de inclusión débil fuzzy de J. Fan y no es una medida de inclusión fuerte en el sentido de la definición anterior, pues no cumple (c).

$$\sigma_F(A, B) = \frac{\vee\left(\sum_{i=1}^{n}(1-\mu_A(x_i)), \sum_{i=1}^{n}\mu_B(x_i)\right)}{\sum^{n}\vee(\mu_A(x_i), 1-\mu_A(x_i), \mu_B(x_i), 1-\mu_B(x_i))}$$

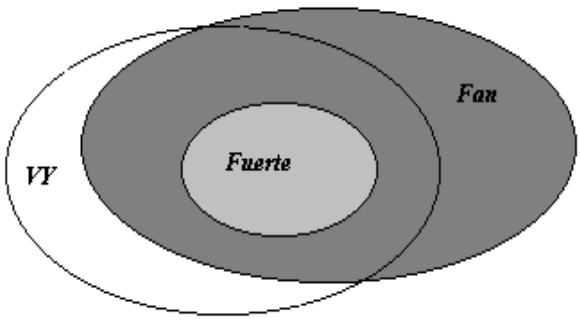


Fig. 1

De la definición anterior es fácil ver:

1) Si $A = 0$, entonces $\sigma_S(A, B) = 1$;

2) Si $B = 1$, entonces $\sigma_S(A, B) = 1$.

3) Toda medida de inclusión fuerte cumple los Axiomas 3 y 4 de Sinha y Dougherty.

Corolario 1. *Sea σ_S una medida de inclusión fuerte, los siguientes items se cumplen:*

i) σ_S satisface los Axiomas 9 y 11 de Sinha y Dougherty;

ii) σ_S satisface las desigualdades:

$$\sigma_S(A \vee B, C) \leq \wedge\{\sigma_S(A, C), \sigma_S(B, C)\}$$

$$\sigma_S(A, B \wedge C) \leq \wedge\{\sigma_S(A, B), \sigma_S(A, C)\}$$

$$\sigma_S(A \wedge B, C) \geq \vee\{\sigma_S(A, C), \sigma_S(B, C)\}.$$

Del corolario anterior deducimos que las medidas de inclusión fuerte fuzzy no cumplen el Axioma 7, pero cumplen la desigualdad:

$$\sigma_S(A \vee B, C) \leq \wedge\{\sigma_S(A, C), \sigma_S(B, C)\}.$$

Tampoco cumplen el Axioma 8 pero satisfacen la siguiente desigualdad:

$$\sigma_S(A, B \wedge C) \leq \wedge\{\sigma_S(A, B), \sigma_S(A, C)\}.$$

Proposición 1. *Sean $\sigma_{S1}, \cdots, \sigma_{Sn}$ n medidas de inclusión fuerte sobre X y sea $M : [0,1]^n \to [0,1]$ tal que cumple A1, A2 y A3. En estas condiciones*

$$\sigma_S(A, B) = \overset{n}{M}_{i=1} \sigma_{Si}(A, B)$$

es una medida de inclusión fuerte sobre X.

Ejemplo 2. Sean $\sigma_{S1}, \cdots, \sigma_{Sn}$ n medidas de inclusión fuerte sobre X y sea

$$M(x_1, \cdots, x_n) = \frac{1}{n} \sum_{i=1}^{n} x_i.$$

La siguiente expresión es una medida de inclusión fuerte.

$$\sigma_S(A, B) = \sum_{i=1}^{n} \sigma_{Si}(A, B).$$

Ejemplo 3. Sean $\sigma_{S1}, \cdots, \sigma_{Sn}$n medidas de inclusión fuerte sobre X y sea

$$M(x_1, \cdots, x_n) = \frac{\bigvee_{i=1}^{n} x_i}{\bigvee_{i=1}^{n} x_i + \bigvee_{i=1}^{n} 1 - x_i}.$$

La siguiente expresión es una medida de inclusión fuerte sobre X.

$$\sigma_S(A, B) = \frac{\bigvee_{i=1}^{n} \sigma_{Si}(A, B)}{\bigvee_{i=1}^{n} \sigma_{Si}(A, B) + \bigvee_{i=1}^{n} \sigma_{Si}(A, B)}.$$

Proposición 2. *Sean $\sigma_{S1}, \cdots, \sigma_{Sn}$ n medidas de inclusión fuerte sobre X y sea $M : [0,1]^n \to [0,1]$ tal que cumple A2, A3 y la siguiente propiedad:*

$M(x_1, \cdots, x_n) = 0$ si y sólo si existe al menos un $x_i = 0$.

En estas condiciones

$$\sigma_S(A, B) = \overset{n}{M}_{i=1} \sigma_{Si}(A, B)$$

es una medida de inclusión fuerte sobre X.

Ejemplo 4. Sean $\sigma_{S1}, \cdots, \sigma_{Sn}$ n medidas de inclusión fuerte sobre X. Entonces

$$\sigma_S(A, B) = \bigwedge_{i=1}^{n} \sigma_{Si}(A, B)$$

$$\sigma_S(A, B) = \prod_{i=1}^{n} \sigma_{Si}(A, B)$$

son medidas de inclusión fuerte.

Proposición 3. *Sea σ'_S una medida fuzzy de inclusión fuerte sobre X,entonces los siguientes items se cumplen:*

i) $\sigma_S(A,B) = \wedge(\sigma'_S(A,B), \sigma'_S(B_c, A_c))$ *es una medida de inclusión fuerte sobre X;*

ii) $\sigma_S(A,B) = \sigma'_S(A,B) \cdot \sigma'_S(B_c, A_c)$ *es una medida de inclusión fuerte sobre X.*

3. Costrucción de medida fuzzy de inclusión fuerte sobre X

En esta sección presentamos un método para construir medidas de inclusión fuerte sobre X.

Proposición 4. *Sea $M : [0,1]^n \to [0,1]$ tal que cumple $A1$ y $A3S$,y sean las funciones*

$$g, h : [0,1]^2 \to [0,1] \text{ tales que}$$

(1) $g(x,y) \leq h(x,y)$ *para todo* $x, y \in [0,1]$;

(2) $g(x,y) = h(x,y)$ *si y sólo si* $x = 0$ *o* $y = 1$;

(3) $g(x,y) = 0$ *si y sólo si* $x = 1$ *e* $y = 0$;

(4) *Si* $x \leq y$, *entonces* $\begin{cases} g(z,x) \leq g(z,y) \\ g(y,z) \leq g(x,z) \\ h(z,y) \leq h(z,x) \\ h(x,z) \leq h(y,z). \end{cases}$

En estas condiciones

$$\sigma_S : \mathcal{F}(X) \times \mathcal{F}(X) \to [0,1] \text{ dada por}$$

$$\sigma_S(A,B) = \frac{\overset{n}{\underset{i=1}{M}}(g(\mu_A(x_i), \mu_B(x_i)))}{\overset{n}{\underset{i=1}{M}}(h(\mu_A(x_i), \mu_B(x_i)))},$$

es una medida de inclusión fuerte sobre X.

Demostración. Bajo las hipótesis de la proposición nunca puede suceder que

$$h(\mu_A(x_i), \mu_B(x_i)) = 0.$$

Si fuera cierto, es decir, si $h(\mu_A(x_i), \mu_B(x_i) = 0$, entonces por (1) sabemos que

$$g(\mu_A(x_i), \mu_B(x_i)) = 0,$$

por tanto por (2) $\mu_A(x_i) \leq \mu_B(x_i)$ se cumple y por (3) tenemos $\mu_A(x_i) = 1$ and $\mu_B(x_i) = 0$. Es decir, contradicción.

(a) (Necesidad) Si $\sigma_S(A,B) = 1$, entonces

$$\overset{n}{\underset{i=1}{M}}(g(\mu_A(x_i), \mu_B(x_i))) = \overset{n}{\underset{i=1}{M}}(h(\mu_A(x_i), \mu_B(x_i))).$$

Por hipótesis sabemos que $g(x,y) \leq h(x,y)$. Como M satisface $A3S$ tenemos que si existe un x_i tal que $g(\mu_A(x_i), \mu_B(x_i)) < h(\mu_A(x_i), \mu_B(x_i))$, entonces

$$\overset{n}{\underset{i=1}{M}}(g(\mu_A(x_i), \mu_B(x_i))) < \overset{n}{\underset{i=1}{M}}(h(\mu_A(x_i), \mu_B(x_i))).$$

Por tanto para todo $i \in \{1, \cdots, n\}$ tenemos

$g(\mu_A(x_i), \mu_B(x_i)) = h(\mu_A(x_i), \mu_B(x_i))$, por (2) tenemos $\mu_A(x_i) = 0$ or $\mu_B(x_i) = 1$ para todo $i \in \{1, \cdots, n\}$.

(Suficiencia) Si $\mu_A(x_i) = 0$ ó $\mu_B(x_i) = 1$ para todo $i \in \{1, \cdots, n\}$, entonces por (2) sabemos $g(\mu_A(x_i), \mu_B(x_i)) = h(\mu_A(x_i), \mu_B(x_i))$, por tanto $\sigma_S(A,B) = 1$.

(b) Como M satisface $A1$ tenemos

$\sigma(A,B) = 0$ sí y sólo sí $\overset{n}{\underset{i=1}{M}}(g(\mu_A(x_i), \mu_B(x_i))) = 0$ sí y sólo sí $g(\mu_A(x_i), \mu_B(x_i)) = 0$ para todo $i \in \{1, \cdots, n\}$ sí y sólo sí $\mu_A(x_i) = 1$ y $\mu_B(x_i) = 0$ para todo $i \in \{1, \cdots, n\}$.

(c) Si $A \leq B$, entonces por (4) tenemos

$$\sigma(A,C) = \frac{\overset{n}{\underset{i=1}{M}}(g(\mu_A(x_i), \mu_C(x_i)))}{\overset{n}{\underset{i=1}{M}}(h(\mu_A(x_i), \mu_C(x_i)))} \geq \frac{\overset{n}{\underset{i=1}{M}}(g(\mu_B(x_i), \mu_C(x_i)))}{\overset{n}{\underset{i=1}{M}}(h(\mu_A(x_i), \mu_C(x_i)))} \geq$$

$$\frac{\overset{n}{\underset{i=1}{M}}(g(\mu_B(x_i), \mu_C(x_i)))}{\overset{n}{\underset{i=1}{M}}(h(\mu_B(x_i), \mu_C(x_i)))} = \sigma(B,C).$$

$\sigma(C,A) =$

$$\frac{\underset{i=1}{\overset{n}{M}}(g(\mu_C(x_i),\mu_A(x_i)))}{\underset{i=1}{\overset{n}{M}}(h(\mu_C(x_i),\mu_A(x_i)))} \leq \frac{\underset{i=1}{\overset{n}{M}}(g(\mu_C(x_i),\mu_B(x_i)))}{\underset{i=1}{\overset{n}{M}}(h(\mu_C(x_i),\mu_A(x_i)))} \leq$$

$$\frac{\underset{i=1}{\overset{n}{M}}(g(\mu_C(x_i),\mu_B(x_i)))}{\underset{i=1}{\overset{n}{M}}(h(\mu_C(x_i),\mu_B(x_i)))} = \sigma(C,B). \qquad \blacksquare$$

Ejemplo 5. Sean las funciones

$$g(x,y) = \frac{1-x+x\cdot y}{4}$$

$$h(x,y) = \begin{cases} \frac{1}{4} \text{ Si } x=0 \text{ o } y=1 \\ \frac{1}{2} \text{ en otro caso} \end{cases}$$ Es fácil ver que

estas funciones cumplen las condiciones (1)-(4) de la proposición anterior. Si tomamos $M(x_1,\cdots,x_n) = \frac{1}{n}\sum_{i=1}^{n} x_i$, esta función cumple $A1$ y $A3S$, entonces

$$\sigma_S(A,B) = \frac{\frac{1}{n}\sum_{i=1}^{n}\frac{1-\mu_A(x_i)+\mu_A(x_i)\cdot\mu_B(x_i)}{4}}{\frac{1}{n}\sum_{i=1}^{n}\begin{cases} \frac{1}{4} \text{ Si } \mu_A(x_i)=0 \text{ o } \mu_B(x_i)=1 \\ \frac{1}{2} \text{ en otro caso} \end{cases}}$$

4. Conclusiones.

A partir de las distintas definiciones que existen de medida de inclusión, y de la idea introducida por Dubois y Prade de conjunto fuertemente incluido, presentamos una nueva definición de medida de inclusión fuerte sobre X. Probamos que esta nueva medida de inclusión es siempre una medida de inclusión débil en el sentido de V. Young y de J. Fan. El recproco no es cierto. También probamos que estas nuevas medidas satisfacen ciertos axiomas de Sinha y Dougherty y las desigualdades que Dubois y Prade analizan para las medidas de inclusión.

Probamos que si agregamos medidas de inclusión fuerte, lo que obtenemos es también una medida de este tipo.

Además presentamos un método de construcción de estas medidas a partir de operadores de agregación y de funciones de $[0,1]^2$ en $[0,1]$.

Dejamos para futuros trabajos realizar un estudio similar al presentado con las medidas de inclusión de Kitainik.

Referencias

[1] D. Dubois, W. Ostasiewicz, H. Prade, Fuzzy sets: History and basic notions, in: D. Dubois, H. Prade (Eds), Fundamentals of Fuzzy Sets, Kluwer, Boston, MA, 2000, pp. 21-124.

[2] D. Dubois and H. Prade, Fuzzy Sets and Systems: Theory and Applications, New York: Academic, 1980.

[3] J. Fan, X. Xie and J. Pei, Subsethood measures: new definitions. *Fuzzy Sets and Systems*, **106** (1999), 201-209.

[4] J. A. Goguen, The logic of inexact concepts, *Synthese*, **19**, (1969) 325-373.

[5] B. Kosko, Fuzzy entropy and conditioning. *Inform. Sci.*, **40** (1986), 165-174.

[6] D. Sinha and E.R. Dougherty, Fuzzification of set inclusion: Theory and applications, *Fuzzy Sets and Systems* **55** (1993) 15-42.

[7] F. Soria, "Axiomatización de las medidas de inclusión difusas de V. Young. Construcción a partir de operadores de agregación e implicación. Otros índices.", Ph. D. Thesis, Universidad Pública de Navarra, (2003).

[8] R. Willmott, Mean measures of containment and equality between fuzzy sets. Proc. 11th Internat. Symp. on Multiple-Valued Logic, Oklahoma City, Oklahoma, 1981, pp. 183-190.

[9] V. R. Young, Fuzzy subsethood. *Fuzzy Sets and Systems*, **77** (1996), 371-384.

[10] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility *Fuzzy Sets and Systems* **1** (1978), 3-28.

An Approach for Implementing Fuzzy Relational Databases in Classic DBMS*

José Galindo
Dpto. Lenguajes y Ciencias de la Computación, Univ. de Málaga, Spain
ppgg@lcc.uma.es

Angélica Urrutia
Dpto. de Computación e Informática, Univ.Católica del Maule, Chile
aurrutia@spock.ucm.cl

Mario Piattini
Escuela Superior de Informática, Univ. de Castilla la Mancha, Spain
Mario.Piattin@uclm.es

Abstract

In this article we present how to implement fuzzy databases based on the relational model. This approach includes many fuzzy attribute types, which can express the most of fuzzy knowledge types. These fuzzy attribute types include imprecise attributes, fuzzy attributes associated to one or more attributes or with an independent meaning. In order to represent such fuzzy information we must study two aspects of fuzzy information: how to represent fuzzy data and how to represent fuzzy metaknowledge data. This second information is very important and it must be considered in any fuzzy database. This article studies the fuzzy metaknowledge data for any fuzzy attribute and how to represent both in a relational database.

Keywords: Fuzzy relational databases, Fuzzy attributes, Fuzzy degrees, Fuzzy metaknowledge, Representation of Fuzzy Knowledge.

1. Introduction

The relational model was developed by E.F. Codd of IBM and published in 1970. This model is the most used at present, in fact, actually the 90% of databases are relational. On a theoretical level, there exists many Fuzzy Relational Database models that, based on the relational model, they extend it in order to allow storing and/or treating vague and uncertain information [12].

On the other hand, the FuzzyEER model [6][8][9][13][14] is an extension of the EER model to create conceptual schemas with fuzzy semantics and notations. This extension is a good eclectic synthesis among the different models and it provides new and useful definitions: fuzzy attributes, fuzzy entities, fuzzy relationships, fuzzy specializations…

In this paper we propose to incorporate the FuzzyEER concepts in a relational DBMS. Fuzzy EER model includes fuzzy attributes (see Section 2), fuzzy aggregations, fuzzy entities, fuzzy weak entities, fuzzy relationships, fuzzy specializations and fuzzy constraints. Our aim is to present this extension as simple and useful as possible. Then, we have extended the FIRST definitions [4][10][11], which have been used in some applications [1][3].

The next section defines the fuzzy attributes included in the FuzzyEER model. After, we define how to represent fuzzy data and how to represent fuzzy metaknowledge data. Finally, concluding remarks and future developments are discussed.

2. Fuzzy Attributes

In order to model fuzzy attributes we distinguish between two classes of fuzzy attributes: Fuzzy attributes whose fuzzy values are fuzzy sets and fuzzy attributes whose values are fuzzy degrees.

2.1. Fuzzy Sets as Fuzzy Values

These fuzzy attributes may be classified in four types, based on the definitions of [4][9][10][11]. This classification is performed taking into account the type of referential or underlying domain. In all of them the values Unknown, Undefined, and Null are included:

- **Type 1**: These are attributes with "precise data", classic or crisp (traditional, with no imprecision). However, they can have linguistic labels defined over them and we can use them in fuzzy queries. This type of attribute is represented in the same way as precise data, but can be transformed or manipulated using fuzzy conditions. This type is useful for extending traditional databases allowing fuzzy queries to be made about classic data.
  Example: the salary must be a crisp attribute. However, we could be interested in fuzzy queries on this attribute: Enquiries of the kind "Give me employees that earn a lot more than the minimum salary".

* Partially supported by TIC2002-00480 project.

- **Type 2**: These are attributes that gather "*imprecise data over an ordered referential*". These attributes admit both crisp and fuzzy data, in the form of possibility distributions over an underlying ordered dominion (fuzzy sets). It is an extension of the Type 1 that does, now, allow the storage of imprecise information. For the sake of simplicity the most complex of these fuzzy sets are supposed to be a trapezoidal function (Figure 1).
  Example: fuzzy attribute height can store fuzzy information such as: "he is approximately 2 metres tall".

- **Type 3**: They are attributes over "*data of discreet non-ordered dominion with analogy*". In these attributes some labels are defined ("blond", "ginger", "brown", etc.) that are scalars with a similarity (or proximity) relationship defined over them, so that this relationship indicates to what extent each pair of labels resemble each other. They also allow possibility distributions (or fuzzy sets) over this dominion.
  Example: Let us suppose a fuzzy attribute hair_color. This attribute may store, for example, the value {1/dark, 0.4/brown}, which expresses that a certain person is more likely to be dark than brown-haired. Note that underlying domain of these fuzzy sets are the set of labels and this set is non-ordered.

- **Type 4**: These attributes are original and they are defined in the same way as Type 3 attributes, without it being necessary for a similarity relationship to exist between the labels. In this case, we suppose that we do not need the similarity relationship (or it does not exist).

2.2. Fuzzy Degrees as Fuzzy Values

The domain of these degrees can be found in the interval [0,1], although other values are also permitted, such as a possibility distribution (usually over this unit interval). In order to keep it simple, we will only use degrees in the interval [0,1], because the other option offers no great advantages.

The meaning of these degrees is varied and depends on their use. The processing of the data will be different depending on the meaning. The most important possible meanings of the degrees used by some authors are [4][5]: Fulfillment degree, Uncertainty degree, Possibility degree and Importance degree. Of course, we can define and use other meanings.

The ways of using these fuzzy degrees are classified in two families: Associated and non-associated degrees.

Associated degrees are associated to a specific value to which they incorporate imprecision. These degrees may be associated to different concepts [5]:

- **Degree in each value of an attribute (**we will call it as **Type 5):** Some attributes may have a fuzzy degree associated to them. This implies that each value of this attribute (in every tuple or instance) has an associated degree, that measures the level of fuzziness of that value. In order to interpret it, we need to know the meaning of the degree and the meaning of the associated attribute.

- **Degree in a set of values of different attributes (Type 6):** Here, the degree is associated to some attributes. Whilst this is an unusual case, it can sometimes be very useful. It joins the fuzziness of some attributes in only one degree.

- **Degree in the whole instance of the relation (Type 7):** This degree is associated to the whole tuple of the relation and not exclusively to the value of a specific attribute of the tuple (or instance). Usually, it can represent something like the "membership degree" of this tuple (or instance) to the relation (or table) of the database.

Non-associated degrees (Type 8): There are cases in which the imprecise information, which we wish to express, can be represented by using only the degree, without associating this degree to another specific value or values. For example, the dangerousness of a medicine may be expressed by a fuzzy degree.

In this paper we do not aim to demonstrate the usefulness of these degrees and their different meanings. Several authors who have used these degrees have already done so.

3. Representation of Fuzzy Attributes

This representation is different according to the fuzzy attribute Type. Fuzzy attributes Type 1 are represented as usual attributes, because they do not allow fuzzy values. Fuzzy attributes Type 2 need five classic attributes: One stores the kind of value (Table 1) and the others four store the crisp values representing the fuzzy value. Note, in Figure 1 and Table 1, that trapezoidal fuzzy values need the others four values. An approximate value (approximately d, $d\pm$margin) is represented with a triangular function centered in d (degree 1) and with degree 0 in $d-$margin and $d+$margin, where value margin depend on the context (Figure 1 with b=c and b−a=d−c=margin).

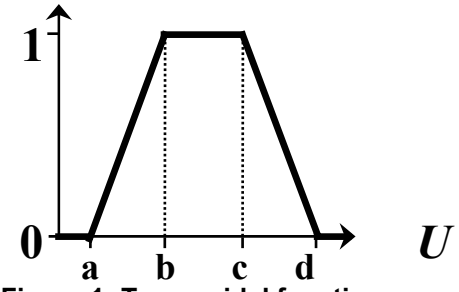


Figure 1: Trapezoidal function.

Number	Kind of values
0, 1, 2	**UNKNOWN, UNDEFINED, NULL**
3	**CRISP:** *d*
4	**LABEL:** label_identifier
5	**INTERVAL:** [*n*,*m*]
6	**APPROXIMATE VALUE:** *d*
7	**TRAPEZOIDAL:** [a,b,c,d]

Table 1: Kind of values of fuzzy attributes Type 2.

Number	Kind of values
0, 1, 2	**UNKNOWN, UNDEFINED, NULL**
3	**SIMPLE:** Degree/Label
4	**POSSIBILITY DISTRIBUTION:** $Degree_1/label_1 + ... + Degree_n/Label_n$

Table 2: Kind of values of fuzzy attributes Type 3 and 4.

Fuzzy attributes Type 3 need a variable number of attributes: One stores the kind of value (Table 2). Note, in Table 2, that number 3 needs only two values, but number 4 need 2*n* values, where *n* is the maximum length for possibility distributions for each fuzzy attribute. Value *n* must be defined for each fuzzy attribute Type 3, and it is stored in the FMB (see following section).

Fuzzy attributes Type 4 are represented just like Type 3. The different between them is shown in the next section. Fuzzy degrees (Types 5, 6, 7 and 8) are represented using a classic numeric attribute, because their domain is the interval [0,1].

4. Representation of Fuzzy Metaknowledge Data: The FMB

Fuzzy metaknowledge data are the necessary knowledge about the fuzzy database (fuzzy attributes specially). This information is stored in relational format in the so-called FMB (Fuzzy Metaknowledge Base). First, we define the information stored in the FMB, and then we explain the structure of it (i.e., the relations of the FMB).

4.1. Information in the FMB

The FMB include the following information:

1. Attributes with fuzzy capabilities: fuzzy attributes and fuzzy degrees (Types 1 to 8).

2. The metaknowledge of each attribute is different according to its type:

   - Types 1 and 2: These fuzzy attributes store in the FMB the definition (fuzzy set) of each linguistic label, the "margin" for approximate values, and the minimum distance to consider two values as very separated (so-called "much"). This last value is used in comparisons like "much greater than".

   - Types 3 and 4: Value *n* (explained above), name of linguistic labels and, only for Type 3, the similarity relationship between whatever two labels (symmetrical or non-symmetrical).

   - Types 5 and 6: Meaning of the degree and attribute (Type 5) or attributes (Type 6) to which the degree is associated.

   - Types 7 and 8: Meaning of the degree.

3. Other objects or information: These objects include fuzzy qualifiers (associated to an attribute and used to set the threshold in queries) and fuzzy quantifiers (associated to a relation or to an attribute). Fuzzy quantifiers are used in queries (for example "Give me employees who belong to *most* of projects"), and in fuzzy constraints (for example "An employee must work in *many* projects").

   If two fuzzy attributes (Types 1, 2, 3 or 4) need the same definitions we can register these two attributes as compatibles. This simplifies data in the FMB.

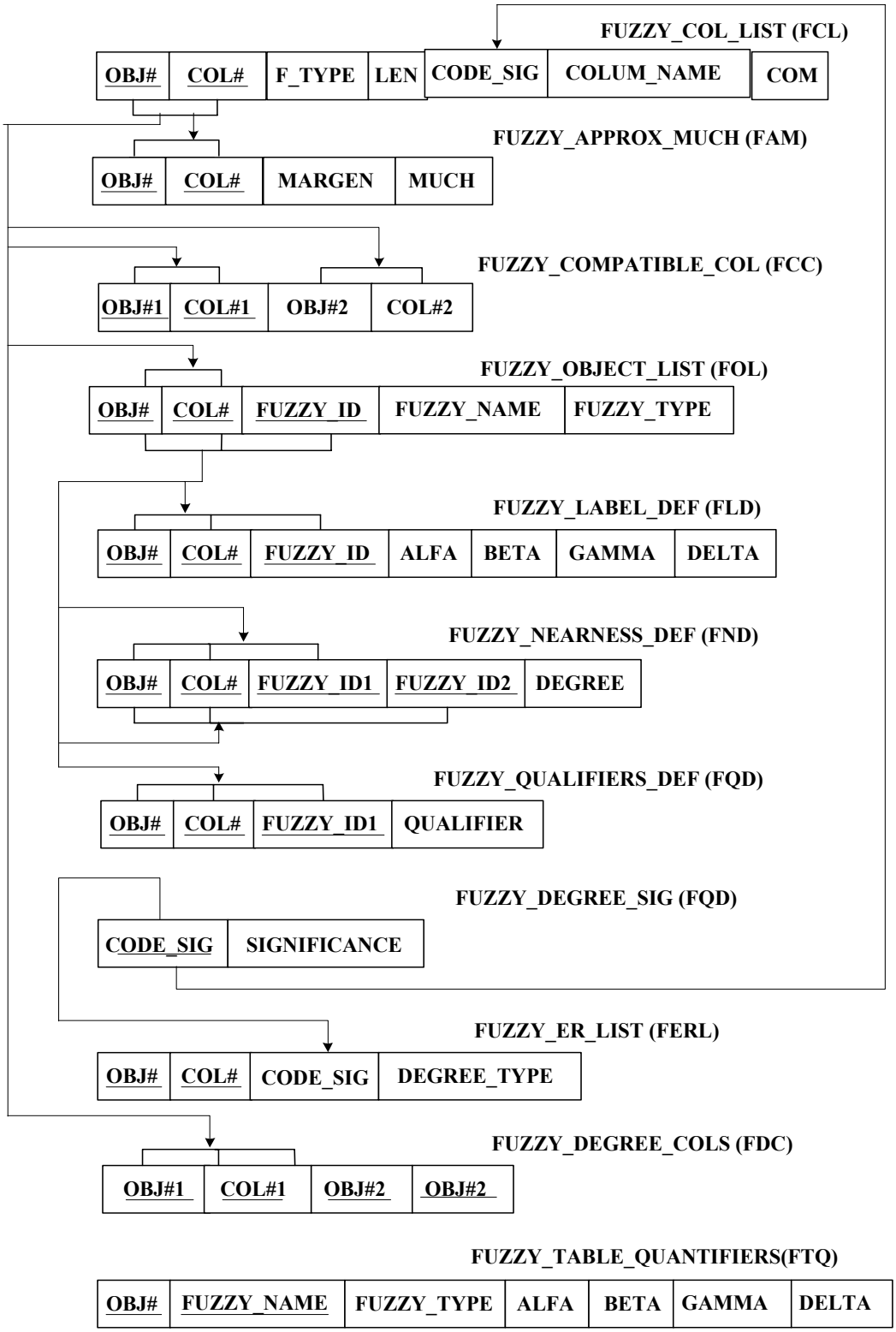


Figure 2: FMB tables in FIRST-2.

4.2. Relations in the FMB

Figure 2 shows the FMB relations (or tables), its attributes, its primary keys (underlined) and its foreign keys (with arrows). We use OBJ# as the relation identifier, and COL# as the column or attribute identifier (just like Oracle). We cannot explain all attributes of all FMB relations for lack of space. Then we only try to give an idea about the usefulness of each relation:

- **FUZZY_COL_LIST:** It describes fuzzy attributes identified by (OBJ#,COL#). F_TYPE set the fuzzy type (from 1 to 8). LEN is the value *n*. CODE_SIG indicates the degree meaning when F_TYPE∈[5,8]. This table should include the unit of measurement of the attribute.

- **FUZZY_DEGREE_SIG:** It stores all the degree meanings of our database.

- **FUZZY_OBJECT_LIST:** This relation contains declarations of fuzzy objects related with fuzzy attributes. These fuzzy objects are: linguistic labels, qualifiers and fuzzy quantifiers. Linguistic labels are classified according to the fuzzy attribute type. Labels of fuzzy attributes Type 3 are also classified according to their similarity relation: symmetrical or non-symmetrical (proximity relation). Fuzzy quantifiers are classified in 10 types according to their scope (absolute or relative), to their number of arguments (0, 1 or 2) and to their building type (sum or product). Building type sets how to obtain the final quantifiers starting from the values [α, β, γ, δ] and the arguments *x* and *y*. See Table 3, where each type may also be absolute or relative.

Num. of Arguments	Building Type	Final Quantifier
0	-	[α, β, γ, δ]
1	Sum	[α+*x*, β+*x*, γ+*x*, δ+*x*]
1	Product	[α**x*, β**x*, γ**x*, δ**x*]
2	Sum	[α+*x*, β+*x*, γ+*y*, δ+*y*]
2	Product	[α**x*, β**x*, γ**y*, δ**y*]

Table 3: Fuzzy Quantifiers Types.

- **FUZZY_LABEL_DEF:** It defines the linguistic labels using trapezoidal functions (Figure 1).
- **FUZZY_APPROX_MUCH:** Values "margin" and "much" for Types 1 and 2.
- **FUZZY_NEARNESS_DEF:** Similarity relationships for Type 3.
- **FUZZY_COMPATIBLE_COL:** Compatible fuzzy attributes, i.e., attributes which use the same linguistic labels.
- **FUZZY_QUALIFIERS_DEF:** It defines fuzzy qualifiers.
- **FUZZY_DEGREE_COLS:** This relation sets the attributes (or columns) associated to fuzzy degrees (only for Type 5 and 6). Note that a Type 5 degree has only one associated attribute, a Type 6 degree has some attributes and an attribute may have many degrees associated to it (but these degrees must be Type 5 or 6). Of course Type 7 and 8 degrees do not use this table.
- **FUZZY_ER_LIST:** Using FuzzyEER words, this relation stores fuzzy entities and fuzzy relationships. DEGREE_TYPE take 'M' for fuzzy entities, 'C' for fuzzy entities with degrees computed automatically, 'E' and 'I' for fuzzy weak entities (dependency on existence or dependency on identification) and, finally, 'R' for fuzzy relationships represented by the table OBJ#.
- **FUZZY_TABLE_QUANTIFIERS:** Definition of quantifiers associated to a relation or table (not to an attribute). These quantifiers are used in fuzzy constraints and they may be of ten types, as it was indicated previously.

We can include another table for the "system quantifiers". The system quantifiers are global quantifiers, i.e., quantifier which definition is global to the system. Some quantifiers are very dependent on the context, and so they are not good system quantifiers. System quantifiers should be relative, or absolute with one or two arguments and type product because these types are not very dependent on the context (particularly the relative quantifiers).

5. Conclusions and Future Lines

This article presents how to store fuzzy knowledge of fuzzy databases in a classic relational database. This allows us to implement fuzzy databases modeled with the FuzzyEER model [6][8][13][14]. It should be stressed the fuzzy attributes types, which can express the most of fuzzy knowledge types.

This research studies how to represent fuzzy data, the necessary metaknowledge about these fuzzy data and how to represent this fuzzy metaknowledge data. This second information is very important and must be considered in any fuzzy database. Actually, we have developed fuzzy databases with some of these characteristics [4]. Certain approaches sharing this objective, such as [2], have focused on queries, instead of on representation issues.

Besides, FSQL (Fuzzy SQL) language may be used in those databases [3][9]. In addition, we have developed a

free software called Visual FSQL [7], witch facilitates the access to fuzzy databases through FSQL.

The future is extend FSQL language, in order to treat with all fuzzy attribute types presented here. In [9] new versions of FuzzyEER and FSQL will be presented, including a better explanation about these concepts.

Acknowledgements

The authors would like to thank MCYT project TIC2002-00480 for their partial support, which contributed to the final formulation of this work.

References

[1] Blanco I., Cubero J.C., Pons O., Vila M.A. (2000): "An Implementation for Fuzzy Deductive Relational Databases". In "Recent Issues on Fuzzy Databases", Ed. G. Bordogna and G. Pasi. Physica-Verlag (Studies in Fuzziness and Soft Computing), pp. 183-207.

[2] Bosc P., Galibourg M., "Indexing principles for a fuzzy data base". Inf. Systems, Vol. 14-6, pp. 493-499, 1989.

[3] Galindo J., Medina M., Pons O., Cubero J. C., "A Server for Fuzzy SQL Queries". In "Flexible Query Answering Systems", Eds. T. Andreasen, H. Christiansen and H.L. Larsen, Lecture Notes in Artificial Intelligence (LNAI) 1495, pp. 164-174. Ed. Springer, 1998.

[4] Galindo J., "Tratamiento de la Imprecisión en Bases de Datos Relacionales: Extensión del Modelo y Adaptación de los SGBD Actuales". Tesis Ph. Doctoral Universidad de Granada, España, 1999 (www.lcc.uma.es).

[5] Galindo J., Medina J.M., Cubero J.C., García M.T., "Relaxing the Universal Quantifier of the Division in Fuzzy Relational Databases". International Journal of Intelligent Systems, 16(6), pp. 713-742, 2001.

[6] Galindo J., Urrutia A., Carrasco R., Piattini M., "Fuzzy Constraints using the Enhanced Entity-Relationship Model". Proceedings published by IEEE-CS Press of the XXI Int. Conf. of the Chilean Computer Science Society (SCCC 2001), pp. 86-94. Punta Arenas (Chile), 2001.

[7] Galindo J., Oliva R.F., Carrasco R.A., "Acceso Web a Bases de Datos Difusas: Un Cliente Visual de Fuzzy SQL". XIII Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF'2004). Jaén (Spain), September 2004.

[8] Galindo J., Urrutia A., Carrasco R.A., Piattini M., "Relaxing Constraints in Enhanced Entity-Relationship Models using Fuzzy Quantifiers". To appear in IEEE Transactions on Fuzzy Systems, 2004/2005.

[9] Galindo J., Urrutia A., Piattini M., "Fuzzy Databases: Modeling, Design and Implementation". To publish by Idea Group Publishing Hershey, USA, 2005.

[10] Medina J.M., "Bases de datos Relacionales Difusas: Modelo Teórico y Aspectos de su Implementación", Ph. Tesis Doctoral, Universidad de Granada, España, 1994 (decsai.ugr.es).

[11] Medina J.M., Pons O., Vila A., "FIRST. A Fuzzy Interface for Relational SysTems". VI International Fuzzy Systems Association World Congress (IFSA 1995). Sao Paulo (Brasil), 1995.

[12] Petry F.E., "Fuzzy Databases: Principles and Applications". *International Series in Intelligent Technologies*. Ed. H.J. Zimmermann. Kluwer Academic Publ. (KAP), 1996.

[13] Urrutia A., Galindo J., Piattini M., "Modeling Data Using Fuzzy Attributes". Proceedings published by IEEE Computer Society Press of the XXII International Conference of the Chilean Computer Science Society (SCCC 2002), pp. 117-123. Copiapo (Chile), 2002. ISBN: O-7695-1867-2.

[14] Urrutia A., "Definición de un Modelo Conceptual para Bases de Datos Difusas". Ph. Doctoral Thesis, University of Castilla la Mancha (Spain), 2003.

Acceso Web a Bases de Datos Difusas: Un Cliente Visual de Fuzzy SQL

José Galindo
Dpto. Lenguajes y Ciencias de la Computación, Universidad de Málaga
ppgg@lcc.uma.es

Rafael F. Oliva
Ingeniero Superior en Informática, por la Universidad de Málaga
rafaoliva@terra.es

Ramon A. Carrasco
Universidad de Granada
dmFSQL@terra.es

Resumen

En este artículo presentamos el desarrollo de una aplicación web para acceder a ella por internet y a través de la cual se puedan realizar consultas a una base de datos difusa localizada en un servidor remoto. Con esta aplicación podremos conectarnos al servidor de Bases de Datos y realizar consultas difusas mediante el lenguaje FSQL (Fuzzy SQL). Esto no significa necesariamente que el usuario deba conocer dicho lenguaje, puesto que las consultas se realizarán a través de una interfaz visual fácil de utilizar, por lo que se posibilita notablemente que cualquier persona pueda manejar el programa. También podremos realizar consultas directamente en SQL y FSQL para el que así lo desee. La consulta en FSQL es procesada por el Servidor FSQL que la analiza y la traduce a SQL.

Palabras clave: Bases de Datos Difusas, Fuzzy SQL, Consultas Difusas, Consultas Flexibles, Consultas Visuales.

1. INTRODUCCIÓN

Las bases de datos difusas han sido ampliamente estudiadas [4][8][9][13]. No obstante, aún queda mucho por avanzar en este campo, como lo demuestran recientes investigaciones en el ámbito, por ejemplo, del modelado conceptual [7][14].

Trabajos anteriormente desarrollados nos permiten disponer de un lenguaje de consulta difuso basado en el popular SQL, que llamamos FSQL o Fuzzy SQL. Además, disponemos de un Servidor FSQL para Oracle que nos permite utilizar este lenguaje [3][4][6][8]. Este desarrollo se basa en el modelo de bases de datos difusas GEFRED y su implementación FIRST [9][10], para SGBD relacionales clásicos.

La utilidad del lenguaje FSQL y del Servidor FSQL ha sido demostrada en algunas publicaciones [4][5].

Uno de los problemas con los que nos encontramos es la dificultad en el acceso a bases de datos difusas por parte de usuarios habituales de las bases de datos. Aunque los conceptos elementales sobre lógica difusa y bases de datos difusas son bastante intuitivos, surgen nuevos conceptos, nuevas técnicas y nuevos elementos que el usuario debe aprender para sacar el máximo partido a una base de datos difusa, obteniendo las ventajas que este tipo de bases de datos aporta respecto a las bases de datos tradicionales. Existen propuestas en bases de datos tradicionales, usando SQL [1].

Para solucionar ese problema surgió el proyecto Visual FSQL que aquí presentamos [12]. Visual FSQL es una aplicación web diseñada para el acceso a bases de datos difusas. Su objetivo es ir guiando al usuario por los pasos necesarios para generar una consulta FSQL sin necesidad de prácticamente escribir nada. A golpe de ratón se irá confeccionando la sentencia SELECT de FSQL para, finalmente proceder a su ejecución y a la obtención de los resultados pertinentes. El programa Visual FSQL es gratuito y puede encontrarse en la web que se indica en la referencia bibliográfica [12], incluyendo tanto el programa como su ayuda y su documentación, así como una introducción a la lógica difusa, a las bases de datos difusas y a la creación de programas para internet.

En la siguiente sección exponemos brevemente las principales características del programa Visual FSQL. A continuación nos centraremos en la utilidad que consideramos más importante: la realización de consultas difusas visuales. Finalmente presentamos unas conclusiones, trabajos futuros y las referencias bibliográficas.

2. PRINCIPALES CARACTERÍSTICAS DE VISUAL FSQL

Nos parece importante comentar que Visual FSQL es un "cliente" FSQL, es decir, es un programa que efectúa sentencias FSQL que requieren llamadas al Servidor FSQL, el cual debe estar instalado. Sin embargo, desde otro punto de vista Visual FSQL es un programa "servidor", ya que se instala en un servidor web y da servicio a los distintos navegadores clientes que accedan a la aplicación.

A modo de resumen podemos decir que las principales características del programa Visual FSQL son las siguientes:

- Es un programa para ejecución a través de internet, lo que implica la disponibilidad universal sin necesidad de instalación de ningún cliente ni componente.
- Es un programa de plataforma abierta, por lo que tenemos un elevado potencial de clientes que se pueden conectar y acceder al sistema. No tenemos la limitación de una plataforma concreta, puesto que cualquier navegador web con soporte Javascript es totalmente operativo.
- Accesibilidad multiusuario y concurrencia soportada.
- Muy flexible en cuanto a escalabilidad y mantenimiento, puesto que cualquier modificación se realiza de forma centralizada en el servidor de la aplicación, sin tener que actualizar los clientes que existan.
- Mínimos requisitos de hardware, únicamente supeditados a las necesidades de Oracle si se instala éste en el mismo servidor. Para el Visual FSQL son necesarios un mínimo de 5 Mb. de espacio en disco y 128 Mb. de RAM, basado fundamentalmente en los requisitos recomendados para el Servidor Web IIS "Internet Information Server" de Microsoft, que es el encargado de dar servicio a la Aplicación, aunque puede usarse otro.
- También existe la posibilidad de balancear la carga del sistema separándolo en 2 máquinas: por un lado el servidor web IIS y la Aplicación Visual FSQL y por otro el Servidor de Bases de Datos Oracle y el sistema Servidor FSQL.
- Identificación mediante clave de acceso (gestión de seguridad), basada en el SGBD Oracle.
- Fácil acceso a "información de la conexión al SGBD": versión actual del SGBD, datos del usuario, roles y privilegios del usuario...
- Interfaz de usuario sencilla y guiada a través de asistentes y ayudas para la construcción de consultas difusas ó clásicas. Esto incluye cuestiones tales como:
  - Fácil selección de los atributos (o columnas) y de las tablas o vistas involucradas en la consulta. Se permiten también el uso de pseudocolumnas, funciones y expresiones.
  - Un mecanismo cómodo para saber los comparadores que tenemos disponibles según el tipo de la columna seleccionada (sea o no difusa y el tipo de ésta). Los comparadores difusos de FSQL varían según el tipo difuso del atributo en el que se usen, pero en general los que tiene FSQL definidos pueden verse en la Tabla 1, tanto en su versión de posibilidad como de necesidad.
  - Posibilidad de utilizar funciones de Oracle de forma cómoda, estando agrupadas según su utilidad (numéricas, de manejo de cadenas, de fechas...).
  - Fácil incorporación de la función CDEG de FSQL, la cual permite mostrar el grado de cumplimiento, en el intervalo [0,1], con el que cada fila del resultado satisface las condiciones difusas de la consulta.
  - Fácil visualización de las etiquetas lingüísticas definidas para cada atributo difuso.
  - Fácil realización de operaciones de Reunión Natural entre dos tablas.
  - Fácil uso de funciones de grupo y de condiciones sobre los grupos.
  - Fácil ordenación de los resultados según distintos criterios y columnas.

Tabla 1: Comparadores Difusos del Lenguaje FSQL para los Tipos Difusos 1 y 2.

Posibilidad	Necesidad	Significado: Posiblemente/Necesariamente...
FEQ	**NFEQ**	Igual Difuso
FGT	**NFGT**	Mayor Difuso
FGEQ	**NFGEQ**	Mayor o Igual Difuso
FLT	**NFLT**	Menor Difuso
FLEQ	**NFLEQ**	Menor o Igual Difuso
MGT	**NMGT**	Mucho Mayor Difuso
MLT	**NMLT**	Mucho Menor Difuso

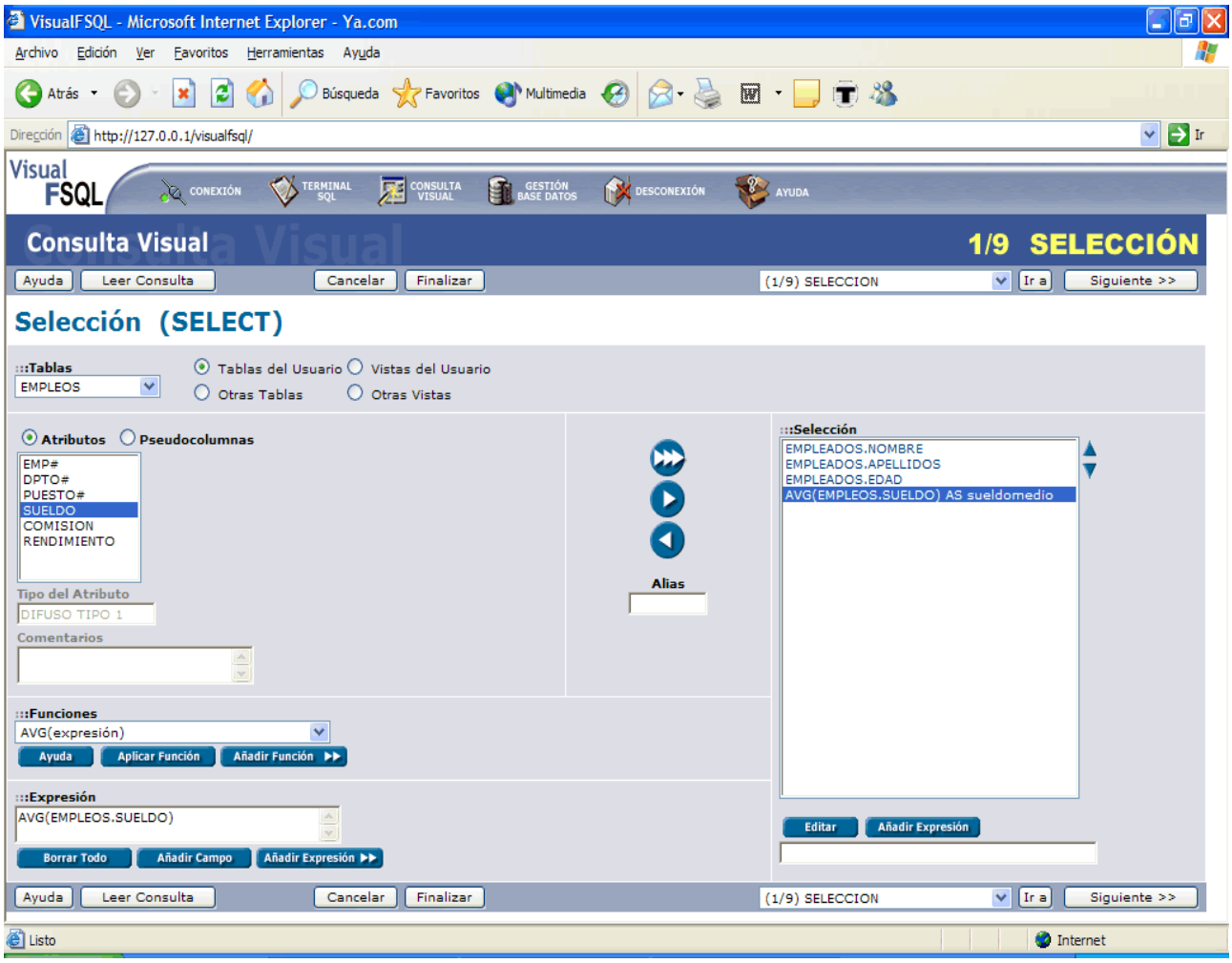


Figura 1: Pantalla de Selección.

- Posibilidad de escribir las consultas directamente en lenguaje SQL, FSQL o dmFSQL [1], un lenguaje de *Data Mining* basado en FSQL que permite usar esta herramienta para clustering, clasificación, dependencias globales difusas...
- Sistema de almacenamiento y recuperación de consultas generadas en el programa. Un usuario puede guardar sus propias consultas para lanzarlas cuando lo desee sin necesidad de construirlas de nuevo.
- Gestión global y genérica de cualquier tabla de la base de datos: Incluyendo operaciones de inserción (INSERT), borrado (DELETE) y modificación (UPDATE).
- Ayuda contextual incluida en el programa.

Otros programas similares que hemos podido encontrar, como [11], no disponen de la mayoría de estas características.

3. CONSULTAS DIFUSAS VISUALES

Esta es la parte principal y más extensa de la aplicación. A través de una sucesión de pantallas consecutivas vamos a ir construyendo mediante asistentes visuales lo que finalmente será una consulta difusa en FSQL.

En total hay nueve pantallas o fases, las cuales podemos enumerar como sigue:

1) Selección de columnas.
2) Relaciones entre tablas.
3) Condiciones simples.
4) Organizar condiciones.
5) Agrupar resultados.
6) Organizar agrupaciones.
7) Ordenar resultados.
8) Finalizar: ver resultados.
9) Editar consulta final.

Para una consulta sólo es necesaria la primera pantalla, en la que se seleccionan los atributos o columnas que queremos seleccionar, y la octava pantalla, en la que se visualizan los resultados.

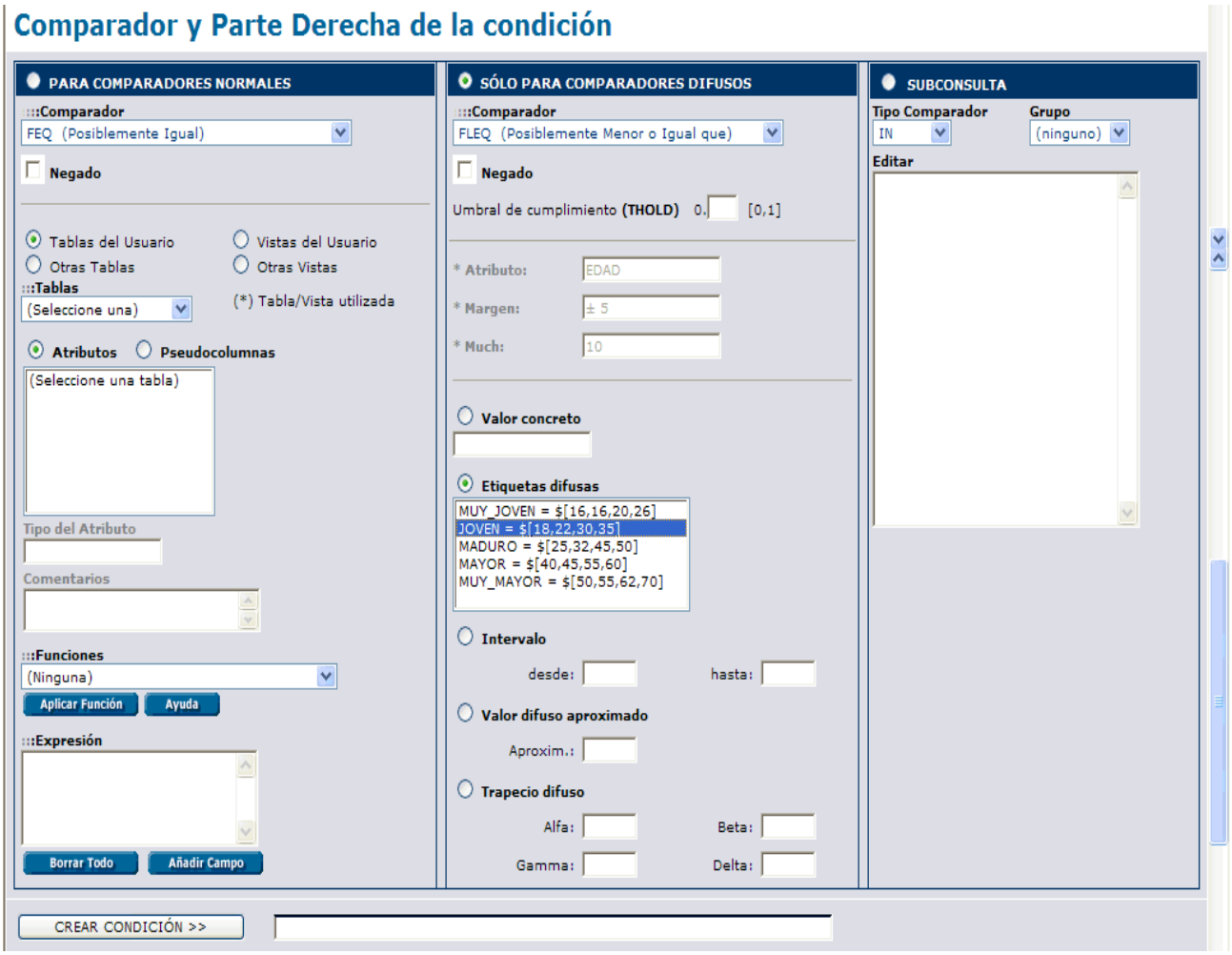


Figura 2: Parte derecha de una condición simple.

El usuario puede saltar de una pantalla a otra libremente, evitando ciertas fases o volviendo para atrás para rectificar lo que desee.

El paso de una pantalla a otra o, incluso, la selección de alguna opción dentro de una pantalla puede redundar en una consulta a la base de datos, por lo que puede suponer un retardo dependiendo del acceso a internet que se disponga en ese momento. Cada pantalla ha sido pensada para minimizar el efecto negativo que estas esperas pueden suponer.

A continuación vamos a explicar algunas de las pantallas más interesantes.

3.1. Pantalla de Selección (1/9)

Esta pantalla puede verse en la Figura 1. Para cada columna que deseemos seleccionar hay que indicar primero la tabla a la que pertenece. Esto es simple de hacer usando el menú desplegable de la izquierda. A su derecha, el usuario puede seleccionar si desea ver las tablas o vistas del usuario o todas a las que tiene acceso.

Una vez seleccionada la tabla (tabla EMPLEOS en el ejemplo de la Figura 1), aparecen sus atributos en la lista inferior. Ahí pueden seleccionarse y pasarse a la lista de atributos seleccionados de la derecha (usando el botón redondo de flecha a la derecha que hay en el centro de la pantalla). Puede aplicarse un alias a ese atributo, añadir todos los atributos, eliminar atributos, aplicarles alguna función, usar pseudocolumnas y generar expresiones más complejas (con operaciones aritméticas, funciones...).

En esta fase el tratamiento de los atributos difusos es exactamente el mismo que el de los atributos no difusos, ya que el sistema esconde que realmente los atributos difusos se almacenan usando varios atributos clásicos [4].

Es importante destacar que cuando se selecciona un atributo aparece en la parte inferior el tipo de dicho atributo y el comentario que hubiera en el Diccionario de Datos del sistema. En la Figura 1 puede verse que el atributo SUELDO es difuso de Tipo 1.

Tras esta fase puede utilizarse el botón finalizar para ver los resultados, o bien, ir a cualquiera de las siguientes fases. Si se han empleado dos o más tablas, casi será obligatorio pasar a la fase 2 para establecer las condiciones de reunión entre dichas tablas.

3.2. Pantalla de Condiciones Simples (3/9)

En esta fase hay que establecer aquellas condiciones que deseemos establecer en nuestra consulta. Una condición simple en FSQL tiene el siguiente formato:

<atrib.> <fcomp> <atrib. | cte.> THOLD <umbral>

donde <fcomp> es un comparador (incluyendo los difusos de la Tabla 1). La palabra THOLD (*thresthold*) es la abreviatura usada por FSQL para establecer el umbral de cumplimiento mínimo que se exige en esa condición simple. Dicho umbral debe ser un valor del intervalo [0,1].

Esta pantalla es extensa y en la Figura 2 puede verse sólo la segunda parte. En la primera parte hay que escoger la parte izquierda de una condición simple y el formato es similar a la primera pantalla.

Para establecer la segunda parte de una condición simple hay que elegir una opción de las 3 columnas que aparecen en la Figura 2. La columna izquierda se utiliza si la parte derecha de una condición utiliza un atributo, una pseudocolumna o una expresión que utilice atributos, pseudocolumnas, funciones o constantes. La opción central se utiliza sólo si la parte derecha de la condición simple es una constante difusa. En la Figura 2 podemos ver que aparecen las etiquetas (muy_joven, joven...) del atributo seleccionado como parte izquierda (edad). También podemos optar por poner una constante de tipo clásica, un intervalo, un valor aproximado o un conjunto difuso trapezoidal, dando sus 4 valores característicos. En la parte superior de estas dos primeras opciones aparece un menú desplegable para escoger el comparador a usar. Si el atributo de la parte izquierda es difuso puede escogerse un comparador difuso (Tabla 1).

Para cada condición simple puede establecerse un umbral de cumplimiento difuso distinto. Esto se hace rellenando el campo que indica "Umbral de Cumplimiento (THOLD)".

Por último, la parte derecha de la pantalla se utiliza si queremos usar subconsultas dentro de esa condición simple (utilizando comparadores como IN, ALL, ANY...).

En la siguiente fase se pueden organizar las condiciones simples como se desee utilizando los operadores AND y OR.

3.2. Agrupar Resultados (5/9)

En esta quinta fase se pueden establecer las cláusulas GROUP BY y HAVING de SQL. Tal y como es deseable, para la primera cláusula sólo pueden escogerse columnas que se hayan insertado en la lista de selección (la cual se formó en la primera pantalla).

En la cláusula HAVING se establecen condiciones sobre los grupos. Estas condiciones pueden incluir tanto funciones de grupo (que se aplican individualmente a cada grupo), tales como COUNT, MAX, MIN, SUM... como funciones de grupo difusas, que se aplican a los grados de cumplimiento difuso de cada columna que haya participado en una condición difusa, o a los grados globales de cumplimiento de las tuplas del grupo, los cuales eran calculados y mostrados por la función CDEG de FSQL, tal y como se comentó anteriormente. No obstante, hay que indicar que las funciones de grupo difusas no han sido implementadas aún en el Servidor FSQL.

El resto de pantallas son fáciles de usar y sin demasiado interés científico. No obstante nos gustaría resaltar el hecho de que cada fila del resultado puede llevar asociado un valor que indica el grado con el que esa fila satisface la condición difusa impuesta. Esa columna podrá visualizarse en la pantalla de resultados (pantalla 8). Finalmente, en la novena y última pantalla el usuario puede opcionalmente editar la consulta FSQL para comprobarla o modificarla. También es posible ver la traducción que el Servidor FSQL hace de la consulta FSQL a una sentencia SQL con llamadas a funciones propias del Servidor FSQL.

4. CONCLUSIONES

Este programa [12] ha aportado una gran funcionalidad en cuanto a la gestión y consulta de Bases de Datos Difusas basadas en FSQL [3][4][6]. El asistente visual de creación de consultas difusas facilita el acceso a este tipo de Bases de Datos a un mayor número de usuarios potenciales, por lo que el abanico de aplicaciones prácticas se incrementa enormemente.

También es destacable que pueda acceder cualquier usuario desde Internet, únicamente con las propias limitaciones de seguridad que queramos establecer. Además la aplicación está totalmente centralizada en un Servidor web y no es necesaria la instalación en los clientes de ningún programa ni componente adicional, únicamente con un Explorador de Internet por parte de los clientes es suficiente. Esto facilita muchísimo la escalabilidad y mantenimiento del sistema, puesto que las modificaciones, actualizaciones ó ampliaciones sólo serían necesarias en el servidor.

Debido a la propia funcionalidad del sistema, que es muy interactivo, se ha optimizado el rendimiento para que el tiempo de respuesta sea el mínimo posible (teniendo en cuenta el tipo de conexión a Internet). Para ello se ha analizado la cantidad de información que el servidor debe enviar en cada una de las peticiones que los clientes realizan a través de Internet, y así intentar reducir el trafico de datos para agilizar el tiempo de respuesta.

También se ha incluido un sistema de Gestión *online* de la Base de Datos, que nos permite insertar, modificar y borrar datos de tipo *crisp* en todas las tablas. Igualmente es de destacar que disponemos de Ayuda online en todas las pantallas de la aplicación.

En cuanto a instalación del sistema, el servidor de Bases de Datos lo podemos tener en la misma máquina que el servidor Web de Internet ó bien utilizar otro equipo dedicado para ello, con lo cual se gana en escalabilidad y balanceo de carga. El sistema es flexible a cualquier arquitectura de servidores.

5. TRABAJOS FUTUROS

Este trabajo tiene bastantes ampliaciones futuras posibles, entre las que podemos destacar las siguientes:

- Extender la Aplicación para poder trabajar con otros SGBDR distintos de ORACLE (Microsoft SQL-Server 2000, Informix, Sybase, Interbase, MySQL, etc.), aún cuando se accedan a Bases de Datos Relaciones no Difusas. Sin embargo, esto sería algo muy complejo de desarrollar ya que el Servidor FSQL habría que reprogramarlo.
- Gestión *online* de la Base de Datos Difusa. Es decir tener un sistema que nos permita gestionar tablas y registros en la Base de Datos Difusa (operaciones de inserción, actualización y borrado).
- Posibilidad de usar el Sistema para generar resultados de consultas reutilizables en otras aplicaciones, permitiendo así, enlazar el resultado de una consulta en FSQL con procesadores de texto, hojas de cálculo y otras bases de datos, y diversos formatos (texto, csv, html, xml, etc.).
- Visualización de las etiquetas de los atributos difusos mediante elementos más visuales (gráficos).
- Gestión más visual de las subconsultas en las condiciones simples.
- Reprogramación completa de la Aplicación en otros lenguajes (p.e. PHP) para poderla instalar en Servidores Web Linux, ampliando de esta manera el ámbito de operatividad del Sistema.
- Incluir la extensión dmFSQL para Data Mining basado en FSQL [2], de forma visual.
- Incluir las novedades de FSQL que se presentarán en [8].

Agradecimientos

Nos gustaría agradecer la colaboración del proyecto TIC2002-00480 del MCYT (Ministerio de Ciencia y Tecnología) de España.

REFERENCIAS

[1] Aversano L., Canfora G., De Lucia A., Stefanucci S., "Understanding SQL through Iconic Interfaces". 26th Annual Int. Computer Software and Applications Conference, pp. 703-710, 2002.

[2] Carrasco R.A., "Lenguajes e Interfaces de Alto Nivel para Data Mining con Aplicación Práctica en Entornos Financieros". Ph. Doctoral Thesis, University of Granada (Spain), 2003.

[3] Galindo J., Medina M., Pons O., Cubero J. C., "A Server for Fuzzy SQL Queries". In "Flexible Query Answering Systems", Eds. T. Andreasen, H. Christiansen and H.L. Larsen, Lecture Notes in Artificial Intelligence (LNAI) 1495, pp. 164-174. Ed. Springer, 1998.

[4] Galindo J., "Tratamiento de la Imprecisión en Bases de Datos Relacionales: Extensión del Modelo y Adaptación de los SGBD Actuales". Tesis Ph. Doctoral Universidad de Granada, España, 1999 (www.lcc.uma.es).

[5] Galindo J., Aranda M.C., Guevara A., Caro J.L., Aguayo A., "Applying Fuzzy Databases and FSQL to the management of rural accommodation". Tourist Management Journal, 23-6, pp. 623-629, 2002.

[6] Galindo J., Medina J.M., "FSQL: Consultas Difusas a Bases de Datos Tradicionales ó Difusas". Revista Cuore (Círculo de Usuarios de Oracle España), 24, suplemento Vivat Academia 6, pp. XI-XIX, Octubre 2003.

[7] Galindo J., Urrutia A., Carrasco R.A., Piattini M., "Relaxing Constraints in Enhanced Entity-Relationship Models using Fuzzy Quantifiers". To appear in IEEE Transactions on Fuzzy Systems, 2004/2005.

[8] Galindo J., Urrutia A., Piattini M., "Fuzzy Databases: Modeling, Design and Implementation". To publish by Idea Group Publishing Hershey, USA, 2005.

[9] Medina J.M., "Bases de datos Relacionales Difusas: Modelo Teórico y Aspectos de su Implementación", Ph. Tesis Doctoral, Universidad de Granada, España, 1994 (decsai.ugr.es).

[10] Medina J.M., Pons O., Vila A., "FIRST. A Fuzzy Interface for Relational SysTems". VI International Fuzzy Systems Association World Congress (IFSA 1995). Sao Paulo (Brasil), 1995.

[11] Medina J.M., Rashid D., Torralbo J.L., Vela R., "A Visual Client for Querying a Fuzzy Database Server". VI International Conference on Information Processing and Management in Knowledge-Based Systems, IPMU'1996, Vol. I, pp. 7-12. Granada (Spain), 1996.

[12] Oliva R.F., "Visual FSQL: Gestión Visual de Bases de Datos Difusas en ORACLE a través de Internet usando FSQL". Proyecto Fin de Carrera, dirigido por J. Galindo, de Ingeniería Superior en Informática de la Universidad de Málaga, 2003. http://www.lcc.uma.es/~ppgg/PFC

[13] Petry F.E., "Fuzzy Databases: Principles and Applications". *International Series in Intelligent Technologies*. Ed. H.J. Zimmermann. Kluwer Academic Publ. (KAP), 1996.

[14] Urrutia A., "Definición de un Modelo Conceptual para Bases de Datos Difusas". Ph. Doctoral Thesis, Univ. of Castilla la Mancha (Spain), 2003.

Uso de Etiquetas Lingüísticas en la Definición de Jerarquías en DataCubos

Miguel Delgado, Carlos Molina*, Daniel Sánchez y M. Amparo Vila
Universidad de Granada
Dpto. Ciencias de la Computación e I.A.
E.T.S.I. Informática
18071, Granada, España
*e-mail: carlosmo@decsai.ugr.es

Resumen

La incorporación del conocimientos de expertos en la definición de DataCubos puede enriquecer las posibilidades de análisis. En muchas ocasiones, a los expertos les será más fácil utilizar etiquetas lingüísticas para expresar este conocimiento. Lo que proponemos es la extensión de un modelo, que permite la definición de conceptos difusos, para que las relaciones jerárquicas puedan modelarse mediante etiquetas lingüísticas para facilitar la integración de esta información.

Palabras Clave: Agregación, Etiquetas lingüísticas, OLAP

1 INTRODUCCIÓN

Desde el nacimiento de OLAP ([3]), se han definido estructuras específicas para dar soporte a esta tecnología. En la actualidad, dada su aplicación en otros campos (p.e. datos hospitalarios) y el uso de fuentes de datos semi-estructuradas (p.e. XML) y no-estructuradas (p.e. texto plano), surgen nuevas necesidades. Entre estas cabe destacar el manejo de información imprecisa y estructuras menos rígidas que permitan un modelado y uso más intuitivo. Se han propuesto modelos para el tratamiento de la imprecisón en diferentes aspectos ([6, 7, 8, 4]).

Una fuente que puede enriquecer los análisis es el conocimiento de expertos en el dominio. Esta información puede extender las jerarquías de los esquemas multidimensionales para mejorar las posibilidades de análisis. En estos casos, utilizar jerarquías imprecisas puede ayudar a incorporar dicho conocimiento. Sin embargo, los expertos no siempre podrán especificar un valor difuso, sino una expresión lingüística de ese valor. El modelo propuesto en [4] permite modelar conceptos mediante el uso de etiquetas lingüísticas. Esta definición lingüística puede también presentarse en las relación jerárquica entre los elementos. Lo que proponemos en este trabajo es extender el modelo de manera que sea capaz de representar estas relaciones.

La solución debe, además, cumplir los siguientes requerimientos:

- Trabajar conjuntamente con valores concretos y etiquetas lingüísticas.
- Permita la coexistencia de diferentes conjuntos de etiquetas. Es posible que diferentes relaciones sean definidas con distintos conjuntos de etiquetas.
- Sea eficiente. A la hora de operar, las operaciones sobre etiquetas no deben de suponer una sobrecarga excesiva, ya que implicaría pérdida en operatividad.

2 MODELO MULTIDIMENSIONAL DIFUSO

En esta sección presentaremos brevemente las partes de la estructura y operaciones del modelo multidimensional a las que afecta la extensión. Debido a las limitaciones de espacio no es posible presentar el modelo con detalle. Una deficición más detallada puede encontrarse en [4].

2.1 ESTRUCTURA MULTIDIMENSIONAL

La extensión que proponemos afecta a las relaciones existentes entre los elementos de niveles en una dimensión. Antes de abordarlas, necesitamos una definición previa.

Definición 1 *Para cada l_i el conjunto*

$$H_i = \{l_j / l_j \neq l_i \wedge l_j \leq_d l_i \wedge \neg \exists l_k \; l_j \leq_d l_k \leq_d l_i\} \quad (1)$$

y lo llamaremos conjunto de hijos del nivel l_i

Definición 2 *Por cada par de niveles l_i y l_j tal que $l_j \in H_i$, tenemos la relación $\mu_{ij} : l_i \times l_j \rightarrow [0,1]$, y la denominaremos* relación de parentesco.

Esta relación establece la jerarquía entre los valores de niveles contiguos. Necesitamos extenderla para cualesquiera niveles en una dimensión.

Definición 3 *Para cada par de niveles l_i y l_j de la dimensión d, tal que $l_j \leq_d l_i \wedge l_j \neq l_i$, tenemos la relación $\eta_{ij} : l_i \times l_j \rightarrow [0,1]$ definida como*

$$\eta_{ij}(a,b) = \begin{cases} \mu_{ij}(a,b) & si\ l_j \in H_{l_i} \\ \bigoplus_{l_k \in H_{l_i}} \bigotimes_{c \in l_k} (\mu_{ik}(a,c) \otimes \eta_{kj}(c,b)) & en\ otro\ caso \end{cases} \quad (2)$$

donde $\otimes$ y $\oplus$ son una t-norma y una t-conorma, respectivamente. Esta relación la denominaremos relación de parentesco extendido.

2.2 OPERACIONES

Sobre el modelo se han definido las operaciones roll-up (subir en la jerarqua de las dimensiones), drill-down (bajar en las jerarquías), slice (reducir la dimensionalidad del esquema), dice (restringir los valores según una condición) y pivot (modificar el orden de acceso a las dimensiones). De éstas, la única que se ve afectada por el cambio en las relaciones es roll-up, dado que al aumentar el nivel de detalle se ha de obtener el conjunto de hechos y su valor de pertenencia para calcular los hechos del nuevo DataCubo. Antes de poder introducirla, necesitamos una definicón previa.

Definición 4 *Para cada valor a en la dimensión d_i el conjunto*

$$F_a = \begin{cases} \bigcup_{l_i \in H_{l_i}} F_b / b \in l_j \wedge \mu_{ij}(a,b) > 0 & si\ l_i \neq l_b \\ \{h / h \in F \wedge \exists \overrightarrow{a}(\overrightarrow{a}) = h\} & si\ l_i = l_b \end{cases} \quad (3)$$

Este conjunto representa el conjunto de todos los hechos relacionados con el valor a.

Definición 5 *El resultado de aplicar* roll-up *en la dimensión d_i, nivel l_r ($l_r \neq l_\perp$), usando el operador de agregación G sobre el DataCubo $C = (D, l_b, F, A, H)$ es otro DataCubo $C' = (D, l'_b, F', A', H')$ donde $l'_b = (l_{1b}, ..., l_r, ..., l_{nb})$, $A'(a_1, ..., a_n) = G(\{(b, \alpha \otimes \eta_{rb}(a,c)) / (b,\alpha) \in F_a \wedge A(a_1, ..., a_n) = (b,\alpha)\})$, F' es la imagen de A', y $H' = (A, l_b, F, G, H)$.*

3 PROPUESTAS ANTERIORES

En la literatura se han presentado diferentes propuestas para abordar la necesidad de operar conjuntamente

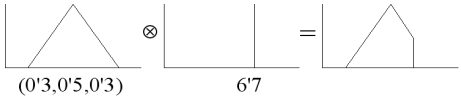


Figura 1: Ejemplo de aplicación del principio de extensión al *mínimo* como *t-norma*

con valores concretos y etiquetas lingüísticas. Las presentaremos brevemente:

- **Pasar etiquetas lingüísticas a valores concretos**. Esta manera de trabajar consiste en calcular un *valor característico* que represente a la etiqueta lingüística, utilizando este valor para las operaciones necesarias. Ejemplos del cálculo del valores característicos son el *centro de gravedad* o el *máximo valor*. Esta aproximación lo que hace es prefijar unos valores concretos (los valores característicos de las etiquetas) y trabajar sobre ellos. Esto implica que no se considere la imprecisión que implica el número difuso subyacente a la etiqueta, por lo que se hace una simplificación de la representación de los valores.

- **Trabajar sólo con etiquetas linguísticas**. Ahora lo que hacemos es justo lo contrario: pasamos las valores concretos a las etiquetas a las que mejor se aproximen y trabajamos siempre este nivel. El problema es que, para estos valores concretos, introducimos incertidumbre que antes no existía y la complejidad en el caso de tener que operar con conjuntos de etiquetas lingüísitcas distintas (p.e. agregar etiquetas sobre el intervalo unidad cuando en un caso se han utilizado 5 etiquetas sobre el dominio y en otro 7). En estos casos tendríamos que definir las tablas de verdad para las posibles agregaciones. Con esto complicaríamos el diseño y perderíamos eficiencia en la resolución de consultas.

- **Aplicar principio de extensión a t-conormas y t-normas**. El problema de esta opción es que el resultado de operar los números difusos que representan las etiquetas no tiene por que ser un número difuso (figura 1), y menos aún una etiqueta del conjunto de partida. Esto haría que la complejidad de las operaciones fuera creciendo conforme se agregan valores, con lo que empeoraría la eficiencia en las consultas.

Por ello, necesitamos una solución nueva que cumpla con las necesidades particulares de nuestro problema.

4 FAMILIA DE OPERADORES OWA

En esta sección presentaremos brevemente la familia de operadores OWA propuesta por Yager ([10]).

Definición 6 *Un vector* $v = [v_1, ..., v_n]$ *es un* vector de pesos *de dimensión* n *si y solo si* $v_i \in [0,1]$ *y* $\sum_i v_i = 1$.

Definición 7 *Siendo* w *un* vector de pesos *de dimensión* n*, la función* $OWA_w : R^n \to R$ *es un* operador de media ponderada (OWA) *de dimensión* n *si*

$$OWA(a_1, ..., a_n) = \sum_i w_i a_{\sigma_i} \tag{4}$$

donde $\{\sigma_1, ..., \sigma_n\}$ *es una permutación tal que* $a_{\sigma_{i-1}} \geq a_{\sigma_i}$ *para todo* $i \in \{2, ..., n\}$.

Dependiendo del vector de pesos que utilicemos, el comportamiento del operador será distinto. Si $w_1 = 1$ y $w_i = 0$ para $i \in \{2, ..., n\}$, el resultado será el máximo de los valores contemplados. Desde un punto de vista lógico, la interpretación sería quedarse con el mayor grado de cumplimiento, es decir, buscar que al menos uno cumpla la propiedad correspondiente (O-lógico). Si por el contrario tenemos, $w_n = 1$ y $w_i = 0$ para $i \in \{1, ..., n-1\}$, el comportamiento sería quedarse con el menor grado con el que se cumple la propiedad (Y-lógico).

Definición 8 *Para un operador OWA con vector de pesos* w *definimos la función* $\alpha : [0,1]^n \to [0,1]$ *como*

$$\alpha(w) = \frac{1}{n-1} \sum_i^n (n-i) w_i \tag{5}$$

Esta función nos da una medida del carácter del operador: cuanto más cercano es este valor a 1, más se asemeja el operador al comportamiento del O-lógico, si por el contrario, este valor está cerca de 0, el comportamiento predominante será el Y-lógico.

Para medir la cantidad de información que utiliza un operador OWA se utiliza la medida de dispersión o entropía.

Definición 9 *Dado un vector de pesos* w*, la dispersión o entropía del operador OWA asociado es*

$$disp(w) = -\sum_i w_i ln w_i \tag{6}$$

4.1 OWA DIFUSO

En nuestro modelo lo que necesitaremos será agregar etiquetas lingísticas. Por esto, necesitamos extender el comportamiento de la familia de operadores OWA para que pueda trabajar sobre valores difusos, y más concretamente, sobre números difusos subyacentes a las etiquetas.

En la literatura se han propuesto extensiones del operador OWA que utilizan informacin difusa ([9]) pero no se adaptan a nuestro problema. Una de las características más importante de los operadores OWA es la ordenación que se debe realizar de los valores para realizar la media ponderada. En el caso difuso, tendremos también que realizar la ordenación de valores. Este problema ha sido ampliamente tratado en la literatura, existiendo múltiples propuestas ([5, 2]). Así pues, con cada método que utilicemos, obtendremos una familia de operadores OWA difusos distinta.

Definición 10 *Siendo* w *un* vector de pesos *de dimensión* n*, y* OM *un método de ordenación de números difusos, la función* $\widetilde{OWA}_w^{OM} : \widetilde{R}^n \to \widetilde{R}$ *es un* operador de media ponderada difuso (FOWA) *de dimensión* n *si*

$$\widetilde{OWA}^{OM}(\tilde{a}_1, ..., \tilde{a}_n) = \bigoplus_i w_i \widetilde{a}_{\sigma_i} \tag{7}$$

donde $\{\sigma_1, ..., \sigma_n\}$ *es una permutación tal que* $\tilde{a}_{\sigma_{i-1}} \leq_{OM} \tilde{a}_{\sigma_i}$ *para todo* $i \in \{2, ..., n\}$ *y* $\bigoplus$ *es la suma extendida.*

La caraterización de estos operadores (α) no varía en su definición, dado que ésta dependen del vector de pesos (w) que no se ha visto modificado al extenderlo a valores difusos.

4.2 OPERADOR A_β^{OM}

Basado en el FOWA definieremos un operador para realizar las agregaciones de las operaciones de roll-up.

Definición 11 *Un operador de agregación* A_β^{OM} *será una función* $A_\beta^{OM} : \widetilde{[0,1]}^n \to \widetilde{[0,1]}$ *definida como*

$$A_\beta^{OM}(a_1, ..., a_n) = \widetilde{OWA}_w^{OM}(a_1, ..., a_n) \tag{8}$$

donde $w_1 = \beta$*,* $w_n = 1 - \beta$*, y* $w_i = 0$ *para todo* $i \in \{2, ..., n-1\}$.

Propiedad 1 *Un operador* A_β^{OM} *es un operador OWA con* $\alpha(w) = \beta$.

Demostración. El vector de pesos correspondiente al operador A_β^{OM} es $w = [w_1, ..., w_n]$ tal que $w_1 = \beta$ y $w_n = 1 - \beta$, teniendo el resto de los pesos a cero. Si calculamos el valor de $\alpha(w)$ obtenemos

$$\alpha(w) = \tfrac{1}{n-1} \textstyle\sum_i^n (n-i) w_i =$$

$$= \frac{(n-1)w_1+(n-2)w_2+...+1w_{n-1}+0w_n}{n-1} =$$
$$= \frac{(n-1)\beta+(n-2)0+...+1\cdot 0+0(1-\beta)}{n-1} = \frac{(n-1)\beta}{n-1} = \beta$$

Dependiendo del valor de β, el operador tendrá un comportamiento más cercano al O-lógico o el Y-lógico. Para notar que nos referimos a un operador de agregación con un comportamiento más cercano al O-lógico escribiremos A_β^{OM}. Si, por el contrario, queremos notar que el comportamiento es más cercano al Y-lógico, escribiremos $A_{1-\beta}^{OM}$.

Definición 12 *Dados dos operadores A_β^{OM} y $A_{\beta'}^{OM}$, diremos que son duales si $\beta' = 1 - \beta$.*

Definición 13 *Dado un operador de agregación A_β^{OM}, la dispersión $disp(A_\beta^{OM})$ tiene la expresión*

$$disp(A_\beta^{OM}) = -\beta ln\beta - (1-\beta)ln(1-\beta) \qquad (9)$$

Esta expresión se obtiene directamente del cálculo de la dispersión del operador $\widetilde{OWA}$ que utilicemos. La dispersión de este será

$$disp(w) = -\sum_i w_i lnw_i = -w_1 lnw_1 - ... - w_n lnw_n =$$
$$-\beta ln\beta - 0ln0 - ... - (1-\beta)ln(1-\beta) =$$
$$-\beta ln\beta - (1-\beta)ln(1-\beta).$$

Propiedad 2 *Dados dos operadores de agregación, A_β^{OM} sobre n valores y $A_{\beta'}^{'OM}$ sobre m valores, si $\beta = \beta'$ o $\beta = 1 - \beta'$ entonces $disp(A_\beta^{OM}) = disp(A_{\beta'}^{'OM})$.*

La demostración es directa de la definición 13.

5 EXTENSIÓN DEL MODELO MULTIDIMENSIONAL DIFUSO

Una vez que hemos definido el operador de agregación A_β^{OM}, ya podemos extender el modelo multidimensional difuso para poder utilizar etiquetas lingüísticas en la relación entre los valores de diferentes niveles en cada dimensión. Tendremos que extender la definición de la estructura así como las operaciones que aplicamos sobre ésta.

5.1 ESTRUCTURA MULTIDIMENSIONAL

La estructura multidimensional se ha de extender para manejar etiquetas lingüísticas en la relación entre elementos en la jerarquía. La *relación de parentesco* y la *relación de parentesco extendida* son las que se ven afectadas.

Definición 14 *Para cada par de niveles l_i y l_j tal que $l_j \in H_{l_i}$ tenemos la relación*

$$\widetilde{\mu}_{ij} : l_i \times l_j \to \widetilde{[0,1]} \qquad (10)$$

A esta relación la denominaremos relación de parentesco difusa.

Definición 15 *Para cada par de niveles l_i y l_j de la dimensión d, tal que $l_j \leq_d l_i \wedge l_j \neq l_i$, tenemos la relación $\widetilde{\eta}_{ij} : l_i \times l_j \to \widetilde{[0,1]}$ definida como*

$$\widetilde{\eta}_{ij}(a,b) = \begin{cases} \widetilde{\mu}_{ij}(a,b) & si\ l_j \in H_{l_i} \\ A_\beta^{OM}(P_{l_1},...,P_{l_n}) & en\ otro\ caso \end{cases} \qquad (11)$$

donde $l_k \in H_{l_i}$ y $P_{l_k} = A_\beta^{OM}(\delta_{c_1},...,\delta_{c_m}) \forall c_i \in l_k$, siendo $\delta_c = A_{1-\beta}^{OM}(\widetilde{\mu}_{ik}(a,b), \widetilde{\eta}_{kj}(c,b))$. A esta relación la denominaremos relación de parentesco extendido difuso.

5.2 OPERACIONES

Como ya hemos comentado, al permitir definir las jerarquías con etiquetas lingüísticas se ve modificado la operación de roll-up y el conjunto F_a. En esta sección veremos como se extienden en su definición utilizando el operador A_β^{OM} sobre números difusos. Comenzaremos presentando $\tilde{F}_a$.

Definición 16 *para cada valor a perteneciente a d_i tenemos el conjunto*

$$\tilde{F}_a = \begin{cases} \bigcup_{l_i \in H_{l_i}} \tilde{F}_b / b \in l_j \wedge \widetilde{\mu}_{ij}(a,b) \neq \tilde{0} & si\ l_i \neq l_b \\ \{h/h \in F \wedge \exists \overrightarrow{a} A(\overrightarrow{a}) = h\} & si\ l_i = l_b \end{cases} \qquad (12)$$

El conjunto $\tilde{F}_a$ representa a todos los hechos relaciones con el valor a.

En la operación de roll-up vamos a necesitar una definición previa.

Definición 17 *D_f es una función definida como $D_f : \widetilde{[0,1]} \to [0,1]$.*

Como podemos ver, D_f se comporta como una función de defuzzificación, aunque restringida al intervalo [0, 1]. Así pues, cualesquiera de los métodos propuestos para la defuzzificación será válido como función D_f.

Definición 18 *El resultado de aplicar* roll-up *sobre la dimensión d_i, nivel l_r ($l_r \neq l_\perp$), utilizando el operador de agregación G sobre el DataCubo $C = (D, l_b, F, A, H)$ es otro DataCubo $C' = (D, l'_b, F', A', H')$*
donde $l'_b = (l_{1b},...,l_r,...,l_{nb})$, $A'(a_1,...,a,...,a_n) = G(\{(b, D_f(A_{1-\beta}^{OM}(\alpha, \widetilde{\eta}_{rb}(a,c))))/(b,\alpha) \in \tilde{F}_a \wedge A(a_1,...,c,...,a_n) = (b,\alpha)\})$, F' es la imagen A', y $H' = (A, l_b, F, G, H)$.

En las definiciones utilizamos operadores duales (β y $1-\beta$). En ocasiones puede ser interesante que esto

Tabla 1: Hechos del DataCubo

No	Prod.	Fecha	Edad	Precio	Cant.	α
1	leche	23-dic	19	10	1	1
2	carne	7-ene	40	18	3	1
3	pan	10-ene	45	1	5	1
4	zumo	28-dic	75	2	2	1
5	queso	3-ene	20	5	1	1
6	leche	10-ene	20	1	5	1
7	pan	25-dic	22	3	1	1
8	pan	1-ene	55	5	2	1
9	zumo	28-dic	23	4	3	1
10	pan	6-ene	75	6	4	1
11	leche	23-dic	78	3	3	1
12	carne	29-dic	40	18	2	1
13	pan	6-ene	17	3	2	1
14	carne	22-dic	65	6	3	1
15	queso	2-ene	52	10	2	1
16	pan	27-dic	66	5	2	1
17	queso	04-ene	70	5	3	1
18	pan	24-dic	60	3	6	1
19	pan	10-ene	65	4	4	1
20	leche	03-ene	64	5	2	1
21	queso	10-ene	15	5	5	1
22	queso	28-dic	40	3	5	1
23	pan	02-ene	65	4	5	1
24	leche	26-dic	23	5	5	1

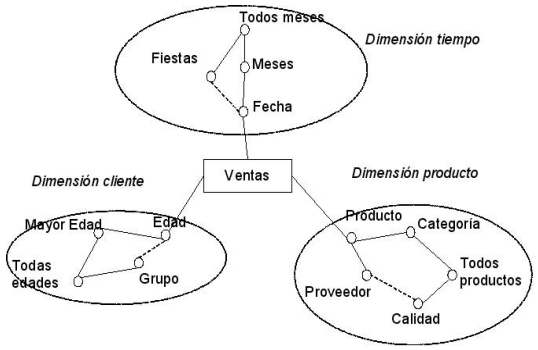


Figura 2: Esquema multidimensional

no sea así (β y $1-\beta'$ con $\beta \neq \beta'$). El valor para β y β' será una decisión en el diseño del DataCubo y dependerá del dominio del análisis y de las necesidades del decisor.

6 EJEMPLO

Para aclarar los conceptos, los aplicaremos sobre un ejemplo. Utilizaremos para ello el esquema multidimensional presentado en la figura 2. Con éste queremos analizar el comportamiento de las ventas dentro de una empresa. Las líneas discontinuas representan relaciones entre niveles difusos. En el caso de los niveles *Calidad* y *Proveedor* esta se ha realizado utilizando las etiquetas lingüísticas representadas en la figura 3, dado que ha sido proporcionada por un experto. Los datos de las dimensiones vienen recogidas en la figura 4.

Sobre este esquema construiremos un DataCubo con los hechos recogidos en la tabla 1. En este DataCubo

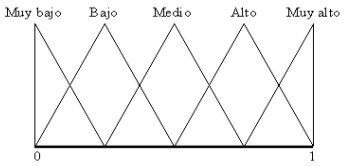


Figura 3: Etiquetas lingüísticas

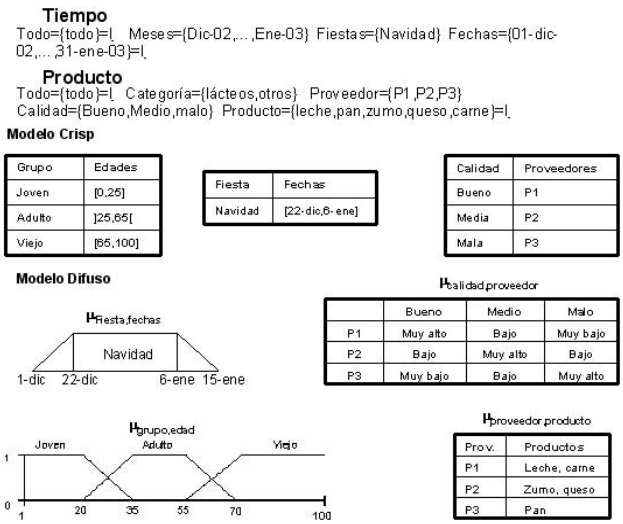


Figura 4: Estructura de las dimensiones

resolveremos la consulta *"obtener la cantidad media de ventas en Navidad según los grupos de edades y la calidad de los proveedores"*. Esta consulta implica, primero, realizar *dice* en dimensión *Tiempo* y nivel *Fiesta* con condición $\beta(x) =$ "x es Navidad", y, segundo, *roll-up* en la dimensión *Tiempo*, dimensión *Producto*, nivel *Calidad*, y *Cliente* y nivel *Grupo*, utilizando el operador de agregación *media* sobre la cantidad.

Para poder realizar la consulta necesitamos establecer los parámetros del operador A_β^{OM}. Utilizaremos el método de ordenación de números difusos propuesto por Delgado et al. en [5] y dos valores de β para mostrar las diferencias: $\beta = 1,0$ (equivaldría al O-lógico), y $\beta = 0,9$. Como función D_f utilizaremos la *media de los máximos*. De cara al usuario utilizaremos el operador de *Resumen Lingüístico (RL)* definido en [1].

El resultado de la consulta puede verse en la tabla 2. Para ambos valores, el kernel de los conjuntos es el mismo, las variaciones se encuentran en los valores con pertenencia menor que 1. En el caso de utilizar $\beta = 1,0$ vemos como los grados de pertenencia son mayores. Con valor 0,9 las agregaciones tienen un comportamiento más restrictivo (tiene un valor α menor) de tal forma que los grados de pertenencia son menores. Con un valor β menor, se potenciaría este

Tabla 2: Resultados de la consulta

Grupo	β	Bueno		Medio		Malo	
		C'	C'_{RL}	C'	C'_{RL}	C'	C'_{RL}
Joven	1,0	{1'0/1'0, 0'9/3'0, 0'4/3'7, 0'2/3'3},1	(1,1,0,1'3) Mayor que 1	{1'0/1'0 0'6/3'0 0'2/2'9},1	(1,1,0,1'4) Mayor que 1	{1'0/2'0 0'6/1'5 0'2/2'4},1	(2,2,0'3,0'4) En torno a 2
	0,9	{1'0/1'0, 0'9/3'0, 0'5/3'7, 0'4/3'3, 0'2/2'9, 0'1/3'1 },1	(1,1,0,1'2) Mayor que 1	{1'0/1'0, 0'6/3'0, 0'4/2'9, 0'1/3'2},1	(1,1,0,1'4) Mayor que 1	{1'0/2'0, 0'6/1'5, 0'4/2'4, 0'2/2'9, 0'1/3'1},1	(2,2,0'5,0'3) En torno a 2
Adulto	1,0	{1'0/2'0, 0'9/3'3, 1,0 0'6/3'4, 0'3/3'3, 0'2/3'3},1	(2,2,0,0'54) Algo mayor que 2	{1'0/3'5, 0'6/3'3 0'3/3'3, 0'2/3'3},1	(3'5,3'5,0'2,0) Algo menor que 3'5	{1'0/2'0, 0'8/4'0, 0'5/3'8, 0'4/3'3, 0'3/3'3, 0'2/3'3},1	(2,2,0,1'6) Mayor que 2
	0,9	{1'0/2'0, 0'9/3'3, 0'6/3'4, 0'3/3'3, 0'2/3'3, 0'1/3'1},1	(2,2,0,0'6) Algo mayor que 2	{1'0/3'5, 0'6/3'3, 0'4/3'3, 0'3/3'3, 0'1/3'2},1	(3'5,3'5,0'3,0) Algo menor que 3'5	{1'0/2'0, 0'8/4'0, 0'5/3'3, 0'3/3'3, 0'2/3'3, 0'1/3'0},1	(2,2,0,1'6) Mayor que 2
Viejo	1,0	{1'0/3'0, 0'5/2'7, 0'2/2'6},1	(3,3,0'37,0) Algo menor que 3	{1'0/2'0, 0'8/2'5, 0'3/3'7, 0'2/3'4},1	(2,2,0,1'7) Mayor que 2	{1'0/4'0, 0'6/3'0, 0'5/3'8, 0'3/4'2, 0'2/3'7},1	(4,4,0'8,0,2) Menor que 4
	0,9	{1'0/3'0, 0'5/2'7, 0'4/2'6, 0'2/3'4, 0'1/3'1},1	(3,3,0'36,0'15) En torno a 3	{1'0/2'0, 0'8/2'5, 0'4/3'0, 0'3/3'4, 0'1/3'2},1	(2,2,0,1'27) Mayor que 2	{1'0/4'0, 0'6/3'0, 0'5/3'8, 0'4/3'7, 0'2/3'4, 0'1/2'9},1	(4,4,0'8,0) Menor que 4

comportamiento restrictivo.

7 CONCLUSIONES

En los sistemas OLAP puede ser interesante incluir información proporcionada por expertos para enriquecer los esquemas de los DataCubos. El uso de expresiones lingüísticas es más natural de cara al usuario que trabajar con valores concretos, que no siempre es fácil de obtener. Por esto, para poder realizar la integración de la información proporcionada por expertos y los esquemas multidimensionales, puede ser muy útil que el modelo multidimensional permita la utilización de etiquetas lingüísticas de forma directa.

Lo que hemos presentado en este trabajo es la extensión de un modelo multidimensional que permite la utilización de valores difusos tanto en la jerarquías como en los hechos, para que contemple la definición de las relaciones jerárquicas mediante etiquetas lingüísticas. Para ello, hemos utilizado una extensión difusa de los operadores OWA que permite agregar los valores de estas relaciones para poder operar eficientemente.

Referencias

[1] I. Blanco, D. Sánchez, J.M. Serrano, and M.A. Vila. A new proposal of aggregation functions: the linguistic summary. In *Proceedings of IFSA'2003*, Istanbul (Turkey), 2003.

[2] G. Bortolan and R. Degani. A review of some methods for ranking fuzzy subsets. *Fuzzy Sets and Systems*, 15:1–19, 1985.

[3] E.F. Codd. Providing OLAP (On-line Analytical Processing) to user-analysts: An IT mandate. Technical report, E.F. Codd and Associates, 1993.

[4] M. Delgado, C. Molina, D. Sánchez, M. A. Vila, and L. Rodríguez-Ariza. A flexible approach to the multidimensional model: the Fuzzy Datacube. In *CAEPIA-TTIA 2003*, number 3040 in LNAI, pages 26–36. Springer-Verlag, 2004.

[5] M. Delgado, J.L. Verdegay, and M.A. Vila. A procedure for ranking fuzzy numbers using fuzzy relations. *Fuzzy Sets and Systems*, 26(1):49–62, 1998.

[6] C. Dyreson. Information retrieval from an incomplete data cube. In *Proceeding of the 22nd Int. Conf. on VLDB*, pages 532–543, Istanbul (Turkey), 1996. Morgan Kaufman Publishers.

[7] A. Laurent, B. Bouchon-Meunier, and A. Doucet. Flexible unary multidimensional queries and their combinations. In *Proceeding of IPMU 2002*, Annecy (France), 2002.

[8] T.B. Pedersen, C.S. Jensen, and C.E. Dyreson. A foundation for capturing and querying complex multidimensional data. *Information Systems*, 26:383–483, 2001.

[9] P.A. Schaefer and H.B. Mitchell. A generalized OWA operator. *International Journal of Intelligent Systems*, 14:123–143, 1999.

[10] R.R. Yager. On ordered weighted averaging aggregtion operators in multicriteria decison making. *IEEE Transactions on Systems, Man and Cybernetics*, 18:183–190, 1988.

ANÁLISIS Y DISEÑO DE CONTROLADORES BORROSOS ROBUSTOS EN TIEMPO DISCRETO

M. García, A. Barreiro
Departamento de Ingeniería de Sistemas y Automática
Escuela Superior de Ingenieros Industriales y de Minas
Campus Universitario Lagoas-Marcosende, 36200, Vigo, España
e-mail: mgrivera@uvigo.es

Resumen

Este artículo analiza la robustez de sistemas de control borroso en tiempo discreto. Un sistema continuo es muestreado y se analiza su estabilidad mediante la teoría de Lyapunov. Se proporciona una medida de la estabilidad del sistema en forma de margen de ganancia. Posteriormente se realiza un rediseño del sistema original para aumentar dicho margen de ganancia. Estos conceptos se ilustran con un servomotor como ejemplo.

Palabras Clave: Control borroso, Estabilidad de Lyapunov, Tiempo discreto, Robustez.

1 INTRODUCCIÓN

El desarrollo del microprocesador en los años 90 [1], originó que el control digital se aplicara extensamente en la industria. Por lo tanto, es interesante estudiar el compartamiento de un sistema continuo cuando se le aplica un muestreo con un computador.

La estabilidad de Lyapunov [4, 5, 6] está relacionada con el comportamiento de las trayectorias de un sistema, cuando su estado inicial se encuentra cerca de un equilibrio. Desde el punto de vista práctico este punto es importante, ya que las perturbaciones que afectan al sistema tienden a separarlo del equilibrio. Dentro del organigrama general de las teorías de estabilidad, Lyapunov se engloba en el área que trata con las variables de estado del sistema, y por tanto, tiene que ver con la representación interna de los sistemas.

El estudio de la robustez surge cuando controladores diseñados para ser estabilizantes de la planta nominal, producen acciones desestabilizadoras ante ciertas perturbaciones del modelo del sistema, y por consiguiente, estos controladores no son utilizables en la práctica. La solución pasa por considerar expresamente la incertidumbre que afecta a la planta nominal.

Este artículo parte de un modelo nominal de la planta y un controlador borroso en tiempo continuo diseñado para cumplir unos objetivos. Posteriormente, el control se realiza mediante muestreo con un computador. La sección 2 analiza la estabilidad de dicho sistema discreto mediante la teoría de Lyapunov. En la sección 3 se desarrolla una metodología para calcular el margen de robustez del control. Finalmente, todas estas ideas se aplican al rediseño de un controlador borroso en la sección 4 para mejorar la robustez.

2 ESTABILIDAD DE LYAPUNOV

Sea un sistema continuo en el cual la dinámica de la planta y el controlador está definida por:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + bKu = A\boldsymbol{x} + bKH(\boldsymbol{x}) \qquad (1)$$

donde $\boldsymbol{x} \in \mathcal{R}^n$ es el estado de la planta, $u \in \mathcal{R}$ la entrada controlada, $A \in \mathcal{R}^{nxn}$ y $b \in \mathcal{R}$ modelan la planta, $K \in \mathcal{R}$ es una ganancia variable que representa una perturbación o imprecisión del modelo, y $H(\boldsymbol{x}) : \mathcal{R}^n \rightarrow \mathcal{R}$ el controlador borroso.

El sistema muestreado con período h correspondiente a planta dada por la ecuación (1) es el siguiente [1]:

$$\boldsymbol{x}_{k+1} = \Phi\boldsymbol{x}_k + \Gamma K u_k = \Phi\boldsymbol{x}_k + \Gamma K H(\boldsymbol{x}_k), \qquad (2)$$

donde $\Phi = \Phi(h) = e^{Ah}$, $\Gamma = \Gamma(h) = \int_0^h e^{Ah}\, b\, ds$.

La condición de estabilidad de Lyapunov en tiempo discreto para sistemas dados por (2), es:

$$\begin{aligned} V(\boldsymbol{x}_{k+1}) - V(\boldsymbol{x}_k) = \boldsymbol{x}'_{k+1}P\boldsymbol{x}_{k+1} - \boldsymbol{x}'_kP\boldsymbol{x}_k = \\ [\Phi\boldsymbol{x}_k + \Gamma KH(\boldsymbol{x}_k)]'P[\Phi\boldsymbol{x}_k + \Gamma KH(\boldsymbol{x}_k)] - \boldsymbol{x}'_kP\boldsymbol{x}_k = \\ \boldsymbol{x}'_k\Phi'P\Phi\boldsymbol{x}_k + \boldsymbol{x}'_k\Phi'P\Gamma KH(\boldsymbol{x}_k) + \Gamma'KH(\boldsymbol{x}_k)P\Phi\boldsymbol{x}_k + \\ \Gamma'KH(\boldsymbol{x}_k)P\Gamma KH(\boldsymbol{x}_k) - \boldsymbol{x}'_kP\boldsymbol{x}_k = \\ \boldsymbol{x}'_k(\Phi'P\Phi - P)\boldsymbol{x}_k + 2\Gamma'P\Phi\boldsymbol{x}_kKH(\boldsymbol{x}_k) + \\ \Gamma'P\Gamma K^2H^2(\boldsymbol{x}_k) < 0, \quad \forall\boldsymbol{x}_k \neq 0 \end{aligned} \qquad (3)$$

donde Φ y Γ representan la planta en tiempo discreto, y P es una matriz real, simétrica y definida positiva. Según el segundo método de Lyapunov el sistema será estable, si existe alguna matriz P que cumpla (3) para cualquier $\boldsymbol{x}_k$ no nulo.

Reorganizando la ecuación (3) en dos términos, $X(\boldsymbol{x}_k)$ que no depende del controlador, e $Y(\boldsymbol{x}_k)$ que sí depende, obtenemos:

$$V(\boldsymbol{x}_{k+1}) - V(\boldsymbol{x}_k) = -X(\boldsymbol{x}_k) + Y(\boldsymbol{x}_k) \tag{4}$$

donde

$$X(\boldsymbol{x}_k) = -\boldsymbol{x}_k'(\Phi' P \Phi - P)\boldsymbol{x}_k = \boldsymbol{x}_k'(P - \Phi' P \Phi)\boldsymbol{x}_k \tag{5}$$

e

$$Y(\boldsymbol{x}_k) = 2\Gamma' P \Phi \boldsymbol{x}_k K H(\boldsymbol{x}_k) + \Gamma' P \Gamma K^2 H^2(\boldsymbol{x}_k) \tag{6}$$

Los términos $X(\boldsymbol{x}_k)$ e $Y(\boldsymbol{x}_k)$ son curvas, superficies o hipersuperficies dependiendo de la dimensión del sistema. La expresión (4) obtenida a partir del segundo método de Lyapunov, puede entonces interpretarse de forma que el sistema será estable, si para todos los posibles valores de $\boldsymbol{x}_k$ no nulos existe alguna matriz P para la cual $-X(\boldsymbol{x}_k) + Y(\boldsymbol{x}_k) < 0$, o lo que es lo mismo

$$X(\boldsymbol{x}_k) > Y(\boldsymbol{x}_k), \;\; \forall \boldsymbol{x}_k \neq 0 \tag{7}$$

Por lo tanto, para que el sistema sea estable $X(\boldsymbol{x}_k)$ debe delimitar superiormente a $Y(\boldsymbol{x}_k)$ para cualquier valor de $\boldsymbol{x}_k$ distinto de cero. Si la dimensión del sistema es 2, $X(\boldsymbol{x}_k)$ e $Y(\boldsymbol{x}_k)$ serán superficies $(X, Y : U \rightarrow \mathcal{R}, U \subseteq \mathcal{R}^2)$, por lo que para poder representar y comparar más fácilmente ambas gráficas, se ha optado por una representación plana de ambas superficies mediante el siguiente desarrollo.

Supongamos que se realiza un cambio de escala en el dominio de entrada definido por $\lambda \boldsymbol{x}_k$, donde $\lambda > 0$. Entonces la ecuación (5) se transforma en

$$\begin{aligned} X(\lambda \boldsymbol{x}_k) = \lambda \boldsymbol{x}_k'(P - \Phi' P \Phi)\lambda \boldsymbol{x}_k = \\ \lambda^2 \boldsymbol{x}_k'(P - \Phi' P \Phi)\boldsymbol{x}_k = \lambda^2 X(\boldsymbol{x}_k) \end{aligned} \tag{8}$$

es decir, la superficie $X(\boldsymbol{x}_k)$ original se ve afectada por un cambio de escala λ^2, o lo que es lo mismo, la relación $X(\boldsymbol{x}_k)/\boldsymbol{x}_k$ es igual a la relación $X(\lambda \boldsymbol{x}_k)/\lambda^2 \boldsymbol{x}_k$. Por lo tanto, toda la superficie $X(\boldsymbol{x}_k)$ se puede caracterizar eligiendo adecuadamente sólo una parte de ella. En particular, si se elige un subconjunto del dominio de entrada tal que $||\boldsymbol{x}_k|| = 1$, es decir:

$$\boldsymbol{x}_k(\theta) = \begin{pmatrix} cos\ \theta \\ sen\ \theta \end{pmatrix}, \; 0 \leq \theta \leq 2\pi$$

entonces toda la superficie $X(\boldsymbol{x}_k)$ queda caracterizada por su intersección con el subconjunto elegido, que en este caso es el equivalente a un cilindro de radio unidad y origen el centro. Dicha intersección se muestra en la figura 1 y es la curva $X_n(\theta)$ que rodea todo el cilindro,

$$X_n(\theta) = X(\boldsymbol{x}_k(\theta)), \; 0 \leq \theta \leq 2\pi$$

Dicha curva admite una representación plana siendo el eje de abscisas la dirección del vector que apunta a cada uno de los puntos de la curva. Dicha dirección varía entre 0 y 2π.

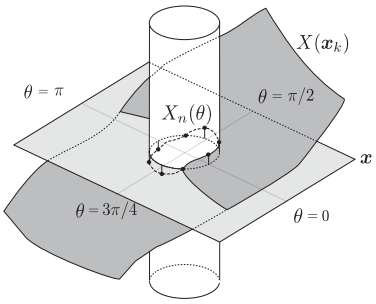


Figura 1: Obtención de la curva $X_n(\theta)$.

El caso de la superficie $Y(\boldsymbol{x}_k)$ dada por (6) no es el mismo, ya que no cumple una relación similar a la de la ecuación (8). Lo que se hace entonces es proyectar sobre el cilindro unidad todos los puntos de $Y(x_k)$ que se encuentran en una dirección determinada θ_i, es decir, aquellos que cumplen:

$$\boldsymbol{x}_k(\lambda, \theta) = \lambda \begin{pmatrix} cos\ \theta \\ sen\ \theta \end{pmatrix}, \text{ donde } \;\; \theta = \theta_i \; y \; \lambda > 0$$

Es decir, existe una doble intersección: por una parte, la intersección del plano que determina la dirección específica con la superficie $Y(x_k)$, que da lugar a una curva; y por otra, la intersección de dicho plano con el cilindro unidad, que da lugar a una recta vertical. En la figura 2 se indican dichas intersecciones. A partir de aquí se proyectan los puntos de la curva de intersección sobre la recta vertical por medio de $Y(\boldsymbol{x}_k)/||\boldsymbol{x}_k||$, dando lugar a un segmento para una dirección dada. Repitiendo el proceso para todas las direcciones entre 0 y 2π, y mediante una representación equivalente a la mencionada en el párrafo anterior, se obtiene una banda plana $Y_n(\theta)$ que caracteriza totalmente a la superficie $Y(\boldsymbol{x}_k)$:

$$\frac{Y(\boldsymbol{x}_k(\lambda, \theta))}{\lambda} \in Y_n(\theta)$$

Por lo tanto, $X_n(\theta)$ e $Y_n(\theta)$, la curva normalizada de $X(\boldsymbol{x}_k)$ y la banda normalizada de $Y(\boldsymbol{x}_k)$ respectivamente, obtenidas como se ha indicado anteriormente, transforman la condición de estabilidad dada por (7) en la siguiente:

$$X_n(\theta) > Y_n(\theta), \;\; 0 \leq \theta \leq 2\pi \tag{9}$$

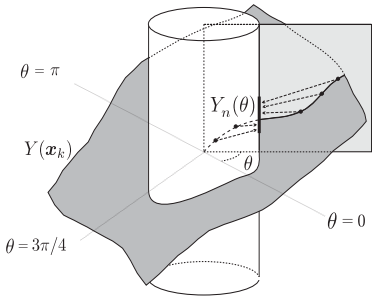


Figura 2: Obtención de la banda plana $Y_n(\theta)$.

es decir, $X_n(\theta)$ es una cota superior de $Y_n(\theta)$ para cualquier dirección θ. Ambas funciones se representan conjuntamente en la figura 3. Esta gráfica es utilizada en la sección siguiente para verificar la estabilidad de un servomotor y su controlador de lógica borrosa.

Una de las características del controlador borroso considerado es su simetría, por lo que la función que lo define es impar $H(-\boldsymbol{x}_k) = -H(\boldsymbol{x}_k)$ y por lo tanto, los términos $X(\boldsymbol{x}_k)$ y $Y(\boldsymbol{x}_k)$ son funciones pares:

$$X(-\boldsymbol{x}_k) = (-\boldsymbol{x}_k')(P - \Phi' P \Phi)(-\boldsymbol{x}_k) = \boldsymbol{x}_k'(P - \Phi' P \Phi)\boldsymbol{x}_k = X(\boldsymbol{x}_k)$$

e

$$\begin{aligned} Y(-\boldsymbol{x}_k) &= 2\Gamma' P \Phi(-\boldsymbol{x}_k) K H(-\boldsymbol{x}_k) + \Gamma' P \Gamma K^2 H^2(-\boldsymbol{x}_k) \\ &= 2\Gamma' P \Phi(-\boldsymbol{x}_k) K(-H(\boldsymbol{x}_k)) + \Gamma' P \Gamma K^2 (-H(\boldsymbol{x}_k))^2 \\ &= 2\Gamma' P \Phi \boldsymbol{x}_k K H(\boldsymbol{x}_k) + \Gamma' P \Gamma K^2 H^2(\boldsymbol{x}_k) = Y(\boldsymbol{x}_k) \end{aligned}$$

Es por ello que para comprobar la estabilidad, será suficiente verificar la condición de estabilidad en la mitad de la region dada por (9)

$$X_n(\theta) > Y_n(\theta), \quad 0 \leq \theta \leq \pi \tag{10}$$

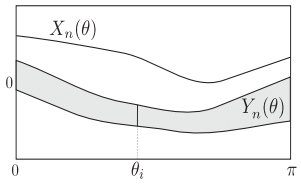


Figura 3: Representación conjunta de $X_n(\theta)$ y $Y_n(\theta)$.

3 Análisis

En primer lugar hay que caracterizar la estabilidad del equilibrio en el origen para el sistema nominal, es decir, el compuesto por la planta sin perturbar, o perturbada con una ganancia variable K de valor nominal 1, y el controlador de lógica borrosa con su diseño original.

Como hemos visto en la sección anterior, la condición de estabilidad de Lyapunov se resume en (10), la cual para el sistema nominal sólo depende de la matriz P, puesto que el resto de los términos se encuentran perfectamente definidos. Es necesario encontrar alguna matriz P que haga que se verifique tal expresión, para demostrar que el equilibrio en el origen es estable.

Por simplicidad seguiremos tratando un sistema de dimensión 2. Para que la matriz de Lyapunov correspondiente

$$P = \begin{pmatrix} p_1 & p_3 \\ p_3 & p_2 \end{pmatrix}$$

donde p_1, p_2 y p_3 son números reales, sea definida positiva ha de cumplir que $p_1 > 0$ y $p_1 p_2 - p_3^2 > 0$. Teniendo en cuenta que para un $\lambda > 0$, si P es de Lyapunov entonces λP también lo es, se puede fijar $p_1 = 1$, con lo que las condiciones anteriores se transforman en $p_2 > 0$ y $|p_3| < \sqrt{p_2}$.

El primer paso para calcular el margen de ganancia es encontrar todas las posibles matrices de Lyapunov, es decir, todos los pares (p_2, p_3) que cumplan la ecuación (10) para el valor nominal de la ganancia K. Se define entonces $\mathcal{P}^K$ como el conjunto de todas las matrices de Lyapunov válidas para un determinado K de ganancia. Dicho conjunto posee las siguientes propiedades:

- $\mathcal{P}^K$ es convexo. Supongamos dos matrices P y $\tilde{P}$ que dan lugar a funciones de Lyapunov en (3), es decir:

$$\boldsymbol{x}_k' \Phi' P \Phi \boldsymbol{x}_k + 2\Gamma' P \Phi \boldsymbol{x}_k K H(\boldsymbol{x}_k) + \Gamma' P \Gamma K^2 H^2(\boldsymbol{x}_k) - \boldsymbol{x}_k' P \boldsymbol{x}_k < 0$$

$$\boldsymbol{x}_k' \Phi' \tilde{P} \Phi \boldsymbol{x}_k + 2\Gamma' \tilde{P} \Phi \boldsymbol{x}_k K H(\boldsymbol{x}_k) + \Gamma' \tilde{P} \Gamma K^2 H^2(\boldsymbol{x}_k) - \boldsymbol{x}_k' \tilde{P} \boldsymbol{x}_k < 0$$

Sumando ambas expresiones tenemos que

$$\boldsymbol{x}_k' \Phi' (P + \tilde{P}) \Phi \boldsymbol{x}_k + 2\Gamma' (P + \tilde{P}) \Phi \boldsymbol{x}_k K H(\boldsymbol{x}_k) + \Gamma' (P + \tilde{P}) \Gamma K^2 H^2(\boldsymbol{x}_k) - \boldsymbol{x}_k' (P + \tilde{P}) \boldsymbol{x}_k < 0$$

también es una función de Lyapunov, por lo que la suma de dos matrices de Lyapunov, $P + \tilde{P}$, da lugar a otra matriz de Lyapunov. Por otra parte, αP y $(1 - \alpha)\tilde{P}$, con $0 \leq \alpha \leq 1$, también son matrices de Lyapunov y como se acaba de comprobar, su suma, $\alpha P + (1 - \alpha)\tilde{P}$, es una matriz de Lyapunov. Esta última suma representa una combinación convexa, de forma que si $P, \tilde{P} \in \mathcal{P}^K$, todos los puntos del segmento que une P con $\tilde{P}$, también pertenenecen a $\mathcal{P}^K$, y son matrices de Lyapunov.

- El conjunto de matrices de Lyapunov válidas para todos los valores intermedios de la ganancia desde un

mínimo hasta un máximo, $\mathcal{P}^{[K_{min}K_{max}]}$, es igual a la intersección de los conjuntos de matrices de Lyapunov correspondientes a dichos valores mínimo y máximo de la ganancia, $\mathcal{P}^{K_{min}} \cap \mathcal{P}^{K_{max}}$. La ecuación (3) puede reescribirse como:

$$\begin{aligned} V(\boldsymbol{x}_{k+1}) - V(\boldsymbol{x}_k) &= \boldsymbol{x}_k'(\Phi' P \Phi - P)\boldsymbol{x}_k + \\ &(2\Gamma' P \Phi \boldsymbol{x}_k H(\boldsymbol{x}_k))K + (\Gamma' P \Gamma H^2(\boldsymbol{x}_k))K^2 \\ &= a(\boldsymbol{x}_k)K^2 + b(\boldsymbol{x}_k)K + c(\boldsymbol{x}_k) < 0 \end{aligned} \qquad (11)$$

Si una cierta matriz P verifica que es de Lyapunov en K_{min} y K_{max}, la expresión (11) es una parábola abierta hacia arriba para cada $\boldsymbol{x}_k$ en función de K pues $a(\boldsymbol{x}_k) > 0$, $\forall \boldsymbol{x}_k \neq 0$, con dos puntos situados debajo del eje de abscisas correspondientes a K_{min} y K_{max}. Entonces, para cualquier valor de K en el intervalo $[K_{min}\ K_{max}]$, K sigue estando debajo del eje de abscisas y la ecuación (11) es también negativa, y por tanto esa matriz P sigue siendo de Lyapunov.

Teniendo en cuenta las propiedades anteriores, se calcula el conjunto $\mathcal{P}^{K_{nom}}$ correspondiente a la ganancia nominal K_{nom}. A partir de aquí se aumenta progresivamente el valor de la ganancia y se determina el conjunto $\mathcal{P}^K$ correspondiente. Mientras el conjunto intersección entre $\mathcal{P}^{K_{nom}}$ y $\mathcal{P}^K$ no sea vacío se continúa con este proceso. Para un determinado valor de la ganancia, dicho conjunto intersección sólo estará formado por un elemento, que da lugar a una matriz que llamaremos P_{max}. El valor de K para dicha situación, K_{max}, es la ganancia máxima permitida:

$$K_{max} = \sup\{K : \mathcal{P}^{K_{nom}} \cap \mathcal{P}^K \neq \emptyset\}$$

Para valores de K superiores a K_{max}, el conjunto intersección es vacío, y no existe ninguna combinación de (p_2, p_3) que cumpla la condición de Lyapunov dada por (10). Por lo tanto, siempre que $K \in [K_{nom}, K_{max}]$, el sistema será exponencialmente estable independientemente del valor que tome K en cada intervalo de muestreo.

A continuación se aplican estas ideas al servomotor. Su dinámica se rige por la siguiente expresión:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0.7 \end{pmatrix} u$$

donde x_1 y x_2 son respectivamente la posición y la velocidad del motor. Las funciones de pertenencia de los antecendentes y de los consecuentes del controlador son las indicadas en la figura 4, y la base de reglas se representa en la tabla 1.

Puesto que se ha modelado la incertidumbre del sistema mediante una ganancia variable, para conocer el grado de robustez del servomotor, es decir, el margen de ganancia, se va a calcular la ganancia máxima,

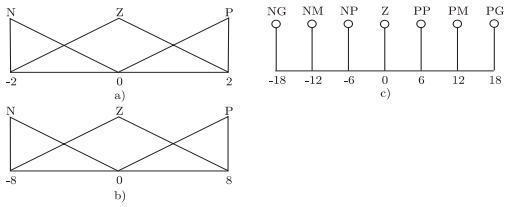


Figura 4: Funciones de pertenencia para el servomotor: a) x_1; b) x_2; c) u.

K_{max}, que puede ser aplicada al sistema haciendo que este siga siendo estable. El sistema en tiempo discreto considerado tiene un tiempo de muestreo de $h = 0.1$ s.

En primer lugar hay que analizar la estabilidad en el origen para el sistema nominal, es decir, la planta sin perturbar o con una K de valor nominal $K_{nom} = 1$. Para ello se halla el conjunto de matrices de Lyapunov correspondiente, $\mathcal{P}^{K_{nom}}$, primero buscando aquellas matrices que son definidas positivas, y después comprobando si estas cumplen la condición de estabilidad (10). Se puede tener en cuenta el hecho de que $\mathcal{P}^K$ es convexo, para así simplificar la búsqueda.

El conjunto $\mathcal{P}^{K_{nom}}$ calculado para el servomotor muestreado con $h = 0.1s$, es decir, los posibles (p_2, p_3) que dan lugar a funciones de Lyapunov para el valor nominal $K_{nom} = 1$, puede verse en la figura 5. Supuesto ahora que la ganancia variable K que per-

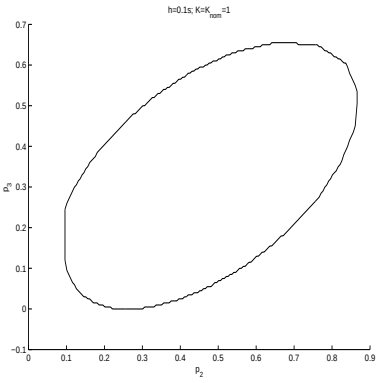


Figura 5: Conjunto $\mathcal{P}^{K_{nom}}$ calculado para $h = 0.1$ s.

$x_1 \backslash x_2$	N	Z	P
N	NP (u_{-1-1})	PM (u_{-10})	PG (u_{-11})
Z	NM (u_{0-1})	Z (u_{00})	PM (u_{01})
P	NG (u_{1-1})	NM (u_{10})	PP (u_{11})

Tabla 1: Base de reglas para el servomotor.

turba la planta se separa de su valor nominal, interesa conocer hasta qué valor puede llegar sin que el sistema deje de ser estable, es decir, el margen de ganancia. Teniendo en cuenta la interpretación de Lyapunov buscamos la ganancia máxima, que denotaremos por K_{max}, correspondiente al máximo valor de K para el cual existe una única matriz $P \in \mathcal{P}^{K_{nom}}$, denominada P_{max}, que permita que se verifique la expresión (10). En estas condiciones, se garantiza la estabilidad en el rango de valores de K entre K_{nom} y K_{max}, puesto que la matriz P_{max} cumple la ecuación (10) en dicho rango. Por lo tanto, el sistema será exponencialmente estable independientemente de como varíe K en cada instante de muestreo.

Partiendo del conjunto $\mathcal{P}^{K_{nom}}$, el valor nominal de K se incrementa y se determina el conjunto $\mathcal{P}^K$ correspondiente. Este proceso continúa en tanto la intersección entre $\mathcal{P}^{K_{nom}}$ y $\mathcal{P}^K$ no sea vacía. Para un determinado valor de ganancia, dicho conjunto intersección sólo estará formado por un elemento, que da lugar a una matriz que llamaremos P_{max}. El valor de K para dicha situación, K_{max}, es el margen de ganancia. Para que el valor del margen de ganancia obtenido tenga una buena precisión, es necesario que los incrementos sucesivos de K sean lo suficientemente pequeños.

Los resultados pueden verse en la figura 6. La matriz de Lyapunov P_{max} correspondiente a $(p_2, p_3) = (0.135, 0.041)$, es matriz de Lyapunov para cualquier K desde K_{nom} hasta $K_{max} = 2.558$. Las figuras 7.a y 7.b muestran como la curva $X_n(\theta)$ acota superiormente la banda $Y_n(\theta)$ para ambos valores de K.

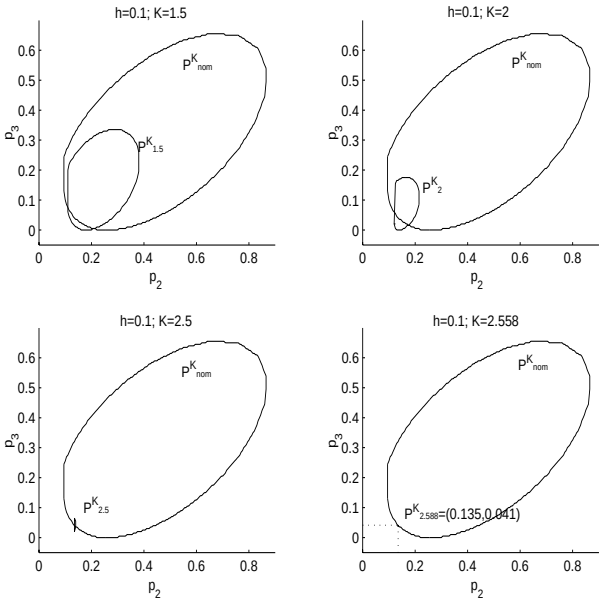


Figura 6: $\mathcal{P}^K$ para K igual a 1.5, 2, 2.5 y 2.558.

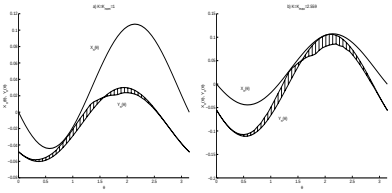


Figura 7: $X_n(\theta)$ e $Y_n(\theta)$ para $(p_2, p_3) = (0.135, 0.041)$. a) $K = K_{nom} = 1$. b) $K = 2.558$.

4 Diseño Robusto

Supongamos que es necesario incrementar el margen de ganancia hasta un nuevo valor K_{MAX}, tal que $K_{MAX} > K_{max}$. Como no existe ningún (p_2, p_3) que verifique las condiciones de Lyapunov para el nuevo rango de valores de K, comprendido ahora entre K_{nom} y K_{MAX}, la solución es modificar el controlador de lógica borrosa original, $H(\boldsymbol{x}_k)$, hasta un nuevo $\tilde{H}(\boldsymbol{x}_k)$ tal que cumpla

$$X_n(\theta) > \tilde{Y}_n(\theta) \tag{12}$$

donde $\tilde{Y}(\theta)$ es por el único término que depende del nuevo controlador $\tilde{H}(\boldsymbol{x}_k)$. La verificación de (12) no sólo depende de $\tilde{H}(\boldsymbol{x}_k)$ sino también de P_{MAX}. Para ello igual que en [3] se toma $P_{MAX} = P_{max}$. La modificación más aconsejable sobre el controlador de lógica borrosa es la de los consecuentes de la base de reglas. El procedimiento de rediseño consiste en recorrer todos los rectángulos en que se ha dividido el dominio de entrada, y buscar para cada uno de ellos, las combinaciones de consecuentes que cumplan la condición de estabilidad para K_{nom} y K_{MAX} simultáneamente. Si existe tal combinación, puesto que cumple las dos condiciones de estabilidad utilizando la misma P_{MAX}, también la cumplirá para cualquier valor de ganancia entre K_{nom} y K_{MAX} en dicho rectángulo. Por lo tanto, para R_1 se calcularán las combinaciones de consecuentes $\{u_{-1-1}, u_{-10}, u_{01}\}$ estables para las dos ganancias, y para R_2, de igual forma las combinaciones $\{u_{-1-1}, u_{-10}, u_{0-1}\}$. El consecuente u_{00} es fijo y de valor nulo. Dado que el controlador es simétrico, los consecuentes necesarios para determinar el controlador son u_{-1-1}, u_{-10}, u_{-11} y u_{01}. Por lo tanto, los controladores resultantes, $\{u_{-1-1}, u_{-10}, u_{-11}, u_{01}\}$, son aquellos obtenidos combinando las soluciones estables $\{u_{-1-1}, u_{-10}, u_{01}\}$ y $\{u_{-1-1}, u_{-10}, u_{0-1}\}$, con la condición de que u_{10} sea igual en ambas y que u_{0-1} sea igual a $-u_{01}$. Para un $K_{MAX} = 3.75$, siguendo el algoritmo descrito anteriormente, se obtienen 5 posibles combinaciones $(u_{00}, u_{-10}, u_{01}, u_{-1,-1})$, todas ellas nos dan un controlador borroso estable. Finalmente, nos quedamos con aquel controlador más parecido al controlador original. El grado de semejanza es medido mediante la diferencia cuadrática entre las com-

	u_{-1-1}	u_{-10}	u_{-11}	u_{01}	distancia
original	NP	PM	PG	PM	0
1	Z	PM	NP	NG	42
2	Z	PM	NM	NM	42
3	PP	PM	NP	NG	45
4	PP	PM	NM	NM	45
5	PM	PM	NP	NG	50

Tabla 2: Controladores estables para $K_{MAX} = 3.75$.

binaciones obtenidas y la original. En la tabla 2 se encuentran los 5 nuevos controladores estables más semejantes al controlador original.

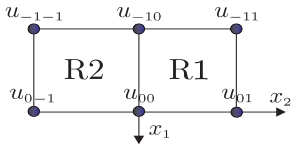


Figura 8: Secuencia de elección de centroides.

Como comprobación de los resultados obtenidos en las figura 9 puede verse el cumplimiento de la condición de estabilidad para K_{nom} y K_{MAX}. Además, en las figuras 10 y 11 se visualizan las trayectorias en el plano de fase de los controladores original y rediseñado para valores de $K = K_{nom} = 1$ y $K = K_{MAX} = 3.75$. Se observa que en el caso del diseño original las trayectorias se alejan del origen para K_{MAX}, lo cual significa que el sistema es inestable. Sin embargo, para el controlador rediseño robusto las trayectorias convergen en el origen y por consiguiente el sistema es estable, para ambas ganancias.

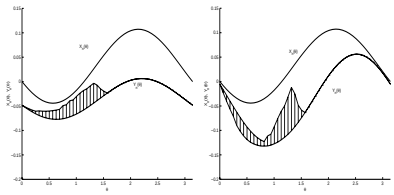


Figura 9: Condición de estabilidad para controlador rediseñado. a) $K = K_{nom} = 1$. b) $K = K_{MAX} = 3.75$

5 Conclusiones

En este artículo se presenta una herramienta basada en la teoría de la estabilidad de Lyapunov. Esta permite analizar la estabilidad de un sistema con control borroso, además de la robustez frente a imprecisiones y perturbaciones de la planta. También puede aplicarse

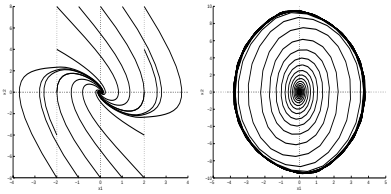


Figura 10: Plano de fase controlador original. a) $K = K_{nom} = 1$. b) $K = K_{MAX} = 3.75$

para la mejora de la robustez mediante un rediseño del controlador.

Agradecimientos

Este trabajo ha sido realizado dentro los proyectos DPI00-1218 y DPI02-4401 del M.C.Y.T.

Referencias

[1] K. J. Åström, B. Wittenmark, (1997) Computer-Controlled Systems: Theory and Design, 3rd ed, Prentice-Hall, Inc, Englewood Cliffs, N. Y.

[2] D. Driankov, H. Hellendoorn, M. Reinfrank, (1993) An Introduction to Fuzzy Control, Springer-Verlag, Berlin.

[3] A. Espada, (1996) Estabilidad y Robustez de Sistemas no Lineales de Control. Análisis y Diseño de Controladores Borrosos, Tesis Doctoral. Departamento de Ingeniería de Sistemas y Automática, Universidad de Vigo, España.

[4] W. M. Haddad, D. S. Bernstein, (1993) "Explicit Construction of Quadratic Lyapunov Functions for the Small Gain, Positivity, Circle, and Popov Theorems and Their Application to Robust Stability Part I: Continuous-Time Theory",Int. J. Robust and Nonlinear Control, Vol.3, pp.313-339.

[5] J. E. Slotine, W. Li, (1991) Applied Nonlinear Control, Prentice-Hall, Englewood Cliffs, N.Y.

[6] M. Vidyasagar, (1993) Nonlinear System Analysis, Prentice-Hall, Inc, Englewood Cliffs, N. Y.

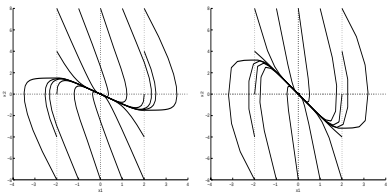


Figura 11: Plano de fase controlador rediseñado. a) $K = K_{nom} = 1$. b) $K = K_{MAX} = 3.75$

Algoritmo para el Cálculo de la Matriz Jacobiana de un Sistema de Control Borroso en Lazo Cerrado

José M. Andujar
Dpto. de Ing. Electrónica, Sistemas Informáticos y Automática, Universidad de Huelva,
Escuela Politécnica Superior, Carretera Huelva-La Rábida, 21071-Palos de la Frontera (Huelva), España
andujar@diesia.uhu.es

Antonio J. Barragán
Dpto. de Ing. Electrónica, Sistemas Informáticos y Automática, Universidad de Huelva,
Escuela Politécnica Superior, Carretera Huelva-La Rábida, 21071-Palos de la Frontera (Huelva), España
antonio.barragan@diesia.uhu.es

Resumen

La estabilidad de los sistemas no lineales ha de ser investigada sin realizar aproximaciones lineales de los mismos. Para este fin existen varias técnicas basadas en el segundo método de Lyapunov; por ejemplo, el método de Krasovskii permite probar la condición suficiente para la estabilidad asintótica de sistemas no lineales. Este método exige el cálculo de la matriz Jacobiana. En este artículo se presenta un modelo matemático equivalente de un sistema de control no lineal multivariable basado en la teoría de la lógica borrosa. Posteriormente se emplea este modelo para calcular la matriz Jacobiana de un sistema de control borroso en lazo cerrado. Finalmente, se propone un algoritmo para la resolución de la matriz Jacobiana. Este algoritmo emplea una metodología basada en la extensión del vector de estado. Este trabajo puede ser útil para el análisis y diseño de sistemas de control borroso.

Palabras Clave: Algoritmo, Control Borroso, Estabilidad, Matriz Jacobiana.

1 Introducción

Una de las mayores críticas al empleo de la lógica borrosa en el diseño de sistemas de control es su escasa formalización matemática. Frecuentemente se adoptan soluciones empíricas que requieren la inclusión del conocimiento de un experto, y son usuales las soluciones *ad-hoc*, sólo válidas para un problema concreto. Los autores de este artículo trabajan en la formalización de los sistemas de control borroso [1, 2, 3, 7], de forma que les sean de aplicación las herramientas matemáticas de análisis y síntesis aceptadas en la comunidad científica.

La teoría de Lyapunov puede ser utilizada para el análisis de la estabilidad de los sistemas TSK (Takagi-Sugeno-Kang) [24, 28], no obstante, el problema es siempre el mismo: encontrar una función de Lyapunov adecuada.

La mayoría de los métodos de diseño para probar la estabilidad de los sistemas de control borroso están basados en la teoría clásica de control [17]: criterio de estabilidad de Popov [10, 12], criterio del círculo [8, 12, 13], método directo de Lyapunov [9, 11, 12, 20, 21, 23, 25], espacio de fase [8, 14], función descriptiva [5], índices de estabilidad y robustez de sistemas [5, 6, 8], teoría de estabilidad de entrada/salida [5, 8, 12, 13, 15], criterio de conicidad [4, 5, 8], hiperestabilidad de Popov [18, 19] y métodos heurísticos [5, 26, 30].

En este trabajo se presenta un algoritmo general para el cálculo de la matriz Jacobiana de un sistema de control borroso en lazo cerrado, como paso fundamental en gran cantidad de técnicas de análisis y síntesis de sistemas de control, como por ejemplo en el método de Krasovskii. Así mismo, se deja abierta la puerta para sintetizar, mediante el teorema de Krasovskii, sistemas de control borroso que aseguren la estabilidad asintótica global.

Este artículo está organizado en las siguientes secciones: la primera, *Modelo borroso no lineal multivariable de un sistema de control*, describe el desarrollo matemático hasta su expresión final de un modelo general de planta identificada mediante técnicas de lógica borrosa, así como la aplicación a la misma de un controlador borroso también general. A continuación se desarrolla el *Cálculo de la matriz jacobiana de un sistema de control borroso en lazo cerrado*. Seguidamente, se utilizan las expresiones desarrolladas para el diseño de un algoritmo. Por último, se proponen algunas conclusiones.

2 Modelo borroso no lineal multivariable de un sistema de control

Considérese una planta multidimensional descrita por:

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, ..., x_n; u_1, u_2, ..., u_m) \\ &\cdots \\ \dot{x}_n &= f_n(x_1, x_2, .., x_n; u_1, u_2, ..., u_m) \end{aligned} \quad (1)$$

donde el controlador multivariable se define como:

$$u_1 = \phi_1(x_1, x_2, \ldots, x_n); \ldots ; u_m = \phi_m(x_1, x_2, \ldots, x_n) \quad (2)$$

Las expresiones (1) y (2) indican un modelo general tanto para la planta como para el controlador, sin ningún tipo de restricciones, por lo que pueden ser no lineales. El sistema puede ser escrito de forma simplificada como en (3).

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{3}$$

$$\dot{\mathbf{x}} = (\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_n)^T;\ \mathbf{x} = (x_1, x_2, \ldots, x_n)^T \text{ y } \mathbf{u} = (u_1, u_2, \ldots, u_m)^T$$

Un modelo borroso equivalente múltiple-entrada-múltiple-salida del proceso puede ser representado mediante el siguiente conjunto de reglas [16, 22]:

$$\begin{aligned} &R^{(l)}\text{: SI } x_1 \text{ es } A_{1i}^l \text{ y } x_2 \text{ es } A_{2i}^l, \ldots, \text{ y } x_n \text{ es } A_{ni}^l \text{ y} \\ &u_1 \text{ es } B_{1i}^l \text{ y } u_2 \text{ es } B_{2i}^l, \ldots, \text{ y } u_m \text{ es } B_{mi}^l \\ &\text{ENTONCES } \dot{x}_i^l \text{ es } g_i^l(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}_i^l) \end{aligned} \tag{4}$$

Donde l=1...M es el número de reglas del modelo borroso de la planta; A_{ki}^l el conjunto borroso definido en el universo de discurso de la variable de estado x_k, k=1... n y la i-ésima ecuación diferencial del proceso; y B_{ji}^l el conjunto borroso definido en el universo de discurso de la j-ésima señal de control u_j, j=1... m, y la i-ésima ecuación diferencial del proceso. Considerando en la expresión (4) que el consecuente es TSK con término afín:

$$g_i^l(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}_i^l) = a_{0i}^l + a_{1i}^l x_1 + \cdots + a_{ni}^l x_n + b_{1i}^l u_1 + \ldots + b_{mi}^l u_m \tag{5}$$

Donde a_{ki}^l y b_{ji}^l representan, respectivamente, el coeficiente constante para la variable de estado x_k y para la j-ésima señal de control de la ecuación diferencial i que representa el proceso. El vector $\boldsymbol{\theta}_i^l$ representa el conjunto de parámetros adaptables:

$$\boldsymbol{\theta}_i^l = (a_{0i}^l, a_{1i}^l, \ldots, a_{ni}^l;\ b_{1i}^l, b_{2i}^l, \ldots, b_{mi}^l)^T \tag{6}$$

Considerando un sistema borroso TSK con desborrosificador centro-promedio y método de inferencia producto, la expresión (3) puede ser escrita como [28]:

$$\dot{x}_i = \sum_{l=1}^{M} w_i^l g_i^l(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}_i^l) \Big/ \sum_{l=1}^{M} w_i^l \tag{7}$$

Donde w_i^l es el grado de cumplimiento de la regla l:

$$w_i^l = \prod_{k=1}^{n} \mu_{F_k}^l(x_k, \boldsymbol{\theta}_i^l) \prod_{j=1}^{m} \mu_{F_j}^l(u_j, \boldsymbol{\theta}_i^l) \tag{8}$$

Siendo $\mu_{F_k}^l(x_k, \boldsymbol{\theta}_i^l)$ y $\mu_{F_j}^l(u_j, \boldsymbol{\theta}_i^l)$ las funciones de pertenencia definidas en sus correspondientes universos de discurso X y U para el conjunto borroso $F \in (X \cup U)$ representado por un conjunto de pares ordenados $\left\{\left(x_k, \mu_{F_k}^l(x_k, \boldsymbol{\theta}_i^l)\right) : x_k \in X\right\}$ y $\left\{\left(u_j, \mu_{F_j}^l(u_j, \boldsymbol{\theta}_i^l)\right) : u_j \in U\right\}$.

Si el consecuente de la regla tiene la forma definida en (5), entonces se puede escribir (7) como en (9) o (10):

$$\dot{x}_i = \frac{1}{\sum_{l=1}^{M} w_i^l} \left[\begin{aligned} &\sum_{l=1}^{M} w_i^l a_{0i}^l + \cdots + \sum_{l=1}^{M} w_i^l a_{ni}^l x_n + \\ &+ \sum_{l=1}^{M} w_i^l b_{1i}^l u_1 + \cdots + \sum_{l=1}^{M} w_i^l b_{mi}^l u_m \end{aligned} \right] \tag{9}$$

$$\dot{x}_i = a_{0i}^t + \sum_{k=1}^{n} a_{ki}^t x_k + \sum_{j=1}^{m} b_{ji}^t u_j \tag{10}$$

Donde a_{0i}^t, a_{ki}^t y b_{ji}^t son coeficientes variables [29] :

$$a_{0i}^t = \sum_{l=1}^{M} w_i^l a_{0i}^l \Big/ \sum_{l=1}^{M} w_i^l;\ a_{ki}^t = \sum_{l=1}^{M} w_i^l a_{ki}^l \Big/ \sum_{l=1}^{M} w_i^l;\ k = 1, \ldots, n \tag{11}$$

$$b_{ji}^t = \sum_{l=1}^{M} w_i^l b_{ji}^l \Big/ \sum_{l=1}^{M} w_i^l;\ j = 1, \ldots, m \tag{12}$$

Obtenido el modelo equivalente borroso de la planta, se procederá de forma similar con el controlador. Para ello se considera que su diseño es completamente independiente del proceso de identificación del modelo de la planta, de forma que la partición del universo de discurso de los estados para la planta y controlador no tienen porqué coincidir. Por lo mismo, sus modelos borrosos no tienen porqué tener el mismo número de reglas. En consecuencia, el controlador borroso puede ser representado por el siguiente conjunto de reglas:

$$\begin{aligned} &R^{(r)}\text{: SI } x_1 \text{ es } C_{1j}^r \text{ y } x_2 \text{ es } C_{2j}^r, \ldots, \text{ y } x_n \text{ es } C_{nj}^r \\ &\text{ENTONCES } u_j^r = c_{0j}^r + c_{1j}^r x_1 + \cdots + c_{nj}^r x_n \end{aligned} \tag{13}$$

Donde C_{kj}^r representa el conjunto borroso de las k = 1…n variables de estado, con j = 1...m acciones de control y r = 1...N reglas.

Procediendo de forma similar a la mostrada en (8), (9) y (10), las señales de control pueden expresarse por (14), o de forma más simplificada, por (15),

$$u_j = \left[\sum_{r=1}^{N} \omega_j^r c_{0j}^r + \sum_{r=1}^{N} \omega_j^r c_{1j}^r + \cdots + \sum_{r=1}^{N} \omega_j^r c_{nj}^r x_n \right] \Big/ \sum_{r=1}^{N} \omega_j^r \tag{14}$$

$$u_j = c_{0j}^t + c_{1j}^t x_1 + c_{2j}^t x_2 + \cdots + c_{nj}^t x_n = c_{0j}^t + \sum_{k=1}^{n} c_{kj}^t x_k \tag{15}$$

donde los coeficientes variables vienen determinados por (16) y el grado de activación de la regla r del controlador, por (17).

$$c_{0j}^t = \sum_{r=1}^{N} \omega_j^r c_{0j}^r \Big/ \sum_{r=1}^{N} \omega_j^r;\ c_{kj}^t = \sum_{r=1}^{N} \omega_j^r c_{kj}^r \Big/ \sum_{r=1}^{N} \omega_j^r \tag{16}$$

$$\omega_j^r = \prod_{k=1}^{n} \mu_{F_k}^r(x_k, \boldsymbol{\varphi}_j^r) \tag{17}$$

$\boldsymbol{\varphi}_j^r$ es el vector de parámetros adaptables del controlador:

$$\boldsymbol{\varphi}_j^r = (c_{0j}^r, c_{1j}^r, \ldots, c_{nj}^r) \tag{18}$$

Sustituyendo la expresión (15) en (10), se obtiene la ecuación del sistema no lineal en lazo cerrado:

$$\dot{x}_i = a_{0i}^t + \sum_{j=1}^{m} b_{ji}^t c_{0j}^t + \sum_{k=1}^{n}\left(a_{ki}^t + \sum_{j=1}^{m} b_{ji}^t c_{kj}^t \right) x_k \qquad (19)$$

Con objeto de simplificar la ecuación (19) el vector de estados se extiende en una coordenada, de manera que:

$$\tilde{\mathbf{x}} = (1, \mathbf{x})^T = (x_0, x_1, \ldots, x_n)^T ;\ x_0 = 1 \qquad (20)$$

Empleando en (19) el vector de estados extendido:

$$\dot{x}_i = \sum_{k=0}^{n}\left(a_{ki}^t + \sum_{j=1}^{m} b_{ji}^t c_{kj}^t \right) \tilde{x}_k \qquad (21)$$
$$k = 0,1,\ldots,n,\ i = 1,\ldots,n \text{ y } j = 1,\ldots,m$$

3 Cálculo de la Matriz Jacobiana de un sistema de control borroso en lazo cerrado

El sistema de control no lineal multivariable definido por la ecuación (21), puede ser representado en la forma:

$$\dot{x}_1 = \xi_1(x_1, x_2, \ldots, x_n); \ldots; \dot{x}_n = \xi_n(x_1, x_2, \ldots, x_n) \qquad (22)$$

Si $\mathbf{x}^e = (x_1^e, x_2^e, \ldots, x_n^e)^T$ es un estado de equilibrio para el sistema (19), su desarrollo en serie de Taylor será:

$$\frac{d}{dt}(x_i - x_i^e) = (x_1 - x_1^e)\left.\frac{\partial \xi_i(x_1^e, x_2^e, \ldots, x_n^e)}{\partial x_1}\right|_{\mathbf{x}^e} + \\ + \ldots + (x_n - x_n^e)\left.\frac{\partial \xi_i(x_1^e, x_2^e, \ldots, x_n^e)}{\partial x_n}\right|_{\mathbf{x}^e} + R_i(\mathbf{x}^e) \qquad (23)$$

S el término no lineal $R_i(\mathbf{x}^e)$ cumple (24), siendo K_i una constante lo suficientemente pequeña, entonces la matriz jacobiana del sistema determina las propiedades del estado de equilibrio $\mathbf{x}^e$.

$$\left|R_i(\mathbf{x}^e)\right| \le K_i\left[(x_1 - x_1^e)^2 + \ldots + (x_n - x_n^e)^2\right] \qquad (24)$$

La matriz jacobiana se calcula mediante la expresión:

$$\mathbf{J}_{iq}(\mathbf{x})\Big|_{\mathbf{x}^e} = \begin{pmatrix} \left(\partial \xi_1(\mathbf{x})/\partial x_1\right)\Big|_{\mathbf{x}^e} & \cdots & \left(\partial \xi_1(\mathbf{x})/\partial x_n\right)\Big|_{\mathbf{x}^e} \\ \vdots & \cdots & \vdots \\ \left(\partial \xi_n(\mathbf{x})/\partial x_1\right)\Big|_{\mathbf{x}^e} & \cdots & \left(\partial \xi_1(\mathbf{x})/\partial x_n\right)\Big|_{\mathbf{x}^e} \end{pmatrix} \qquad (25)$$

En la expresión anterior es necesario calcular las derivadas parciales de la expresión (21) con respecto a cada una de las variables de estado. Esta ecuación se puede dividir en dos sumandos:

$$\dot{x}_i = \sum_{k=0}^{n}\left(a_{ki}^t + \sum_{j=1}^{m} b_{ji}^t c_{kj}^t \right) \tilde{x}_k = \sum_{k=0}^{n} a_{ki}^t \tilde{x}_k + \sum_{k=0}^{n}\sum_{j=1}^{m} b_{ji}^t c_{kj}^t \tilde{x}_k \qquad (26)$$

Las derivadas parciales serán obtenidas de forma general para las $q = 1 \ldots n$ variables de estado y las $i = 1 \ldots n$ ecuaciones diferenciales de primer orden.

3.1. Derivada del primer sumando

La derivada parcial del primer sumando de (26) es:

$$\frac{\partial\left(\sum_{k=0}^{n} a_{ki}^t \tilde{x}_k\right)}{\partial x_q} = \sum_{k=0}^{n} \frac{\partial(a_{ki}^t \tilde{x}_k)}{\partial x_q} = \sum_{k=0}^{n}\left(\frac{\partial a_{ki}^t}{\partial x_q} \tilde{x}_k \right) + a_{qi}^t \qquad (27)$$

Teniendo en cuenta (11):

$$\frac{\partial a_{ki}^t}{\partial x_q} = \frac{\partial}{\partial x_q}\left(\sum_{l=1}^{M} w_i^l a_{ki}^l \Big/ \sum_{l=1}^{M} w_i^l \right) \qquad (28)$$

pero

$$\frac{\partial\left(\sum_{l=1}^{M} w_i^l a_{ki}^l\right)}{\partial x_q} = \sum_{l=1}^{M} \frac{\partial w_i^l}{\partial x_q} a_{ki}^l, \text{ y} \qquad (29)$$

$$\frac{\partial\left(\sum_{l=1}^{M} w_i^l\right)}{\partial x_q} = \sum_{l=1}^{M} \frac{\partial w_i^l}{\partial x_q}. \qquad (30)$$

Sustituyendo (29) y (30) en (28) y simplificando:

$$\frac{\partial a_{ki}^t}{\partial x_q} = \sum_{l,p=1}^{M}\left(\frac{\partial w_i^l}{\partial x_q} w_i^p \left(a_{ki}^l - a_{ki}^p\right) \right) \Big/ \left(\sum_{l=1}^{M} w_i^l \right)^2 \qquad (31)$$

Finalmente, la ecuación (27) puede ser escrita como:

$$\frac{\partial\left(\sum_{k=0}^{n} a_{ki}^t \tilde{x}_k\right)}{\partial x_q} = a_{qi}^t + \sum_{k=0}^{n}\left(\frac{\sum_{l,p=1}^{M}\left(\frac{\partial w_i^l}{\partial x_q} w_i^p (a_{ki}^l - a_{ki}^p) \right)}{\left(\sum_{l=1}^{M} w_i^l \right)^2} \tilde{x}_k \right) \qquad (32)$$

La expresión (32) necesita el cálculo previo de la derivada del grado de activación de las reglas de la planta, dada por la ecuación (8). Este punto se resuelve en el apartado 3.4.

3.2. Derivada del segundo sumando

La derivada del segundo sumando está dada por:

$$\frac{\partial\left(\sum_{k=0}^{n}\sum_{j=1}^{m} b_{ji}^t c_{kj}^t \tilde{x}_k\right)}{\partial x_q} = \sum_{k=0}^{n}\left(\frac{\partial\left(\sum_{j=1}^{m} b_{ji}^t c_{kj}^t\right)}{\partial x_q} \tilde{x}_k \right) + \sum_{j=1}^{m} b_{ji}^t c_{qj}^t \qquad (33)$$

siendo

$$\frac{\partial\left(\sum_{j=1}^{m} b_{ji}^t c_{kj}^t\right)}{\partial x_q} = \sum_{j=1}^{m}\left(\frac{\partial b_{ji}^t}{\partial x_q} c_{kj}^t + \frac{\partial c_{kj}^t}{\partial x_q} b_{ji}^t \right) \qquad (34)$$

Dada la similitud de las expresiones para el cálculo de a_{ki}^t y b_{ji}^t, es fácil deducir que:

$$\frac{\partial b_{ji}^t}{\partial x_q} = \sum_{l,p=1}^{M}\left(\frac{\partial w_i^l}{\partial x_q} w_i^p (b_{ji}^l - b_{ji}^p)\right) \Bigg/ \left(\sum_{l=1}^{M} w_i^l\right)^2 \tag{35}$$

La derivada correspondiente al controlador es:

$$\frac{\partial c_{kj}^t}{\partial x_q} = \frac{\dfrac{\partial\left(\sum_{r=1}^{N}\omega_j^r c_{kj}^r\right)}{\partial x_q}\sum_{r=1}^{N}\omega_j^r - \sum_{r=1}^{N}\omega_j^r c_{kj}^r \dfrac{\partial\left(\sum_{r=1}^{N}\omega_j^r\right)}{\partial x_q}}{\left(\sum_{r=1}^{N}\omega_j^r\right)^2} \tag{36}$$

donde:

$$\frac{\partial\left(\sum_{r=1}^{N}\omega_j^r c_{kj}^r\right)}{\partial x_q} = \sum_{r=1}^{N}\frac{\partial \omega_j^r}{\partial x_q} c_{kj}^r \tag{37}$$

$$\frac{\partial\left(\sum_{r=1}^{N}\omega_j^r\right)}{\partial x_q} = \sum_{r=1}^{N}\frac{\partial \omega_j^r}{\partial x_q} \tag{38}$$

Sustituyendo las expresiones (37) y (38) en (36):

$$\frac{\partial c_{kj}^t}{\partial x_q} = \sum_{r,s=1}^{N}\left(\frac{\partial \omega_j^r}{\partial x_q}\omega_j^s (c_{kj}^r - c_{kj}^s)\right) \Bigg/ \left(\sum_{r=1}^{N}\omega_j^r\right)^2 \tag{39}$$

Para evaluar la expresión (39) es necesario calcular la derivada del grado de activación de las reglas del controlador, lo cual se resuelve en el apartado 3.3.

Una vez obtenidas las expresiones intermedias (35) y (39) éstas pueden sustituirse en (34):

$$\frac{\partial\left(\sum_{j=1}^{m} b_{ji}^t c_{kj}^t\right)}{\partial x_q} = \sum_{j=1}^{m}\left(\begin{array}{l} \dfrac{\sum_{l,p=1}^{M}\left(\frac{\partial w_i^l}{\partial x_q} w_i^p (b_{ji}^l - b_{ji}^p)\right)}{\left(\sum_{l=1}^{M} w_i^l\right)^2} c_{kj}^t + \\ + \dfrac{\sum_{r,s=1}^{N}\left(\frac{\partial \omega_j^r}{\partial x_q}\omega_j^s (c_{kj}^r - c_{kj}^s)\right)}{\left(\sum_{r=1}^{N}\omega_j^r\right)^2} b_{ji}^t \end{array}\right) \tag{40}$$

Sustituyendo (40) en (33) se obtiene:

$$\begin{aligned} \frac{\partial\left(\sum_{k=0}^{n}\sum_{j=1}^{m} b_{ji}^t c_{kj}^t \tilde{x}_k\right)}{\partial x_q} &= \sum_{j=1}^{m} b_{ji}^t c_{qj}^t + \\ &+ \sum_{k=0}^{n}\left(\sum_{j=1}^{m}\left(\begin{array}{l} \dfrac{\sum_{l,p=1}^{M}\left(\frac{\partial w_i^l}{\partial x_q} w_i^p (b_{ji}^l - b_{ji}^p)\right)}{\left(\sum_{l=1}^{M} w_i^l\right)^2} c_{kj}^t + \\ + \dfrac{\sum_{r,s=1}^{N}\left(\frac{\partial \omega_j^r}{\partial x_q}\omega_j^s (c_{kj}^r - c_{kj}^s)\right)}{\left(\sum_{r=1}^{N}\omega_j^r\right)^2} b_{ji}^t \end{array}\right)\tilde{x}_k\right) \end{aligned} \tag{41}$$

3.4. Derivada del grado de activación de las reglas del controlador

Dada la expresión (17) que define el grado de activación de las reglas del controlador, la derivada parcial de esta expresión con respecto a cada variable de estado es:

$$\frac{\partial \omega_j^r}{\partial x_q} = \sum_{v=1}^{n}\left(\frac{\partial \mu_{F_v}^r (x_v, \boldsymbol{\varphi}_j^r)}{\partial x_q}\prod_{\substack{k=v \\ k\neq v}}^{n}\mu_{F_k}^r (x_k, \boldsymbol{\varphi}_j^r)\right) \tag{42}$$

Teniendo en cuenta que:

$$\frac{\partial f(x_i)}{\partial x_j} = 0, \quad \forall i \neq j \tag{43}$$

La ecuación (42) puede ser escrita como:

$$\frac{\partial \omega_j^r}{\partial x_q} = \frac{\partial \mu_{F_q}^r (x_q, \boldsymbol{\varphi}_j^r)}{\partial x_q}\prod_{\substack{k=1 \\ k\neq q}}^{n}\mu_{F_k}^r (x_k, \boldsymbol{\varphi}_j^r) \tag{44}$$

3.3. Derivada del grado de activación de las reglas de la planta

Dada la expresión (8) que define el grado de activación de las reglas de la planta, su derivación parcial es

$$\frac{\partial w_i^l}{\partial x_q} = \frac{\partial}{\partial x_q}\left(\underbrace{\prod_{k=1}^{n}\mu_{F_k}^l (x_k, \boldsymbol{\theta}_i^l)}_{w_i^l(\mathbf{x})}\underbrace{\prod_{j=1}^{m}\mu_{F_j}^l (u_j, \boldsymbol{\theta}_i^l)}_{w_i^l(\mathbf{u})}\right) \tag{45}$$

donde

$$\begin{aligned} \frac{\partial w_i^l(\mathbf{u})}{\partial x_q} &= \frac{\partial \mu_{F_1}^l (u_1, \boldsymbol{\theta}_i^l)}{\partial x_q}\prod_{\substack{j=1 \\ j\neq 1}}^{m}\mu_{F_j}^l (u_j, \boldsymbol{\theta}_i^l) + \\ &+ \ldots + \frac{\partial \mu_{F_m}^l (u_m, \boldsymbol{\theta}_i^l)}{\partial x_q}\prod_{\substack{j=1 \\ j\neq m}}^{m}\mu_{F_j}^l (u_j, \boldsymbol{\theta}_i^l) \end{aligned} \tag{46}$$

Con lo cual,

$$\frac{\partial w_i^l(\mathbf{u})}{\partial x_q} = \sum_{v=1}^{m}\left(\frac{\partial \mu_{F_v}^l (u_v, \boldsymbol{\theta}_i^l)}{\partial x_q}\prod_{\substack{j=1 \\ j\neq v}}^{m}\mu_{F_j}^l (u_j, \boldsymbol{\theta}_i^l)\right) \tag{47}$$

Teniendo en cuenta (43):

$$\frac{\partial w_i^l(\mathbf{x})}{\partial x_q} = \frac{\partial \mu_{F_q}^l (x_q, \boldsymbol{\theta}_i^l)}{\partial x_q}\prod_{\substack{k=1 \\ k\neq q}}^{n}\mu_{F_k}^l (x_k, \boldsymbol{\theta}_i^l) \tag{48}$$

Sustituyendo las ecuaciones (47) y (48) en (45):

$$\begin{aligned} \frac{\partial w_i^l}{\partial x_q} &= \frac{\partial \mu_{F_q}^l (x_q, \boldsymbol{\theta}_i^l)}{\partial x_q}\prod_{j=1}^{m}\mu_{F_j}^l (u_j, \boldsymbol{\theta}_i^l)\prod_{\substack{k=1 \\ k\neq q}}^{n}\mu_{F_k}^l (x_k, \boldsymbol{\theta}_i^l) + \\ &+ \prod_{k=1}^{n}\mu_{F_k}^l (x_k, \boldsymbol{\theta}_i^l)\sum_{v=1}^{m}\left(\frac{\partial \mu_{F_v}^l (u_v, \boldsymbol{\theta}_i^l)}{\partial x_q}\prod_{\substack{j=1 \\ j\neq v}}^{m}\mu_{F_j}^l (u_j, \boldsymbol{\theta}_i^l)\right) \end{aligned} \tag{49}$$

Finalmente, $\partial w_i^l(\mathbf{u})/\partial x_q$ exige la derivada parcial del vector de control respecto a cada variable de estado. Aplicando el vector de estados extendido a (15):

$$u_j = \sum_{k=0}^{n} c_{kj}^t \tilde{x}_k \tag{50}$$

Luego,

$$\frac{\partial u_j}{\partial x_q} = \frac{\partial\left(\sum_{k=0}^{n} c_{kj}^t \tilde{x}_k\right)}{\partial x_q} = c_{qj}^t + \sum_{k=0}^{n}\left(\frac{\partial c_{kj}^t}{\partial x_q}\tilde{x}_k\right) \tag{51}$$

Sustituyendo ahora $\partial c_{ki}^t/\partial x_q$ dado por (39):

$$\frac{\partial u_j}{\partial x_q} = c_{qj}^t + \sum_{k=0}^{n}\left(\frac{\sum_{r,s=1}^{N}\left(\frac{\partial \omega_j^r}{\partial x_q}\omega_j^s(c_{kj}^r - c_{kj}^s)\right)}{\left(\sum_{r=1}^{N}\omega_j^r\right)^2}\tilde{x}_k\right) \tag{52}$$

3.5. Matriz Jacobiana

Cada elemento de la matriz jacobiana viene determinado por la derivada de la expresión (26) respecto a cada una de las variables de estado. Los dos sumandos de dicha expresión son las ecuaciones (32) y (41). De esta forma:

$$\mathbf{J}_{iq}(\mathbf{x}) = \frac{\partial \xi_i(\mathbf{x})}{\partial x_q} = a_{qi}^t + \sum_{j=1}^{m} b_{ji}^t c_{qj}^t + \\ +\sum_{k=0}^{n}\left\{\left[\begin{array}{l}\sum_{l,p=1}^{M}\left(\frac{\partial w_i^l}{\partial x_q}w_i^p(a_{ki}^l - a_{ki}^p)\right)\Big/\left(\sum_{l=1}^{M} w_i^l\right)^2 + \\ +\sum_{j=1}^{m}\left(\begin{array}{l}\frac{\sum_{l,p=1}^{M}\left(\frac{\partial w_i^l}{\partial x_q}w_i^p(b_{ji}^l - b_{ji}^p)\right)}{\left(\sum_{l=1}^{M} w_i^l\right)^2}c_{kj}^t + \\ b_{ji}^t\frac{\sum_{r,s=1}^{N}\left(\frac{\partial \omega_j^r}{\partial x_q}\omega_j^s(c_{kj}^r - c_{kj}^s)\right)}{\left(\sum_{r=1}^{N}\omega_j^r\right)^2}\end{array}\right)\end{array}\right]\tilde{x}_k\right\} \tag{53}$$

Con la intención de no complicar la expresión anterior, no se han sustituido las expresiones de $\partial \omega_j^r/\partial x_q$ y $\partial w_i^l/\partial x_q$ dadas por las ecuaciones (44) y (49) respectivamente.

Nótese que las expresiones (44) y (49), y por consiguiente la expresión (53), son dependientes del tipo de función de pertenencia que se escoja para los modelos borrosos de la planta y el controlador.

4 Algoritmo

Sea el sistema definido por (1) y (2) de equivalente borroso TSK con término afín dado por (10). El controlador, considerando igualmente consecuente TSK con término afín, está dado por (15). Sea el modelo borroso en lazo cerrado dado por (21). Dado un punto $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$, los pasos para calcular la matriz jacobiana del sistema en lazo cerrado en dicho punto son:

1. Calcular el grado de activación de las reglas del controlador borroso a través de (17).
2. Calcular los coeficientes variables del controlador borroso mediante la segunda expresión de la ecuación (16), con $k = 0,1,\ldots,n$.
3. Obtener el valor de cada señal de control generada por el controlador empleando (15).
4. Calcular en grado de activación de las reglas del modelo borroso de la planta a utilizando (8).
5. Para la *i*-ésima ecuación del proceso y la *q*-ésima variable de estado:
   - **5.1.** Calcular los coeficientes variables de modelo de la planta mediante la segunda ecuación de (11) y (12), con $k = 0,1,\ldots,n$.
   - **5.2.** Obtener el segundo sumando de la ecuación (53).
   - **5.3.** Calcular la derivada parcial del grado de activación de las reglas del controlador empleando (44).
   - **5.4.** Calcular la derivada parcial del grado de activación de las reglas de la planta mediante (52). Para completar esta expresión es necesario utilizar (52).
   - **5.5.** Obtener el tercer sumando de (53)
   - **5.6.** Calcular el término J_{iq} de la matriz jacobiana.
6. Repetir el paso 5 hasta completar todos los términos de la matriz jacobiana evaluada en el punto **x**.

5 Conclusiones

En este artículo se ha obtenido un modelo matemático equivalente de un sistema de control multivariable no lineal, basado en la teoría de la lógica borrosa. Posteriormente, este modelo ha sido empleado para calcular la matriz jacobiana del sistema de lazo cerrado. Por último, se ha propuesto un algoritmo para resolver dicho cálculo. Este algoritmo emplea una metodología basada en la extensión del vector de estados y mejora al presentado en [1], por ser más general, simple y eficiente.

Los autores de este artículo han diseñado una función de MATLAB® que implementa el algoritmo propuesto, junto con una serie de funciones adicionales que facilitan su utilización.

La resolución de la matriz jacobiana propuesta posibilita el diseño de controladores borrosos que aseguren la estabilidad asintótica. Para ello, se propone utilizar el teorema de Krasovskii, el cual demuestra que si la matriz $\mathbf{J}(\mathbf{x})^T + \mathbf{J}(\mathbf{x})$ resulta definida negativa en un entorno del punto de equilibrio, éste es asintóticamente estable en sus proximidades; además, si $\boldsymbol{\xi}(\mathbf{x})^T\boldsymbol{\xi}(\mathbf{x}) \to \infty$ cuando

$\|\mathbf{x}\| \to \infty$, la estabilidad asintótica es global. En este momento estamos trabajando en este punto y ya hemos obtenido resultados.

6 Referencias

[1] J. M. Andújar, A. J. Barragán, Matriz Jacobiana de un Sistema de Control Borroso en Lazo Cerrado. Aplicación a la Síntesis de Sistemas de Control Borroso. Proc. XXIV Jornadas de Automática, León, España, (2003).

[2] J. M. Andújar, P. Salmerón, O. Sánchez, Control borroso multivariable de una planta de aguas residuales, Proc. SAAEI'2000, Barcelona, España, (2000) 663-666.

[3] J. M. Andújar, Sistemas borrosos multivariables. Modelización neuro-borrosa, control y análisis de estabilidad, Tesis, Universidad de Huelva, España, (2000).

[4] J. Aracil, A. García, A. Ollero, Fuzzy control of dynamical systems. Stability analysis based on the conicity criterion, Proc. 4th International Fuzzy Systems Association Congress, Brussels, Belgium, (1991) 5-8.

[5] J. Aracil, F. Gordillo, Stability issues in fuzzy control, Physica-Verlag, Heidelberg-New York, (2000).

[6] J. Aracil, A. Ollero, A. García, Stability Indices of the Global Analysis of Expert Control Systems, IEEE Transactions on Systems, Man and Cybernetics 19 (5) (1987) 998-1007.

[7] J. M. Bravo, O. Sánchez, J. M. Andújar, E. Fernández, Stability analisys and synthesis of fuzzy systems using intervalar arithmetic, Proc. 15th IFAC World Congress, Barcelona, Spain, (2002).

[8] D. Driankov, H. Hellendoor, M. Reinfrank, An Introduction to Fuzzy Control, Springer-Verlag, Berlin, (1993).

[9] G. Feng, S. G. Cao, N. W. Rees, Stable adaptative control of fuzzy dynamics systems, Fuzzy Sets and Systems 131 (2002) 217-224.

[10] C. C. Fuh, P. C. Tung, Robust stability analysis of fuzzy control systems, Fuzzy Sets and Systems 88 (1997) 289-298.

[11] T. Hung, M. Sugeno, R. Tong, R. R. Yager, Theoretical aspects of fuzzy control, New York: John Wiley and Sons Inc., (1995).

[12] H. R. Khalil, Nonlinear Systems, New York: Macmillan Publishing Company, (1992).

[13] H. R. Lin, W. J. Wang, L_2-stability analysis of fuzzy control systems, Fuzzy Sets and Systems 100 (1998) 159-172.

[14] M. Maeda, S. Murakami, Stability Analysis of fuzzy control systems using phase planes, Japanese Journal of Fuzzy Theory and Systems 3 (2) (1991) 149-160.

[15] H. A. Malki , H. Li, G. Chen, New design and stability analysis of fuzzy proportional-derivative control systems, IEEE Transactions on Fuzzy Systems 2 (4) (1994) 245-254.

[16] T. Nguyen, M. Sugeno, R. Tong, R. R. Yager, Theoretical aspects of fuzzy control, John Wiley Sons, (1995).

[17] A. Piegat, Fuzzy Modeling and control, Phisica-Verlag Heidelberg, (2001).

[18] V. M. Popov, The solution of a new stability problem for controlled systems, Automatic and remoted control 24 (1963) 1-23.

[19] V. M. Popov, Hyperstability of control Systems, Berlin: Springer Verlag, (1973).

[20] C. I. Siettos, G. V. Bafas, Semiglobal stabilization of nonlinear systems using fuzzy control and singular perturbation methods, Fuzzy Sets and Systems 129 (2002) 275-294.

[21] Q. Sun, R. Li, P. Zhang, Stable and optimal adaptative fuzzy control of complex systems using fuzzy diynamic model, Fuzzy Sets and Systems 133 (2003) 1-17.

[22] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Transaction on Systems, Man and Cybernetics 15 (1985) 116-132.

[23] K. Tanaka, M. Sugeno, Stability analysis and design of fuzzy control systems, Fuzzy Sets and Systems 45 (1992) 135-156.

[24] K. Tanaka, Stability and Stabilizability of Fuzzy-Neural-Linear Control Systems, IEEE Transactions on Fuzzy Systems 3 (4) (1995) 438-447.

[25] S. Tong, H. H. Li, Observer-based robust fuzzy control of nonlinear system with parametric uncertainties, Fuzzy Set and Systems 131 (2002) 165-184.

[26] H. O. Wang, K. Tanaka, M. Griffin, An approach to fuzzy control of nonlinear system: stability and design issues, IEEE Transaction on Fuzzy Systems 4 (1) (1996) 14-23.

[27] L. X. Wang, Adaptative fuzzy systems and control, Prentice-Hall, (1994).

[28] L. X. Wang, A Course in Fuzzy Systems and Control, Prentice-Hall, New Jersey, USA, (1997).

[29] L. Wong, F. Leung, P. Tam, Satability design of TS model based fuzzy systems, Proc. Six IEEE international conference on fuzzy systems, Barcelona, Spain, (1997) 83-86.

[30] H. Ying, Practical design of nonlinear fuzzy controllers with stability analysis for regulating processes with unknown mathematical models, Automatica 30 (7) (1994) 1185-1195.

SÍNTESIS DE SISTEMAS DIFUSOS A PARTIR DE VHDL

S. Sánchez-Solano **A. Barriga** **P. Brox** **I. Baturone**
Instituto de Microelectrónica de Sevilla, CNM-CSIC, Sevilla, España.
santiago@imse.cnm.es barriga@imse.cnm.es brox@imse.cnm.es lumi@imse.cnm.es

Resumen

En esta comunicación se presenta una estrategia de diseño para sistemas difusos basada en la utilización del lenguaje de descripción de hardware VHDL. Dicha estrategia consiste en aprovechar la flexibilidad que proporciona este lenguaje para describir los sistemas difusos utilizando un formato muy similar al lenguaje natural, y donde las definiciones de la estructura del sistema y de los operadores difusos se realizan por separado. La técnica utilizada permite diferenciar claramente el comportamiento funcional del sistema de los detalles de implementación, facilita las tareas de modelado de los sistemas y permite enlazar directamente con herramientas de síntesis automática de circuitos microelectrónicos. En la comunicación se analizan los resultados obtenidos mediante esta estrategia y se comparan con otros basados en arquitecturas específicas de los sistemas difusos.

Palabras Clave: Hardware Fuzzy, VHDL, Herramientas CAD.

1 INTRODUCCIÓN

El número de aplicaciones que incluyen elementos de control inteligente basados en lógica difusa se ha incrementado considerablemente en los últimos años. En muchos de estos sistemas se utilizan circuitos integrados con objeto de acelerar la ejecución de lo procesos de inferencia y reducir el tamaño y el consumo de las soluciones adoptadas. Estas características han motivado el desarrollo de numerosas herramientas de ayuda al diseño que exploran diferentes técnicas para la implementación microelectrónica de sistemas difusos [1], [2].

Al igual que ocurre con otro tipo de sistemas, una metodología de diseño de circuitos difusos debe incluir un conjunto de herramientas que faciliten las diferentes etapas de desarrollo del sistema: descripción de especificaciones, verificación de comportamiento, síntesis automática, etc. Con objeto de manejar de forma adecuada la complejidad del sistema y minimizar la posibilidad de introducir errores, en las distintas etapas del proceso de diseño se utilizan diferentes niveles de abstracción para describir el sistema y se direccionan solo aquellos aspectos del diseño que resultan significativos en cada nivel.

En el flujo de diseño de sistemas difusos se pueden distinguir dos niveles de descripción claramente diferenciados. En el nivel algorítmico se especifica el comportamiento funcional del sistema. El objetivo de este nivel es definir la forma de las funciones de pertenencia, la base de reglas, el mecanismo de inferencia y la estrategia de defuzzificación que mejor se adaptan al problema planteado. Los entornos de desarrollo de sistemas difusos suelen emplear herramientas y lenguajes de descripción específicos para facilitar esta etapa de diseño. Por otra parte, en el nivel de circuitos se debe seleccionar una arquitectura de implementación eficiente, diseñar los bloques básicos del sistema y verificar el comportamiento lógico y temporal del mismo.

Una estrategia común en muchos entornos de desarrollo consiste en utilizar lenguajes de descripción de hardware de alto nivel, como VHDL, para facilitar la realización de esta etapa de diseño [3]. Además de proporcionar herramientas de descripción y modelado similares a las empleadas en el diseño de otros tipos de circuitos [4], el uso de VHDL facilita el acceso directo a muchas herramientas de síntesis automática, lo que supone una ventaja importante a la hora de acelerar el ciclo de desarrollo de nuevos productos [5].

Al aplicar procedimientos de síntesis automática surge siempre una posible disyuntiva. Elegir una arquitectura predefinida para el sistema a sintetizar o dejar que la propia herramienta de síntesis escoja la opción arquitectural más adecuada para minimizar los criterios de síntesis (velocidad o área) definidos por el diseñador al

[1] Trabajo financiado por el proyecto TIC2001-1726 (CICYT).

invocar la herramienta. La primera alternativa presenta la ventaja de proporcionar soluciones eficientes y bien adaptadas a la arquitectura elegida, pero tiene el inconveniente de condicionar al diseñador a emplear, en la etapa de descripción algorítmica del sistema, solo aquellas funcionalidades del lenguaje de especificación que son directamente trasladables a la arquitectura objetivo.

En esta comunicación se explora una estrategia de diseño para sistemas difusos basada en la segunda alternativa. Para ello, utilizando las facilidades que proporciona el lenguaje VHDL para definir nuevas funciones y operadores, se propone una sintaxis que permite expresar la base de conocimiento de un sistema difuso en VHDL con un estilo de descripción muy similar al lenguaje natural. Esta descripción de la estructura del sistema difuso a nivel de comportamiento es independiente de la descripción de los operadores difusos que determinan la realización del proceso de inferencia. El estilo de descripción facilita, por otra parte, el uso de modificadores lingüísticos y operadores relacionales que permiten compactar las bases de reglas e incrementar la interpretabilidad lingüística de las mismas.

En la Sección 2 se introduce el estilo de descripción utilizado y se ilustra con algunos ejemplos. La inclusión de esta etapa de diseño en el flujo de desarrollo del entorno *Xfuzzy* se describe en la Sección 3. En la Sección 4 se presentan algunos resultados de síntesis de sistemas difusos sobre FPGAs siguiendo esta estrategia de diseño y se comparan con los obtenidos empleando descripciones estructurales de los mismos sistemas. Por último la Sección 5 recoge las principales conclusiones del trabajo.

2 MODELADO VHDL DE SISTEMAS DIFUSOS

Uno de los objetivos de este trabajo es generar una herramienta de síntesis automática para la nueva versión del entorno de desarrollo de sistemas difusos *Xfuzzy* [6]. La versión 3 de dicho entorno incluye un lenguaje de especificación formal, XFL3, que permite la utilización de modificadores lingüísticos y operadores relacionales en los antecedentes de las reglas, así como la selección y definición por parte del usuario de las funciones de pertenencia y de los distintos operadores difusos [7]. Las nuevas herramientas de identificación y aprendizaje permiten, asimismo, la utilización de familias de funciones de pertenencia que simplifican las etapas de ajuste de parámetros y síntesis hardware de los sistemas.

Con objeto de mantener la correspondencia con la descripción algorítmica del sistema, el estilo de descripción utilizado a nivel VHDL contempla también la definición por separado de la estructura del sistema (base de reglas y funciones de pertenencia) y de los operadores difusos (conectivos y modificadores lingüísticos y método de defuzzificación) que definen el proceso de inferencia.

2.1. ESTRUCTURA DEL SISTEMA

La Figura 1 muestra un ejemplo de la especificación VHDL del cuerpo de la arquitectura de un sistema difuso con 9 reglas. Cada regla contiene dos partes: el antecedente de la regla y su conclusión o consecuente. El antecedente es una expresión que relaciona las variables de entrada del sistema con sus valores lingüísticos, mientras que el consecuente expresa el valor lingüístico que toma la salida de la regla. Cada una de las reglas puede interpretarse de la siguiente forma:

if x *is* **low** *and* y *is* **low** *then* z *is* **half** (1)

donde x, y, z son las variables de entradas y salida, y *low* y *half* son etiquetas lingüísticas.

La descripción VHDL corresponde a una especificación de alto nivel ya que no se asume ningún criterio de implementación específico, es decir, no se establecen los mecanismos de procesado de las operaciones difusas *is* (=), conectivo *and* e inferencia *then* (rule(,)), sino que tan solo se describe la estructura de la base de reglas que define el comportamiento del sistema.

La Figura 2 muestra un segundo ejemplo en el que las reglas se agrupan mediante el conectivo *or* y se hace uso de los modificadores lingüísticos *mayor que* (>) y *menor que* (<). En este caso se consigue reducir el número de reglas. Así la base de reglas equivalente a la de la Figura 1 contendrá 3 reglas en lugar de las 9 originales.

```
architecture specification of example1 is
. . .
begin
   R(1) <= rule( (x = low)
            and (y = low),    half);
   R(2) <= rule( (x = low)
            and (y = middle), strong);
   R(3) <= rule( (x = low)
            and (y = high),   strong);
   R(4) <= rule( (x = middle)
            and (y = low),    weak);
   R(5) <= rule( (x = middle)
            and (y = middle), half);
   R(6) <= rule( (x = middle)
            and (y = high),   strong);
   R(7) <= rule( (x = high)
            and (y = low),    weak);
   R(8) <= rule( (x = high)
            and (y = middle), weak);
   R(9) <= rule( (x = high)
            and (y = high),   half);

   Z  <= defuzz(R);
end specification;
```

Figura 1: Descripción VHDL de la base de reglas.

```
architecture specification of example2 is
. . .
begin
  R(1) <= rule(((x = low) and (y = low))
          or ((x = middle) and (y = middle))
          or ((x = high) and (y = high)),
               half);
  R(2) <= rule(((x = middle) and (y = low))
          or ((x = high) and (y < high)),
               strong);
  R(3) <= rule(((x = low) and (y > low))
          or ((x = middle) and (y = high)),
               weak);

  Z  <= defuzz(R);

end specification;
```

Figura 2: Base de reglas con modificadores lingüísticos.

La especificación de la base de conocimiento del sistema difuso se completa con la definición de las funciones de pertenencia asociadas a las etiquetas lingüísticas utilizadas en las reglas. Dichas etiquetas representan un rango de valores dentro del universo de discurso de las variables de entradas y salida. Se describen mediante funciones que indican el grado de pertenencia a la etiqueta para un determinado valor de la variable.

La Figura 3 muestra cómo se han definido en VHDL diferentes funciones de pertenencia de tipo triangular y trapezoidal, así como singularidades difusas representadas por valores enteros. Una función de pertenencia de tipo triángulo o trapezoide se caracteriza por su identificador (etiqueta lingüística) y los tres puntos que definen el triángulo o los cuatro puntos para el trapezoide. En el ejemplo de la figura hemos definido tres constantes que corresponderán a las funciones de pertenencia asociadas a las variables de entradas x e y (*low*, *middle* y *high*) y 5 funciones de pertenencia de tipo entero (*very_weak, weak, half, strong* y *very_strong*) correspondientes a la variable de salida.

```
architecture specification of ejemplo1 is
 -- MF for x and y
 constant low: trapezoid := (0, 0, 15, 31);
 constant middle: triangle := (15, 31, 47);
 constant high: trapezoid:= (31, 47, 63, 63);

 -- MF for z
 constant very_weak:   integer := 0;
 constant weak:        integer := 15;
 constant half:        integer := 31;
 constant strong:      integer := 47;
 constant very_strong: integer := 63;
. . .
begin
. . .
end specification;
```

Figura 3: Definición VHDL de las funciones de pertenencia.

```
package xfvhdfunc is

 type triangle is array (1 to 3) of integer
      range 0 to TOP;
 type trapezoid is array (1 to 4) of integer
      range 0 to TOP;

 function "=" (x: integer; y: triangle)
               return integer;
 function ">" (x: integer; y: triangle)
               return integer;
 function ">=" (x: integer; y: triangle)
               return integer;
 function "<" (x: integer; y: triangle)
               return integer;
 function "<=" (x: integer; y: triangle)
               return integer;
 function "and" (x,y: integer)return integer;
 function "or" (x,y: integer) return integer;
 function "not" (x: integer) return integer;
 function rule (x: integer; y: integer)
               return two_integer_vector;
 function defuzz(x: consec)
               return two_integer_vector;
end xfvhdfunc;
```

Figura 4: Paquete VHDL con las definiciones de tipos de datos y funciones.

2.2. OPERADORES DIFUSOS

La definición de tipos de datos y operadores difusos se lleva a cabo en VHDL de una manera independiente a la descripción del sistema. Dichas definiciones pueden encapsularse en paquetes VHDL (Figura 4) que formarán parte de la librería encargada de personalizar la implementación del sistema difuso. Los operadores difusos se definen en la cabecera del paquete VHDL

```
package body xfvhdfunc is

 function "=" (x: integer; y: triangle)
              return integer is
   variable sal: integer := 0;
 begin
   if x <= y.point(1) then sal := 0;
   elsif x > y.point(1) and x < y.point(2)
       then
         sal := y.slope(1)*(x - y.point(1));
   elsif x = y.point(2) then sal := DISCRET;
   elsif x > y.point(2) and x < y.point(3)
       then
         sal := y.slope(2)*(y.point(3) - x);
   else sal := 0; end if;
   return sal;
 end "=";

 function ">" (x: integer; y: triangle)
              return integer is
   variable sal: integer := 0;
 begin
   if x > y.point(2) and x < y.point(3) then
         sal := y.slope(2)*(x - y.point(2));
   elsif x >= y.point(3) then sal := DISCRET;
   else  sal := 0; end if;
   return sal;
 end ">";
```

Figura 5: Cuerpo del paquete VHDL que contiene las definiciones de funciones.

haciendo uso de funciones que en algunos casos sobrecargan a los operadores definidos en el lenguaje VHDL. Por último las definiciones de las funciones que implementan las etiquetas, los conectivos *and*, *or* y *not*, los modificadores lingüísticos y los métodos de inferencia y defuzzificación se incluyen en el cuerpo del paquete VHDL, como se muestra en el ejemplo de la Figura 5.

3 INTEGRACIÓN EN EL FLUJO DE DISEÑO DE *XFUZZY*

La Figura 6 ilustra el flujo de diseño de sistemas difusos mediante el entorno de desarrollo *Xfuzzy*. El punto de comienzo del proceso de diseño es una descripción funcional del sistema en el lenguaje de especificación XFL3. Dicha descripción puede generarse de forma manual, a partir del conocimiento previo del sistema, o de manera automática, utilizando las herramientas de identificación proporcionadas por el entorno. En ambos casos la verificación y el ajuste de dicha especificación pueden llevarse a cabo con ayuda de herramientas de simulación y aprendizaje supervisado.

Una vez que la especificación del sistema ha sido validada puede procederse a realizar la etapa de síntesis. La herramienta de síntesis *xfvhdl* [8] traslada la especificación XFL3 en una descripción VHDL de acuerdo con la estrategia de realización seleccionada (estructural o de comportamiento). Si se selecciona una estrategia de realización de tipo comportamiento *xfvhdl* genera una descripción VHDL similar a la descrita en la Sección 2. Cuando se elige una estrategia de realización basada en una arquitectura específica [1] *xfvhdl* utiliza una librería de celdas que contiene versiones parametrizadas de los diferentes bloques básicos de dicha arquitectura. Tanto el código usado en la descripción de las celdas de la librería como el generado por la herramienta cumplen las restricciones de implementación impuestas por la mayoría de las herramientas de síntesis automática a partir de VHDL.

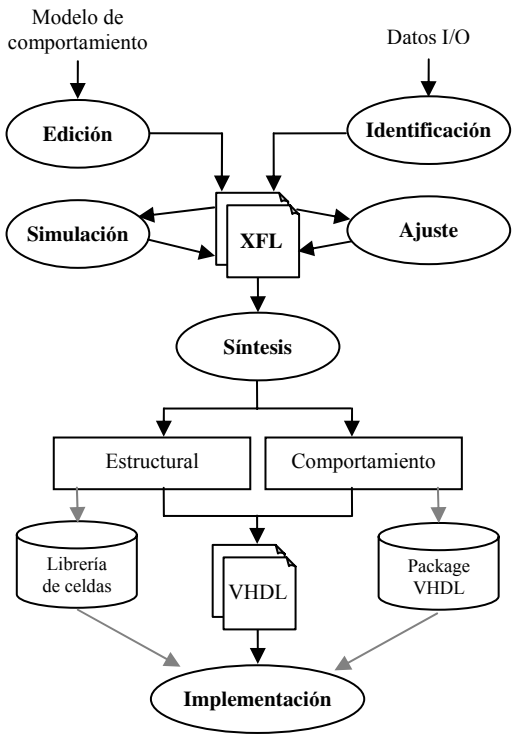


Figura 6: Flujo de diseño de sistemas difusos con *Xfuzzy*.

La validación de las descripciones VHDL puede realizarse mediante simulación con ayuda del simulador *ModelSim* de Mentor Graphics. La posterior etapa de síntesis lógica, realizada con las herramientas *XST* de Xilinx o *FPGA Compiler 2* y *FPGA Express* de Synopsys, genera la descripción del circuito que implementa el sistema difuso. Con objeto de acelerar la realización de estas dos etapas de diseño, *xfvhdl* proporciona dos ficheros adicionales: un fichero de *TestBench* que facilita la simulación del sistema difuso y un fichero de comandos que dirige los procesos de síntesis e implementación con las herramientas de FPGAs de Xilinx.

4 RESULTADOS DE SÍNTESIS

Con objeto de evaluar la viabilidad de la técnica de diseño propuesta se han realizado diferentes implementaciones de sistemas difusos de dos entradas y una salida capaces de aproximar con distintos grados de precisión la función mostrada en la Figura 7, dada por la siguiente expresión:

$$z = \frac{1}{1 + e^{10(x-y)}} \qquad (2)$$

Se han considerado diferentes bases de reglas con 3, 5, 7 y 9 funciones de pertenencia (MFs) para las variables de entrada y 3, 5 y 7 MFs para la variable de salida. Todos los sistemas han sido implementados para precisiones de 5 a 10 bits en las variables y grados de pertenencia.

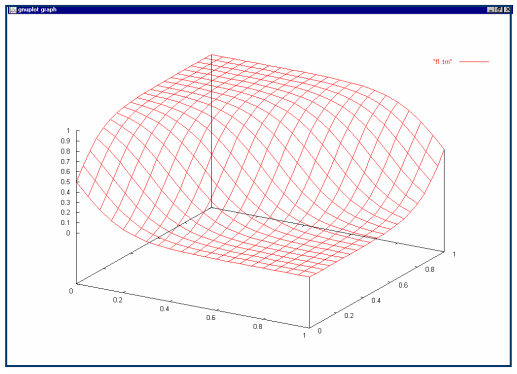


Figura 7: Función correspondiente a la expresión (2).

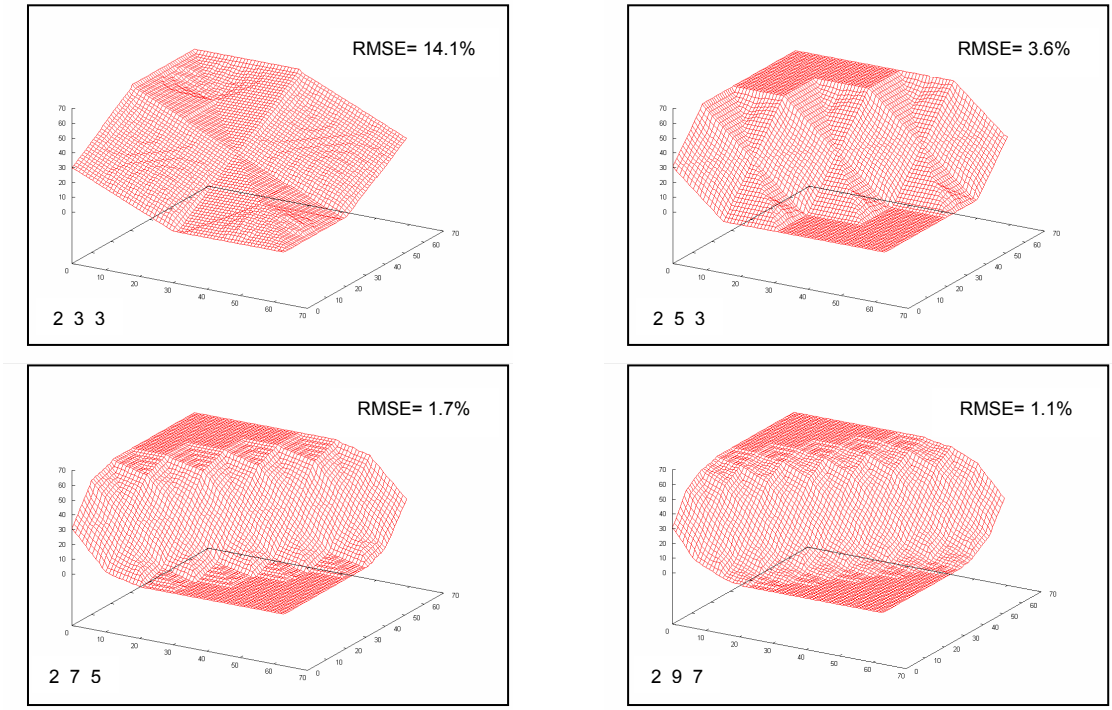


Figura 8: Aproximación de la función (2) utilizando sistemas con diferente número de (entradas_MFs_reglas).

Las superficies de la Figura 8 muestran las funciones aproximadas por las diferentes bases de reglas. Estos resultados permiten comparar realizaciones de sistemas con diferentes grados de complejidad y precisión. Las bases de reglas iniciales fueron obtenidas por dos métodos diferentes con ayuda de las herramientas del entorno *Xfuzzy*. Una primera alternativa consistió en partir de una superficie de control plana y utilizar la herramienta de aprendizaje supervisado que proporciona el entorno para ajustar los parámeros de las funciones de pertenencia de los antecedentes y consecuentes de las reglas. Como un método alternativo se utilizó la herramienta de identificación de sistemas a partir de datos numéricos como paso previo al ajuste de los parámetros de los sistemas. Los resultados proporcionados por ambos procedimientos fueron prácticamente idénticos, obteniéndose bases de reglas de 9, 25, 49 y 81 reglas, respectivamente. A continuación el numero de funciones de pertenencia de salida fue reducido empleando las técnicas de clustering proporcionadas por las herramientas de simplificación de *Xfuzzy*. Posteriormente las bases de reglas fueron compactadas usando los conectivos y modificadores lingüísticos disponibles en el lenguaje XFL3. Finalmente todos los sistemas fueron implementados en FPGAS de la familia de dispositivos Spartan 2E de Xilinx.

En la Figura 9 se muestran los resultados de implementación más significativos. Comparando los

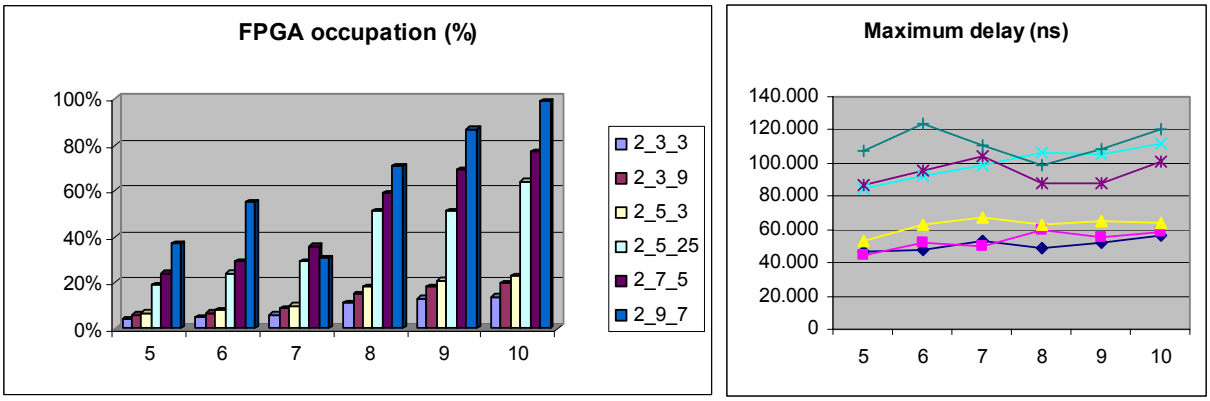


Figura 9: Ocupación de la FPGA y retrasos máximos para diferentes bases de reglas y bits de precisión.

resultados obtenidos se observa una reducción importante, tanto en términos de recursos de hardware como en los tiempos máximos de procesado, cuando el número de reglas es compactado mediante el uso de conectivos y modificadores lingüísticos. Esto es debido a que en este último caso se simplifica la etapa de agregación de reglas que es implementada mediante la suma en paralelo de las salidas de la reglas por medio de un bloque combinacional.

Como consecuencia, el empleo de bases de reglas simplificadas proporciona mejores resultados de implementación en términos de área y velocidad, ya que este tipo de sistemas poseen bases de reglas más compactas que facilitan la optimización de las etapas de salida de los circuitos (agregación de reglas y defuzzificación). Puede concluirse, por tanto, que la reducción del tamaño de la base de reglas debe ser uno de los principales criterios de diseño de sistemas difusos.

Por último, en la Figura 10 se comparan los resultados obtenidos mediante esta estrategia de implementación con los correspondientes a una realización basada en la arquitectura específica para sistemas difusos descrita en [1]. Concretamente, utilizando circuitos aritméticos para generar las funciones de pertenencia y almacenando en memoria ROM la base de conocimiento del sistema. Como puede observarse en la figura, para el caso particular usado a los largo del texto el uso de la arquitectura específica resulta ventajoso sólo cuando la precisión utilizada es superior a 8 bits. Un análisis similar puede permitir en cada caso establecer un compromiso adecuado entre coste y precisión para distintos casos particulares.

5 CONCLUSIONES

El empleo de una descripción a alto nivel de un sistema difuso permite que el diseñador se centre en la estructura del sistema y sus especificaciones de comportamiento en lugar de en los detalles de implementación de las funciones de procesado.

La utilización de un lenguaje de descripción de hardware en combinación con una librería de funciones predifinidas presenta la ventaja de proporcionar un procedimiento directo para la obtención del circuito difuso mediante herramientas de síntesis automática.

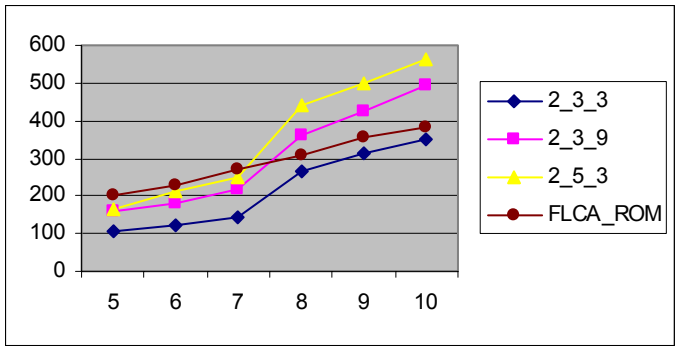


Figura 10: Comparación de los resultados de síntesis obtenidos mediante estrategias estructurales y de comportamiento.

Referencias

[1] I. Baturone, A. Barriga, S. Sánchez-Solano, C. J. Jiménez, D. R. López, Microelectronic Design of Fuzzy Logic-Based Systems, CRC Press, 2000.

[2] A. Costa, A. De Gloria, P. Faraboschi, A. Pagni, G. Rizzotto, Hardware Solutions fot Fuzzy Control, *Proceedings of the IEEE*, vol. 83, n. 3, pp. 422-434, 1995.

[3] A. Zamfirescu, C. Ussery, VHDL and Fuzzy Logic If-Then Rules, Proc. EURO-VHDL´92, pp. 636-641, Hamburg 1992.

[4] D. Galán, C. J. Jimenez, A. Barriga, S. Sánchez-Solano, VHDL Package for Description of Fuzzy Logic Controllers, Proc. EURO-VHDL'95 pp. 528-533, Brighton, 1995.

[5] T. Hollstein, S. K. Halgamuge, M. Glesner, Computer-Aided Design of Fuzzy Systems Based on Generic VHDL Specifications, *IEEE Trans. on Fuzzy Systems*, vol. 4, no. 4, Nov. 1996.

[6] Xfuzzy home page: http://www.imse.cnm.es/Xfuzzy

[7] F. J. Moreno-Velo, S. Sánchez Solano, A. Barriga, I. Baturone, D. R. López, XFL3: A new fuzzy system specification language. Proc. CSCC'2001, pp. 361-366, Creta, 2001

[8] E. Lago, C. J. Jiménez, D. R. López, S. Sánchez-Solano, A. Barriga, Xfvhdl: A Tool for the Synthesis of Fuzzy Logic Controllers, Proc. DATE'98, pp. 102-107, Paris, 1998.

UNA ALTERNATIVA AL ÍNDICE DE LIOU Y WANG PARA LA ORDENACIÓN DE NÚMEROS DIFUSOS

M. Socorro García
Depto de Electrónica, Tecnología
de Computadoras y Proyectos.
E.T.S de Ingeniería Industrial.
Universidad de Politécnica de Cartagena. Murcia, Spain
socorro.garcia@upct.es

M. Teresa Lamata
Depto de Ciencias de la Computación
e Inteligencia Artificial.
E.T.S de Ingeniería Informática.
Universidad de Granada.18071.Granada, Spain
mtl@decsai.ugr.es

Resumen

El presente trabajo muestra una alternativa al índice de Liou y Wang para la ordenación de números difusos. Este método de ordenación, presenta la inclusión del índice de optimismo del decisor, el cual resulta muy interesante pero por si solo no es capaz de discriminar en todos los casos, con lo que proponemos un método de ordenación que incluya no solo el índice de optimismo apuntado por Liou y Wang, sino que contenga también un índice de modalidad, que represente la centralidad del decisor.

Palabras Clave: ordenación de números difusos, índice de optimismo, índice de modalidad.

1. INTRODUCCIÓN

Los números difusos no siempre muestran un conjunto totalmente ordenado como lo hacen los números reales. En aplicaciones de problemas de decisión multiatributo, cuando las valoraciones finales son difusas, es muy difícil distinguir la mejor alternativa posible y la ordenación de estas.

Para resolver el problema de ordenación de las utilidades difusas muchos investigadores han propuesto métodos de ordenación difusa que pueden utilizarse para comparar números difusos.

Bortolan y Degani [3], fueron los primeros que propusieron no solo una revisión sistemática de las investigaciones en ordenación difusa sino que también presentaron resultados de la comparación entre varios métodos de ordenación, comprobando la disparidad de resultados entre ellos.

Entre los métodos estudiados por Bortolan y Degani destacan, tres de lo índices propuestos por Yager [19,20,21], el índice de preferencia-α de Adamo [1], el índice de Baas y Kwakernaak [2], el método propuesto por Kerre[13], el índice de Jain [11,12] así como los grados de dominación propuestos por Dubois y Prade [9]

Chen y Hwang [6], fueron los primeros en presentar una clasificación de una veintena de métodos existentes, agrupándolos en grandes clases de acuerdo al medio que cada método utiliza: métodos de relaciones de preferencia, métodos de media y amplitud difusa, métodos de puntuación difusa y métodos lingüísticos. Cada una de estas clases principales está subdividida de acuerdo a la técnica utilizada en cada caso.

De la clasificación presentada por Chen y Hwang, entre los que se encuentran los ya apuntados por Bortolan y Degani, destacan las aproximaciones de Buckley y Chanas [4], Mabuchi [16], Delgado [8], Lee y Li [14], etc

Asimismo, Wang y Kerre [17,18], presentaron una treintena de índices de ordenación organizados en varias líneas para investigarlos más eficientemente, clasificándolos en tres categorías.

En la primera clase, cada índice esta asociado con la representación F del conjunto de cantidades difusas a la recta real $\Re$ para transformar las cantidades difusas en números reales. Las cantidades difusas se comparan entonces de acuerdo a los correspondientes números reales. Dentro de esta clasificación cabe destacar la inclusión de la aproximación de Choobineh y Li [7], Fortemps y Roubens [10], L. Campos y A. Gonzalez [5] y Liou y Wang [15]. Esta última es objeto de estudio en este artículo.

En la segunda clase, se determinan unos conjuntos de referencia y todas las cantidades difusas a ordenar se comparan con los conjuntos de referencia.

En la última clase, se construye una relación de referencia para hacer las comparaciones binarias entre las cantidades difusas involucradas. Estas comparaciones binarias sirven como base para obtener la ordenación final.

Una vez clasificados los índices, propusieron algunos axiomas que sirven como propiedades razonables para establecer la racionalidad de un proceso de ordenación.

Liou y Wang propusieron un método de ordenación de números difusos con valor integral. El método que es independiente del tipo de funciones de pertenencia utilizadas y de la normalidad o no de las funciones, puede ordenar más de dos número difusos simultáneamente. Es relativamente simple en computación, especialmente en la ordenación de números difusos triangulares y trapezoidales. Además incluye la utilización de un índice de optimismo para reflejar la aptitud ante el riesgo por parte del decisor.

Este planteamiento resulta interesante a la hora de ordenar números difusos, pero como podremos comprobar, el índice de optimismo por si solo no es lo suficientemente discriminatorio en todos los casos, especialmente en los que no están muy claramente definidos. Por ello, nosotros proponemos la extensión de este índice introduciendo un nuevo concepto que es el grado de modalidad del decisor.

El artículo se desarrolla de la siguiente forma. En el apartado 2 introducimos de forma breve y concisa los números difusos y sus propiedades para poder definir el índice de ordenación que es el motivo de la propuesta. Finalizamos con un estudio relativo a tres números difusos que son indiferentes si aplicamos otros índices como el de Yager, el de Liou y Wang u otros, y que si tienen un claro orden con nuestro método.

2. FUNCIONES DE PERTENENCIA DE NUMEROS DIFUSOS

El concepto de número difuso se define como sigue:

Definición 1: Un número real difuso A se describe como un subconjunto difuso de la recta por una función de pertenencia f_A que tiene las siguientes propiedades.

1. f_A es una aplicación continua de R en el intervalo cerrado [0,1]
2. f_A(x)=0, para todo x $\in$(-∞,a].
3. f_A es estrictamente creciente en [a,b]
4. f_A(x) =1, para todo x$\in$[b,c]
5. f_A es estrictamente decreciente en [c,d]
6. f_A(x)=0, para todo x$\in$[d, ∞)

donde a,b,c y d son números reales. El número difuso [a,b,c,d;1] se representará por los cuatro números de la definición y por el uno que indica la propiedad de normalidad.

A menos que especifiquemos otra cosa asumiremos que A es convexo, normal y acotado, (es decir -∞<a, d<∞.)

Definición 2: La función de pertenencia podrá ser expresada también como:

$$f_A(x) = \begin{cases} f_A^L(x) & (a \le x \le b) \\ 1 & (b \le x \le c) \\ f_A^R(x) & (c \le x \le d) \\ 0 & otherwise \end{cases} \qquad (1)$$

donde $f_A^L(x):[a,b]\to[0,1]$ y $f_A^R(x):[c,d]\to[0,1]$.

A partir de la Definición 2, es obvio que $f_A^L(x)$ es la función de pertenencia izquierda del número difuso A, es continua y estrictamente creciente en [*a,b*], y $f_A^R(x)$ es la función de pertenencia derecha del número difuso A, es continua y estrictamente decreciente en [*c,d*].

En este articulo, usaremos números difusos triangulares, aunque su definición se puede extender a cualquier otro número difuso con solo tener en cuenta la Definición 4. El número difuso A será triangular si su función de pertenencia f_A está dada por:

$$f_A(x) = \begin{cases} \dfrac{(x-a)}{(b-a)} & (a \le x \le b) \\ \dfrac{(x-c)}{(b-c)} & (b \le x \le c) \\ 0 & otherwise \end{cases} \qquad (2)$$

donde a,b y c son números reales. En este caso el número difuso se expresa con la tripleta [a,b,c;1] y el mismo significado que para los números difusos trapezoidales.

3. ORDENACIÓN DE NÚMEROS DIFUSOS EN FUNCIÓN DE TRES ÁREAS

Definición 3: Sea A un número difuso definido mediante las funciones continuas $f_A^L(x)$ y $f_A^R(x)$ en [0,1], con funciones inversas asociadas $f_A^{L^{-1}}(x)$ y $f_A^{R^{-1}}(x)$ respectivamente, funciones que también son continuas e integrables en [0,1], entonces definimos el área engendrada por $f_A^L(x)$ en $[-\infty, b]$ como:

$$S_I(A_i) = a + \int_a^b f_A^L(x)\,dx = \int_0^1 f_A^{L^{-1}}(y)\,dy \quad (3)$$

esta área estaría relacionada con la parte izquierda del número difuso.

De igual forma definimos el área engendrada por $f_A^R(x)$ en $[-\infty, d]$ como:

$$S_D(A_i) = b + \int_b^c f_A^R(x)\,dx = \int_0^1 f_A^{R^{-1}}(y)\,dy \quad (4)$$

esta área estaría relacionada con la parte derecha del número difuso

Definición 4: Dado un número difuso A, definimos el área hasta la altura del número difuso como:

$$S_H(A_i) = b \qquad (5)$$

esta área estaría relacionada con la moda del número difuso.

Estas tres áreas estan reflejadas en la figura 1.

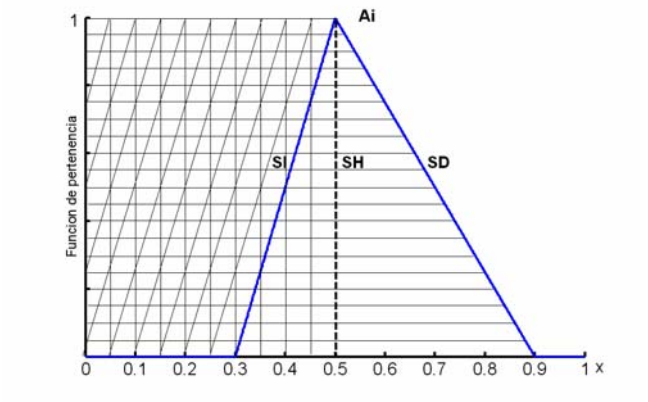


Figura. 1: Representación de las áreas S_I, S_D, S_H

Definición 5: Consideramos que $S = \{A_1, A_2, ..., A_n\}$ es un conjunto de números difusos convexos, que la función de ordenación R es una aplicación de S a los números reales $\Re$. Para cualquiera A_i, $A_j \in$ S, la función de ordenación tiene las siguientes propiedades:

i. Si $R(A_i) < R(A_j)$, entonces $A_i < A_j$

ii. Si $R(A_i) = R(A_j)$, entonces $A_i = A_j$

iii. Si $R(A_i) > R(A_j)$, entonces $A_i > A_j$

Liou y Wang definieron el siguiente índice de ordenación [15,17] :

$$LW_i^{\lambda}(A_i) = \lambda S_D(A_i) + (1-\lambda) S_I(A_i) \quad (6)$$

siendo $\lambda \in [0,1]$ el índice que refleja el grado de optimismo del decisor. Cuanto mayor es λ, el grado de optimismo es mayor. Más específicamente, cuando λ=0 representa un decisor pesimista, si λ=1 el decisor es optimista y si λ=0,5 el decisor es moderado.

Este índice tiene bastantes problemas de discriminación en casos no muy claramente definidos, no siendo suficiente la inclusión del índice de optimismo a la hora de la ordenación.

Dados por ejemplo los números A_1, A_2 y A_3 de la figura 2, que se presenta posteriormente, el índice de Liou y Wang no es capaz de discriminar entre ellos, para cualquiera que sea el índice de optimismo considerado, como se refleja en la tabla 1.

Debido a esta limitación del índice de Liou y Wang presentamos un índice que mejore la discriminación, manteniendo el índice de optimismo e incluyendo asimismo un nuevo índice que llamaremos índice de modalidad $\alpha \in [0,1]$ y que representa cuanto más modal es el decisor. Es decir, α nos da la ponderación del valor central y (1-α) la de los extremos. Así si el número difuso se tratase como 'crisp' α sería igual a 1. Mientras que el parámetro λ pondera el extremo superior del número difuso con respecto al inferior.

De esta manera, definimos un índice, función de las tres áreas anteriormente descritas $I(A_i) = f(S_I, S_D, S_H)$, como sigue:

Definición 6: Si A es un número difuso con función de pertenencia f_A, como se define en (1), entonces el

índice de ordenación en función de las áreas, se define como:

$$I(A_i) = (1-\alpha)\cdot\left[LW_i^{\lambda}(A_i)\right] + \alpha \cdot S_H(A_i)$$
$$I(A_i) = (1-\alpha)\cdot\left[\lambda \cdot S_D(A_i) + (1-\lambda)\cdot S_I(A_i)\right] + \alpha \cdot S_H(A_i) \quad (7)$$

donde $S_D(A_i)$ es el área derecha del número difuso, $S_I(A_i)$ el área izquierda del número difuso y $S_H(A_i)$ el área hasta la altura del número difuso, α es el grado de modalidad del decisor α ∈ [0,1] y λ el grado de optimismo del decisor λ ∈ [0,1].

Donde α significa el grado de importancia dado al valor modal con respecto a los extremos del número difuso, mientras que λ indica el optimismo/pesimismo del decisor al dar más o menos peso al área mayor o menor, respectivamente

Como se puede comprobar en la definición 6 el índice de Liou y Wang $LW_i^{\lambda}(A_i)$, sería un caso particular del índice de ordenación presentado cuando $\alpha = 0$.

3.1. CASOS PARTICULARES

- Si $\lambda = 0$ (pesimista):

$$I(A_i) = (1-\alpha)\cdot S_I(A_i) + \alpha \cdot S_H(A_i)$$

1. Si $\alpha = 0$; $I = S_I(A_i)$ (totalmente pesimista)
2. Si $\alpha = 0.5$; $I = 0.5\cdot S_I(A_i) + 0.5\cdot S_H(A_i)$
3. Si $\alpha = 1$; $I = S_H(A_i)$ (modal)

- Si $\lambda = 1$ (optimista):

$$I(A_i) = (1-\alpha)\cdot S_D(A_i) + \alpha \cdot S_H(A_i)$$

4. Si $\alpha = 0$; $I = S_D(A_i)$ (totalmente optimista)
5. Si $\alpha = 0.5$; $I = 0.5\cdot S_D(A_i) + 0.5\cdot S_H(A_i)$
6. Si $\alpha = 1$; $I = S_H(A_i)$ (modal)

- Si $\lambda = 0.5$ (moderado):

$$I(A_i) = (1-\alpha)\left[0.5S_D(A_i) + 0.5S_I(A_i)\right] + \alpha S_H(A_i)$$

7. Si $\alpha = 0$:
   $I(A_i) = 0.5\cdot S_D(A_i) + 0.5\cdot S_I(A_i)$ representa el índice de Yager
8. Si $\alpha = 0.5$:
   $I(A_i) = 0.25\cdot S_D(A_i) + 0.25\cdot S_I(A_i) + 0.5\cdot S_H(A_i)$
9. Si $\alpha = 1$; $I = S_H(A_i)$ (modal)

- Si $\lambda = 0.5$ y $\alpha = 1/3$:

$$I_{1/3}(A_i) = 1/3\, S_D(A_i) + 1/3\, S_H(A_i) + 1/3\, S_I(A_i)$$

que sería el caso en el que las tres áreas tienen el mismo peso a la hora de la evaluación por parte del decisor y correspondería al decisor neutral.

4. EJEMPLO COMPARATIVO

Dados los números difusos A_1, A_2 y A_3, "alrededor de 0.6", "algo menos de 0.7" y "algo más de 0.5" respectivamente, con la representación de la figura 2:

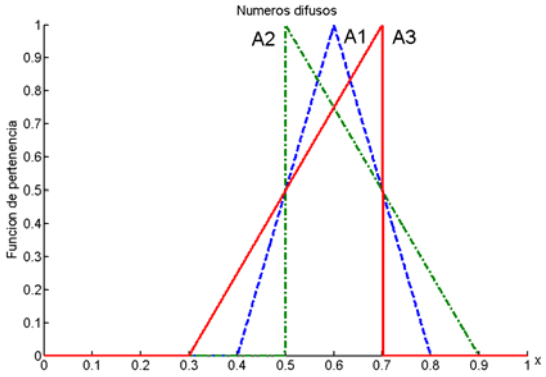


Figura. 2: Ejemplo comparativo

Como se puede apreciar según el índice Liou y Wang para cualquier grado de optimismo no discrimina entre los tres números difusos propuestos, como se puede comprobar en la tabla 1.

Vemos como ejemplo particular del apartado $\forall \alpha \neq 0$, el caso del decisor neutral para el que $I_{1/3}(A_1)$=1.8/3, $I_{1/3}(A_2)$=1.7/3, $I_{1/3}(A_3)$=1.9/3 luego para estos valores de los pesos comprobamos que:

$$A_3 \succ A_1 \succ A_2$$

Con el índice propuesto tenemos $A_3 \succ A_1 \succ A_2$ en cualquier caso a excepción del caso en que $\alpha = 0$ (que es corresponde con el propio índice de Liou y Wang). Con lo que se puede apreciar que el índice propuesto discrimina como mínimo igual que Liou y Wang y mejora los casos en que este no lo hace.

A_i		S_I	S_H	S_D	$LW_i^{\lambda}(A_i)$		$I(A_i)$	
A_1	(0.4,0.6,0.8)	0.50	0.6	0.70	$\lambda = 0$	0.50	$\forall \alpha \neq 0$	$(0.5, 0.6]$
					$\lambda = 0.5$	0.60		0.6
					$\lambda = 1$	0.70		$[0.6, 0.7)$
A_2	(0.5,0.5,0.9)	0.50	0.50	0.70	$\lambda = 0$	0.50		0.5
					$\lambda = 0.5$	0.60		$[0.5, 0.6)$
					$\lambda = 1$	0.70		$[0.5, 0.7)$
A_3	(0.3,0.7,0.7)	0.50	0.70	0.70	$\lambda = 0$	0.50		$(0.5, 0.7]$
					$\lambda = 0.5$	0.60		$(0.6, 0.7]$
					$\lambda = 1$	0.70		0.7

Tabla 1: Comparación entre el índice de Liou y Wang y el índice propuesto

5. CONCLUSIONES

El método propuesto es sencillo de aplicar al estar definido como la suma de tres áreas, áreas que están asociadas a tres funciones continuas y acotadas en [0,1]

La aproximación planteada presenta todas las ventajas, que ya presentaba el método de ordenación de Liou y Wang, mejorando este en algunos de los casos en que el índice de Liou y Wang no es suficientemente discriminatorio.

Esta propuesta produce resultados satisfactorios, no solamente en problemas bien definidos, como ya lo hacia también Liou y Wang, sino que da solución a algunos de los casos que este método no podía discriminar, asegurándose como mínimo la discriminación planteada por el índice de Liou y Wang.

Asimismo, este método incluye la posibilidad de ordenar *n* números difusos a la vez, problema que presentan algunos métodos de ordenación, donde funcionan bien para dos números difusos, pero que no lo hacen tanto, con más de dos, dándose en algunos casos, ciertas incongruencias.

Finalmente, esta metodología no requiere un esfuerzo computacional muy grande para problemas de gran escala. Siendo en caso de utilización de números difusos triangulares muy simple y rápida. Lo que asegura su implementabilidad en problemas prácticos.

Agradecimientos. Este artículo se ha elaborado bajo el proyecto TIC2002-03411 y TIC2002-11942-E financiado por la DGICYT.

Referencias

[1] J.M. Adamo, Fuzzy decision trees, Fuzzy Sets and Systems 4 (1980) 207-219

[2] S.M. Baas and H. Kwakernaak, Rating and ranking of multiple-aspect alternatives using fuzzy sets. Automatica 13 (1977) 47-58

[3] G. Bortolan and R. Degani, A review of some methods for ranking fuzzy subsets, Fuzzy Sets and Systems 15 (1985) 1-19

[4] J.J. Buckley and S. Chanas, A fast method of ranking alternatives using fuzzy numbers, Fuzzy Sets and Systems, 30 (1989) 337-339

[5] L. Campos, A. Gonzalez, Further contributions to the study of the average value for ranking fuzzy numbers, International Journal of Approximate Reasoning (1994), 10: 135-153

[6] S.J. Chen, C.L. Hwang, Fuzzy Multiple Attribute Decision Making: Methods and Applications, Springer-Verlang Berlin (1992)

[7] F. Choobineh, H. Li, An index for ordering fuzzy numbers, Fuzzy Sets and Systems 54 (1993) 287-294

[8] M. Delgado, J.L. Verdegay and M.A. Vila, A procedure for ranking fuzzy numbers using fuzzy relation, Fuzzy Sets and Systems, 26 (1988)

[9] D. Dubois and H. Prade, Ranking of fuzzy numbers in the setting of possibility theory, Inform. Sci 30 (1983) 183-224

[10] P. Fortemps, M. Roubens, Ranking and defuzzification methods based on areas compensation, Fuzzy Sets and Systems 82 (1996) 319-330

[11] R. Jain, Decision-making in the presence of fuzzy variables, IEEE Trans. Systems Man Cybernet. 6 (1976) 698-703

[12] R. Jain, A procedure of multiple-aspect decision making using fuzzy sets. Internat. J. Systems Sci. 8 (1977) 1-7

[13] E.E. Kerre, The use of fuzzy set theory in electrocardiological diagnostics, in : M.M. Gupta and E. Sanchez, Eds., Approximate Reasoning in Decision Analysis (North-Holland, Amsterdam, 1982) 277-282

[14] E. S. Lee and R, L, Li, Comparison of fuzzy numbers based on the probability measure of fuzzy events, Computer and Mathematics Applications, 15 (1988) 887-896

[15] T-S. Liou, M-J. Wang, Ranking fuzzy numbers with integral value. Fuzzy Sets and Systems 50 (1992) 247-255

[16] S. Mabuchi, An approach to the comparison of fuzzy subsets with an α-cut dependent index, IEEE Trans. On Systems , Man, And Cybernetics, SCM-18 (1988) 264-272

[17] X. Wang, E.E. Kerre, Reasonable properties for the ordering of fuzzy quantities (I). Fuzzy Sets and Systems 118 (2001) 375-385

[18] X. Wang, E.E. Kerre, Reasonable properties for the ordering of fuzzy quantities (II), Fuzzy Sets and Systems 118 (2001) 387-405

[19] R.R. Yager, Ranking fuzzy subsets over the unit interval, Proc 1978 CDC (1978) 1435-1437

[20] R.R. Yager, On choosing between fuzzy subsets, Kybernetes 9 (1980) 151-154

[21] R.R. Yager, A procedure for ordering fuzzy subsets of the unit interval, Inform Sci. 24 (1981) 143-161

Generation of strict preference relations from fuzzy orderings

Bonifacio Llamazares
Dpt. of Applied Economics
University of Valladolid
Avda. Valle de Esgueva 6
E-47011 Valladolid, Spain
boni@eco.uva.es

Bernard De Baets
Dpt. of Applied Mathematics,
Biometrics and Process Control
University of Ghent
Coupure links 653
B-9000 Ghent, Belgium
Bernard.DeBaets@UGent.be

Abstract

In this paper we obtain and characterize some fuzzy strict preference relations from fuzzy partial orders. The definition of these orders is based on an equivalence relation and, in the paper, we consider that the indifference relation plays this role. Moreover, we analyze the transitivity of the fuzzy strict preference relations obtained.

Keywords: Fuzzy orderings, strict preference relations, indifference relations.

1 Introduction

In the classical theory of preference modelling, it is usual to represent the agents' preferences on a set of alternatives A by means of a reflexive relation R, called weak preference relation, where aRb is interpreted as "a is at least as good as b". From this relation, it is possible to obtain a strict preference relation P (the asymmetric part of R) and an indifference relation I (the symmetric part of R). The relation P is used to describe strict preference between the alternatives, i.e. aPb means that "a is better than b". The relation I is used to show the absence of preference, i.e. aIb means that "a and b are indifferent". These relations can be expressed as $P = R \cap (R^{-1})^c$ and $I = R \cap R^{-1}$.

When the relation R is a partial order then I is the minimal reflexive relation, $I = \{(a,a) \mid a \in A\}$, and $P = R \setminus I$. Furthermore P is transitive and I is an equivalence relation.

In the fuzzy framework a fuzzy partial order is also a reflexive, antisymmetric and transitive fuzzy relation. However, there exists different ways for generalizing the crisp properties. In this paper we consider a definition of fuzzy partial order introduced by Höhle and Blanchard [5] and different t-norms and t-conorms in order to obtain and characterize some fuzzy strict preference relations. Moreover, we study the transitivity of the fuzzy strict preference relations obtained.

The paper is organized as follows. In Section 2 we introduce notation and basic definitions, while Section 3 contains the main results of the paper.

2 Preliminaries

Let A be a non-empty set of alternatives with $|A| \geq 2$. Subsets of A will be called *crisp* subsets of A. A *crisp binary relation Q on A* is a crisp subset of $A \times A$. We will use aQb to denote $(a,b) \in Q$. The crisp relation Δ is defined by $\Delta = \{(a,a) \mid a \in A\}$. Given a crisp relation Q on A, the relations Q^{-1} and Q^c are defined by

$$\begin{aligned} Q^{-1} &= \{(a,b) \in A \times A \mid (b,a) \in Q\}, \\ Q^c &= \{(a,b) \in A \times A \mid (a,b) \notin Q\}. \end{aligned}$$

A crisp relation Q on A is:

1. *reflexive* if $\Delta \subseteq Q$.
2. *symmetric* if $Q \subseteq Q^{-1}$.
3. *asymmetric* if $Q \cap Q^{-1} = \emptyset$.
4. *antisymmetric* if $Q \cap Q^{-1} \subseteq \Delta$.
5. *transitive* if (aQb and bQc) $\Rightarrow$ aQc for all $a,b,c \in A$.
6. *complete* if $Q \cup Q^{-1} = A \times A$.

A crisp relation is an *equivalence relation* if it reflexive, symmetric and transitive. On the other hand, a crisp relation is a *partial order* if it is reflexive, antisymmetric and transitive.

A crisp relation R on A is a *crisp weak preference relation* if it is reflexive. Any crisp weak preference relation can be factorized into a *crisp strict preference*

relation, $P = R \cap (R^{-1})^c$, and a *crisp indifference relation*, $I = R \cap R^{-1}$. Moreover, the crisp relations P and I are characterized through the following properties: P is asymmetric, I is symmetric, $P \cap I = \emptyset$ and $R = P \cup I$.

A *fuzzy subset* B *of* A is defined through its *membership function*, $\mu_B : A \longrightarrow [0,1]$, where $\mu_B(a)$ is the grade of membership of a in B. Given two fuzzy subsets B, C of A, $B \subseteq C$ if $\mu_B(a) \leq \mu_C(a)$ for all $a \in A$.

The intersection and the union of fuzzy sets are defined by means of triangular norms and conorms, respectively. A function $T : [0,1]^2 \longrightarrow [0,1]$ is a *triangular norm* (*t-norm*) if it is increasing, commutative, associative and satisfies $T(1,x) = x$ for all $x \in [0,1]$. A function $S : [0,1]^2 \longrightarrow [0,1]$ is a *triangular conorm* (*t-conorm*) if it is increasing, commutative, associative and satisfies $S(0,x) = x$ for all $x \in [0,1]$. It is easy to check that $T(x,0) = 0$ and $S(x,1) = 1$ for all $x \in [0,1]$.

Given a t-norm T and a t-conorm S, the intersection and the union of two fuzzy subsets B, C of A are defined as follows:

1. $\mu_{(B \cap_T C)}(a) = T(\mu_B(a), \mu_C(a))$ for all $a \in A$.
2. $\mu_{(B \cup_S C)}(a) = S(\mu_B(a), \mu_C(a))$ for all $a \in A$.

Next we show some t-norms and t-conorms which are very usual in the literature:

1. $T_{\mathbf{M}}(x,y) = \min(x,y)$.
   $S_{\mathbf{M}}(x,y) = \max(x,y)$.
2. $T_{\mathbf{P}}(x,y) = xy$.
   $S_{\mathbf{P}}(x,y) = x + y - xy$.
3. $T_{\mathbf{L}}(x,y) = \max(x + y - 1, 0)$.
   $S_{\mathbf{L}}(x,y) = \min(x + y, 1)$.

A t-norm T has *zero divisors* if there exist $x, y \in]0,1[$ such that $T(x,y) = 0$. In the previous examples, only $T_{\mathbf{L}}$ has zero divisors.

A *fuzzy binary relation* Q *on* A is a fuzzy subset of $A \times A$. The value $\mu_Q(a,b)$ will be denoted by $Q(a,b)$. If $Q(a,b) \in \{0,1\}$ for all $a, b \in A$ then Q is a crisp relation. On the other hand, given a fuzzy relation Q on A, we will use the following notation:

$$\begin{aligned} M_Q(a,b) &= S_{\mathbf{M}}(Q(a,b), Q(b,a)), \\ m_Q(a,b) &= T_{\mathbf{M}}(Q(a,b), Q(b,a)). \end{aligned}$$

Given a fuzzy relation Q on A, the relations Q^{-1} and Q^c are defined by $Q^{-1}(a,b) = Q(b,a)$ and $Q^c(a,b) = 1 - Q(a,b)$, for all $a, b \in A$.

Given a t-norm T and a t-conorm S, a fuzzy relation Q on A is:

1. *reflexive* if $Q(a,a) = 1$ for all $a \in A$.
2. *symmetric* if $Q(a,b) = Q(b,a)$ for all $a, b \in A$.
3. T*-asymmetric* if $T(Q(a,b), Q(b,a)) = 0$ for all $a, b \in A$.
4. T*-antisymmetric* if $T(Q(a,b), Q(b,a)) = 0$ for all $a, b \in A, a \neq b$.
5. T*-transitive* if $T(Q(a,b), Q(b,c)) \leq Q(a,c)$ for all $a, b, c \in A$.
6. S*-complete* if $S(Q(a,b), Q(b,a)) = 1$ for all $a, b \in A$.

A fuzzy relation is a T*-equivalence* if it is reflexive, symmetric and T-transitive. On the other hand, a fuzzy relation is a partial order with respect to T, for brevity T*-ordering*, if it is reflexive, T-antisymmetric and T-transitive.

However, some authors such as Höhle and Blanchard [5] and Bodenhofer [1] have considered the following definition of fuzzy partial order.

Definition 1 *A fuzzy relation R is called a fuzzy partial order with respect to a t-norm T and a T-equivalence I, for brevity T-I-ordering, if it is:*

1. *I-reflexive: $I(a,b) \leq R(a,b)$ for all $a, b \in A$.*
2. *T-I-antisymmetric: $T(R(a,b), R(b,a)) \leq I(a,b)$ for all $a, b \in A$.*
3. *T-transitive.*

T-orderings are particular cases of T-I-orderings and they are obtained when I is the crisp equality ($I = \Delta$), i.e.

$$I(a,b) = \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{if } a \neq b. \end{cases}$$

It is obvious that if I is a T-equivalence, then R is a T-I-ordering if and only if R is T-transitive and for all $a, b \in A$

$$T(R(a,b), R(b,a)) \leq I(a,b) \leq m_R(a,b). \qquad (1)$$

When the t-norm is $T_{\mathbf{L}}$, the previous condition appears usually in the literature on factorizations of fuzzy preferences (see, for instance, Fodor and Roubens [4], De Baets and Fodor [3] and Llamazares [6]). For this reason, in the next section we study how to obtain fuzzy strict preference relations from T-equivalences and T-I-orderings.

3 The results

Firstly, we introduce the concept of a strict preference relation associated with a T-equivalence and a T-I-ordering. Note that we do not consider an incomparability relation although it is possible that the T-I-ordering is not S-complete.

Definition 2 *Let T be a t-norm, S a t-conorm, I a T-equivalence and R a T-I-ordering. A fuzzy relation P is called a strict preference relation associated with (R, I, T, S) if $R = P \cup_S I$, $P \cap_T I = \emptyset$ and $P \cap_T P^{-1} = \emptyset$.*

Next we begin our analysis. The simplest case is when the T-equivalence I is the crisp equality. Note that in this case, from (1) we have $T(R(a,b), R(b,a)) = 0$ for all $a, b \in A$, $a \neq b$.

Theorem 1 *Let T be a t-norm, S a t-conorm, I the crisp equality and R a T-I-ordering. A fuzzy relation P is a strict preference relation associated with (R, I, T, S) if and only if for all $a, b \in A$*

$$P(a,b) = \begin{cases} R(a,b), & \text{if } a \neq b, \\ 0, & \text{if } a = b. \end{cases}$$

Proof

Suppose first that P is a strict preference relation associated with (R, I, T, S). Then, given $a, b \in A$, $a \neq b$,

$$P(a,b) = S(P(a,b), 0) = S(P(a,b), I(a,b)) = R(a,b).$$

Moreover,

$$P(a,a) = T(P(a,a), 1) = T(P(a,a), I(a,a)) = 0.$$

On the other hand, it is easy to check that the converse statement is true. ■

Now we study the transitivity of this relation.

Theorem 2 *Let T be a t-norm, S a t-conorm, I the crisp equality and R a T-I-ordering. If P is a strict preference relation associated with (R, I, T, S) then P is T-transitive.*

Proof

Given $a, b, c \in A$ we distinguish three cases:

1. If a, b, c are different, then

   $$T(P(a,b), P(b,c)) \leq P(a,c)$$

   because R is T-transitive.

2. If $a = b$ or $b = c$ then,

   $$T(P(a,b), P(b,c)) = 0 \leq P(a,c).$$

3. If $a = c$ and $a \neq b$, then

   $$\begin{aligned} T(P(a,b), P(b,c)) &= T(R(a,b), R(b,a)) \\ &= 0 = P(a,c). \end{aligned}$$ ■

When we consider t-norms without zero divisors we obtain a strict preference relation well known in the literature; it was given by Ovchinnikov [7].

Theorem 3 *Let T be a t-norm without zero divisors, S a t-conorm, I a T-equivalence and R a T-I-ordering. A fuzzy relation P is a strict preference relation associated with (R, I, T, S) if and only if for all $a, b \in A$ the relation R satisfies*

$$m_R(a,b) = 0 \ \text{ if } \ m_R(a,b) < M_R(a,b)$$

and I and P are defined by

$$I(a,b) = \begin{cases} R(a,b), & \text{if } R(a,b) = R(b,a), \\ 0, & \text{if } R(a,b) \neq R(b,a), \end{cases}$$

$$P(a,b) = \begin{cases} R(a,b), & \text{if } R(a,b) \neq R(b,a), \\ 0, & \text{if } R(a,b) = R(b,a). \end{cases}$$

Proof

Suppose first that P is a strict preference relation associated with (R, I). Given $a, b \in A$, since T has not zero divisors, I is symmetric, $T(P(a,b), I(a,b)) = 0$, $T(P(b,a), I(b,a)) = 0$ and $T(P(a,b), P(b,a)) = 0$, we have that only one of $P(a,b)$, $P(b,a)$ and $I(a,b)$ can be greater than zero. Now we distinguish two cases:

1. If $R(a,b) = R(b,a)$, then

   $$R(a,b) = S(P(a,b), I(a,b)) = S(P(b,a), I(a,b)).$$

   Consequently, the only possibility is $P(a,b) = P(b,a) = 0$ and $I(a,b) = R(a,b)$.

2. If $R(a,b) \neq R(b,a)$ and we suppose that $R(a,b) > R(b,a)$, then

   $$S(P(a,b), I(a,b)) > S(P(b,a), I(a,b)).$$

   Thus, the only possibility is $I(a,b) = P(b,a) = R(b,a) = 0$ and $P(a,b) = R(a,b)$. Analogously, if we suppose $R(b,a) > R(a,b)$, we obtain $I(a,b) = P(a,b) = R(a,b) = 0$ and $P(b,a) = R(b,a)$.

On the other hand, it is easy to check that the converse statement is true. ■

Note that the relation P can also be written as

$$P(a,b) = \begin{cases} R(a,b), & \text{if } R(a,b) > R(b,a), \\ 0, & \text{otherwise.} \end{cases}$$

Next we study the transitivity of this relation.

Theorem 4 *Let T be a t-norm without zero divisors, S a t-conorm, I a T-equivalence and R a T-I-ordering. If P is a strict preference relation associated with (R, I, T, S) then P is T-transitive.*

Proof

We consider that the relation P is given by

$$P(a,b) = \begin{cases} R(a,b), & \text{if } R(a,b) > R(b,a), \\ 0, & \text{otherwise.} \end{cases}$$

Since $T(x,0) = T(0,x) = 0$ for all $x \in [0,1]$ and R is T-transitive, given $a, b, c \in A$ we only need to prove that $T(P(a,b), P(b,c)) \leq P(a,c)$ when $P(a,b) = R(a,b) > 0$, $P(b,c) = R(b,c) > 0$ and $P(a,c) = 0$. In this case, it is also satisfied $R(b,a) = R(c,b) = 0$ and $R(a,c) \leq R(c,a)$.

As R is T-transitive, then

$$T(R(b,c), R(c,a)) \leq R(b,a) = 0.$$

Because T has not zero divisors and $R(b,c) \neq 0$, we have $R(c,a) = 0$. Again, by the transitivity of R,

$$\begin{aligned} T(P(a,b), P(b,c)) &= T(R(a,b), R(b,c)) \leq R(a,c) \\ &\leq R(c,a) = 0 = P(a,c). \end{aligned}$$ ■

From now on we only consider the t-norm $T_{\mathbf{L}}$ and we obtain fuzzy strict preference relations with respect to three t-conorms: $S_{\mathbf{L}}$, $S_{\mathbf{M}}$ and $S_{\mathbf{P}}$.

Theorem 5 *Let I be a $T_{\mathbf{L}}$-equivalence and R a $T_{\mathbf{L}}$-I-ordering. A fuzzy relation P is a strict preference relation associated with $(R, I, T_{\mathbf{L}}, S_{\mathbf{L}})$ if and only if for all $a, b \in A$*

$$P(a,b) = R(a,b) - I(a,b).$$

Proof

Suppose first that P is a strict preference relation associated with $(R, I, T_{\mathbf{L}}, S_{\mathbf{L}})$. Since $P \cap_{T_{\mathbf{L}}} I = \emptyset$ and $R = P \cup_{S_{\mathbf{L}}} I$, then for all $a, b \in A$ the following hold:

$$P(a,b) + I(a,b) \leq 1 \tag{2}$$

$$R(a,b) = \min(P(a,b) + I(a,b), 1) \tag{3}$$

Therefore, $R(a,b) = P(a,b) + I(a,b)$, i.e. $P(a,b) = R(a,b) - I(a,b)$.

Assume now that $P(a,b) = R(a,b) - I(a,b)$ for all $a, b \in A$. Since $P(a,b) + I(a,b) = R(a,b) \leq 1$, (2) and (3) are trivially satisfied. We are going to prove that $P \cap_{T_{\mathbf{L}}} P^{-1} = \emptyset$, i.e. for all $a, b \in A$

$$P(a,b) + P(b,a) \leq 1 \tag{4}$$

Since R is a T-I-ordering, we have

$$R(a,b) + R(b,a) - 1 \leq I(a,b).$$

Therefore, $R(a,b) + R(b,a) - 1 \leq 2I(a,b)$, or equivalently,

$$R(a,b) - I(a,b) + R(b,a) - I(b,a) \leq 1,$$

i.e., (4) is satisfied. ■

Dasgupta and Deb [2] have proven that if R is $S_{\mathbf{L}}$-complete and $I(a,b) = T_{\mathbf{L}}(R(a,b), R(b,a))$ or $I(a,b) = m_R(a,b)$, then P is $T_{\mathbf{L}}$-transitive. However, if I is different from the previous cases it is possible that P is not $T_{\mathbf{L}}$-transitive, even when R is $S_{\mathbf{L}}$-complete. For example, if $A = \{a, b, c\}$ and we denote R, P and I with the usual matricial notation, it is sufficient to consider

$$R = \begin{pmatrix} 1 & 0.8 & 0.6 \\ 0.4 & 1 & 0.8 \\ 0.6 & 0.4 & 1 \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 0.2 & 0.6 \\ 0.2 & 1 & 0.3 \\ 0.6 & 0.3 & 1 \end{pmatrix},$$

$$P = \begin{pmatrix} 0 & 0.6 & 0 \\ 0.2 & 0 & 0.5 \\ 0 & 0.1 & 0 \end{pmatrix}.$$

Since $P(a,b) + P(b,c) - 1 > P(a,c)$, P is not $T_{\mathbf{L}}$-transitive.

Now, we consider the t-conorm $S_{\mathbf{M}}$.

Theorem 6 *Let I be a $T_{\mathbf{L}}$-equivalence and R a $T_{\mathbf{L}}$-I-ordering. A fuzzy relation P is a strict preference relation associated with $(R, I, T_{\mathbf{L}}, S_{\mathbf{M}})$ if and only if for all $a, b \in A$ the relation R satisfies*

$$R(a,b) + R(b,a) \leq 1 \;\; \textit{if} \;\; I(a,b) < M_R(a,b)$$

and P is defined by

$$P(a,b) = R(a,b), \;\; \textit{if} \;\; R(a,b) - I(a,b) > 0,$$
$$P(a,b) \in [0\ ,\ \min(R(a,b), 1 - R(a,b))], \;\; \textit{otherwise.}$$

Proof

Suppose first that P is a strict preference relation associated with $(R, I, T_{\mathbf{L}}, S_{\mathbf{M}})$. Since $R = P \cup_{S_{\mathbf{M}}} I$, $P \cap_{T_{\mathbf{L}}} I = \emptyset$ and $P \cap_{T_{\mathbf{L}}} P^{-1} = \emptyset$, then for all $a, b \in A$ the following hold:

$$R(a,b) = \max(P(a,b), I(a,b)) \tag{5}$$

$$P(a,b) + I(a,b) \leq 1 \tag{6}$$

$$P(a,b) + P(b,a) \leq 1 \tag{7}$$

Given $a, b \in A$, since the previous relationships are also satisfied for the ordered pair (b,a), we have

$$R(b,a) = \max(P(b,a), I(a,b)) \tag{8}$$

$$P(b,a) + I(a,b) \leq 1 \tag{9}$$

We distinguish two cases:

1. If $R(a,b) = R(b,a)$, then from (5) we have again two cases:

   (a) If $I(a,b) = R(a,b)$, then (5) and (8) are always satisfied. From these equations we have $P(a,b) \leq R(a,b)$ and $P(b,a) \leq R(a,b)$. And by conditions (6) and (9) we also obtain $P(a,b) \leq 1-R(a,b)$ and $P(b,a) \leq 1-R(a,b)$. Therefore,

   $$P(a,b) \leq \min(R(a,b), 1-R(a,b)),$$

   $$P(b,a) \leq \min(R(a,b), 1-R(a,b)),$$

   and (7) is always satisfied.

   (b) If $I(a,b) < R(a,b)$, then

   $$P(a,b) = P(b,a) = R(a,b)$$

   by (5) and (8). From (7) we have

   $$R(a,b) + R(b,a) \leq 1,$$

   i.e. $R(a,b) \leq 1/2$. Since $I(a,b) < R(a,b)$, conditions (6) and (9) are always satisfied because $R(a,b) \leq 1 - R(a,b)$.

2. If $R(a,b) \neq R(b,a)$, suppose, for instance, that $R(b,a) < R(a,b)$. From (8) we distinguish two cases:

   (a) If $I(a,b) = R(b,a)$, then (8) is satisfied and $P(a,b) = R(a,b)$ by (5). From (6) we have

   $$R(a,b) + R(b,a) \leq 1.$$

   Since $P(b,a) \leq R(b,a)$ by (8), conditions (7) and (9) are always satisfied because

   $$R(b,a) \leq 1 - R(a,b) \leq 1 - R(b,a).$$

   (b) If $I(a,b) < R(b,a)$, then $P(a,b) = R(a,b)$ and $P(b,a) = R(b,a)$ by (5) and (8), respectively. From (7) we have

   $$R(a,b) + R(b,a) \leq 1.$$

   Since $I(a,b) < R(b,a)$, conditions (6) and (9) are always satisfied because

   $$R(b,a) \leq 1 - R(a,b) \leq 1 - R(b,a).$$

   Analogously, if $R(a,b) < R(b,a)$ we obtain that $P(b,a) = R(b,a)$, $R(a,b) + R(b,a) \leq 1$ and

   $$\max(I(a,b), P(a,b)) = R(a,b).$$

From these cases we can obtain the conditions given in the formulation of this Theorem.

On the other hand, it is easy to check that the converse statement is true. ■

In Theorem 6, if R is $S_{\mathbf{M}}$-complete it is easy to check that R, I and P are crisp relations. So, since R is $T_{\mathbf{L}}$-transitive then P is also $T_{\mathbf{L}}$-transitive. However, if R is not $S_{\mathbf{M}}$-complete then it is possible that P is not $T_{\mathbf{L}}$-transitive. For example, it is sufficient to consider

$$R = \begin{pmatrix} 1 & 0.6 & 0.6 \\ 0.2 & 1 & 0.5 \\ 0.6 & 0.3 & 1 \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 0.2 & 0.6 \\ 0.2 & 1 & 0.3 \\ 0.6 & 0.3 & 1 \end{pmatrix},$$

$$P = \begin{pmatrix} 0 & 0.6 & 0 \\ 0.2 & 0 & 0.5 \\ 0 & 0.1 & 0 \end{pmatrix}.$$

Finally, we consider the t-conorm $S_{\mathbf{P}}$. We will use the following notation: $Z(a,b) = \sqrt{M_R(a,b) - 3/4}$.

Theorem 7 *Let I be a $T_{\mathbf{L}}$-equivalence and R a $T_{\mathbf{L}}$-I-ordering. A fuzzy relation P is a strict preference relation associated with $(R, I, T_{\mathbf{L}}, S_{\mathbf{P}})$ if and only if for all $a, b \in A$:*

1. *The relations R and I satisfy the following condition:*
   If $M_R(a,b) > 3/4$, then

   $$1/2 + Z(a,b) \leq m_R(a,b) \quad \textit{and}$$

   $$I(a,b) \in [1/2 + Z(a,b)\ ,\ m_R(a,b)]$$

   or

   $$m_R(a,b) \leq 3/2 - M_R(a,b) - Z(a,b) \quad \textit{and}$$

   $$I(a,b) \in [T_{\mathbf{L}}(R(a,b), R(b,a))\ ,\ 1/2 - Z(a,b)]\,.$$

2. *P is defined by*

   $$P(a,b) = \begin{cases} \dfrac{R(a,b) - I(a,b)}{1 - I(a,b)}, & \text{if } I(a,b) < 1, \\ 0, & \text{if } I(a,b) = 1. \end{cases}$$

Proof

Suppose first that P is a strict preference relation associated with $(R, I, T_{\mathbf{L}}, S_{\mathbf{P}})$. Since $R = P \cup_{S_{\mathbf{P}}} I$, $P \cap_{T_{\mathbf{L}}} I = \emptyset$ and $P \cap_{T_{\mathbf{L}}} P^{-1} = \emptyset$, then for all $a, b \in A$ the following hold:

$$R(a,b) = P(a,b) + I(a,b) - P(a,b)I(a,b) \quad (10)$$

$$P(a,b) + I(a,b) \leq 1 \quad (11)$$

$$P(a,b) + P(b,a) \leq 1 \quad (12)$$

From (10) we have

$$P(a,b) = \frac{R(a,b) - I(a,b)}{1 - I(a,b)} \quad (13)$$

if $I(a,b) < 1$. Since (11) is satisfied, if $I(a,b) = 1$ then $P(a,b) = 0$.

Moreover, the equations (11) and (12) have also to be satisfied when P is given by (13). Thus, from (12) we obtain

$$\frac{R(a,b) - I(a,b)}{1 - I(a,b)} + \frac{R(b,a) - I(a,b)}{1 - I(a,b)} \leq 1,$$

i.e., $R(a,b) + R(b,a) - 1 \leq I(a,b)$, but this condition is always verified because R is a $T_{\mathbf{L}}$-I-ordering. From (11) we have

$$\frac{R(a,b) - I(a,b)}{1 - I(a,b)} + I(a,b) \leq 1,$$

or equivalently,

$$R(a,b) \leq I^2(a,b) - I(a,b) + 1. \tag{14}$$

Since $x^2 - x + 1 \geq 3/4$ for all $x \in [0,1]$, we have that (14) is always true if $R(a,b) \leq 3/4$. On the other hand, if $R(a,b) > 3/4$ then

$$I^2(a,b) - I(a,b) + (1 - R(a,b)) \geq 0$$

if and only if

$$I(a,b) \leq 1/2 - \sqrt{R(a,b) - 3/4}$$

or

$$I(a,b) \geq 1/2 + \sqrt{R(a,b) - 3/4}.$$

Since these relationships are also true for the ordered pair (b,a), we have that if $M_R(a,b) > 3/4$ then (14) is satisfied if and only if

$$I(a,b) \leq 1/2 - Z(a,b)$$

or

$$I(a,b) \geq 1/2 + Z(a,b).$$

Since R is a $T_{\mathbf{L}}$-I-ordering, $I(a,b)$ satisfies

$$T_{\mathbf{L}}(R(a,b), R(b,a)) \leq I(a,b) \leq m_R(a,b).$$

Therefore, if $M_R(a,b) > 3/4$ then (14) is satisfied if and only if

$$T_{\mathbf{L}}(R(a,b), R(b,a)) \leq I(a,b) \leq 1/2 - Z(a,b) \tag{15}$$

or

$$1/2 + Z(a,b) \leq I(a,b) \leq m_R(a,b). \tag{16}$$

Since $1/2 - Z(a,b) \geq 0$, the condition (15) is satisfied if and only if

$$M_R(a,b) + m_R(a,b) - 1 \leq 1/2 - Z(a,b),$$

or, equivalently

$$m_R(a,b) \leq 3/2 - M_R(a,b) - Z(a,b).$$

On the other hand, it is easy to check that the converse statement is true. ■

In Theorem 7, if R is $S_{\mathbf{P}}$-complete it is easy to check that R, I and P are crisp relations. So, since R is $T_{\mathbf{L}}$-transitive then P is also $T_{\mathbf{L}}$-transitive. However, if R is not $S_{\mathbf{P}}$-complete then it is possible that P is not $T_{\mathbf{L}}$-transitive. For example, it is sufficient to consider

$$R = \begin{pmatrix} 1 & 0.68 & 0.6 \\ 0.36 & 1 & 0.65 \\ 0.6 & 0.37 & 1 \end{pmatrix},\ I = \begin{pmatrix} 1 & 0.2 & 0.6 \\ 0.2 & 1 & 0.3 \\ 0.6 & 0.3 & 1 \end{pmatrix},$$

$$P = \begin{pmatrix} 0 & 0.6 & 0 \\ 0.2 & 0 & 0.5 \\ 0 & 0.1 & 0 \end{pmatrix}.$$

Acknowledgements

This work is partially financed by Junta de Castilla y León (Proyecto VA057/02), Ministerio de Ciencia y Tecnología (Plan Nacional de Investigación, Desarrollo e Innovación Tecnológica (I+D+I), Proyecto BEC2001-2253, and National Network in Decision Making, Preference Modelling and Aggregation, Ref. TIC-2002-11942-E) and ERDF.

References

[1] U. Bodenhofer. A similarity-based generalization of fuzzy orderings preserving the classical axioms. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 8, pp. 593–610, 2000.

[2] M. Dasgupta, R. Deb. Factoring fuzzy transitivity. *Fuzzy Sets and Systems* 118, pp. 489–502, 2001.

[3] B. De Baets, J. Fodor. Additive fuzzy preference structures: the next generation. In B. De Baets, J. Fodor (Eds.), *Principles of Fuzzy Preference Modelling and Decision Making*. Academia Press, Gent, pp. 15–25, 2003.

[4] J. Fodor, M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers, Dordrecht, 1994.

[5] U. Höhle, N. Blanchard. Partial ordering in L-underdeterminate sets. *Information Sciences* 35, pp. 133–144, 1985.

[6] B. Llamazares. Characterization of fuzzy preference structures through Łukasiewicz triplets *Fuzzy Sets and Systems* 136, pp. 217–235, 2003.

[7] S. Ovchinnikov. Structure of fuzzy binary relations. *Fuzzy Sets and Systems* 6, pp. 169–195, 1981.

Preórdenes Duales, Operadores de Clausura y Contextos Borrosos

J. Jacas
Sec. Matemàtiques i Informàtica
ETSAB
Diagonal 649
Univ. Politècnica de Catalunya
08028 Barcelona
joan.jacas@upc.es

J. Recasens
Sec. Matemàtiques i Informàtica
ETSAV
Pere Serra 1-15
Univ. Politècnica de Catalunya
08190 Sant Cugat del Vallès
j.recasens@upc.es

Resumen

Este trabajo relaciona los operadores de clausura en contextos borrosos con los operadores de clausura generados por preórdenes.

Palabras Clave: T-preorden, T-preórdenes duales, Operador de clausura, Contexto borroso.

1 Introducción

Un contexto borroso es una terna (X, A, R) donde X y A son conjuntos y R una relación borrosa $R : X \times A \to [0, 1]$. X se interpreta como un conjunto de objetos, A como un conjunto de atributos y $R(x, a)$ como el grado en que x posee el atributo a. R genera sendos operadores de clausura $''$ en $[0, 1]^X$ y $[0, 1]^A$. R también genera T-preórdenes duales en X y A que inducen operadores de clausura. La siguiente sección relaciona los operadores de clausura generados por T-preórdenes con el conjunto de sus generadores, que a su vez pueden interpretarse como filtros borrosos. En la Sección 3 se define el concepto de T-preórdenes duales y se enuncian algunas de sus propiedades que serán usadas en la Sección 4 para relacionar los operadores de clausura en un contexto borroso con los de los preórdenes duales generados por la relación en el contexto.

2 Preórdenes, filtros, ideales y operadores de clausura

Definición 2.1. Dada una t-norma T, un *T-preorden* P en un conjunto X es una relación borrosa $P : X \times X \to [0, 1]$ que satisface

2.1.1. $P(x, x) = 1 \ \forall x \in X$

2.1.2. $T(P(x, y), P(y, z)) \leq P(x, z) \ \forall x, y, z \in X$.

Si P satisface además

2.1.3. $P(x, y) = P(y, x) \ \forall x, y \in X$,

entonces P es una *T-indistinguibilidad*.

Definición 2.2. Dada una relación borrosa R en un conjunto X, su inversa R^{-1} es la relación borrosa

$$R^{-1}(x, y) = R(y, x) \ \forall x, y \in X$$

Teorema de Representación de T-preórdenes 2.3. [9] Una relación borrosa P en un conjunto X es un T-preorden en X si, y sólo si, existe una familia $\{h_i\}_{i\in I}$ de subconjuntos borrosos de X tal que para todo $x, y \in X$

$$P(x, y) = \inf_{i\in I} \hat{T}(h_i(x)|h_i(y)).$$

Nota: $\hat{T}$ representa la residuación de la t-norma T

$$\hat{T}(x|y) = x \to_T y = \sup(\alpha|T(\alpha, x) \leq y).$$

Un conjunto borroso perteneciente a alguna familia como la del teorema anterior se denomina un generador de P. Es claro que h es un generador de P sii

$$\hat{T}(h(x)|h(y)) \geq P(x, y) \ \forall x, y \in X. \qquad (1)$$

Definición 2.4. Dada un T-preorden P en un conjunto X, ϕ^i_P y ϕ^f_P son dos operadores tales que para cada $h \in [0, 1]^X$

$$\phi^i_P(h(x)) = \sup_{y\in X} T(h(y), P(x, y))$$
$$\phi^f_P(h(x)) = \sup_{y\in X} T(h(y), P(y, x)).$$

En un conjunto ordenado $(X, \leq)$, un ideal y un filtro se definen de la siguiente manera:

$I \subset X$ es un ideal de $(X, \leq)$ sii $\forall x \in X, \forall y \in I, x \leq y$ implica $x \in I$.

$F \subset X$ es un filtro de $(X, \leq)$ sii $\forall x \in X, \forall y \in F, x \geq y$ implica $x \in F$.

Dado un conjunto $A \subset X$, el ideal I_A y el filtro F_A generados por A son los siguientes ideal y filtro respectivamente.

$$\begin{aligned} I_A &= \{x \in X \mid \exists y \in A \wedge x \leq y\} \\ F_A &= \{x \in X \mid \exists y \in A \wedge x \geq y\} \end{aligned}$$

Cambiando en I_A y F_A $\exists y$ por $\sup_{y \in X}$, $\wedge$ por T y $\leq$ por P, vemos que la definición 2.4. fuzzifica estos dos conceptos por lo que $\phi_P^i(h)$ y $\phi_P^f(h)$ pueden considerarse el ideal y el filtro borroso generados por el subconjunto borroso h.

Definición 2.5. Dado un conjunto parcialmente ordenado M, un operador $c : M \to M$ es un operador de clausura sii para todo $x, y \in M$ satisface

2.5.1. $x \leq c(x)$

2.5.2. $x \leq y$ implica $c(x) \leq c(y)$

2.5.3. $c(c(x)) = c(x)$.

Proposición 2.6. El operador ϕ_P^f es un operador de clausura en $[0,1]^X$.

Corolario 2.7. Dado un T-preorden P en X, El operador $\phi_{P^{-1}}^i$ es un operador de clausura en $[0,1]^X$ y coincide con ϕ_P^f.

Proposición 2.8. Dado un T-preorden P en X, un subconjunto borroso h de X es un generador de P sii $\phi_P^f(h) = h$

Este resultado puede traducirse del siguiente modo: Un subconjunto borroso h es un generador de P sii es un filtro borroso.

Corolario 2.9. Sea F_P el conjunto de generadores de P. $F_P = \phi_P^f([0,1]^X)$.

En este punto es interesante observar que los resultados obtenidos en [8] para T-indistinguibilidades se generalizan sin más a T-preórdenes. Las demostraciones de los siguientes dos resultados conciden con las de los Teoremas 3.2 y 3.3 de [8].

Proposición 2.10. Sea P un T-preorden en X, F_P el conjunto de generadores o filtros borrosos de P, $A \subset F_P$, h un generador de P y $\alpha \in [0,1]$.

1. $\bigvee A \in F_P$
2. $\bigwedge A \in F_P$
3. $T(\alpha, h) \in F_P$
4. $\hat{T}(h|\alpha) \in F_P$
5. $\hat{T}(\alpha|h) \in F_P$

Teorema 2.10 Sea F un conjunto de subconjuntos borrosos de X que satisface las propiedades de la proposición anterior. En este caso existe un T-preorden P de modo que F es el conjunto de sus generadores. P queda determinada por $P(x,y) = \inf_{h \in F} \hat{T}(h(x)|h(y))$ para todo $x, y \in X$.

3 T-Preórdenes duales

Definición 3.1. Dos T-preórdenes P y Q en X y A respectivamente son *duales* si y sólo si existe $R : X \times A \to [0,1]$ tal que

$$P(x,y) = \inf_{a \in A} \hat{T}(R(x,a)|R(y,a)) \ \forall x, y \in X$$
$$Q(a,b) = \inf_{x \in X} \hat{T}(R(x,a)|R(x,b)) \ \forall a, b \in A.$$

Ejemplo 3.2. Sea P un T-preorden en X. P y P^{-1} son T-preórdenes duales con respecto a la relación $R = P$.

Ejemplo 3.3 Sea P un T-preorden en X, F el conjunto de sus generadores y $R : X \times F \to [0,1]$ la relación $R(x,h) = h(x)$ $\forall x \in X$, $\forall h \in F$. El T-preorden P^* en F definido por

$$\begin{aligned} P^*(h,l) &= \inf_{x \in X} \hat{T}(R(x,h)|R(x,l)) = \\ &= \inf_{x \in X} \hat{T}(h(x)|l(x)) \qquad \forall h, l \in F \end{aligned}$$

es dual de P respecto a la relación R.

Proposición 3.4. Si P y Q son dos T-preórdenes en X, A respectivamente duales respecto a R, entonces $\{R(\cdot, a)\}_{a \in A}$ y $\{R(x, \cdot)\}_{x \in X}$ son familias de generadores de P y Q respectivamente.

Sean P y Q dos T-preórdenes en X, A respectivamente duales respecto a R, F_P el conjunto de generadores de P e i la aplicación

$$\begin{aligned} i : A &\to F_P \\ a &\to R(\cdot, a). \end{aligned}$$

Entonces

$$Q(a,b) = P^*(i(a), i(b)). \tag{2}$$

Por tanto,

Proposición 3.5. Sea P un T-preorden en X y Q un T-preorden en A. P yQ son duales si, y sólo si, existe una aplicación $i : A \to F_P$ tal que $i(A)$ es una familia

de generadores de P y $(Q(a,b) = P^*(i(a), i(b)) \; \forall a, b, \in A$. La relación R es $R(x,a) = i(a)(x) \; \forall x \in X, a \in A$.

Demostración.

$$\inf_{a\in A} \hat{T}(R(x,a)|R(y,a)) =$$
$$= \inf_{a\in A} \hat{T}(i(a)(x)|i(a)(y)) = P(x,y).$$

$$\inf_{x\in X} \hat{T}(R(x,a)|R(x,b)) =$$
$$= \inf_{x\in x} \hat{T}(i(a)(x)|i(b)(x)) = P^*(i(a), i(b)) = Q(a,b).$$

De forma totalmente análoga se pueden definir operadores de T-indistinguibilidad duales respecto a una relación borrosa R.

4 Contextos borrosos

Un *contexto* (clásico) es una terna (X, A, R) donde X y A son dos conjuntos y R una relación $(R \subset X \times A)$. X se puede interpretar como un conjunto de objetos, A un conjunto de atributos y $(x, a) \in R$, que como es usual escribiremos xRa, como la relación "el objeto x posee el atributo a".

Para un subconjunto de objetos $Y \subset X$ y un conjunto de atributos $B \subset A$, se define

$$\begin{aligned} Y' &= \{a \in A \mid xRa \; \forall x \in Y\} \\ B' &= \{x \in x \mid xRa \; \forall a \in B\} \end{aligned}$$

Es decir, Y' es el conjunto de atributos compartidos por todos los objetos de Y y B' es el conjunto de objetos que poseen todos los atributos de B.

En un contexto un *concepto* se define como un par (Y, B) con $Y \subset X$ y $B \subset A$ tal que $Y' = B$ y $B' = Y$.

El traslado al caso borroso se consigue sustituyendo la relación clásica R por otra borrosa.

Definición 4.1. Un *contexto borroso* es una terna (X, A, R) con X, A conjuntos y $R : X \times A \to [0, 1]$.

$R(x, a)$ es el grado en que el objeto x posee el atributo a.

Definición 4.2. Dados h y μ sendos subconjuntos borrosos de X y A con (X, A, R) un contexto borroso, se definen los subconjuntos borrosos h' y μ' de A y X respectivamente del siguiente modo:

$$\begin{aligned} h'(a) &= \inf_{x\in X} \hat{T}(h(x)|R(x,a)) \; \forall a \in A \\ \mu'(x) &= \inf_{a\in A} \hat{T}(\mu(a)|R(x,a)) \; \forall x \in X. \end{aligned}$$

Definición 4.3. Un *concepto borroso* en un contexto borroso (X, A, R) es un par (h, μ), h subconjunto borroso de X y μ subconjunto borroso de A tal que $h' = \mu$ y $\mu' = h$.

Proposición 4.4. Si (h, μ) es un concepto, entonces $(h')' = h$ y $(\mu')' = \mu$.

Los dos operadores $'$ en un contexto son un caso particular de conexiones de Galois.

Definición 4.5. Sean M, N dos conjuntos parcialmente ordenados. Un par $(^{\triangleright}, ^{\triangleleft})$ de aplicaciones $^{\triangleright} : M \to N$ y $^{\triangleleft} : N \to M$ son una *conexión de Galois* sii para todo $m \in M, n \in N$

$$m^{\triangleright} \leq n \iff m \leq n^{\triangleleft}. \tag{3}$$

Lema 4.6. En un contexto borroso (X, A, R), los operadores $'$ son una conexion de Galois entre $[0, 1]^X$ y $[0, 1]^A$.

Lema 4.7. Sea $(^{\triangleright}, ^{\triangleleft})$ una conexión de Galois entre M y N. Los operadores

$$c_M = ^{\triangleright\,\triangleleft} : M \to M \; y \; c_N = ^{\triangleleft\,\triangleright} : N \to N \tag{4}$$

son operadores de clausura.

Corolario 4.8. Sea (X, A, R) un contexto borroso. Los operadores

$$'' : [0,1]^X \to [0,1]^X \quad y \quad '' : [0,1]^A \to [0,1]^A \tag{5}$$

son operadores de clausura.

La forma explícita de estos operadores es

$$(h')'(x) = \inf_{a\in A} \hat{T}(\inf_{y\in X} \hat{T}(h(y)|R(y,a))|R(x,a)) \quad \forall x \in X \tag{6}$$

y

$$(\mu')'(a) = \inf_{x\in X} \hat{T}(\inf_{b\in A} \hat{T}(\mu(b)|R(x,b))|R(x,a)) \quad \forall a \in A \tag{7}$$

respectivamente.

Tomando un subconjunto de X con un solo elemento $\{x_0\}$ con función caracerística h_{x_0} ($h_{x_0}(x) = 1$, si $x = x_0$ y $h_{x_0}(x) = 0$, si $x \neq x_0$), la igualdad (6) se convierte en

$$(h'_{x_0})'(x) = \inf_{a\in A} \hat{T}(R(x_0,a)|R(x,a)) \; \forall x \in X \tag{8}$$

ya que

$$h'_{x_0}(a) = \inf_{x\in X} \hat{T}(h_{x_0}(x)|R(x,a)) = R(x_0,a) \quad \forall a \in A \tag{9}$$

Proposición 4.9. La relación borrosa P en X definida por

$$P(x,y) = (h'_x)'(y) \ \forall x, y \in X \tag{10}$$

es un T-preorden.

Demostración.

En el teorema 2.3. tomar la familia $\{R(\cdot,a)\}_{a\in A}$.

Por consiguiente, P genera un operador de clausura ϕ^f_P en $[0,1]^X$:

$$\phi^f_P(h(x)) = \sup_{y\in X} T(h(y), P(y,x)) \tag{11}$$

o de forma más explícita

$$\begin{aligned}&\phi^f_P(h(x)) = \\ &= \sup_{y\in X} T(h(y), \inf_{a\in A} \hat{T}(R(y,a)|R(x,a))) = \\ &= \sup_{y\in X} \inf_{a\in A} T(h(y), \hat{T}(R(y,a)|R(x,a)))\end{aligned}$$

Proposición 4.10. $\phi^f_P(h) \leq (h')' \ \forall h \in [0,1]^X$.

Demostración.

La expresión (6) puede escribirse

$$(h')'(x) = \inf_{a\in A} \sup_{y\in X} \hat{T}(\hat{T}(h(y)|R(y,a))|R(x,a)) \quad \forall x \in X. \tag{12}$$

y el resultado se sigue comparando esta expresión con (12) gracias al siguiente lema.

Lema 4.11. [2] Para todo $x, y, z \in [0,1]$,

$$\hat{T}(\hat{T}(x|y)|z) \geq T(x, \hat{T}(y|z)). \tag{13}$$

En general no se tiene la igualdad en la proposición anterior. Por ejemplo, si h es el conjunto vacío y existe $x_0 \in X$ con $R(x_0,a) = 1 \ \forall a \in A$, entonces $(h')'(x_0) = 1$ mientras que $\phi^f_P(h)(x_0) = 0$.

Tenemos pues en X dos operadores de clausura definidos de forma natural a partir de la relación R que coinciden en los singletones clásicos pero que en general difieren en el resto de subconjuntos tanto clásicos como borrosos de X.

Un estudio similar permite definir un preorden borroso en A con su operador de clausura.

En efecto, si μ_{a_0} es la función característica de $\{a_0\} \subset A$, se tiene

$$(\mu'_{a_0})'(a) = \inf_{x\in X} \hat{T}(R(x,a_0)|R(x,a)) \quad \forall a \in A \tag{14}$$

y $Q(a,b) = (\mu'_a)'(b)$ es un T-preorden borroso. El operador de clausura ϕ^f_Q generado per Q coincide con $''$ en los singletones clásicos de A y es menor que $''$ ($\phi^f_Q(\mu) \leq (\mu')' \ \forall \mu \in [0,1]^A$).

Agradecimientos

Este trabajo ha sido parcialmente financiado por el proyecto número TIC2003-04564 del DGICYT.

Referencias

[1] U. Bodenhofer. A Similarity-Based Generalization of Fuzzy Orderings. *Schriftenreihe der Johannes Kepler Universität* **26**. Universitätsverlag Rudolf Trauner, Linz, 1999.

[2] D. Boixader. Some Properties Concerning the Quasi-inverse of a t-norm. *Mathware & Soft Computing* **5** Pág.5-12, 1998.

[3] A. Burrusco, R. Fuentes-Ganzález. The Embedding of the Formal Concept Analysis into the *L*-Fuzzy Concept Theory. *Mathware & Soft Computing* **5** Pág.13-22, 1998.

[4] J.L. Castro, E. Trillas. Tarski's fuzzy consequences. *Proc. Internat. Fuzzy Eng. Symp'91* Pág.70-81, 1991.

[5] B.A. Davey, H.A. Priestley. Introduction to Lattices and Order. Cambridge University Press. 2002.

[6] J. Elorza, P. Burillo. Connecting fuzzy preorders, fuzzy consequence operators and fuzzy closure and co-closure systems. *Fuzzy Sets and Systems* **139** Pág.601-613, 2003.

[7] E. Hernández, J. Recasens. Generating Indistinguishability Operators from Prototypes. *Int. J. Intelligent Systems* **17** Pág.1131-1142, 2002.

[8] F. Klawonn, J.L. Castro. Similarity in Fuzzy Reasoning. *Mathware & Soft Computing* **5** Pág.197228, 1995.

[9] L. Valverde. On the Structure of F-Indistinguishability Operators. *Fuzzy Sets and Systems* **17** Pág.313-328, 1985.

Reglas de Asociación sobre Dominios Imprecisos aplicando Relaciones Difusas de Similitud

D. Sánchez, M.A. Vila
Dpto. Ciencias de la Computación
e Inteligencia Artificial
Universidad de Granada (España)
{daniel,vila}@decsai.ugr.es

J.R. Sánchez
Servicio de las Tecnologías
de la Información y la Comunicación
Universidad de Almería (España)
jrsanche@ual.es

J.M. Serrano[1]
Dpto. Informática
Universidad de Jaén (España)
jschica@ujaen.es

Resumen

Las reglas de asociación constituyen una herramienta eficaz para la extracción de relaciones entre valores dentro de una base de datos. Existen distintas variaciones en función de las fuentes de datos utilizadas y según el problema en concreto. En este trabajo nos centraremos en la extracción de reglas de asociación difusas, extendiendo un modelo ya existente para dar la posibilidad de aplicar la herramienta sobre una base de datos relacional difusa, sobre la que se puedan haber definido relaciones de similitud entre valores.

Palabras Clave: Bases de datos difusas, Relaciones difusas de similitud, Reglas de asociación difusas, Minería de datos difusos.

1 INTRODUCCIÓN

En la actual sociedad de la información, las grandes bases de datos pueden considerarse como una importante fuente de información. Existen casos en los que lo realmente interesante no radica en la recuperación de la información tal y como ésta se halla almacenada en una base de datos, sino en la posibilidad de obtener nuevo conocimiento a partir de la información conocida. Existen multitud de herramientas, agrupadas comúnmente bajo la denominación de Extracción de conocimiento en Bases de datos, destinadas a llevar a cabo dichas funciones de forma eficiente.

Por otro lado, al enfrentarnos con problemas reales, suele ser común el hecho de que aparezca un cierto grado de imprecisión o incertidumbre sobre los datos. Por otro lado, nos puede interesar manejar información precisa mediante herramientas flexibles, con el objeto de disminuir la granularidad o definir el contenido semántico de los resultados. En ambos casos, la teoría de los subconjuntos difusos nos proporciona una ayuda inestimable para la representación de información afectada de incertidumbre, imprecisión o vaguedad en general [27]. Al igual que existen modelos para representar y manejar datos que admiten este tipo de características, es necesario extender las herramientas de extracción de conocimiento para tratar con datos difusos. Existen distintos avances al respecto, como se verá más adelante.

En este trabajo, mostramos una extensión de las herramientas conocidas como reglas de asociación que permite trabajar con valores definidos sobre dominios imprecisos en una base de datos relacional difusa. Mediante este nuevo enfoque, definiendo una relación difusa entre valores, es posible enriquecer el modelo estableciendo un parecido semántico entre los datos, que pueda verse reflejado en las reglas obtenidas.

El trabajo está organizado de la siguiente forma. Tras la introducción, procedemos a enumerar herramientas y extensiones ya existentes para abordar el problema. A continuación, aportaremos la definición de nuestro enfoque, comentando aplicaciones y mostrando su utilidad en un ejemplo. Por último, destacamos las conclusiones del trabajo y las futuras líneas a seguir.

2 TRABAJOS RELACIONADOS

2.1 Reglas de asociación

Dado un conjunto de ítems, I, y un conjunto de transacciones T (al que también se le suele llamar T-set), donde cada transacción es un subconjunto de I, se dice que una regla de asociación [1] es una "implicación" de la forma $A \Rightarrow C$ que denota la presencia de los ítemsets (o conjuntos de ítems) A y C en algunas

[1]Autor para notificaciones. Tlf.: +34 953 026585, Fax: +34 953 002420

de las transacciones de T, asumiendo que $A, C \subset I$, $A \cap C = \emptyset$ y $A, C \neq \emptyset$.

Las medidas usuales propuestas en [1] para establecer la bondad y el interés de una regla de asociación son la *confianza* ($Conf(A \Rightarrow C)$, la probabilidad condicional $p(C|A)$) y el *soporte* ($Supp(A \Rightarrow C)$, la probabilidad conjunta $p(A \cup C)$).

Sin embargo, algunos autores apuntan algunas desventajas de la confianza, por lo que una solución alternativa es la que aparece en [3, 4], donde se propone en su lugar el *factor de certeza* de Shortliffe y Buchanan [29]. El factor de certeza de la regla $A \Rightarrow C$ se calcula como

$$CF(A \Rightarrow C) = \frac{(Conf(A \Rightarrow C)) - Supp(C)}{1 - Supp(C)} \quad (1)$$

si $Conf(A \Rightarrow C) > Supp(C)$, y

$$CF(A \Rightarrow C) = \frac{(Conf(A \Rightarrow C)) - Supp(C)}{Supp(C)} \quad (2)$$

si $Conf(A \Rightarrow C) < Supp(C)$, o 0 en otro caso.

El factor de certeza toma valores en $[-1, 1]$, indicando el grado en el que nuestra creencia de que el consecuente es cierto varía cuando el antecedente es también cierto. Se mueve desde 1, el máximo incremento (esto es, si A es cierto, C también lo es), hasta -1, que indica el máximo decremento.

Por definición, las reglas de asociación se encuentran definidas sobre conjuntos de transacciones. Dado que lo más común en una base de datos es trabajar con tuplas en lugar de transacciones, se han propuesto diversas soluciones a dicho problema. En caso de trabajar con bases de datos relacionales, lo usual es considerar que un ítem es cada par $\langle atributo, valor \rangle$, y que cada transacción se corresponde con una tupla en una tabla. Por ejemplo, diremos que el ítem $\langle A, a_0 \rangle$ está en la transacción asociada a una tupla t sií $t[A] = a_0$. Sin embargo esta variante presenta algunos inconvenientes. En el caso de atributos numéricos, el número de ítems asociado a cada posible par $\langle atributo, valor \rangle$ podría ser muy alto. Algunas soluciones propuestas son las Reglas de asociación Generalizadas ([30]), donde se permite la definición de taxonomías entre los valores de atributos, o las Reglas de asociación Cuantitativas ([31, 36]), donde a cada ítem se le hace corresponder un par $\langle atributo, intervalo \rangle$, esto es, se particiona el dominio del atributo en intervalos.

Otra posibilidad es la de dotar de un cierto grado de imprecisión a los valores, y extraer Reglas de asociación difusas.

2.2 Reglas de asociación difusas

En la literatura, encontramos diversas propuestas de reglas de asociación difusas como una generalización de las reglas de asociación cuando los datos de partida son difusos o se han fuzzyficado previamente [25, 2, 23, 24, 15]. Aunque la mayoría de estos enfoques se aplican sobre bases de datos relacionales, casi todos las medidas y algoritmos propuestos pueden emplearse en un marco más general. Se puede encontrar un interesante y completo estudio, con referencias bibliográficas sobre las extensiones a los casos de atributos cuantitativos o jerarquías de ítems, en [14]. Otros enfoques adicionales al problema son los aparecidos en [11, 21, 33, 18, 20, 19, 26, 12, 8]. En [17], se estudian diversas medidas para minería de datos difusos.

En [15], se parte de la definición de transacciones difusas como subconjuntos difusos de ítems, y será la idea en la que nos apoyaremos más adelante. Así, sea $I = \{i_1, \ldots, i_m\}$ un conjunto de ítems y T' un conjunto de transacciones difusas, en el que cada transacción difusa es un subconjunto difuso de I. Dada la transacción $\tilde{\tau} \in T'$, notaremos por $\tilde{\tau}(i_k)$ al grado de pertenencia de i_k en $\tilde{\tau}$. Una regla de asociación difusa es una implicación de la forma $A \Rightarrow C$, tal que $A, C \subset I$ y $A \cap C = \emptyset$.

Resulta inmediato que el conjunto de transacciones donde aparece un cierto ítem es un conjunto difuso, al que se denomina *representación* del ítem. Para el ítem i_k en T', tenemos el siguiente subconjunto difuso de T':

$$\tilde{\Gamma}_{i_k} = \sum_{\tilde{\tau} \in T'} \tilde{\tau}(i_k)/\tilde{\tau} \quad (3)$$

Esta representación puede extenderse a un ítemset de la forma siguiente: Dado $I_0 \subset I$ un ítemset, su representación es el siguiente subconjunto de T':

$$\tilde{\Gamma}_{I_0} = \bigcap_{i \in I_0} \tilde{\Gamma}_i = min_{i \in I_0} \tilde{\Gamma}_i \quad (4)$$

Para medir el interés y la bondad de una regla de asociación difusa, se han de aplicar herramientas de razonamiento aproximado, debido a la imprecisión que puede afectar a las transacciones difusas. De acuerdo con lo expuesto en [15], se puede usar un enfoque semántico basado en la evaluación de sentencias cuantificadas [35]. De acuerdo con [13], Q es un cuantificador difuso coherente si verifica las siguientes propiedades:

- $Q(0) = 0$ y $Q(1) = 1$
- Monotonía: Si $x < y$, $Q(x) \leq Q(y)$.

Dado Q, un cuantificador difuso coherente,

- El soporte de un ítemset I_0 en el FT-set T' es igual al resultado de evaluar la sentencia cuantificada Q de T' son $\tilde{\Gamma}_{I_0}$.
- El soporte de la regla de asociación $A \Rightarrow C$ en el FT-set T', $Supp(A \Rightarrow C)$, equivale a la evaluación de la sentencia cuantificada Q de T' son $\tilde{\Gamma}_{A \cup C} =$ Q de T' son $(\tilde{\Gamma}_A \cap \tilde{\Gamma}_C)$.
- La confianza de la regla de asociación $A \Rightarrow C$ en el FT-set T', $Conf(A \Rightarrow C)$, se corresponde con el resultado de la evaluación de la sentencia cuantificada Q de $\tilde{\Gamma}_A$ son $\tilde{\Gamma}_C$.

El factor de certeza, $CF(A \Rightarrow C)$, podemos seguir calculándolo de la misma forma en que lo hacíamos en el caso crisp, a partir del soporte y la confianza.

Las anteriores sentencias pueden evaluarse, por ejemplo, mediante el método GD, definido en [16] como

$$GD_Q(G/F) = \sum_{\alpha_i \in \triangle(G/F)} (\alpha_i - \alpha_{i+1}) Q\left(\frac{|(G \cap F)_{\alpha_i}|}{|F_{\alpha_i}|}\right) \quad (5)$$

donde $\triangle(G/F) = \wedge(G \cap F) \cup \wedge(F)$, siendo $\wedge(F)$ el conjunto de niveles de F, y $\triangle(G/F) = \{\alpha_1, \ldots, \alpha_p\}$ con $\alpha_i < \alpha_{i+1}$ para cada $i \in \{1, ..., p\}$. Suponemos que el conjunto F está normalizado. De no ser así, F se normaliza y el factor de normalización se aplica a $G \cap F$.

Según se extrae de [15], el método propuesto cuenta con la ventaja de ser una generalización del enfoque clásico para la estimación de la bondad de una regla de asociación.

2.3 Relaciones difusas

Dados dos elementos, no necesariamente incluidos en el mismo conjunto, es posible establecer una conexión entre ellos. Si esa conexión tiene un carácter impreciso o gradual, podemos formularla como una relación difusa. En lo que concierne a este trabajo, nos vamos a centrar especialmente en las relaciones difusas de similitud.

Una relación difusa sim definida sobre $X \times X$ es una relación de similitud [34] sií es una relación de semejanza (esto es, verifica las propiedades reflexiva y simétrica) y verifica además la propiedad Max-min transitiva, esto es:

$$sim(x, z) \geq max_{y \in X}\{min(sim(x, y), sim(y, z))\} \quad (6)$$

Las relaciones de similitud son una extensión del concepto clásico de relación de equivalencia. De hecho, los α-cortes de una relación difusa de similitud son relaciones crisp de equivalencia.

3 REGLAS DE ASOCIACIÓN BASADAS EN RELACIONES DIFUSAS DE SIMILITUD

Como indicamos en la introducción, la lógica difusa puede ser una herramienta efectiva para la representación de información heterogénea. En particular, las relaciones difusas de similitud nos permiten establecer nexos semánticos entre valores.

En el apartado anterior citábamos varias definiciones de reglas de asociación difusas que pueden hallarse en la literatura. Ninguna de ellas, hasta donde nosotros sabemos, contempla la posibilidad de establecer relaciones difusas entre los valores.

3.1 Definición

Continuaremos con la misma notación empleada en el apartado 2.2, pero asociando a cada ítem un par $\langle atributo, valor \rangle$. Así, dados dos ítems $i_0 = \langle At, a_0 \rangle$ e $i_1 = \langle At, a_1 \rangle$, y un grado de similitud $S_{At}(a_0, a_1) = \alpha$, nos centraremos en estudiar cómo el soporte de un ítem se ve afectado por las apariciones de ítems similares.

Extendemos la definición previa de regla de asociación difusa de la siguiente forma. Sea RE un esquema relacional, r una instancia de RE y $At \in RE$ un atributo, para el que definimos $dom(At) = \{a_1, \ldots, a_p\}$ como el conjunto de posibles valores de At. En un caso más general, podemos considerar incluso que la instancia r de partida es también difusa, y que si para cada tupla $t \in r$, $t[At] = a_i$ nos devuelve el valor tomado por el atributo A en t, mediante $\mu_{t[At]}$ indicaremos el grado en que toma dicho valor. A cada valor $a_i \in dom(At)$ le asociamos una etiqueta lingüística, E_{a_i}, definida como

$$E_{a_i} : dom(At) \rightarrow [0, 1]; E_{a_i}(a) = S_{At}(a_i, a) \quad (7)$$

donde $S_{At}(a_i, a)$ es el grado de similitud entre a_i y a. De esta manera, cada vez que un valor aparece, indicaremos su similitud con otros valores como el grado de compatibilidad que nos devuelve la etiqueta lingüística asociada.

Sea I_{At} el conjunto de ítems donde cada uno de sus elementos se asocia a un par $\langle At, E_{a_i} \rangle$, $|I_{At}| = |dom(At)|$. Transformamos cada tupla $t \in r$ en una transacción difusa $\tilde{\tau}$, de la siguiente forma,

$$\tilde{\tau}(\langle At, E_{a_i}\rangle) = min\{\mu_{t[At]}, E_{a_i}(t[At])\} \qquad (8)$$

Si trabajamos con datos precisos, basta con considerar el grado μ_t constantemente igual a 1. Siguiendo este proceso, obtenemos un conjunto de transacciones difusas sobre el que podemos aplicar la misma metodología propuesta en [15] para extraer reglas de asociación difusas.

3.2 Un ejemplo

Consideremos la tabla 1, extraída de [32], que recoge información sobre un conjunto de empleados (*Categoría profesional*, *Experiencia* y *Salario*). Sobre los valores que pueden tomar los distintos atributos se han definido unas relaciones difusas de similitud, mostradas en la tabla 2. En este ejemplo los grados de pertenencia de los valores a las tuplas son constantemente 1, por lo que no se han considerado.

Tabla 1: Relación difusa de empleados

	Cat.	*Exp.*	*Salario*
1	Vendedor	3	$37K$
2	Ing. Diseño	10	$40K$
3	Ing. Sistemas	5	$45K$
4	Ing. Software	5	$45K$
5	Contable	12	$47K$
6	Contable	5	$50K$
7	Secretaria	10	$53K$
8	Secretaria	15	$55K$

Tabla 2: Relaciones de similitud sobre los dominios de atributos de la tabla 1

CAT.	*Ing.Sw*	*Con*	*Ing.Sis*	*Vend*	*Ing.Dis*
Secr	0.6	0.7	0.6	0.5	0.6
Ing.Sw		0.6	0.8	0.5	0.8
Con			0.6	0.5	0.6
Ing.Sis				0.5	0.8
Vend					0.5

EXP.	5	10	12	15
3	0.9	0.7	0.7	0.5
5		0.7	0.7	0.5
10			0.9	0.7
12				0.7

SAL.	40	45	47	50	53	55
37	0.9	0.7	0.7	0.5	0.5	0.5
40		0.7	0.7	0.5	0.5	0.5
45			0.9	0.5	0.5	0.5
47				0.5	0.5	0.5
50					0.9	0.9
53						0.9

Como resultado de aplicar nuestra metodología sobre dicha tabla, obtenemos un conjunto de reglas de asociación difusas. Mostramos sólo algunas de ellas (restringiendo el soporte por encima del 20% y con factor de certeza superior a 0.6) en la tabla 3.

Tabla 3: Reglas de asociación difusas obtenidas

$[Sal. = 45K] \Rightarrow [Exp. = 5]$, s. 25.0%, FC 1.0
$[Sal. = 45K] \Rightarrow [Cat. = \text{“}Ing.Sw\text{”}]$, s. 22.5%, FC 0.87
$[Sal. = 45K] \Rightarrow [Cat. = \text{“}Ing.Sis\text{”}]$, s. 22.5%, FC 0.87
$[Sal. = 45K] \Rightarrow [Cat. = \text{“}Ing.Dis\text{”}]$, s. 20.0%, FC 0.75
$[Exp. = 10] \Rightarrow [Cat. = \text{“}Ing.Dis\text{”}]$, s. 20.0%, FC 0.75
$[Exp. = 10] \Rightarrow [Cat. = \text{“}Secr.\text{”}]$, s. 20.0%, FC 0.75
$[Exp. = 5] \Rightarrow [Cat. = \text{“}Ing.Sw\text{”}]$, s. 30.0%, FC 0.71
$[Exp. = 5] \Rightarrow [Cat. = \text{“}Ing.Sis\text{”}]$, s. 30.0%, FC 0.71
$[Exp. = 5] \Rightarrow [Cat. = \text{“}Ing.Dis\text{”}]$, s. 27.5%, FC 0.63
$[Exp. = 5] \Rightarrow [Cat. = \text{“}Con.\text{”}]$, s. 27.5%, FC 0.63
$[Cat. = \text{“}Ing.Dis\text{”}, Sal. = 45K] \Rightarrow [Exp. = 5]$, s. 20.0%, FC 1.0
$[Cat. = \text{“}Ing.Sis\text{”}, Sal. = 45K] \Rightarrow [Exp. = 5]$, s. 22.5%, FC 1.0
$[Cat. = \text{“}Ing.Sw\text{”}, Sal. = 45K] \Rightarrow [Exp. = 5]$, s. 22.5%, FC 1.0
$[Exp. = 5, Sal. = 45K] \Rightarrow [Cat. = \text{“}Ing.Sw\text{”}]$, s. 22.5%, FC 0.87
$[Exp. = 5, Sal. = 45K] \Rightarrow [Cat. = \text{“}Ing.Sis\text{”}]$, s. 22.5%, FC 0.87
$[Exp. = 5, Sal. = 45K] \Rightarrow [Cat. = \text{“}Ing.Dis\text{”}]$, s. 20.0%, FC 0.75
$[Cat. = \text{“}Ing.Sis\text{”}, Exp. = 5] \Rightarrow [Sal. = 45K]$, s. 22.5%, FC 0.74
$[Cat. = \text{“}Ing.Sw\text{”}, Exp. = 5] \Rightarrow [Sal. = 45K]$, s. 22.5%, FC 0.74
$[Cat. = \text{“}Ing.Dis\text{”}, Exp. = 5] \Rightarrow [Sal. = 45K]$, s. 20.0%, FC 0.69

El primer hecho a destacar sobre los resultados es que las únicas reglas de asociación clásicas que se podrían extraer serían todas triviales, dado el tamaño de la tabla. En cambio, relajando la semántica entre valores, tal y como hacemos, el conjunto de reglas obtenido se presta a una mejor interpretación.

3.3 Aplicaciones

Algunos de los posibles escenarios en los que nuestra definición de regla de asociación difusa podría resultar útil son los siguientes:

- En el caso de que sólo busquemos la obtención de reglas de asociación clásicas, bastaría con establecer como la igualdad clásica a la relación difusa de similitud. La función asociada $E_{a_i}(a) = 1$ sií $a_i = a$, y 0 en caso contrario. Cada ítem sólo es tenido en cuenta cuando en la tabla aparece el par $\langle atributo, valor\rangle$ correspondiente, lo que se traduce en que al final obtendremos reglas de asociación clásicas.

- Las relaciones de similitud pueden resultar útiles cuando el dominio de un atributo toma valores cuyas semánticas se solapan. Consideremos el atributo *Color del pelo* y supongamos que en la base de datos hallamos valores tales como *“rubio”*, *“amarillo”*, *“claro”*, *“rojo”*, *“anaranjado”*, etc. (una posible causa estaría en que diferentes usuarios hubieran introducido información en la base de datos sin establecer un criterio a la hora de asentar el conjunto de valores que puede tomar el atributo). Si queremos asociar el color del pelo con otros atributos, podría interesarnos tener en

cuenta que *"amarillo"* y *"rubio"* son parecidos hasta cierto punto, entre otros casos. Esto puede llevarse a cabo definiendo una relación de similitud adecuada sobre el dominio del atributo. De esta forma, es de esperar que las reglas en las que intervenga este atributo reflejen en mayor medida las posibles relaciones entre el color del pelo y otras características.

- Otra herramienta bastante estudiada en minería de datos son las dependencias aproximadas [9, 22, 28]. Una dependencia aproximada se puede definir informalmente como una dependencia funcional que admite excepciones. El modelo (o *"teoría"* [28, 7]) de una dependencia aproximada se define como el conjunto de reglas de asociación que describen las relaciones entre los valores del antecedente y el consecuente de la dependencia. En [5] la definición de dependencia aproximada se extiende para poder trabajar con bases de datos relacionales difusas. Aplicando la definición aquí descrita de reglas de asociación difusas, una línea de investigación futura será la de estudiar el modelo de una dependencia aproximada difusa.

4 CONCLUSIONES Y LÍNEAS FUTURAS

En el modelado de un problema real resulta interesante e incluso conveniente la consideración de cierta imprecisión entre los datos, expresada mediante relaciones semánticas. De cara a extraer conocimiento útil a partir de dicha información, es necesario desarrollar nuevas herramientas o extender las ya existentes.

En este trabajo hemos presentado una definición de reglas de asociación difusas en bases de datos relacionales sobre la que hay definidas relaciones de similitud entre valores. El presente resultado cubre un área potencialmente interesante sobre la cual cabe plantearse algunas líneas futuras.

La primera propuesta es la definición del modelo de una dependencia aproximada difusa. En particular, algunos resultados sobre problemas reales, aparecidos en [6, 10], servirán como base sobre la que plantear nuestra metodología.

En un contexto más general, y en ausencia de expertos en un dominio concreto, la definición automatizada de grados de similitud entre valores también se presenta como una idea interesante que será objeto de estudio en un próximo trabajo.

Agradecimientos

Trabajo subvencionado por el proyecto de investigación Fuzzy-KIM CICYT TIC2002-04021-C02-02

Referencias

[1] R. Agrawal, T. Imielinski, A. Swami. *Mining Association Rules between Sets of Items in Large Databases*. Proc. of the 1993 ACM SIGMOD Conf., Washington DC, USA, 1993.

[2] W.H. Au & K.C.C. Chan. *An effective algorithm for discovering fuzzy rules in relational databases*. In *Proc. of IEEE International Conference on Fuzzy Systems*, vol II, 1314–1319, 1998.

[3] F. Berzal, I. Blanco, D. Sánchez, and M.A. Vila. *A new framework to assess association rules*. In Hoffmann, F., editor, *Advances in Intelligent Data Analysis. Fourth International Symposium, IDA'01. Lecture Notes in Computer Science 2189*, pages 95–104. Springer-Verlag, 2001.

[4] F. Berzal, I. Blanco, D. Sánchez, and M.A. Vila. *Measuring the Accuracy and Interest of Association rules: A New Framework*. Intelligent Data Analysis 6 pp. 221–235, 2002.

[5] F. Berzal, I. Blanco, D. Sánchez, J.M. Serrano, and M.A. Vila. *A definition for fuzzy approximate dependencies*. Aceptado para su publicación en Fuzzy Sets and Systems.

[6] F. Berzal, J.C. Cubero, D. Sánchez, J.M. Serrano, M.A. Vila. *Finding Fuzzy Approximate Dependencies within STULONG Data*. Proceedings of the ECML/PKDD 2003, Workshop on Discovery Challenge. Pp 34–46. 22–26 September 2003.

[7] I. Blanco, M.J. Martín-Bautista, D. Sánchez, M.A. Vila. *On the support of dependencies in relational databases: Strong approximate dependencies*. Data Mining and Knowledge Discovery. Sometido.

[8] P. Bosc, D. Dubois, O. Pivert, H. Prade. *On fuzzy association rules based on fuzzy cardinalities*. FUZZY-IEEE, 2001.

[9] P.D. Bra & J. Paredaens. *Horizontal decompositions for handling exceptions to functional dependencies*. Advances in Database Theory, 2:123–144, 1983.

[10] J. Calero, G. Delgado, M. Sánchez-Marañón, D. Sánchez, J.M. Serrano, M.A. Vila. *An Experience in Management of Imprecise Soil Databases by Means of Fuzzy Association Rules and Fuzzy Approximate Dependencies*. Aceptado para su presentación en ICEIS 2004, 14-17 Abril 2004, Oporto (Portugal).

[11] K.C.C. Chan, W.-H. Au. *Mining Fuzzy Association Rules*. CIKM 1997: 209–215, 1997.

[12] G. Chen, Q. Wei. *Fuzzy association rules and the extended mining algorithms*. Information Sciences 147(1-4): 201–228, 2002.

[13] J.C. Cubero, J.M. Medina, O. Pons, M.A. Vila. *The generalized selection: An alternative way for the quotient operations in fuzzy relational databases. Fuzzy Logic and Soft Computing*, B. Bouchon-Meunier, R. Yager, and L.A. Zadeh, Eds. World Scientific Press, 1995.

[14] M. Delgado, N. Marín, D. Sánchez, M.A. Vila. *Mining Fuzzy Association Rules: An Overview. 2003 BISC International Workshop on Soft Computing for Internet and Bioinformatics"*, accepted. 2003.

[15] M. Delgado, N. Marín, D. Sánchez, M.A. Vila. *Fuzzy Association Rules: General Model and Applications*. IEEE Transactions on Fuzzy Systems 11(2), pp. 214–225, 2003.

[16] M. Delgado, D. Sánchez, M.A. Vila. *Fuzzy cardinality based evaluation of quantified sentences*. International Journal of Approximate Reasoning, vol. 23, pp. 23-66, 2000.

[17] D. Dubois, E. Hüllermeier, H. Prade. *A Note on Quality Measures for Fuzzy Association Rules*. Fuzzy Sets and Systems - IFSA 2003, LNAI 2715, 346–353, 2003.

[18] J.M. de Graaf, W.A. Kosters, J.J.W. Witteman. *Interesting Fuzzy Association Rules in Quantitative Databases*. PKDD 2001: 140–151, 2001.

[19] A. Gyenesei. *Interestingness Measures for Fuzzy Association Rules*. PKDD 2001: 152–164, 2001.

[20] E. Hüllermeier. *Implication-Based Fuzzy Association Rules*. PKDD 2001: 241–252, 2001.

[21] M. Kaya, R. Alhajj, F. Polat, A. Arslan. *Efficient Automated Mining of Fuzzy Association Rules*. DEXA 2002: 133–142, 2002.

[22] J. Kivinen, H. Mannila. *Approximate Dependency Inference from Relations*. Theoretical Computer Science 149(1), pp. 129–149, 1995.

[23] C.-M. Kuok, A. Fu, & M. H. Wong. *Mining fuzzy association rules in databases*, *SIGMOD Record 27(1)*:41–46, 1998.

[24] T.P. Hong, C.S. Kuo, & S.C. Chi. *Mining association rules from quantitative data*, *Intelligent Data Analysis 3*:363–376, 1999.

[25] J.H. Lee & H.L. Kwang. *An extension of association rules using fuzzy sets*. In *Proc. of IFSA'97*, Prague, Czech Republic, 1997.

[26] J. Luo, S. Bridges. *Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection*. International Journal of Intelligent Systems 15(8): 687–703, 2000.

[27] W. Pedrycz. *Fuzzy Set Technology in Knowledge Discovery*. Fuzzy Sets and Systems 98, pp.279–290, 1998.

[28] B. Pfahringer & S. Kramer. *Compression-based evaluation of partial determinations*. In *Proc. First Int'l Conf. Knowledge Discovery and Data Mining (KDD'95)*, pages 234–239, 1995.

[29] E. Shortliffe & B. Buchanan. *A model of inexact reasoning in medicine*. Mathematical Biosciences, 23:351–379, 1975.

[30] R. Srikant, R. Agrawal. *Mining Generalized Association Rules*. Procs. of the 21st Int'l Conference on Very Large Databases. Zurich, Suiza, Septiembre 1995.

[31] R. Srikant, R. Agrawal. *Mining Quantitative Association Rules in Large Relational Tables*. Procs. of the ACM SIGMOD Conference on Management of Data, 1996.

[32] S.-L. Wang,J.-S. Tsai, B.-C. Chien. *Mining Approximate Dependencies Using Partitions on Similarity-relation-based Fuzzy Databases*. Proceedings of the IEEE SMC'99, October 1999, Tokyo, Japan, V-871–V-875.

[33] H.-T. Yang, S.-J. Lee. *Mining Fuzzy Association Rules from Sequence databases with Quantitative data and Inter-Transaction Intervals*. FSKD 2002: 606–610.

[34] L.A. Zadeh. *Similarity Relations and Fuzzy Orderings*. Information Sciences, 3:177–200, 1971.

[35] L.A. Zadeh. *A computational approach to fuzzy quantifiers in natural languages*. Computing and Mathematics with Applications, vol. 9, no. 1, pp. 149–184, 1983.

[36] Z. Zhang, Y. Lu, B. Zhang. *An Effective Partitioning-Combining Algorithm for Discovering Quantitative Association Rules*. National Natural Science Foundation of China, 1997.

SERVIDOR DE BASES DE DATOS RELACIONALES DIFUSAS PARA DEDUCCIÓN Y MINERÍA DE DATOS

Ignacio J. Blanco, Carmen Martínez-Cruz, M.Amparo Vila
Dept. Ciencias de la Computación e Inteligencia Artificial
Universidad de Granada.
C/ Periodista Daniel Saucedo Aranda, S/N,18071,
Granada, ESPAÑA
iblanco, cmcruz*, vila@decsai.ugr.es

Jose M. Serrano
Dept. Computer Science
E. U. P. Linares. University of Jaen
C/ Alfonso X El Sabio 28
23700, Linares, Jaen, ESPAÑA
jschica@ujaen.es

Resumen

Este artículo propone una arquitectura e implementación para un servidor relacional multipropósito. Esta arquitectura permite realizar consultas simples, difusas, lógicas, o de minería de datos sobre datos clásicos o difusos. El propósito de esta integración es el poder combinar las diferentes operaciones, proporcionadas por los diferentes servidores ya implementados, permitiendo así tanto las ejecución de consultas complejas, como el almacenamiento de nuevo conocimiento de forma automática sobre la base de datos. Para llevar a cabo esta propuesta es necesaria una combinación de las bases de metaconocimiento existentes y la definición de nuevos elementos del catálogo. Además se introduce un lenguaje para manipular todos estos datos de forma coherente y uniforme sobre la base de las sentencias SQL clásicas.

Palabras Clave: Minería de Datos, Bases de Datos Relacionales Difusas, Bases de Datos Lógicas, Datos Difusos, Clasificación, Clustering, Dependencias Funcionales Difusas, Arquitectura de Bases de Datos.

1. INTRODUCCIÓN

Existen muchas extensiones al Modelo Relacional desde que fue expuesto por Codd en [7]. Muchas de estas extensiones se centran en la integración de la Lógica Difusa al modelo, con el objeto de poder representar valores imprecisos y flexibles [5, 13, 11]. Otras extensiones estan basadas en la representación lógica de una Base de Datos Relacional [9] y la posibilidad de realizar deducciones usando mecanismos de inferencia clásicos. La consecuencia fue la combinación de ambas extensiones en un nuevo modelo, usado en [1, 2].

Actualmente existe una nueva tendencia, denominada Segunda generación de DBMining, que propone un Sistema de Minería de Bases de Datos y Descubrimiento de Información(KDDB System). Esta nueva tendencia, planteada por Imielinski [10] fusiona dos disciplinas, los procesos de minería de datos con las Bases de Datos. Basado en esto, Carrasco [6] propone integrar operaciones de minería de datos con una extensión difusa sobre los modelos relacionales de Bases de Datos.

El objetivo de este artículo es combinar la gestión de minería de datos y la representación de información imprecisa, y lógica, para aumentar el potencial de las consultas sobre una base de datos. Para ello se propone una extensión del SGBDR que permita la inclusión de las nuevas arquitecturas y aumente la escalabilidad del sistema.

Todas las arquitecturas e implementaciones mostradas en este artículo se basan el un modelo concreto de Base de Datos Relacional Difusa llamado GEFRED [13]. Sin embargo, cada una ha sido desarrollada para realizar ciertas funciones concretas (difusas, lógicas o de minería de datos). Con la nueva arquitectura se pretende aumentar el poder del modelo relacional, de forma que se puedan ejecutar consultas complejas mediante la combinación de las operaciones implementadas en el sistema, como pueda ser el combinar procesos de Minería de Datos(DM) sobre relaciones definidas sobre Reglas Lógicas.

La sección 2 va a repasar las arquitecturas anteriormente mencionadas para en la sección 3 describir la arquitectura propuesta que integra a todas, mediante la creación de un servidor único que gestiona el flujo de información. La sección 4 propone un ejemplo de combinación de información entre arquitecturas y la sección 5 expone las conclusiones y trabajos futuros que se extraen del trabajo.

2. NOCIONES PREVIAS

Las arquitecturas que se han desarrollado y sobre las que se va a realizar el proceso de unificación son las siguientes:

2.1. FIRST

Esta arquitectura, basada en el modelo teórico GEFRED expuesto por Medina en [13], permite la representación y manipulación de datos difusos. Esto es posible gracias a la definición de una serie de tipos de datos adicionales sobre el catálogo del SGBDR y un conjunto de operaciones.
Los tipos de datos difusos definidos son:

- *Tipo difuso 1*, representa datos clásicos que pueden ser consultados de forma difusa.
- *Tipo difuso 2*, representa datos imprecisos pertenecientes a un dominio difuso construido sobre un referecial ordenado y que pueden ser consultados de forma imprecisa.
- *Tipo difuso 3*, representa datos imprecisos pertenecientes a un dominio difuso construido sobre un referencial discreto no ordenado.

Al catálogo del sistema se añaden nuevas relaciones que permiten la interpretación de los tipos difusos. La extensión del catálogo se denomina FMB (*Base de Metaconocimiento Difuso*), y esta formado por las siguientes relaciones:

- *fuzzy_col_list*: contiene la lista de atributos de la BD que son susceptibles de tratamiento difuso.
- *fuzzy_object_list*: almacena los objetos difusos que pertenecen al dominio del atributo en cuestion.
- *fuzzy_label_def*: contiene información sobre las distribuciones de posibilidad trapezoidales que se asocian a etiquetas lingüísticas.
- *fuzzy_aprox_much*: contiene parámetros usados para la comparación difusa de valores.
- *fuzzy_nearness_def*: contiene los valores de similitud entre cada par de valores de los tipos difusos 3.
- *fuzzy_compatible_col*: contiene atributos cuyo dominio es igual que otro atributo difuso de la base de datos.
- *fuzzy_qualifiers_def*: almacena el umbral mínimo que se le asigna a cada cualificador, definido sobre una etiqueta lingüística.

Para poder gestionar estos datos difusos se ha llevado a cabo una extensión del SQL, que se ha denominado FSQL(*Fuzzy SQL*), realizado por Galindo[8] . Por un lado, se ha generado una ampliación del DDL (*Lenguaje de Definicion de Datos*), que permite crear, borrar y gestionar las nuevas estructuras de datos. Por otro lado, la ampliación del DML (*Lenguaje de Manipulación de Datos*), permite manipular los datos que se quieran almacenar en estas relaciones. El proceso de consulta de datos, ahora, se realiza mediante la utilización de la clásica sentencia *SELECT* ahora ampliada, para trabajar con los nuevos tipos de datos.

2.2. FREDDI Extendido

Esta arquitectura tal y como se explica en [12, 2] permite almacenar y ejecutar reglas lógicas en un SGBDR [9, 14]. Para llevar esto a cabo, es necesario una nueva ampliación del catálogo del sistema, que se va a denominar RB (*Base de Reglas*) y esta formado por las siguientes relaciones:

- *ded_intensional_catalog*: almacena un índice de todas las relaciones intensivas de la base de datos.
- *intensional_table_descripcion*: almacena los predicados intensivos definidos como una disyunción de reglas.
- *rule_description*: describe las reglas como una secuencia de predicados y comparaciones.
- *predicate_description*: describe el orden de las variables en cada uno de los predicados.
- *condition_description*: describe la estructura de cada predicado (variable, operadores de comparación, y umbrales).

La gestión de las relaciones extensivas, intensivas, y de las reglas generalizadas se realiza mediante la extensión del lenguaje DDL realizado por Blanco [3]. El calculo de una instancia a partir de una relacion intensiva puede realizarse mediante la implementación de los algoritmos básicos de Prolog, o de Datalog, expuesto por Blanco en [4]

2.3. FIRST* y DmFIRST

Estas arquitecturas, amplian a la anteriormente definida (FIRST) para permitir representar y manipular múltiples tipos de datos (definida por Carrasco en[6]). La finalidad de FIRST* es la de permitir el tratamiento difuso de la diversidad de dominios, que es susceptible de ser tratado por un sistema de Minería de Datos. DmFIRST, es otra arquitectura, definida sobre la base de FIRST*, que propone la integración en el SGBDR

de las siguientes técnicas de DM: Clustering, Caracterirización, Clasificación difusa, Dependencias difusas entre atributos.

En FIRST* se define un nuevo tipo de datos,

Tipo difuso 4 que consiste en una parte de almacenamiento de información acerca del valor del dato y otra de almacenamiento de meta-datos, información acerca del tipo de valor que se almacena.

La extensión del catálogo del sistema para la definición y manipulación del tipo de datos 4 se ha denominado FMB*(Base de Metaconocimiento Difuso*) y esta formado por las siguientes relaciones:

- *dmfsql_label_definition*: almacena información acerca de las etiquetas lingüísticas definidas para el tipo difuso 4.
- dmfsql_col_col: alacena información acerca de todos los atributos capaces de contener o manejar el tipo difuso 4.
- *dmfsql_funcitions*: almacena la referencia a las funciones que definen los comparadores difusos de los atributos de tipo 4, y sus funciones de representación.
- *dmfsql_fuctions_col*: contiene la definición para cada atributo de tipo 4 y para cada comparador su función asociada.
- *dmfsql_col_par*: contiene la interfaz de llamada a las funciones que implementan los comparadores para los tipos difusos 4.

La extensión del catálogo del sistema para la realización de las operaciones de minería de datos sobre el SGBD se ha denominado DmFMB (Base de Metaconocimiento para Minería de Datos), se presenta a continuación:

- *dmfsql_project*: almacenan información general acerca del proceso de minería de datos, como el nombre del proyecto, la tabla origen de datos, es estado del proceso de clustering, etc.
- *dmfsql_col_list*: almacen la información acerca de los atributos de la tabla de origen que son significativos para el proceso de minería de datos, como el tipo de operación, la relación creada para el almacenamiento de resultados, etc.

También se ha creado una nueva extensión del lenguaje FSQL, con el objetivo de gestiónar todos los conceptos y relaciones definidos en esta sección, esto es, *tipo difuso 4*. El DmFSQL incluye funciones para la definición de datos (*DDL*) como *CREATE_MINING*, *REVOKE_MINING*, y modifica la sentencia *SELECT* de manipulación de datos (*DML*) para permitir la realización de consultas que impliquen uno o varios de los procesos de DM, expuestos anteriormente.

3. INTEGRACIÓN DE LAS ARQUITECTURAS

3.1. PROCESO ACTUAL

Sobre la implementación de la arquitectura FIRST que permite la representación y manipulación de datos difusos, se han desarrollado por separado las arquitecturas FREDDI extendido y DmFIRST. Ambas hacen uso tanto, de la base de metaconomiento implementada por FIRST (FMB), como del servidor cuando trabajan con información imprecisa, obviando el uso de la misma, cuando trabajan con datos clásicos. En la figura 1, se puede observar como fluye la información a través de los diferentes módulos cuando se realiza una consulta en cada una de las arquitecturas:

1. Se lanza la consulta desde un cliente hacia el servidor correspondiente. El tipo de consultas que se pueden realizar son las siguientes:
   a Consulta de Minería de Datos, puede consultar por datos clásicos o difusos.
   b Consulta de datos difusos.
   c Consulta para la deducción de datos puede consultar por datos clásicos o difusos.
2. La base de datos le proporciona información a los diferentes servidores, acerca de la estructura de los datos que estan utilizando a través del catálogo del sistema.
3. Es la información que cada servidor le transfiere a la Base de datos, p.ej. *DMFIRST* almacena la tablas intermedias y *FREDDI Extendido* genera tablas temporales.
4. Consulta en SQL cásico que el servidor correspondiente ha generado.
5. Se lanza ejecutor de consultas la sentencia SQL.
6. El resultado de la consulta es devuelto a cada cliente que debe formatearlos.

7,8 Representa el intercambio de información entre el servidor FSQL con el resto. Esto ocurre cuando la consulta utiliza datos difusos.

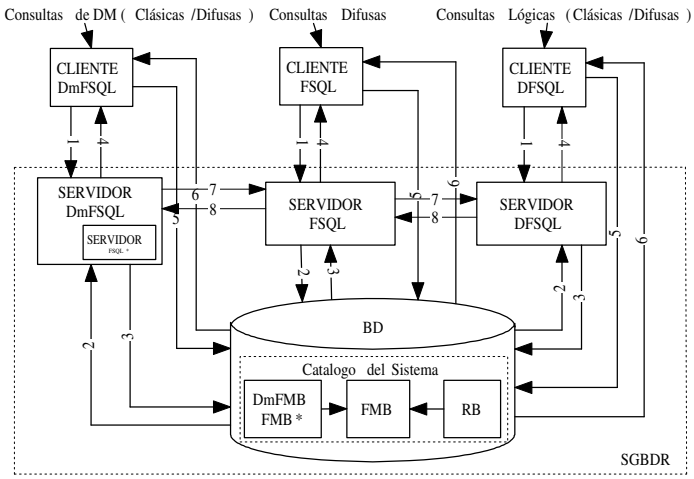


Figura 1: Arquitectura de un SGBD con servidores independiendes.

3.2. ARQUITECTURA PROPUESTA

Se propone una arquitectura única que permita realizar las operaciones de cada una de las arquitecturas expuestas, y las operaciones clásicas del SGBD, sin necesidad de utilizar diferentes clientes. Para ello se desarrolla de un servidor, capaz de distinguir el tipo de consulta que se realiza y lanze la operación al correspondiente subservidor, que actuará sobre la base de datos y devolverá los resultados obtenidos al servidor general, que decide lo que hacer con ellos. En la figura 2 se muestra como se comporta flujo de la información a través del servidor:

a Se lanza la consulta sobre el cliente.

b El cliente envia la consulta al servidor, que decide a que módulo debe enviarla.

c En cada servidor la consulta es analizada y ejecutado el proceso de conversión al SQL.

d La BD proporciona a los servidores información acerca de las estructuras de datos que pueden manipular.

e,f Los servidores intercambian información con el FSQL cuando las consultas incluyen información imprecisa.

g Los servidores envian información a la base de datos para ser almacenada o actualizada.

h Los servidores, (o el SGBD en caso de consultas clásicas) envian a los clientes el resultado de la consulta lanzada.

i El servidor envia al cliente el resultado de la consulta que debe formatearlos.

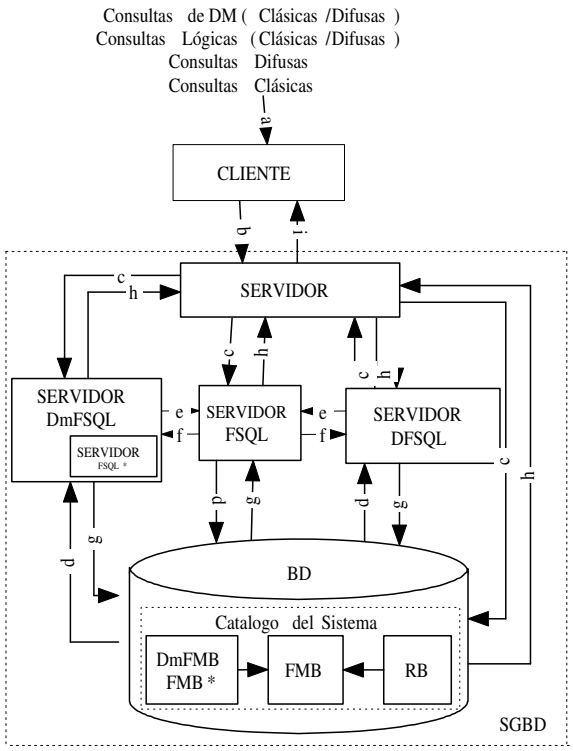


Figura 2: Arquitectura de un SGBD con un único servidor.

Las ventajas de tener un servidor centralizado, son entre otras:

- la posibilidad de combinar operaciones en una misma consulta.
- la capacidad de mantener un lenguaje de consulta unificado.
- La escalabilidad del sistema, que permite la sencilla incorporación de nuevos servidores, o nuevas operaciones a los servidores ya existentes, de forma transparente para el usuario.

3.3. IMPLEMENTACIÓN

Para llevar a cabo esta unificación, hay que realizar una serie de cambios en el sistema actual:

3.3.1. El catálogo del sistema

El catálogo del sistema mantendrá toda la información acerca de la estructura de los datos y las operaciones que se puedan realizar sobre ellos, de forma centralizada, evitando en la medida de lo posible, redundancias

e inconsistencias. Para ello, se propone una nueva extensión del catálogo, llamada MB (Base de Metaconocimiento), que contendrá dos relaciones adicionales a las descritas en las secciones anteriores. Las relaciones son:

- *extended_tab_columns*: proporciona información acerca de las todas las columnas que pueden ser accedidas por el usuario (todos los atributos clasicos almacenados en la relación *all_tab_columns* están incluidos en esta tabla).

- *extended_tables*: almacena todas las relaciones definidas en la base de datos. Aquellas relaciones referentes a tablas almacenadas en *all_objects* están incluidas en esta relación.

3.3.2. El lenguaje de consulta

El lenguaje de consulta, consistirá en la unificación de los lenguajes FSQL, DFSQL y DMFSQL, añadiendo las sentencias de DDL y DML que, por ejemplo, en el caso del DMFSQL existen sin acabar, y nuevas sentencias que han surjido, gracias a las cuales se podrán hacer consultas complejas.

Las consultas lanzadas por el cliente, son enviadas a través del servidor general, al analizador semántico correspondiente, único para cada servidor que traducira la sentencia en el lenguaje SQL clásico. Los analizadores sintáxtico y léxico, por contra, serán únicos para todas las consultas que se deban de realizar sobre la base de datos. La incorporación de nuevas intrucciones, es inmediata mediante la inserción de los nuevos tokens, en un archivo.

4. EJEMPLO DE INTERCONEXION DE ARQUITECTURAS

La unión de las arquitecturas, establece el soporte para la combinación de operaciones que se realizan de manera independiende o para el almacenamiento de resultados parciales, que posteriormente podran ser explotados por otros procesos implementados sobre la base de datos.

El ejemplo que se propone es el de la interconexión de operaciones correspondientes al proceso de minería de datos y de tratamiento de reglas lógicas. En concreto, la búsqueda de Dependencias Funcionales Difusas(DFD) y de las Dependencias Graduales Difusas(DGD), genera (en el caso en que se encuentren) reglas que puede ser almacenadas en la Base de Datos como reglas con grado de cumplimiento.

Problema planteado. Queremos comprobar si se cumple la siguiente Dependencia Funcional Difusa:

Cuadro 1: Relacion NOTASAL.

Alumno	**Calificacion**	**Comportamiento**
Juan	8	bueno
Isabel	10	muy_bueno
Marta	4	malo
Antonio	6	regular
Carlos	2	malo
...	...	...

0.6 - 0.7 DFD CALIFICACION $_{FEQ*FEQ} \longrightarrow$ COMPORTAMIENTO con confianza c y soporte s

Obviamente, lo que pretendemos es averiguar si la nota implica el comportamiento de los alumnos, donde la calificación de los alumnos tenga un grado de cumplimiento de 0.6 y el comportamiento tenga un grado de 0.7.

Para ello se lanza el proceso de DM sobre la tabla 1 con esta sentencia:

```
SELECT_MINING FGLOBAL_DEPENDENCIES ALUMNOS_PRJ
  USING T_NORM THOLD_ANT_CON
```

esta sentencia es aproximadamente equivalente a:

```
SELECT COUNT(*) FROM NOTASAL A1, NOTASAL A2
WHERE(A1.ALUMNO<>A2.ALUMNO) AND
   (A1. CALIFICACION FEQ A2.CALIFICACION
   THOLD 0.6)   AND NOT
   (A1.COMPORTAMIENTO NFEQ A2.COMPORTAMIENTO
   THOLD 0.7)
```

El cálculo del soporte y confianza de la DFD se realiza con una sentencia similar, contando el numero de ocurrencias del antecedente y el consecuente. Si el soporte y confianza son lo suficientemente altos, la DFD se acepta, y podrá ser almacenada en la BD.

A continuación se propone, una forma de cómo se podría almacenar la información sobre esta DFD, en forma de regla:

DFD(CALIFICACION,COMPORTAMIENTO) :-
NOTASAL(X,Y) $\wedge$ (X $=_{0,6}$ CALIFICACION)$\wedge$
(Y $=_{0,7}$ COMPORTAMIENTO)

donde al incluir una nueva tupla a la tabla NOTASAL, primero se compará si se cumplen los umbrales para cada uno de los atributos (antecedente y consecuente de la regla). Si esto ocurre, la nueva tupla será insertada verificando la DFD y aumentando asi la confianza y el soporte de la misma.

No obstante quedan por plantear las operaciones que permitirán llevar a cabo las nuevas consultas que se

plantean tras la unificación de las arquitecturas, para sacar el máximo rendimiento al sistema. Algunos ejemplos de estas nuevas operaciones que surjen son: la creación de las reglas, la comprobación de la integridad de mismas en las actualizaciones, que un proceso de minería de datos pueda ejecutarse sobre una tabla intensiva ...

5. CONCLUSIONES y TRABAJOS FUTUROS

En este artículo se propone la forma de conexión de diferentes servidores implementados de forma independiente, con objetivos o funciones diferentes, para formar uno sólo. Dicho servidor unificado permitirá la combinación de las operaciones y tipos de datos multiplicando así la funcionalidad del sistema. De esta forma, los resultados de los diferentes servidores pueden ser combinados, manipulados, consultados y almacenados sobre cualquiera de las nuevas estructuras de datos, formando nuevas consultas.

Para llevar esto a cabo, se ha propuesto una simple ampliación del catálogo del sistema, que permitirá incorporar nuevos servidores a la arquitectura unificada, integrando sus datos y operaciónes de forma cómoda y sencilla, es decir, haciéndolo más escalable.

Además la sintáxis del FSQL, DFSQL ha sido diseñada en la filosofía del SQL, pero el lenguaje incorporado por DmFSQL debe ser unificado con dicha filosofía y algunas sentencias para la gestión de los tipos de datos, deberán ser completadas.

Esta arquitectura abre la puerta para añadir, de forma inmediata nuevas operaciones relacionadas con la minería de datos, e integrarlas de la misma manera que lo están en el servidor DmFSQL, como puedan ser: reglas de asociación difusas, análisis evolutivos...

Agradecimientos

Esta investigación esta financiada por los proyectos TIC2002-04021-C02-02 and TIC2002-00480 del Ministerio de Ciencia y Tecnología.

Referencias

[1] J. F. Baldwin and S. Q. Zhou. A fuzzy relational inference language. *Fuzzy Sets and Systems*, (14):155–174, 1984.

[2] I. Blanco. *Deducción en Bases de Datos Relationales Difusas*. PhD thesis, E. T. S. I. Informática, Universidad de Granada, Spain, 2001.

[3] I. Blanco, J. C. Cubero, O. Pons, and M. A. Vila. An implementation for fuzzy relational databases. In G. Bordogna and G. Passi, editors, *Recent Research Issues on the Management of Fuzziness in Databases*, Studies in Fuzziness and Soft Computing, pages 183–207. Physica-Verlag, 2000.

[4] I. Blanco, O. Pons, J. M. Serrano, and M. A. Vila. Deduction in a gefred database using datalog. In *International Conference in Fuzzy Logic and Technology EUSFLAT 2003*, pages 550–553, September 2003.

[5] P. Bosc, M. Galibourg, and G. Hamon. Fuzzy querying with sql: Extensions and implementation aspects. *Fuzzy Sets and Systems*, 28:333–349, 1988.

[6] Ramón A. Carrasco. *Lenguajes e Interfaces de Alto Nivel para Data Mining con Aplicación Práctica a Entornos Financieros*. PhD thesis, E. T. S. I. Informática, Universidad de Granada, Spain, 2003.

[7] E. F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 1970.

[8] J. Galindo. *Tratamiento de la Imprecisión en Bases de Datos Relacionales: Extensión del modelo y adaptación de los SGBD actuales*. PhD thesis, Department of Computer Science and Artificial Intelligence, University of Granada, España, 1999.

[9] H. Gallaire, J. Minker, and J. M. Nicholas. Logic and databases: A deductive approach. *Computing Surveys*, 16(2):153–185, June 1984.

[10] T. Imielinski and Heikki Mannila. A database perspective on knowledge discovery. *Communications of the ACM*, 39(11):58–64, 1996.

[11] J. Kacprzyk and S. Zadrozny. Sqlf and fquery for access. In *Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, volume 4, pages 2464–2469, 2001.

[12] J. M. Medina, O. Pons, J. C. Cubero, and M. A. Vila. Freddi: A fuzzy relational deductive database interface. *International Journal of Intelligent Systems*, 12:597–613, 1997.

[13] J. M. Medina, O. Pons, and M. A. Vila. Gefred. a generalized model of fuzzy relational databases. *Information Sciences*, 1994.

[14] M. A. Vila, J. C. Cubero, J. M. Medina, and O. Pons. A logic approach to fuzzy relational databases. *International Journal of Intelligent Systems*, 9:449–460, 1994.

Relaciones intensivo-extensivas: una herramienta para la deducción y la integridad en Bases de Datos Relacionales

Ignacio J. Blanco[*], Carmen Martínez, Olga Pons y M. Amparo Vila
Depto. Ciencias de la Computación e Inteligencia Artificial
C/ Periodista Daniel Saucedo Aranda S/N
18071, Universidad de Granada
Granada, ESPAÑA
e-mail: {iblanco,cmcruz,pons,vila}@decsai.ugr.es

Resumen

En el presente trabajo, proponemos una definición para el concepto de relación intensivo-extensiva, cuyo contenido puede estar explícitamente e implícitamente almacenado dentro de una base de datos relacional, y se justifica la necesidad de dicho concepto. Además, se proponen dos alternativas para la implementación del concepto definido y algunas aplicaciones de el mismo en el ámbito de la normalización y de la deducción.

Palabras clave: bases de datos, modelo relacional, deducción, normalización, dependencias

1 Introducción

A partir de la conexión entre el Modelo Relacional de Bases de Datos [4] y el lenguaje de la Lógica, introducida por Gallaire et al. [5], Blanco et al. [2] definieron los conceptos de relación extensiva y relación intensiva para representar aquellas situaciones en las que el contenido de una relación puede ser explícitamente almacenado en una base de datos relacional o deducido a partir del contenido de otras relaciones, respectivamente.

En un paso posterior, los mismos autores [3] tratan de adaptar los conceptos de la extensión deductiva del Modelo Relacional clásico de modo que puedan ser aplicados a una extensión difusa concreta de dicho modelo llamada *GEFRED* [7], definiendo los conceptos de relaciones extensiva e intensiva difusas.

En este trabajo, proponemos la combinación de los conceptos de relación extensiva y relación intensiva, ya sean clásicas o difusas, en un nuevo concepto que permita almacenar simultáneamente ambos tipos de información (explícita y deducida) para representar nuevas situaciones acerca del conocimiento.

En el siguiente apartado, repasaremos las definiciones básicas para llevar a cabo nuestro cometido. En el Apartado 3 propondremos dos definiciones alternativas para el concepto de relación intensivo-extensiva. Posteriormente, en el Apartado 4, propondremos posibles aplicaciones de este tipo de relaciones para concluir en el Apartado 5.

2 Definiciones previas

El concepto central del Modelo Relacional es el de *relación*, que se ve a continuación.

Definición 1 *Una* **relación** *es un par* (R, r)*, donde*

- R *es un conjunto de atributos conocido como* **esquema**, *y*
- $r \subseteq D(R_1) \times D(R_2) \times \ldots \times D(R_n)$ *es un conjunto de n-uplas, llamado* **instancia**, *donde n es la cardinalidad del esquema, y* $D(R_i), i \in \{1, \ldots, n\}$ *es el dominio de discurso del atributo i-ésimo del esquema.*

Gallaire et al.[5] se establece que una relación clásica del Modelo Relacional puede representarse mediante un predicado $R_E(R_1, R_2, \ldots, R_n)$ en la Lógica de Primer Orden, el cual tiene la misma estructura que el esquema de la relación. Cada una de las n-uplas de la instancia constituyen, a su vez, un hecho del predicado R_E. Considerando la definición clásica de relación y su correspondencia con un predicado de la Lógica, Blanco et al. [2] definen una *relación extensiva* como una relación en el sentido clásico del Modelo Relacional y un *predicado extensivo* como el predicado asociado a dicha relación.

En contraposición al concepto de relación extensiva, Blanco et al. [2] definen el concepto de *relación inten-*

[*]Autor para correspondencia.

siva como aquella relación cuyo contenido es calculado a partir del contenido explícito de la base de datos (es decir, de las relaciones extensivas), de otras relaciones intensivas o de la combinación de ambas.

Definición 2 *Se define una relación intensiva como un par* (R, I) *donde:*

- R *constituye el esquema de la relación desde el punto de vista clásico, e*
- I *es un conjunto de fórmulas para el cálculo de la instancia de la relación, conocido como generador de instancia.*

De este modo, la relación intensiva constituye una relación cuya instancia es un conjunto de n-uplas generado por definición.

Al igual que ocurre con las relaciones extensivas, se define una correspondencia entre la extensión del Modelo Relacional y la Lógica. De esta manera, cualquier relación intensiva $R_I(R_1, R_2, \ldots, R_n)$ tiene asociado un predicado R_I cuya estructura viene determinada por el esquema de la relación asociada, y cuyos hechos han de ser calculados. Este cálculo se obtiene aplicando el conjunto de fórmulas que "definen" la instancia de la relación a los hechos y predicados contenidos en la base de datos mediante un proceso llamado de *deducción.*

Una vez definidos los conceptos extendidos de relación, en Blanco et al. [3], los autores se centra en la definición de "fórmula" para el cálculo de instancias (es decir, de los hechos de un predicado intensivo). La estructura que propone es la de una regla de tipo *Prolog*, en la cual los hechos de un predicado se calculan en base a la existencia o confirmación de una conjunción de hechos que pueden incluir al propio predicado implicado (reglas recursivas). De este modo, la instancia de un predicado se calcula en base a una disyunción de reglas como la que se muestra en la Expresión (1) y donde cada una de las reglas, a su vez, está formada por una conjunción de hechos.

$$\begin{array}{c} P \leftarrow P_1 \vee P_2 \vee \ldots \vee P_m \\ \forall i \in \{1, \ldots, m\}, P_i \leftarrow Q_{i,1} \wedge \ldots \wedge Q_{i,n_m} \end{array} \quad (1)$$

Con posterioridad a la aportación de Gallaire, Medina et al. [7] propusieron una extensión del Modelo Relacional que permitía representar, almacenar y consultar información imprecisa mediante la Teoría de Conjuntos Difusos de Zadeh [9], a la cual llamaron *GEFRED*, y Vila et al.[8] trasladaron la correspondencia planteada por Gallaire entre el Modelo Relacional y la Lógica a la extensión difusa de una base de datos relacional.

Combinando el modelo *GEFRED* con la correspondencia lógica de una base de datos relacional difusa con el modelo lógico, Blanco et al. [3] proponen la definición de *relación extensiva difusa* basada en la definición de *relación generalizada difusa* de Medina.

Definición 3 *Se define una relación extensiva difusa como un par* $(\mathcal{H}, \mathcal{B})$ *donde* $\mathcal{H}$ *es el conjunto llamado cabecera,* $\mathcal{B}$ *es el conjunto llamado cuerpo, y ambos se definen como se muestra en la Expresión (2)*

$$\begin{array}{l} H = \{(A_{G_1} : D_{G_1}[, C_{A_{G_1}}]), \ldots, \\ \quad (A_{G_n} : D_{G_n}[, C_{A_{G_n}}])\} \\ B = \{\{(A_{G1} : \tilde{d}_{i1}[, c_{i1}], \ldots, \\ \quad (A_{Gn} : \tilde{d}_{in}[, c_{in}])\}\} \end{array} \quad (2)$$

Desde un punto de vista más intuitivo, $\mathcal{H}$ es un conjunto de triples atributo-dominio generalizado difuso-grado de compatibilidad, que fijan la estructura de la relación generalizada difusa (es decir, su esquema). Por su parte, $\mathcal{B}$ es el conjunto de n-uplas formadas por n componentes, cada una de las cuales es un triple atributo-valor-grado en el intervalo $[0, 1]$.

Al incrementar el tipo de dominios representables en el Modelo Relacional mediante la extensión difusa del mismo, son necesarios nuevos operadores para comparar valores pertenecientes a estos dominios e, incluso, operadores que permitan la comparación de valores en dominios distintos. Las reglas para el modelo clásico no sirven para este tipo de situaciones, ya que se basan en los operadores de comparación clásicos. Por tanto, antes de definir el concepto de relación intensiva difusa se hace necesario proveer de una definición de regla capaz de trabajar con los nuevos valores y los nuevos operadores de la extensión relacional.

En ese sentido, en Blanco et al. [3], los autores proporcionan los conceptos de *regla generalizada para la deducción con datos difusos* y *regla generalizada difusa con grado de cumplimiento.* En nuestro caso, y para el ámbito de este trabajo, nos centraremos en la definición de la primera de ellas, que se muestra a continuación.

Definición 4 *Sean:*

- P *un predicado n-ario y* $Q_i, i \in \{1, \ldots, m\}$ *predicados* n_i*-arios,*
- $X_i, i \in \{1, \ldots, n\}$ *e* $Y_{j,k}, j \in \{1, \ldots, m\}, k \in \{1, \ldots, n_j\}$ *variables que toman valores en dominios generalizados difusos,*

Se define una **regla generalizada para la deducción con datos difusos***, o simplemente* regla

generalizada, *como una regla con la siguiente estructura:*

$$\begin{array}{l} P(X_1, X_2, \ldots, X_n) : - \\ \quad Q_1(Y_{1,1}, Y_{1,2}, \ldots, Y_{1,n_1}) \wedge \ldots \wedge \\ \quad Q_m(Y_{m,1}, Y_{m,2}, \ldots, Y_{m,n_m}) \wedge \Psi \end{array} \quad (3)$$

donde Ψ *es una fórmula:*

$$\begin{array}{l} \Psi \equiv \bigwedge (X_i =_{\alpha_{i,j,k}} Y_{j,k}) \bigwedge \\ \quad \bigwedge (Y_{j,k} =_{\beta_{j,k,l,p}} Y_{l,p}) \bigwedge \\ \quad \bigwedge \phi_{j,k,l,p}(\Theta^{\theta}_{j,k,l,p}(Y_{j,k}, Y_{l,p}), \gamma_{j,k,l,p}) \end{array} \quad (4)$$

y donde:

- $\bigwedge A_i$ *representa la conjunción* $A_1 \wedge A_2 \wedge \ldots \wedge A_n$,
- *las condiciones* $(X_i =_{\alpha_{i,j,k}} Y_{j,k})$ *representan a los acoplamientos entre variables de la cabeza y el cuerpo de la regla, pudiendo aparecer cualquier variable varias veces,*
- *las condiciones* $(Y_{j,k} =_{\beta_{j,k,l,p}} Y_{l,p})$ *representan a los acoplamientos entre variables del cuerpo de la regla, pudiendo aparecer cualquier variable varias veces,*
- *las expresiones* $\phi_{j,k,l,p}(\Theta^{\theta}_{j,k,l,p}(Y_{j,k}, Y_{l,p}), \gamma_{j,k,l,p})$ *representan a aquellas comparaciones introducidas expresamente por el usuario para restringir la semántica de la regla y en las cuales:*
  - $\phi_{j,k,l,p}$ *es la función que determina si el grado de cumplimiento de la condición sobrepasa un umbral mínimo,*
  - $\Theta^{\theta}_{j,k,l,p}$ *es un comparador generalizado difuso del modelo GEFRED [7] definido sobre el comparador clásico* θ *y*
  - $\gamma_{j,k,l,p}$ *es el grado de mínimo de cumplimiento de la comparación para que se evalúe como verdadera.*
- *cada variable de la cabeza de la regla está relacionada con una o más en el cuerpo de la regla, y*
- *cada variable del cuerpo de la regla está relacionada con, al menos, otra variable del cuerpo o con una o varias de la cabeza de la regla.*

Los autores no incorporan el uso de constantes para las comparaciones o para restringir los resultados de una deducción. No obstante, podemos simular el comportamiento en dicho caso mediante una relación con una única tupla. De este modo, la comparación $(X = a)$, donde a es una variable, puede simularse con $(X = Y) \wedge ConstanteY(Y)$, de modo que $ConstanteY$ es una relación con un único atributo y una única tupla, (a). Esta solución puede antojarse ineficiente, pero permite hacer uso del concepto existente de regla.

Antes de proseguir, es necesario recordar que esta definición generalizada de regla incluye la representación del caso clásico.

Una vez extendido el concepto de regla para el cálculo de una relación intensiva con datos imprecisos, la definición de esta última es inmediata, a partir de la definición en el Modelo Relacional clásico.

Definición 5 *Se define una* **relación intensiva difusa** *como un par* $(\mathcal{H}, \mathcal{I})$ *donde:*

- $\mathcal{H}$ *es un conjunto llamado cabecera, cuya estructura se recoge en la Definición 3 y que representa la estructura de la relación, y*
- $\mathcal{B}$ *es un conjunto de fórmulas del tipo que se muestra en la Definición 4 y que permiten el cálculo de la instancia de la relación.*

Análogamente a la extensión deductiva del Modelo Relacional clásico, cada relación extensiva difusa llevará asociado un predicado con variables que toman valores en dominios difusos. La estructura de este predicado será la determinada por la cabecera de la relación, y se afirmará un hecho del mismo por cada una de las tuplas contenidas en el cuerpo de la relación.

Con respecto a las relaciones intensivas difusas, cada una de ellas llevará asociado un predicado con variables que toman valores en dominios difusos. La estructura del predicado será la que dicte la cabecera de la relación y sus hechos podrán ser calculados por deducción aplicada a los hechos de otros predicados relacionados con la base de datos, siguiendo las pautas marcadas por un conjunto de reglas que definen al predicado intensivo.

3 Relación intensivo-extensiva

Es un hecho conocido que las fuentes de información para las bases de datos son imperfectas (incompletas, imprecisas, etc.), pero es más interesante almacenar alguna información, por imperfecta que sea, que no tener ninguna información. Uno de estos casos es el de las deducciones incompletas por falta de datos.

Nuestra intención es la de definir un nuevo concepto que permita incorporar hechos conocidos al mecanismo de deducción, permitiéndole continuar o expandir dicha deducción de una forma controlada.

Este concepto es el de *relación intensivo-extensiva* y se basa en los conceptos de *relación intensiva* y *relación extensiva*, vistos en el Apartado 2.

A continuación, trataremos de justificar la necesidad de dicho concepto mediante un ejemplo muy simple. Supongamos que se conocen ciertas tuplas, que no pueden ser deducidas de la propia base de datos por falta de conocimiento pero que permitirían efectuar una deducción completa. Como ilustración, emplearemos el clásico ejemplo en la deducción: la relación *antepasado*.

Conocemos una relación *Padre* que tiene la forma mostrada en la Tabla 1 y una relación intensiva *Antepasado*, como se muestra en la Expresión (5), que nos permite construir el árbol genealógico para una serie de personas.

Padre	
Antonio	Pedro
Pedro	Juan
Luis	Manuel

Tabla 1: Relación padre

$$\begin{array}{l} A(X,Y) : -Padre(X,Y) \\ A(X,Y) : -Padre(X,Z) \wedge A(Z,Y) \end{array} \quad (5)$$

El cálculo de la relación $(A, C(I))$ dará como resultado una relación como la que se muestra en la Tabla 2.

Antepasado	
Antonio	Pedro
Pedro	Juan
Antonio	Juan
Luis	Manuel

Tabla 2: Relación antepasado

Supongamos que tenemos datos adicionales, de modo que sabemos que *Juan* no es padre de *Luis*, pero sabemos que *Juan* es antepasado de *Luis*. Este no es un hecho que pueda representarse a través del predicado *Padre* puesto que no representa la relación existente entre las dos entidades, pero tampoco es representable en la relación *Antepasado* porque esta no puede contener hechos.

Si se lograse representar este hecho de alguna forma, la relación resultante sería la que se muestra en la Tabla 3.

Llegados a este punto, nos parece interesante la definición de un nuevo concepto de relación, tanto en la extensión deductiva del Modelo Relacional clásico, como en la del Modelo Relacional Difuso. El concepto al que nos referimos es el de *relación intensivo-extensiva* (ya sea clásica o difusa).

Antepasado	
Antonio	Pedro
Pedro	Juan
Luis	Manuel
Juan	Luis
Antonio	Juan
Pedro	Luis
Juan	Manuel
Antonio	Luis
Pedro	Manuel
Antonio	Manuel

Tabla 3: Relación completa de antepasados

3.1 La relación intensivo-extensiva como la unión de relaciones extensiva e intensiva

La primera alternativa que surge, consiste en reutilizar las definiciones antes expuestas considerando la relación intensivo-extensiva como la unión de una relación intensiva y una extensiva.

Sean (R, r) y (R, I) dos relaciones, extensiva e intensiva, respectivamente, ambas con el mismo esquema. Podemos llamar *relación intensivo-extensiva* a la relación $(R, r \cup C(I))$, donde $C(I)$ es el conjunto de tuplas resultante de aplicar un mecanismo de deducción a la base de datos mediante las reglas contenidas en el generador de instancia I.

Según esta aproximación, los hechos pertenecientes a la "parte extensiva" de la relación intensivo-extensiva no intervendrían en el cálculo de los hechos del predicado sino que serían añadidos *a posteriori*, lo cual no refleja la situación que se muestra en la Tabla 3.

Los hechos "conocidos" del predicado *Antepasado* (parte extensiva de la relación) pueden ser separados en una relación $Antepasado_E$ como se muestra en la Tabla 4.

$Antepasado_E$	
Juan	Luis

Tabla 4: Relación antepasado

Supongamos ahora una modificación de la definición de la relación *Antepasado*, mostrada en la Expresión (5), como la que se muestra en la Expresión (6).

$$\begin{array}{l} A(X,Y) : -Padre(X,Y) \\ A(X,Y) : -Antepasado_E(X,Y) \\ A(X,Y) : -Padre(X,Z) \wedge A(Z,Y) \end{array} \quad (6)$$

Esta situación permite el uso de los hechos de la parte extensiva del predicado para el cálculo de los hechos

de la parte intensiva del mismo. Esta modificación nos permite dar una definición de relación intensivo-extensiva basada en dos relaciones, intensiva y extensiva respectivamente.

Definición 6 *Sean una relación extensiva* (P, p) *y una relación intensiva* (Q, I) *tales que* $P = Q$*. Sean* $P(X_1, \ldots, X_n)$ *y* $Q(X_1, \ldots, X_n)$ *los predicados asociados a las relaciones* P *y* Q*, respectivamente. Se llama* **relación intensivo-extensiva** $Q_{E(P)}$ **basada en P y Q** *a la relación* $(Q_{E(P)}, I')$ *tal que* $Q_{E(P)} = P = Q$ *e* $I' = I \cup \{Q(X_1, \ldots, X_n) : -P(X_1, \ldots, X_n)\}$.

3.2 La relación intensivo-extensiva como un concepto independiente

Sin embargo, entendemos que la situación requiere de una definición independiente de *relación intensivo-extensiva* que se base en las definiciones de relación extensiva y relación intensiva pero que sea una modificación de estas. Es por ello que tratamos de definir un nuevo concepto dentro de los tipos de relaciones.

Definición 7 *Sea* (P, I) *una relación intensiva. Se dice que* (P, I) *es una* **relación intensivo-extensiva** *si* I *contiene reglas de la forma*

$$A(X_1, \ldots, X_n) : -(X_1 = a_1) \wedge \ldots \wedge (X_n = a_n) \tag{7}$$

con $a_i \in D(X_i), \forall i \in \{1, \ldots, n\}$.

Es decir, una relación se dice *intensivo-extensiva* si su generador de instancia contiene hechos para el propio predicado. Dada la forma de definir los hechos en el generador de instancia, estos hechos estarán involucrados en el cálculo de la instancia de la relación intensivo-extensiva.

En el ejemplo visto anteriormente, no es necesario introducir una nueva relación $Antepasado_E$ sino redefinir el generador de instancia para que incorpore el hecho conocido del predicado intensivo, como se muestra en la Expresión (8).

$$\begin{array}{l} A(X, Y) : -Padre(X, Y) \\ A(X, Y) : -(X = \text{“Juan”}) \wedge (Y = \text{“Luis”}) \\ A(X, Y) : -Padre(X, Z) \wedge A(Z, Y) \end{array} \tag{8}$$

4 Aplicaciones

Las últimas tendencias en la *Minería de Datos* (*KDD*, *Knowledge Discovery and Data Mining*) consisten en realizar lo que se conoce como *KDDB* (*Knowledge Discovery and DataBase Mining*), es decir, *Adquisición de Conocimiento y Minería de Bases de Datos*. Con este objetivo, se tratan de definir *Sistemas de Data Mining de Segunda Generación* [6] dotando a los sistemas de bases de datos existentes de la capacidad de representar estructuras resultantes del proceso de minería e, incluso, proporcionarles mecanismos para poder realizar operaciones de minería de forma independiente (acoplamiento fuerte).

La información que proporciona el proceso de minería va desde conceptos clásicos como el de dependencia funcional hasta conceptos complejos como el de dependencias funcionales difusas, además de los comunes en este tipo de procesos como son los *árboles de decisión* y *clusterings*.

En el Modelo Relacional clásico, el concepto de *dependencia funcional* se vuelve demasiado estricto, y se plantean flexibilizaciones del mismo con conceptos como el de *dependencias funcionales parciales* de Berzal et al. [1], concepto con el cual los autores pretenden realizar la normalización de una relación aislando aquellas tuplas que no cumplen la dependencia funcional clásica.

Sin embargo, la inserción de nuevas tuplas en la relación original obliga a evaluar si cumple la dependencia funcional, en cuyo caso se normalizaría. En caso contrario, la nueva tupla sería aislada con todas aquellas que no la cumplen en la llamada *tabla de excepciones*. La aplicación del concepto de relación intensivo-extensiva a este problema es evidente, dado que hay un subconjunto de reglas que cumplen una dependencia (regla) y un subconjunto que no la cumplen (hechos).

La dependencia funcional puede ser representada mediante un predicado intensivo-extensivo, y todas las excepciones pueden formar parte de la parte extensiva de dicho predicado.

Por otra parte, las relaciones intensivas permiten representar la transitividad en las relaciones entre entidades. Si se pueden incorporar directamente relaciones transitivas conocidas entre entidades, se incrementa la capacidad para representar situaciones posibles. Sin embargo, el uso de reglas como las que se muestran en la Definición 4 puede provocar una perdida de información considerable, dado que el grado de satisfacción de la regla tiende progresivamente hacia 1. En estos casos, puede incluirse hechos en la parte intensiva del predicado intensivo-extensivo que limiten el crecimiento del grado de satisfacción a un grado conocido.

5 Conclusiones y trabajos futuros

El proceso de minería de bases de datos permite la extracción de las relaciones existentes entre los datos almacenados en una base de datos relacional. La

forma básica de esas relaciones es la llamada *dependencia funcional*, cuya extracción permite llevar a cabo el proceso conocido como *normalización de una base de datos*. Dicho proceso garantiza que las relaciones cumplen ciertas características. Sin embargo, el proceso de normalización ha de ser llevado a cabo por el desarrollador de la base de datos, y la reconstrucción de la base de datos original (o *no normalizada*) corre de su cuenta.

Con las dependencias parciales ocurre otro tanto. El propio desarrollador tendría que reunir las relaciones normalizadas para obtener aquellas tuplas de la relación original (*no normalizada parcialmente*) que cumplen la dependencia funcional y anexar, posteriormente, las tuplas de la relación original que no la cumplen (almacenadas en lo que se conoce como *tabla de excepciones*). Por este motivo, la incorporación del concepto de relación intensivo-extensiva, junto con la de un motor de inferencia, permite obtener la relación original a partir de una serie de relaciones normalizadas *parcialmente* y de sus *excepciones a la dependencia* de igual manera que una relación habitual.

Partiendo de la definición, nos proponemos estudiar la aplicación de este nuevo concepto a las estructuras de conocimiento resultantes de diversas técnicas de la minería de bases de datos, como puedan ser la *dependencias aproximadas*, *dependencias parciales difusas* o las *reglas de asociación*.

Desde un punto de vista aplicado, incorporaremos esta herramienta a sistemas de bases de datos relacionales o difusos relacionales que ya incorporen dichas técnicas. Con esto, nos proponemos proveerlos de capacidades adaptativas ante la inserción de nuevas tuplas en relaciones que tengan asociadas este tipo de estructuras de modo que las estructuras no queden invalidadas por modificaciones mínimas en la población en la cual se basan.

Reconocimientos

Esta investigación ha sido financiada con fondos del proyecto TIC2002-00480 del Ministerio de Ciencia y Tecnología.

Referencias

[1] F. Berzal, J. C. Cubero, F. Cuenca, and J. M. Medina. Relational decomposition through functional partial dependencies. *Data & Knowledge Engineering*, 43:207–234, 2002.

[2] I. Blanco, J. C. Cubero, O. Pons, and M. A. Vila. An implementation for fuzzy relational databases. In G. Bordogna and G. Passi, editors, *Recent Research Issues on the Management of Fuzziness in Databases*, Studies in Fuzziness and Soft Computing, pages 183–207. Physica-Verlag, 2000.

[3] I. Blanco, M. J. Martin-Bautista, O. Pons, and M. A. Vila. A mechanism for deduction in a fuzzy relational database. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 11:47–66, September 2003.

[4] E. F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 1970.

[5] H. Gallaire, J. Minker, and J. M. Nicholas. Logic and databases: A deductive approach. *Computing Surveys*, 16(2):153–185, June 1984.

[6] T. Imielinski and Heikki Mannila. A database perspective on knowledge discovery. *Communications of the ACM*, 39(11):58–64, 1996.

[7] J. M. Medina, O. Pons, and M. A. Vila. Gefred. a generalized model of fuzzy relational databases. *Information Sciences*, 1994.

[8] M. A. Vila, J. C. Cubero, J. M. Medina, and O. Pons. A logic approach to fuzzy relational databases. *International Journal of Intelligent Systems*, 9:449–460, 1994.

[9] L. A. Zadeh. Fuzzy sets. *Information and Control*, 83:338–353, 1965.

New Developments in Possibilistic Decision Theory

Lluís Godo
IIIA - CSIC
Campus UAB s/n.
08193 Bellaterra, Spain.
godo@iiia.csic.es

Adriana Zapico
Departamento de Computación
Fac. de Cs. Exactas, Fis-Quím. y Naturales
Universidad Nacional de Río Cuarto
CONICET
5800 Río Cuarto, Argentina.
zapico@dc.exa.unrc.edu.ar

Abstract

In Possibilistic Decision Theory, decisions are ranked via their associated possibility distributions on consequences, using two qualitative utility functionals expressed in terms of the uncertainty on situations and of the decision maker's preference assignment to consequences. The model has been extended in several aspects to be applicable in a broad context, for instance generalised utilities functions involving other t-norms than minimum have been introduced, uncertainty and preferences have been allowed to be measured in non-linear structures, and partially inconsistent belief states have been also considered. In this paper, we focus in extending the model along two different directions. We consider "sequential" refinements of the preference rankings in a non-linear framework on the one hand, and on the other hand we also address the concept of conditioning.

Keywords: Qualitative Decision Theory, Refinement orderings.

1 Introduction

In the recent past, Qualitative Decision Theory has had a significant development due to the necessity of dealing with qualitative, non-numerical knowledge, since qualitative representations are closer to the way people reason. Doyle and Thomason summarise in [1] the main qualitative approaches. Among them, we can find models representing uncertainty by possibility distributions. In [4], a qualitative counterpart of Von Neumann and Morgenstern's Expected Utility Theory[12] was proposed. In this model, uncertainty is assumed to be of possibilistic nature, that is, a belief states is represented by a normalised possibility distribution $\pi_0 : S \to V$ on the (finite) set of possible situations S and with values on a (finite) linearly ordered scale V [1]. Decisions are modelled, as usual, as mappings $d : S \to X$ from situations to the (finite) set of possible consequences X. Therefore, each decision d can be associated with a (normalised) possibility distribution on consequences $\pi_d : X \to V$ defining

$$\pi_d(x) = \max\{\pi_0(s) \mid s \in S, d(s) = x\}.$$

The possibilistic model also shares the usual assumption that the decision maker's preferences on the set of possible consequences are representable by a utility function $u : X \to U$, where U is another (finite) linearly ordered scale. A consequence x is preferred to x' whenever $u(x) > u(x')$. The utility scale U is assumed to be commensurate with V, which amounts to assume the existence of an onto, order-preserving mapping $h : V \to U$ linking both scales.

In the model of [4], à la Von Neumann and Morgerstern[2], ranking decisions amounts therefore to rank possibility distributions on X, also called possibilistic lotteries. Let $\mathbf{\Pi}(\mathbf{X}) = \{\pi : X \to V \mid \exists x \in X,\ \pi(x) = 1\}$ denote the set of normalised distributions on consequences. Given a utility assignment $u : X \to U$, these distributions may be ranked applying two qualitative utility functionals, according to an optimistic or a pessimistic criterion, and by means of Sugeno-like integrals. Both criteria are qualitative in the sense that they only involve the minimum, the maximum and an order-reversing operators. Namely,

$$d \preceq^- d' \text{ iff } \mathcal{U}^-(d) \leq \mathcal{U}^-(d')$$
$$d \preceq^+ d' \text{ iff } \mathcal{U}^+(d) \leq \mathcal{U}^+(d')$$

[1] The infimum and the supremum of each set will be denoted by 0 and 1 respectively, although they might be different for each set, for a simpler notation, we use the same notation for them assuming they may be distinguished by the context.

[2] An alternative approach of the possibilistic model à la Savage has been also developed, see e.g. [5].

where

$$\mathcal{U}^-(d) = QU^-(\pi_d \mid u) = \min_{x\in X} \max(n(\pi_d(x)), u(x))$$
$$\mathcal{U}^+(d) = QU^+(\pi_d \mid u) = \max_{x\in X} \min(h(\pi_d(x)), u(x)),$$

where $n = n_U \circ h$, n_U being the reversing involution on U. Since $\mathcal{U}^-(d)$ evaluates to what extent all possible consequences of d are good, $\mathcal{U}^-$ models a pessimistic criterion, while $\mathcal{U}^+(d)$ represents an optimistic behavior by evaluating to what extent at least one possible consequence is good.

Several extensions of the decision model have been proposed. Generalised utility functionals involving general t-norm operations in the uncertainty scale were considered in [2]. For each t-norm-like operation $\top$ on V such that h is coherent w.r.t. $\top$, i.e. h satisfies

$$h(\alpha) = h(\beta) \Rightarrow h(\alpha \top \lambda) = h(\beta \top \lambda) \qquad \forall \lambda \in V,$$

the following generalised qualitative utility functionals were proposed:

$$GQU^-_\top(\pi_d \mid u) = \min_{x\in X} n(\pi_d(x) \top \lambda_x), \tag{1}$$
$$GQU^+_\top(\pi_d \mid u) = \max_{x\in X} h(\pi_d(x) \top \mu_x), \tag{2}$$

with $n(\lambda_x) = u(x) = h(\mu_x)$, and n is as above. In particular, when $\top = \min$ the above original functionals are recovered.

In some decision problems, like in case-based decision or when partially conflicting belief states are considered, non-normalised possibility distributions can be involved. To properly handle these decision problems, another extension of the model is proposed in [9], where adapted utility functionals are defined as follows:

$$\underline{GQU}^-_\top(\pi_d \mid u) = \min(GQU^-_\top(\mathcal{N}(\pi_d) \mid u), n(\mathcal{I}(\pi_d))), \tag{3}$$
$$\underline{GQU}^+_\top(\pi_d \mid u) = \max(GQU^+_\top(\mathcal{N}(\pi_d) \mid u), h(\mathcal{I}(\pi_d))), \tag{4}$$

where $\mathcal{N}(\pi_d)$ denotes the normalisation of π_d, defined as

$$\mathcal{N}(\pi)(x) = \begin{cases} 1, & \text{if } \pi(x) = \mathcal{H}(\pi) \\ \pi(x), & \text{otherwise}^3 \end{cases}$$

with the *height* of a distribution being defined as $\mathcal{H}(\pi) = \max\{\pi(x) \mid x \in X\}$, and $\mathcal{I}(\pi) = n_V(\mathcal{H}(\pi))$ is called the *inconsistency level* of π, where n_V is the reversing involution on V.

Other extensions of the model, described in [13], allow non-linear scales for measuring preferences or/and uncertainty, in particular finite distributive lattices with involution, and they consider generalised utility functionals analogous to (1) and (2) but with minimum and maximum replaced by lattice meet and joint, and h required to be an epimorphism. In that framework, the normalisation is defined as

$$\mathcal{N}(\pi)(x) = \begin{cases} 1, & \text{if } \pi(x) \text{ is maximal} \\ \pi(x), & \text{otherwise,} \end{cases}$$

and the height of a distribution is defined as $\mathcal{H}(\pi) = \bigvee\{\pi(x) \mid x \in X\}$.

One of the drawbacks of the possibilistic decision criteria (both the optimistic and pessimistic) is their lack of discrimination power in some situations. In those situations, in order to break the indifference status among alternative decisions, it is natural to consider possible refinements of the preference ordering. In [3] several possibilities were analysed and the refinements of the ordering induced by one criterion with the orderings induced by other criteria were characterised.

In this paper, we extend the approach of [3] by allowing to work with non-normalised distributions and to measure uncertainty and preferences on finite distributive lattices. In the second part of the paper we also consider the preference orderings resulting from conditioning by an event in the set of states or in the set of consequences.

2 Axiomatic characterisations

In this section we recall the axiomatic characterisations of the generalised utility functions, provided in [2] for linear structures of measurement, and in [13] for lattices measurement sets. For each t-norm $\top$ in V, a possibilistic $\vee$-$\top$-mixture $M_\top$ is considered, it is an internal operation on $\Pi(X)$ defined as $M_\top(\pi, \pi'; \alpha, \beta) = (\alpha \top \pi) \vee (\beta \top \pi')$ $\alpha, \beta \in V$ with $\alpha = 1$ or $\beta = 1$. This mixture operation is a possibilistic counterpart of the convex linear combination of probability distributions, that combines two normal distributions into a new one. Given a preference relation $\sqsubseteq$ defined on $\Pi(X)$, there are different axiomatisation cases taking into account the structure of V and U ($\pi \sim \pi'$ means $\pi \sqsubseteq \pi'$ and $\pi' \sqsubseteq \pi$).

Linear scales In the case both scales V and U are linearly ordered, the following set of axioms $\mathbf{AX}_\top = \{A1, A2, A3_\top, A4_\top\}$ characterises the (pessimistic) preference orderings induced by a $GQU^-_\top$ utility functional, for each t-norm $\top$ on V:

A1 (*structure*): $\sqsubseteq$ is a total pre-order (i.e. $\sqsubseteq$ is reflexive, transitive, total)

A2 (*uncertainty aversion*): if $\pi \leq \pi' \Rightarrow \pi' \sqsubseteq \pi$.

[3] Obviously, if π is normalised, $\mathcal{N}(\pi) = \pi$, and (3) and (4) coincide with (1) and (2) respectively.

A3$_\top$ (*independence*):
$\pi_1 \sim \pi_2 \Rightarrow M_\top(\pi_1, \pi; \alpha, \beta) \sim M_\top(\pi_2, \pi; \alpha, \beta)$.

A4$_\top$ (*continuity*): $\forall \pi \in \Pi(X)\ \exists \lambda \in V$ s.t. $\pi \sim \pi^-_\lambda$.

Here $\overline{x}$ and $\underline{x}$ are a maximal and a minimal element of $(X, \sqsubseteq)$ respectively (they exist by A1). Also, for any $\lambda \in V$, π^-_λ denotes the distribution $M_\top(\overline{x}, \underline{x}; 1, \lambda)$.

For representing an optimistic behaviour induced by a $GQU^+_\top$ functional, $A2$ and $A4_\top$ are respectively replaced by:

A2$^+$ (*uncertainty attraction*): if $\pi \leq \pi' \Rightarrow \pi \sqsubseteq \pi'$.

A4$^+_\top$: $\forall \pi \in \Pi(X)\ \exists \lambda \in V$ s.t. $\pi \sim \pi^+_\lambda$,

where $\pi^+_\lambda = M_\top(\overline{x}, \underline{x}; \lambda, 1)$. $\mathbf{AX}^+_\top$ denotes the set of axioms {A1, A2$^+$, A3$_\top$, A4$^+_\top$}.

In [2] it is shown that a preference relations on $\Pi(X)$ satisfies $AX^-_\top$ (resp. $AX^+_\top$) if and only if it is indeed induced by a $GQU^-_\top$ (resp. $GQU^+_\top$) functional.

These representations have been extended to also characterise preference relations on the set $\mathbf{\Pi^{ex}(X)}$ of non-necessarily normalised possibility distributions on X. Then one considers the extended sets of axioms $\underline{\mathbf{AX}}_\top$ $=\mathbf{AX}_\top \cup \{\mathbf{AP7}\}$ and $\underline{\mathbf{AX}}^+_\top = \mathbf{AX}^+_\top \cup \{\mathbf{AP7}\}$, with

AP7(normalisation): $\pi \sim M_\top(\mathcal{N}(\pi), X; 1, \mathcal{I}(\pi))$,

with the understanding that $\mathbf{AX}_\top$ and $\mathbf{AX}^+_\top$ are assumed to be fulfilled only by the restriction of the preference relation on the subset of $\Pi(X) \subset \Pi^{ex}(X)$ of normalised possibility distributions. It is shown in [9] that these axiomatics characterise those preference relations on $\Pi^{ex}(X)$ induced by the $\underline{GQU^-_\top}$ and $\underline{GQU^+_\top}$ functionals respectively.

Non linear Lattices This is the case of V being a (finite) distributive residuated lattice with involution and U a (finite) distributive lattice with involution. For a pessimistic behaviour of a preference relation $\sqsubseteq$ on $\Pi(X)$ we consider the set of axioms $\mathbf{AXP}_\top$ = {AP1, A2, A3$_\top$, A4$_\top$, AP5$_\top$, AP6$_\top$} with

AP1(*structure*): $(\Pi(X), \sqsubseteq)$ is a pre-lattice[4]

AP5$_\top$ (*preference reversing*):
if $\pi^-_\lambda \sqsubseteq \pi^-_{\lambda'} \Rightarrow \pi^-_{n_V(\lambda)} \sqsupseteq \pi^-_{n_V(\lambda')}$

AP6$_\top$ (*incomparability preservation*):
if $\lambda <> \lambda' \Rightarrow \pi^-_\lambda \sqsubset\sqsupset \pi^-_{\lambda'}$

[4] $(\Pi(X), \sqsubseteq)$ is a *pre-lattice* if $(\Pi(X)/\sim, \leq)$ is a lattice, where the lattice order $\leq$ is defined as $[\pi] \leq [\pi']$ iff $\pi \sqsubseteq \pi'$.

where for any $\lambda \in V$, $\pi^-_\lambda = M_\top(\overline{\pi}, X; 1, \lambda)$ with $\overline{\pi}$ being a maximal element of $(\Pi(X), \sqsubseteq)$.

For an optimistic behaviour we modify axiom $A2$ and the axioms involving π^-_λ. Indeed, we consider $\mathbf{AXP}^+_\top$= {AP1, A2$^+$, A3$_\top$, A4$^+_\top$, AP5$^+_\top$, AP6$^+_\top$ }, with

AP5$^+_\top$ (*preference reversing*):
if $\pi^+_\lambda \sqsubseteq \pi^+_{\lambda'} \Rightarrow \pi^+_{n_V(\lambda)} \sqsupseteq \pi^+_{n_V(\lambda')}$

AP6$^+_\top$ (*incomparability preservation*):
if $\lambda <> \lambda' \Rightarrow \pi^+_\lambda \sqsubset\sqsupset \pi^+_{\lambda'}$

where $\pi^+_\lambda = M_\top(X, \underline{\pi}; \lambda, 1)$, with $\underline{\pi}$ a minimal element on $(\Pi(X), \sqsubseteq)$.

Preference relations on $\Pi(X)$ satisfying $\mathbf{AXP}_\top$ or $\mathbf{AXP}^+_\top$ are shown in [13] to be exactly those induced by lattice extensions of the $GQU^-_\top$ and $GQU^+_\top$ functionals.

3 Some refinements

In order to try to break some indifference status established taking into account the rankings induced by the QU or the GQU, we may sequentially apply some of the qualitative criteria previously mentioned. A first option, very simple and easy to justify, is to use the optimistic criterion to refine the pessimistic one, i.e.

$$\begin{aligned}\pi \sqsubseteq \pi' \iff & \{GQU^-(\pi \mid u) <_U GQU^-(\pi' \mid u)\} \text{ or} \\ & \{[GQU^-(\pi \mid u) = GQU^-(\pi' \mid u)] \text{ and} \\ & [GQU^+(\pi \mid u) \leq_U GQU^+(\pi' \mid u)]\},\end{aligned}$$

where both generalised utility functions are defined over the same set U and with the same preference function u.

There are several alternative combinations for refining. For example, the lexicographic ordering of an arbitrary set of generalised pessimistic/optimistic criteria that might include: (i) different t-norms, and (ii) different preference sets and assignments, may be seen as an easy generalisation of this idea. Indeed, given a set of preference mappings $u_j : X \to U_j$, a set $\top_j$ of t-norms on V, and a set of coherent commensurability mappings $h_j : V \to U_j$, we define :

$$\begin{aligned}&\pi \sqsubseteq \pi' \iff \\ &(\mathcal{U}_1(\pi), \ldots, \mathcal{U}_m(\pi)) \leq_{LEX} (\mathcal{U}_1(\pi'), \ldots, \mathcal{U}_m(\pi')) \quad (5)\end{aligned}$$

with either $\mathcal{U}_j = GQU^-_{\top_j}(\cdot \mid u_j, h_j)$ or $\mathcal{U}_j = GQU^+_{\top_j}(\cdot \mid u_j, h_j)$, where $\leq_{LEX}$ denotes the *lexicographic ordering* on the cartesian product $U_1 \times \ldots \times U_m$: $(u_1, \ldots, u_m) \leq_{LEX} (u'_1, \ldots, u'_m)$ if either $(u_1, \ldots, u_m) = (u'_1, \ldots, u'_m)$ or there is $j \leq m$ such that $u_i = u'_i$ for all $i < j$ and $u_j < u'_j$.

This type of refinements may be useful in decision problems in which we have several utility criteria that can be ranked according to their importance. For example in a safety decision problem we can have a utility function u_1 that accounts for preference for high personal safety levels and another function u_2 that accounts for preference in having low costs. In such a case, the first criterion must be more important than the second one. Notice that lexicographic utility scales have been considered for a long in the framework of Expected Utility Theory by people like Fishburn and LaValle (see e.g. [7, 10]).

Let us introduce a concept that will be useful for simplifying notation and mainly to generalise the results. Given a finite set of binary relations $\mathcal{R} = \{\preccurlyeq_i\}_{i=1,\ldots,k}$ on a set E, each "Boolean" mapping $g : \{0,1\}^k \times \{0,1\}^k \to \{0,1\}$ induces a new binary relation $\preccurlyeq^g_{\mathcal{R}}$ on E by defining:

$$e \preccurlyeq^g_{\mathcal{R}} e' \iff g((\mu_{\preccurlyeq_1}(e,e'),\ldots,\mu_{\preccurlyeq_k}(e,e')), (\mu_{\preccurlyeq_1}(e',e),\ldots,\mu_{\preccurlyeq_k}(e',e))) = 1,$$

where $\mu_{\preccurlyeq_i}$ is the membership of the binary relation $\preccurlyeq_i$. Notice that the *lexicographic ordering* is a particular case of the relations $\preccurlyeq^g_{\mathcal{R}}$. Indeed, if the $\preccurlyeq_i$'s are linear orderings, and let $g_*(\overline{x},\overline{y}) = \max_{i=1,\ldots,k}\ z_i$, with

$$z_i = \begin{cases} \min(x_1, 1-y_1), & i = 1 \\ \min[\min_{j=1,\ldots,i-1}\{\min(x_j,y_j)\}, & \\ \quad \min(x_i, 1-y_i)], & 1 < i < k \\ \min(\min_{j=1,\ldots,k-1}\{\min(x_j,y_j)\}, x_k), & i = k, \end{cases}$$

then $\preccurlyeq^{g_*}_{\mathcal{R}}$ is nothing but the lexicographic ordering on E induced by $\preccurlyeq_1,\ldots,\preccurlyeq_k$, taken them in this sequence. In particular, taking g_* and $E = \Pi(X)$ with $\preccurlyeq_i = \leq_{\mathcal{U}_i}$, the preference ordering (5) on possibility distributions can be simply written as $\pi \preccurlyeq^{g_*}_{\mathcal{R}} \pi'$. Moreover, the Pareto ordering is also a particular example of these $\preccurlyeq^g_{\mathcal{R}}$ orderings [5]. Now, we can proceed to the axiomatic characterisation of refinements of (partially ordered) preference relations involving arbitrary generalised utility functionals on lattice structures. The characterisation is a straightforward generalisation of the characterisations of the partial orderings described on Section 2.

Theorem 1 (Representation Theorem) *Let V be a (finite) residuated distributive lattice with involution and let $\sqsubseteq$ be a preference relation on $\Pi(X)$ (normalised distributions on X over V) for which there exist a set $\mathcal{R} = \{\sqsubseteq_1,\ldots,\sqsubseteq_m\}$ of preference orderings also on $\Pi(X)$ and a boolean mapping g such that $\sqsubseteq = \preccurlyeq^g_{\mathcal{R}}$, i.e.*

$$\pi \sqsubseteq \pi' \iff g((\mu_{\sqsubseteq_1}(\pi,\pi'),\ldots,\mu_{\sqsubseteq_m}(\pi,\pi')), (\mu_{\sqsubseteq_1}(\pi',\pi),\ldots,\mu_{\sqsubseteq_m}(\pi',\pi))) = 1,$$

with each $\sqsubseteq_j$ satisfying either $\mathrm{AXP}_{\top_j}$ or $\mathrm{AXP}^+_{\top_j}$ (we will write $\sqsubseteq^-_j$ or $\sqsubseteq^+_j$ depending on whether $\sqsubseteq_j$ satisfies $\mathrm{AXP}_{\top_j}$ or $\mathrm{AXP}^+_{\top_j}$ respectively.). Then there exist:

(i) *m utility finite distributive lattices with involution $(U_i, \wedge^i, \vee^i, n_{U^i}, 0, 1)$;*

(ii) *m preference functions $u^i{:}X \to (U^i, \leq^i)$, each u^i satisfying either*

$$\{(u^i)^{-1}(1) \neq \emptyset \text{ and } \bigwedge_{x\in X}(u^i)(x) = 0\} \quad \text{or}$$
$$\{(u^i)^{-1}(0) \neq \emptyset \text{ and } \bigvee_{x\in X}(u^i)(x) = 1\},$$

depending whether $\sqsubseteq_i$ satisfies $\mathrm{AXP}_{\top_i}$ or $\mathrm{AXP}^+_{\top_i}$ respectively;

(iii) *m onto join-preserving mappings $h^i{:}V \to U^i$, all satisfying coherence w.r.t $\top_i$, also satisfying*

$$\text{if } \lambda <> \lambda' \text{ then } h^i(\lambda) <> h^i(\lambda'),$$

and $n_{U^i} \circ h^i \circ n_V = h^i$,

in such a way that it holds:

$$\pi \sqsubseteq \pi' \quad \text{iff} \quad \pi \preccurlyeq^g_{\{\preccurlyeq_{GQU^1},\ldots,\preccurlyeq_{GQU^m}\}} \pi',$$

with $\pi \preccurlyeq_{GQU^i} \pi'$ iff

$$GQU^{sgn(i)}_{\top_i}(\pi|u^i,h^i) \leq_{U^i} GQU^{sgn(i)}_{\top_i}(\pi'|u^i,h^i),$$

where $sgn(i) = -$ if $\sqsubseteq_i = \sqsubseteq^-_i$, and $sgn(i) = +$ if $\sqsubseteq_i = \sqsubseteq^+_i$.

4 Conditional preferences

It may be the case that after having ranked the distributions, we get additional knowledge about possible consequences, which can be modelled as an independent possibility distribution that restricts the set of possible consequences. In this case we need to take into account this information to update the preference ranking by conditioning the possibilistic lotteries by this information.

Definition 1 *Let $\sqsubseteq$ be a preference relation on the set of possibly non-normalised distributions on X, $\Pi^{ex}(X)$, a conjunction operation $\wedge$ on $\Pi^{ex}(X)$. For each possibility distribution A on X, we define the* preference relation $\wedge$-conditioned by A, $\sqsubseteq^{\wedge}_A$, *as*

$$\pi \sqsubseteq^{\wedge}_A \pi' \iff \pi \wedge A \ \sqsubseteq\ \pi' \wedge A.$$

[5] It has to be noticed as well that not any g gives raise to a proper ordering, i.e. a reflexive and transitive relation.

Accordingly, one can define their corresponding utility functions as

$$GQU_A^-(\pi \mid u) = \underline{GQU^-}(\pi \wedge A \mid u),$$
$$GQU_A^+(\pi \mid u) = \underline{GQU^+}(\pi \wedge A \mid u).$$

Notice that although π is normalised, $\pi \wedge A$ can be non-normalised. Except for a normalisation factor, $\pi \wedge A$ may be seen as the conditioning of π by the fuzzy event A.

Thanks to the characterisation of preference relations on $\Pi^{ex}(X)$, we are able of providing the representation theorem for the *conditional* preference relations induced by the conditional utility functionals GQU_A^- and GQU_A^+, for any possibility distribution A on X.

Theorem 2 *Let $(\Pi^{ex}(X), \preccurlyeq, M_\top)$ be a preference relation for which there exist a subset A in X, a conjunction operation $\wedge$ on $\Pi^{ex}(X) \times \Pi^{ex}(X)$, and a preference relation $\sqsubseteq$ on $\Pi^{ex}(X)$, s.t.*

$$\pi \preccurlyeq \pi' \iff \pi \sqsubseteq_A^{\wedge} \pi',$$

with $\sqsubseteq$ satisfying the axiom set $\underline{\mathbf{AX}}_\top$. Then there exist:

- **(i)** *a finite linearly ordered preference scale U with $inf(U) = 0$ and $sup(U) = 1$,*
- **(ii)** *a preference function $u : X \to U$ such that $u^{-1}(1) \neq \emptyset \neq u^{-1}(0)$,*
- **(iii)** *an onto order-preserving function $h : V \to U$ coherent w.r.t. $\top$*

in such a way that

$$\pi' \preccurlyeq \pi \quad \textit{iff} \quad GQU_A^-(\pi') \leq_U GQU_A^-(\pi),$$

with $n = n_U \circ h$.

A dual characterisation theorem can be easily obtained as well for the optimistic preference relations induced by GQU_A^+ functionals, just by replacing above the axiom set $\underline{\mathbf{AX}}_\top$ by the axiom set $\underline{\mathbf{AX}}_\top^+$.

There is however another possible way of conditioning when definite information is known in the sense that the actual state is in a given subset of the set S. First, let us recall that given a set of states A, and two acts d, d' the "compound" act of d and d' w.r.t. A is defined as

$$dAd'(s) = \begin{cases} d(s), & \text{if } s \in A \\ d'(s), & \text{otherwise} \end{cases}.$$

Given a preference order among decisions $\preceq$, Savage [11] defines the conditional preference relation $\preceq_A$ on a set of states A as follows:

$$d \preceq_A d' \text{ iff } dAd'' \preceq d'Ad'' \text{ for all } d''.$$

Notice that if $\preceq$ satisfies the Sure Thing Principle then $d \preceq_A d'$ iff $dAd'' \preceq d'Ad''$ for some d''.

Now, given a possibility distribution $\pi_0 : S \to [0,1]$, recall also that we define the possibility distribution π_d associated to a decision d as $\pi_d(x) = \Pi_0(\{s \in S \mid d(s) = x\}) = \Pi_0(d^{-1}(x))$, where Π_0 is the corresponding possibility measure induced by π_0. Therefore, it is easy to check that we have

$$\pi_{dAd'}(x) = max(\pi_d^A(x), \pi_{d'}^{S \setminus A}(x)),$$

where for any $T \subseteq S$ and any decision d, we define $\pi_d^T(x) = \Pi_0(d^{-1}(x) \cap T)$.

Then, following Savage, and given a preference ordering among possibility distributions on consequences $\sqsubseteq$, we can consider the following conditional ordering on a subset A of states:

Definition 2 *$\pi_d \sqsubseteq_A \pi_{d'}$ iff $\pi_{dAd''} \sqsubseteq \pi_{d'Ad''}$ for all d''.*

In the rest of the section, we show how this definition of conditional preference can be made simpler in the case the (unconditional) preference ordering $\sqsubseteq$ is defined by GQU^- and GQU^+ utility functionals. To this end, given a utility mapping $u : X \to [0,1]$, let us define the following orderings:

$\pi \sqsubseteq^- \pi'$ iff $GQU^-(\pi \mid u) \leq GQU^-(\pi' \mid u)$,

$\pi \sqsubseteq^+ \pi'$ iff $GQU^+(\pi \mid u) \leq GQU^+(\pi' \mid u)$.

Further, if V^X denotes the set of mappings $\pi : X \to V$ and if U^X denotes the set of mappings $u : X \to U$, let us define the following mappings $Z^-, Z^+ : V^X \times U^X \to U$ by putting

$$Z^-(\pi \mid u) = \min_{x \in X} n(\pi(x) \top \lambda_x)$$

$$Z^+(\pi \mid u) = \max_{x \in X} h(\pi(x) \top \mu_x)$$

with the usual assumptions on n, h, λ_x and μ_x. Notice that if $\pi \in \Pi(V)$, then $Z^-(\pi \mid u) = GQU^-(\pi \mid u)$ and $Z^+(\pi \mid u) = GQU^+(\pi \mid u)$.

Lemma 1 *Let $\pi_1, \pi_2 \in V^X$. Then one has:*
(i) $Z^-(\max(\pi_1, \pi_2) \mid u) = \min(Z^-(\pi_1 \mid u), Z^-(\pi_2 \mid u))$.
(ii) $Z^+(\max(\pi_1, \pi_2) \mid u) = \max(Z^+(\pi_1 \mid u), Z^+(\pi_2 \mid u))$.

Lemma 2 *Let $\overline{x}, \underline{x} \in X$ be such that $u(\overline{x}) = 1$ and $u(\underline{x}) = 0$. For any $T \subseteq S$ we have:*
(i) $Z^-(\pi_{\overline{x}}^T \mid u) = 1$,
(ii) $Z^+(\pi_{\underline{x}}^T \mid u) = 0$.

Proposition 1
(i) *If* $\overline{x}$ *is such that* $u(\overline{x}) = 1$, *then* $\pi_d \sqsubseteq_A^- \pi_{d'}$ *iff* $\pi_{dA\overline{x}} \sqsubseteq^- \pi_{d'A\overline{x}}$.
(ii) *If* $\underline{x}$ *is such that* $u(\underline{x}) = 0$, *then* $\pi_d \sqsubseteq_A^+ \pi_{d'}$ *iff* $\pi_{dA\underline{x}} \sqsubseteq^+ \pi_{d'A\underline{x}}$.

Proof: We have to prove that if $\pi_{d_1A\overline{x}} \sqsubseteq^- \pi_{d_2A\overline{x}}$ then $\pi_{d_1} \sqsubseteq_A^- \pi_{d_2}$. Assume then $\pi_{d_1A\overline{x}} \sqsubseteq^- \pi_{d_2A\overline{x}}$. This means that $GQU^-(\pi_{d_1A\overline{x}} \mid u) \leq GQU^-(\pi_{d_2A\overline{x}} \mid u)$. But, since $\pi_{dA\overline{x}} = \max(\pi_d^A, \pi_{\overline{x}}^{S\setminus A})$, by Lemma 1, $GQU^-(\pi_{dA\overline{x}} \mid u) = Z^-(\pi_{dA\overline{x}} \mid u) = \min(Z^-(\pi_d^A \mid u), Z^-(\pi_{\overline{x}}^{S\setminus A}) \mid u)$. Now, by Lemma 2, $Z^-(\pi_{\overline{x}}^{S\setminus A} \mid u) = 1$, hence $GQU^-(\pi_{dA\overline{x}} \mid u) = Z^-(\pi_d^A \mid u)$. Therefore, we have $Z^-(\pi_{d_1}^A \mid u) \leq Z^-(\pi_{d_2}^A \mid u)$. Now, for any d, we have:
$GQU^-(\pi_{d_1Ad} \mid u) = Z^-(\pi_{d_1Ad} \mid u) = \min(Z^-(\pi_{d_1}^A \mid u), Z^-(\pi_d^{S\setminus A} \mid u)) \leq \min(Z^-(\pi_{d_2}^A \mid u), Z^-(\pi_d^{S\setminus A} \mid u)) = Z^-(\pi_{d_2Ad} \mid u) = GQU^-(\pi_{d_2Ad} \mid u)$. Thus it means that $\pi_{d_1} \sqsubseteq_A^- \pi_{d_2}$.

The case of $\sqsubseteq_A^+$ is analogous. □

This proposition amounts to state that, if $\preceq^-$ and $\preceq^+$ are preference orderings on acts induced by GQU^- and GQU^+ generalised utilities respectively, then

$$d \preceq_A^- d' \quad \text{iff} \quad dA\overline{x} \preceq^- d'A\overline{x} \ ,$$
$$d \preceq_A^+ d' \quad \text{iff} \quad dA\underline{x} \preceq^+ d'A\underline{x} \ .$$

This means that, in the case of pessimistic criteria, conditioning by A is equivalent to consider the decision gets the best consequence in any state outside A, while in the optimistic case conditioning by A amounts to consider the decision gets the worst consequence in any state outside A.

5 Conclusions

In this paper we have considered two further steps in the foundations of Possibilistic Decision Theory à la Von Neumann and Morgenstern. In this framework, preference relations are defined on possibilistic distributions on the set of consequences. Namely we have considered the definition and characterisation of refinements of partially ordered preference relations and two notions of conditional preference relations. Refinements in the framework of possibilistic decision have also been recently considered by Fargier and Sabbadin in [6] where they show that both (pessimistic and optimistic) preference orderings induced by the possibilistic utility functionals QU^- and QU^+ always admit a refinement which is compatible with the Expected Utility model. On the other hand, Giang and Shenoy propose in [8] a model where both the possibilistic pessimistic and optimistic functionals are combined somehow in a single functional which behaves as a refinement. In future works we plan to establish links between these kind of refinements and those described in the paper. Finally, we think that the notions of conditional preference presented here will allow us to build tighter bridges between the two kinds of axiomatic approaches to possibilistic decision theory, à la Von Neumann and Morgenstern and à la Savage.

References

[1] J. Doyle and R. Thomason. Background to qualitative decision theory. *AI Magazine*, 20(2):55–68,1999.

[2] D. Dubois, L. Godo, H. Prade, and A. Zapico. On the possibilistic-based decision model: From decision under uncertainty to case-based decision. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 7(6):631–670, 1999.

[3] D. Dubois, L. Godo, H. Prade, and A. Zapico. Advances in Qualitative Decision Theory: Refined Rankings. In *7th Ibero-American Conference on Artificial Intelligence(IBERAMIA2000)*, LNAI 1952, Springer, pages 427-436, 2000.

[4] D. Dubois and H. Prade. Possibility theory as a basis for qualitative decision theory. In *14th Int. Joint Conf. on Artificial Intelligence (IJCAI'95)*, pages 1924–1930, Montreal, 1995.

[5] D.Dubois, H. Prade, and R.Sabbadin. Decision-theoretic foundation of qualitative possibility theory *European Journal of Operational Research*, 128,pages 495–478,2001.

[6] H. Fargier and R. Sabbadin. Qualitative Decision under Uncertainty: Back to Expected Utility. In *18th Int. Joint Conf. on Artificial Intelligence (IJCAI'03)*, pages 303–308, Acapulco, 2003.

[7] P.C. Fishburn. Lexicographic orders, utilities and decision rules: A survey. *Management Scicence* , 20, 11, pages 14421471, 1974.

[8] P. H. Giang and P.P. Shenoy. Two Axiomatic Approaches to Decision Making Using Possibility Theory. *European Journal of Operational Research*, (To Appear).

[9] L. Godo and A. Zapico. On the possibilistic-based decision model: Characterisation of preferences relations under partial inconsistency. *Applied Intelligence*, 14(3):319–333, 2001.

[10] I. H. LaValle and P. C. Fishburn. Structuring and assessing linear lexicographic utility. *Journal of Risk and Uncertainty*, 1996.

[11] L. J. Savage. *The Foundations of Statistics.* Dover, New York, 1972.

[12] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton: Princeton University Press, 1944.

[13] A. Zapico. Axiomatic foundations for qualitative/ordinal decisions with partial preferences. In *16th Int. Joint Conf. on Artificial Intelligence (IJCAI'99)*, 1999.

A LINGUISTIC EVALUATION MODEL FOR ENGINEERING SYSTEMS

Luis Martínez, Luis. G. Pérez
Dpto. de Informática Universidad de Jaén
Jaén, Spain
Martin,lgonzaga@ujaen.es

Jun Liu, Jian-Bo Yang
Manchester School of Management,
UMIST, Manchester, UK
j.liu, Jian-bo.yang @umist.ac.uk

Abstract

To make a decision of implementing a design for a large engineering system different design proposals are evaluated and ranked based on different criteria. The knowledge used to evaluate these proposals is usually vague and incomplete, we shall use fuzzy linguistic approaches and the Dempster-Shafer theory of evidence to deal with it. In this contribution we propose an evaluation model for Engineering Systems based on the *Safety*, *Cost* and *Technical performance* criteria. The safety assessments will be obtained using fuzzy rule-based evidential reasoning (FURBER) approach and the cost and technical assessments are directly provided by the experts. The assessments for all the criteria are usually assessed in different utility spaces. The aim of this contribution is to develop a multi-granular linguistic decision model to evaluate different design options for an engineering system

Keywords: Engineering systems, decision making, multi-granular linguistic information.

1 INTRODUCTION

The decision of implementing a design in a large engineering system depends on that the design satisfies technical and economical constraints. Multi-Criteria Decision Making (MCDM) techniques could be applied to obtain a ranking order of different design options. Particularly in the feasibility and concept selection stages of an engineering system. Subjective assessments about safety, cost and technical performance can be studied together to determine the best risk reduction action and to choose the best design/operation option. Multiple analysts can provide their subjective judgments for each design option on these criteria.

Typical safety assessment approaches may be difficult to use in situations where there is a lack of information, past experience, or ill-defined situation in risk analysis [Wan97]. Therefore, linguistic descriptors, such as, "*Likely*", "*Impossible*", are used to describe an event due to the fact they are used commonly by engineers and safety analysts. Hence, the use of the fuzzy linguistic approach [Zad75] is a good model to analyse the safety of engineering systems with incomplete information. Also the estimation of the cost and technical performance are ill-defined situations, therefore the use of the linguistic approach is adequate too. Although they may be conducted in different utility spaces.

The aim of this paper is to develop a multi-granular linguistic decision model that evaluates the different design options for an engineering system according to multiple criteria assessed linguistically in different utility spaces. First we define the linguistic framework used for expressing the linguistic assessments for each criteria:

Safety will be assessed based on fuzzy logic and the evidential reasoning approach, referred to as a *fuzzy rule-based evidential reasoning* (FURBER) approach [Liu03], which is based on the RIMER approach proposed recently in [Yng03]. The synthesis of the safety assessments for each option is expressed and implemented using a linguistic 2-tuple scheme [Her00].

The cost and technical performance assessments of each design option are supplied directly by the experts in terms of linguistic labels.

All these linguistic assessments defined in a multi-granular linguistic domain will be the input values for a Multi-Expert Multi-Criteria Decision Making (MEMC-DM) problem that we shall solve to evaluate and rank the different design options.

In this contribution we shall propose a multi-granular linguistic decision model based on two different proposals presented in [Her00a, Her04], in order to evaluate the different design options.

In order to do so, this contribution is structured as follows: in Section 2 we make a brief review of linguistic tools. In Section 3 we describe the evaluation framework for modelling of large engineering systems. In Section 4 it will be presented the application of the linguistic decision model to evaluate the design options. And finally, some conclusions are pointed out.

2 LINGUISTIC BACKGROUND

Here we review briefly some core concepts about linguistic information as the *Fuzzy Linguistic Approach* and *2-tuple Linguistic model*.

2.1. FUZZY LINGUISTIC APPROACH

Usually, we work in a quantitative setting, where the information is expressed by numerical values. However, many aspects of different activities in the real world

cannot be assessed in a quantitative form, but rather in a qualitative one, i.e., with vague or imprecise knowledge. In that case a better approach may be to use linguistic assessments instead of numerical values. The fuzzy linguistic approach represents qualitative aspects as linguistic values by means of linguistic variables [Zad75].

We have to choose the appropriate linguistic descriptors for the term set and their semantics. In the literature, several possibilities can be found [Her00c]. One possibility of generating the linguistic term set consists of directly supplying the term set by considering all terms distributed on a scale on which a total order is defined. For example, a set of seven terms S, could be:

S ={s_0 = None; s_1 = Very Low; s_2 = Low; s_3 = Medium; s_4 = High; s_5 = Very High; s_6 = Perfect}

Usually, in these cases, it is required that in the linguistic term set there exist:

1. A negation operator: Neg(s_i) = s_j such that j = g-i (g+1 is the cardinality).
2. An order: s_i = s_j <=> i = j.

Therefore, there exists a minimization and a maximization operator.

The semantics of the terms are given by fuzzy numbers defined in the [0,1] interval, which are described by membership functions. A way to characterize a fuzzy number is to use a representation based on parameters of its membership function. Since the linguistic assessments given by the users are just approximate ones, some authors consider that linear trapezoidal membership functions are good enough to capture the vagueness of those linguistic assessments, since it may be impossible and unnecessary to obtain more accurate values [Her00c].

2.2. THE 2-TUPLE LINGUISTIC MODEL

This model was presented in [Her00], for overcoming the drawback of the loss of information presented by the classical linguistic computational models [Her00b]. The 2-tuple fuzzy linguistic representation model is based on the symbolic method and takes as the base of its representation the concept of Symbolic Translation.

Definition 1 The Symbolic Translation of a linguistic term s_i $S = \{s_0,...,s_g\}$ is a numerical value assessed in [-0.5,0.5) that supports the "difference of information" between an amount of information b [0, g] and the closest value in {0,…,g} that indicates the index of the closest linguistic term in S (s_i), being [0,g] the interval of granularity of S.

From this concept a linguistic representation model is developed, which represents the linguistic information by means of 2-tuples $(s_i, a_i), s_i$ S and a_i [0.5,0.5).

This model defines a set of functions between linguistic 2-tuples and numerical values.

Definition 2 Let $S = \{s_0,...,s_g\}$ be a linguistic term set and b [0, g] a value supporting the result of a symbolic aggregation operation, then the 2-tuple that expresses the equivalent information to b is obtained with the following function:

$$: [0, g] \quad S \quad (0.5,.0.5)$$

$$(b) = (s_i, a), with \quad \begin{matrix} s_i & i = round(b) \\ a = b & i \quad a \quad [0.5,0.5) \end{matrix}$$

where s_i has the closest index label to "b" and "a" is the value of the symbolic translation.

Proposition 1. Let $S = \{s_0,...,s_g\}$ be a linguistic term set and (s_i, a_i) be a linguistic 2-tuple. There is always a ^{1} function, such that, from a 2-tuple it returns its equivalent numerical value b [0, g].

Proof. It is trivial, we consider the following function:

$$^{1} : S \quad [0,5,0.5) \quad [0, g]$$

$$^{1}(s_i, a) = i + a = b$$

This model has a computational technique based on the 2-tuples were presented in [Her00]:

1. **Aggregation of 2-tuples**

   The aggregation of linguistic 2-tuples consist of obtaining a value that summarizes a set of values, therefore, the result of the aggregation of a set of 2-tuples must be a linguistic 2-tuple. In [Her00] we can find several 2-tuple aggregation operators.

2. **Comparison of 2-tuples**

   The comparison of information represented by 2-tuples is carried out according to an ordinary lexicographic order (more details [Her00]).

3. **Negation Operator of a 2-tuple**

   The negation operator over 2-tuples is defined as: $Neg(s_i, a) = \quad (g \quad ^{1}(s_i, a))$

where g+1 is the cardinality of S, s_i $S = \{s_0,..., s_g\}$.

3 EVALUATION FRAMEWORK FOR ENGINEERING SYSTEMS

In this section we show briefly how are the assessments for safety using the FURBER approach

[Liu03, Yng03] and how are the other criteria provided by the experts assessed.

3.1. SAFETY EVALUATION FRAMEWORK

A generic framework for modelling system safety estimate using FURBER approach and for safety synthesis is outlined, more details see [Liu03, Yng03].

Step #1: Identification of causes/factors: it can be done by a panel of experts during a brainstorming session at the early concept design stages.

Step #2: Identify and definite fuzzy input and fuzzy output variables (i.e., safety estimates)

The three fundamental parameters used to assess the safety level of an engineering system on a subjective basis are the failure rate (**FR**), consequence severity (**CS**) and failure consequence probability (**FCP**). Subjective assessments are more appropriate for analysis using these three parameters as they are always associated with great uncertainty.

Safety estimates is the only output fuzzy variable used to produce safety evaluation for a particular cause to technical failure. This variable is described linguistically, which is described and determined by the above parameters. In safety it is common to express a safety assessments assessed in the following linguistic term set [Wan95], that we note as, $\mathbf{S_S}$, in this paper:

$$S_S = \{Poor,\ Low,\ Average,\ High, Good\},$$

that are referred to as safety expressions.

Step #3: Construct a fuzzy rule-base

Fuzzy logic systems are knowledge-based or rule-based ones constructed from human knowledge in the form of fuzzy IF-THEN rules.

Let us suppose a linguistic term set with seven labels may be used for ***failure rate*** (i.e., J_1=7); five labels for consequence severity (i.e., J_2=7), seven labels for ***failure consequence probability*** (i.e., J_3=7), described before.

Being *L* the total number of rules, in this case will be used a sample of 245 rules [Liu03].

Step #4: Fuzzy rule-base inference mechanism

Suppose a fuzzy rule-base with the belief structure is given by R= $\{R_1,\ldots, R_L\}$. The kth rule in (1) can be represented as follows:

R_k: IF U is A_k

THEN ***safety estimate*** is *D* with belief degree g_k

where *U* represents the antecedent attribute vector (**FR**, **CS**, **FCP**), A^k the packet antecedents $\{A_1^k, A_2^k, A_3^k\}$ D the consequent vector $(D_1,\ldots, D_N)$, g_k the vector of the belief degrees $(g_{1k},\ldots, g_{Nk})$ and $k \in \{1,\ldots, L\}$.

Once a rule-base is built up, the knowledge contained in it can be used to perform the inference procedure.

In order to reach a safety assessment the fuzzy reasoning system expresses the ***safety estimates*** $S(e_i(a_l))$ as follows for the assessment done by the ith expert e_i on the lth potential cause a_l to a technical failure:

$$S(e_i(a_l)) = \begin{array}{l}(Poor; J_{1i}^l), (Low; J_{2i}^l), (Average; J_{3i}^l,) \\ ; (High; J_{4i}^l,), (Good; J_{5i}^l)\end{array}$$

where e_i represents the ith expert (i=1,…, p) and a_l represents the lth (l=1,…, q) potential cause to a technical failure. J_{ti}^l represents the belief degree to which the safety of al is believed to be assessed to D_t by the expert e_i. The inference procedure is based on fuzzy rule-base and evidential reasoning approach, referred to as a *fuzzy rule-based evidential reasoning approach* – FURBER approach [Liu03]. The final result is still a belief distribution on safety expression, which gives a panoramic view about the safety level for a given input.

In this phase for the synthesis purpose, we transform the safety estimate into a linguistic 2-tuple, i.e., transform the distribution assessment $S(e_i(a_l))$ on the S_S into linguistic 2-tuples over the S_S.

Definition 3. *Let* c_i^l *a function that transforms a distribution assessment in a linguistic term set* S_S *into a numerical value in the interval of granularity of* S_S*, [0, g-1], where g is the cardinality of* S_S*:*

$$c_i^l : S(e_i(a_l)) \qquad [0, g \quad 1]$$

Therefore, applying the D function (Definition 2) to

$$c_i^l(\{(s_t; J_{ti}^l), t = 0,\ldots, g \quad 1\}) = \frac{\sum_{t=0}^{g} t J_{ti}^l}{\sum_{t=0}^{g} J_{ti}^l} = b_i^l$$

b_i^l (i=1,…, p; l=1,…, q) we shall obtain a safety estimate whose values are linguistic 2-tuples (by the lth expert on the lth potential cause to a technical failure), e.g., if b_i^l = 1.2, then its equivalent linguistic 2-tuple representation is:

$$(1.2) = (\text{Low}, 0.2)$$

3.2. COST MODELLING

Cost and safety are two important criteria in design of complex engineering systems, but they are usually in conflict because higher safety normally leads to higher costs. The cost incurred for safety improvement associated with a design option is usually affected by different factors [Wan96].

These factors can be different in each engineering system and often include uncertainties. Therefore, it may

be more appropriate to model cost incurred in safety improvement associated with design options on a subjective basis.

In the literature [Wan95, Wan96] cost was estimated and described using fuzzy sets over the linguistic variables belonging to the linguistic term set, $\mathbf{S_C}$, with seven labels. In our case we shall use linguistic labels belong to a linguistic term set whose linguistic terms have a semantic triangular shaped and symmetrically distributed:

S_C ={*Very Low, Low, Moderately Low, Average, Moderately High, High, Very High*}

3.3. TECHNICAL PERFORMANCE MODELLING

Performance measurement is an area that has become increasingly important, sophisticated and more demanding. So technical performance is taken into account as evaluation criterion to rank the different design options for engineering systems [Sii00]. The technical performance is different in each engineering system and usually include uncertainties. Due to this fact, it is difficult to fix a linguistic term set for measuring technical performance suitable to any engineering system. In this paper we propose the use of a linguistic term set with nine terms, denoted as S_P, to assess the technical performance, but it could be possible to use different linguistic term sets depending on the engineering system.

In our case the experts provide their preferences about the technical performance using the linguistic term set S_P whose syntax and semantics are:

S_P ={*None, Very Unsuitable, Unsuitable, A Little Unsuitable, Suitable, More than Suitable, Very Suitable, Almost Totally Suitable, Perfect*}

At this moment cost, safety and technical performance assessments are expressed by means of linguistic values but in different linguistic utility spaces. In section 4 we shall propose a decision model to deal with this multi-granular linguistic context.

4 DECISION MODEL: RANKING OPTIONS

Our aim is to choose the most suitable design option for an engineering system taking into account the features of safety, cost and technical performance. So far, the assessments of safety are assessed in S_S while the assessments of the cost are assessed in S_C and for technical performance in S_P. Therefore, to rank the options we shall apply a multi-granular linguistic decision model in order to solve our problem. This model is based on the models presented in [Her00a, Her04] to deal with multi-granular linguistic information in decision analysis that consist of two phases:

Aggregation phase: it combines the assessments of safety and cost of the different experts into an overall suitability assessment for each design option. This phase has two steps:

1. Normalization process: it makes the multi-granular linguistic information uniform over a linguistic term set called Basic Linguistic Term Set (BLTS).
2. Aggregation process: it combines the information unified to obtain an overall value of suitability for each design option.

Exploitation phase: it ranks the different design options according to assessments obtained in the aggregation phase by means of a choice degree.

We shall describe in further detail the steps of the decision model used to solve our multi-granular linguistic decision making problem.

Our problem is modelled as a MEMC-DM problem where each expert i, provides assessments for the cost and technical performance and according to his/her opinions the safety assessments are synthesised using the FURBER approach:

Table 1. Expert's assessments

Design Options	*Criteria*		
Expert i	Safety	Cost	Tech. Per.
O_1	(s_{i1}, a)	$(c_{i1}, 0)$	$(p_{i1}, 0)$
:	:	:	:
O_n	(s_{in}, a)	$(c_{in}, 0)$	$(p_{in}, 0)$

Where (s_{ij}, a) are the safety assessments synthesized from the opinions of the expert e_i for the design option o_j, i.e., estimated based on the fuzzy rule-based system produced at lower levels, and then synthesised to obtain the safety assessment of the system by means of linguistic 2-tuples in S_S. While (c_{ij}, a) and (p_{ij}, a) are the assessments for cost and technical performance provided by the expert e_i for each design option o_j, assessed by linguistic values in the linguistic sets S_C and S_P.

4.1. AGGREGATION PHASE

In this phase the individual information of each expert is combined to obtain collective preference values for each design option. But as the criteria are assessed in a multi-granular linguistic context this phase combines the information in two steps.

A) Normalization Process

We are dealing with multi-granular linguistic information, to manage it the model unifies it in a common utility space, the BLTS. In the literature related to engineering systems evaluation [Yng03,Liu03] the utility space used to express the suitability of an engineering system is the following linguistic term set

S_T={Slightly Preferred, Moderately Preferred, Average, Preferred, Greatly Preferred}

So we propose it as BLTS.

Once we have chosen the common utility space to express the suitability of the design options we shall transform all the input assessments into linguistic 2-tuples assessed in the BLTS. To do so firstly, we should notice:

a) Safety assessments are expressed in S_S that is similar to S_T except in the syntax. So the transformation of the safety assessments will consist of using the syntax of the BLTS without any other change.

b) Regarding the cost and technical performance assessments, they are assessed in S_C and S_P linguistic term sets with different granularity, syntax and semantics to the BLTS. So to convert these assessments into linguistic 2-tuples in the BLTS, we shall use the following process based on two different decision models presented in [Her00a, Her04]:

a. *Transforming linguistic values into fuzzy sets in the BLTS*: We use a transformation function which convert any linguistic value assessed S_C or S_P into a fuzzy set in S_T.

Definition 4. *Let* $S = \{s_0, ..., s_p\}$ *and* $S_T = \{c_0, ..., c_g\}$ *be two linguistic term sets, such that, p+1 and g+1 are the granularity of S and* S_T *respectively. Then, a multi-granularity transformation function,* t_{SS_T} *is defined as:*

$$t_{SS_T} : S \quad F(S_T)$$

$$g_k^i = \max_y \min\{m_{s_{ij}}(y), m_{c_k}(y)\}$$

where $m_{s_{ij}}(y)$ *and* $m_{c_k}(y)$ *are the membership functions of the fuzzy sets associated with the terms* s_{ij} *and* c_k*, respectely.*

Using t_{SS_T} such that S will be S_C and S_P we have converted the assessments provided by the experts for cost and technical performance into fuzzy sets in the BLTS. Therefore, at this moment the safety assessments are expressed by linguistic 2-tuples in S_T and cost and technical performance by means of fuzzy sets in S_T. So to unify all the information into linguistic 2-tuples in the BLTS we should transform the fuzzy sets we have just obtained using t_{SS_T} into linguistic 2-tuples.

b. *Transforming fuzzy sets in S_T into linguistic 2-tuples in S_T:* The cost and technical performance assessments expressed by means of fuzzy sets in the BLTS are transformed into linguistic 2-tuples in the BLTS. This transformation is carried out using the c function (Def. 3) and the function (Def. 2). Now all the input assessments (safety, cost and technical performance) are expressed in an uniform way by means of linguistic 2-tuples in the BLTS.

Let us suppose that the expert ***i*** provide the following assessments for cost and technical performance and the synthesised values for safety are the following ones:

Table 2. Input assessments provided by Expert i

	Expert i		
Options	**Safety**	**Cost**	**Tech. Per.**
o_1	**(Poor,0.27)**	**(Moderated High,0)**	**(Suitable,0)**
:	…	…	…
o_n	**(Low,-0.46)**	**(High,0)**	**(Very Suitable,0)**

After the normalization process we shall obtain the following input assessments:

Table 3. Input assessments expressed by linguistic 2-tuples in the BLTS

Options	*Experts*		
	Safety	**Cost**	**Tech. Per.**
o_1	**(** s_0^5 **,0.27)**	**(** s_3^5 **,-0.26)**	**(** s_2^5 **,0.01)**
:			
o_n	**(** s_1^5 **,-0.46)**	**(** s_3^5 **,0.25)**	**(** s_3^5 **,-0.06)**

This transformation is applied to all the expert opinions.

B) Aggregation Process

This process combines the assessments that express the values for the different criteria to obtain a global value for each design option. We want to obtain an evaluation value for each design option according to its values for cost, technical performance and safety expressed by means of linguistic 2-tuples in the BLTS. This global value will be expressed with the syntax of S_T. In this case we propose the use of the 2-tuple weighted aggregation operator [Her00]. We have a set of pairs of assessments $\{(s_i, a), (c_i, a), (t_i, a)\}$ S_T for each design option. Taking into account Remark 1, since the cost assessments have a decreasing interpretation for the suitability, the aggregated value for each design option is obtained using the following expression:

$$W_AM^*((s_i, a), (c_i, a), (t_i, a)) =$$

$$\left({}^1(s_i, a) \cdot w_s + {}^1(Neg(c_i, a)) \cdot w_c + {}^1(t_i, a) \cdot w_p \right)$$

Where $Neg(c_i, a)$ is the assessment for the cost of design option *i* taking into account its decreasing interpretation and (s_i, a), (t_i, a) are the assessments for the safety and technical performance of the option *i* respectively. And w_s, w_c, w_p such that $w_s + w_c + w_p = 1$ being the importance for each criterion safety, cost and technical performance respectively. Suppose a value of $w_s = 0.6$. $w_c = 0.2$ and $w_p = 0.2$. From Table 3 we have:

Table 4. Design Options Utility Assessments

Design Options	
Options i	**Utility**
o_1	**(Moderated Preferred, 0.01)**
:	:
:	:
o_n	**(Moderated Preferred, 0.24)**

At this moment we have a suitability value of each design option expressed by means of a linguistic 2-tuple in S_T for each expert. To obtain a global suitability assessment for each design option we shall apply another aggregation operator to the global assessments of all experts. Now, we could consider that all the experts are equally important (arithmetic mean) or we could assign different weights to each expert (weighted average).

4.2. EXPLOITATION PHASE

Finally the decision process applies a choice degree to obtain a selection set of alternatives. Different choice functions have been proposed in the choice theory literature [Rou89]. The choice functions rank the alternatives according to different possibilities and from the ranking the best one/s are obtained.

In our problem the information is expressed by means of the linguistic 2-tuple representation model that has defined a total order over itself. Then in our problem we shall order the results using this order. If there is an only expert then from Table 4 we could infer that the best design option is $\mathbf{o_n}$.

5 CONCLUSIONS

In this contribution we have presented an evaluation approach for design assessment of complex engineering systems based on a linguistic decision model. The use of a linguistic model is due to the fact that it is difficult to deal with vagueness and uncertainty using traditional probabilistic models and tools. The linguistic assessment approach provides a useful and natural way to support the solution of such complex decision problems. Our proposal is based on the evaluation of cost, technical performance and safety and can deal with multi-granular linguistic information. In the safety assessment, a fuzzy rule-base with the belief structures is used to capture uncertainty and nonlinear causal relationships in safety assessments. The evaluation is implemented by means of a multi-granular linguistic MEMC decision model.

ACKNOWLEDGMENTS

This work has been partially supported by the Research Project TIC 2002-03348, by the "Secretaría de Estado de Educación y Universidades" through the "Programa Nacional de ayudas para la movilidad de profesores de universidad e investigadores españoles y extranjeros" and by the UK Engineering and Physical Sciences Research Council (EPSRC) under Grant No: GR/R30624

REFERENCES

[Her00] F. Herrera, L. Martínez. A 2-tuple Fuzzy Linguistic Representation Model for Computing with Words. *IEEE Transactions on Fuzzy Systems* **8:6** pp.746-752, (2000).

[Her00a] F. Herrera, E. Herrera-Viedma, L. Martínez. A fusion approach for managing multi-granularity linguistic term sets in decision making. *Fuzzy Sets and Systems*. **114** 43-58 (2000).

[Her00b] F. Herrera, L. Martínez. An Approach for Combining Numerical and Linguistic Information based on the 2-tuple fuzzy linguistic representation model in Decision Making. Int. Journal of Uncertainty, *Fuzziness and Knowledge–Based Systems*. **8:5** pp. 539-562 (2000).

[Her04] F. Herrera, E. Herrera-Viedma, L. Martinez, F. Mata, P.J. Sanchez. A multi-granular linguistic decision model for evaluating the quality of network services, in: Int. Sensory Evaluation: Methodologies and Applications, Ruan Da, Zeng Xianyi (Eds.) Springer, pp. 71-92, 2004

[Liu03] J. Liu, J.B. Yang, J. Wang, H.S. Sii and Y.W. Wang , Fuzzy rule-based evidential reasoning approach for safety analysis, *International Journal of General Systems*, In press (2003).

[Rou89] M. Roubens, Some properties of choice functions based on valued binary relations, *European Journal of Operational Research* 40 pp. 309-321, (1989).

[Sii00] H.S. Sii and J. Wang, A subjective design for safety framework for offshore engineering products, Workshops on reliability and Risk Based Inspection Planning and ESRA Technical Committee on Offshore Safety, 14-15 December, Zurich, Switzerland (2000).

[Wan95] J. Wang, J.B. Yang, P. Sen, Safety analysis and synthesis using fuzzy sets and evidential reasoning. *Reliab. Engineeting System Safety* 41 pp. 103-118 (1995).

[Wan96] Wang J., Yang J. B., and Sen P., Multi-person and multi-attribute design evaluations using evidential reasoning based on subjective safety and cost analyses, *Rel. Eng. and System Safety*, 52:2, pp. 113-129 (1996).

[Wan97] Wang J., and Ruxton T., Design for safety, *Jou. of Ame. Society of Safety Engineers*, pp. 23-29 (1997).

[Yng03] J. B. Yang, J. Liu, J. Wang and H. S. Sii, A generic knowledge-base inference methodology using the evidential reasoning approach – RIMER, Submitted to *IEEE Tran. on Systems, Man, and Cybernetics* (2003).

[Zad75] L.A. Zadeh, The Concept of a Linguistic Variable and Its Applications to Approximate Reasoning. Part I, *Information Sciences* **8** 199-249, Part II, *Information Sciences* **8** 301-357, Part III, *Information Sciences* **9** 43-80, (1975).

A decision model for treatment given imprecise clinical diagnoses

Lluís Godo, Josep Puyol-Gruart, Sandra Sandri
Institut d'Investigació en Intel·ligència Artificial, CSIC
08193 Bellaterra, Spain
{*godo,puyol,sandri*}*@iiia.csic.es*

Pilar Barrufet
Hospital de Mataró
09304 Mataró, Spain
pbarrufet@csm.scs.es

Abstract

The goal of this paper is to discuss a model of clinical problem-solving tasks of treatment given an imprecise diagnosis. We propose a general framework that incorporates concepts of causal relations, abductive inference and elements of fuzzy decision theory. The model is exemplified for a pneumonia treatment problem.

Key Words: Diagnosis, Abduction, Treatment

1 Introduction

Assigning a treatment to a patient, even once a diagnosis has been reached, is not a trivial task. One has to consider the context of the patient (age, sex, etc) to determine the medicines, their doses and the amount of time they should be taken. Also, very often the diagnosis for a given patient may be only imperfectly known. On the one hand, a diagnosis may be imprecise, as when only the type of infection is known to be affecting the patient. On the other hand, the diagnosis may be uncertain, when a series of competing hypotheses may equally explain the patient's symptoms. Last but not least, the drugs themselves may interact and so their administration have to be thought of conjunctively. On the overall, one has to propose a treatment that covers all diagnostic possibilities while aiming at avoiding causing new problems to the patient.

Some systems for treatment assignment are known in the literature. The first is probably Mycin [12], for diagnosis and treatment of infectious diseases. TerapIA [2] is a system for treatment of pneumonia infection developed using Milord [9]. In both Mycin and TerapIA, treatment knowledge is modeled by rules of the type IF diagnosis(patient) THEN treatment(patient), and the reasoning parts from the diagnosis, along with the patient context, to the indicated treatment. Other systems are based on frames (HELP [4]), case-based reasoning (ICONS [10, 11]), or probabilistic networks [5].

On the other hand, an abductive approach for solving therapy problems in the plant-health domain, based on the CommonKADS methodology, is described in [14]. Recently, a very related framework for diagnosis, investigation and treatment has been proposed by Massruhá [7], based on Parsimonious Covering Theory (PCT) [8], with focus in diseases that attack plantations [6]. In this framework, in what concerns treatment, knowledge is modeled as graphs in which the arcs are from treatments to diagnoses, along with context information containing (weighted) indications and contraindications, and inference is also abductive.

In this work we propose a framework for treatment that combines ideas taken from [2] and [7], and is primarily aimed at the treatment of human infectious diseases. The knowledge is structured in three components.

(i) The first one links treatments and causal agents, indicating the pharmacological adequacy of a treatment to combat a causal agent. Links are weighted by effectiveness (or adequacy) degrees and inference is abductive. There is also ontological knowledge relating treatment agents with similar chemical structure and properties. This first component mainly consists of pharmacological/chemical knowledge.

(ii) A second component structures the knowledge about compatibility or contextual adequacy (as opposed to risk) of treatments according to different characteristics of the subject to be treated. This kind of knowledge is a combination of medical and pharmacological knowledge and may be imperfect, i.e. tainted with imprecision, vagueness and uncertainty. In this component inference is deductive.

(iii) Finally another component links causal agents and disorders, and is needed for abductive reasoning in the diagnosis phase. This component however, mainly consisting of medical causal knowledge, will not be paid attention in this paper since our working hypothesis here is that the disorder is known and there is partial knowledge about its possible causal agents.

This structured knowledge is meant to be used, given a set of subject characteristics and a partially known diagnosis, to provide a set of alternative treatments, ranked by a global adequacy degree.

This work is organized as follows. In Section 2, we present the components of our generic treatment model and in Section 3 we present the case information representation and the inference mechanism. Section 4 describes an illustrative example and Section 5 finally brings the conclusions.

2 A generic treatment model: components

Treatments for infectious diseases depend first of all on the micro-organism causing the disease (different treatment agents are active for different micro-organisms) but also on particular patient characteristics (for instance, a patient may be allergic to some antibiotics).

In what follows we assume to work with a set D of disorders (diseases), a set T of treatment agents (antibiotics) grouped into a hierarchy of classes H (we assume $T \subseteq H$), a set A of causal agents (micro-organisms) and a set P of subject clinical parameters. Each of the parameters refers to a variable x, defined on a domain U_x. Examples of pairs (x, U_x) that can be found in P are $(gender, \{male, female\})$, $(age, [0, 100])$, $(pregnancy, \{yes, no\})$, etc.

In our model, the knowledge base KB has three components. A first component KB_{pharm} deals with the pharmacological knowledge about the adequacies (or effectiveness or sensitivity) of treatments for micro-organisms and ontological knowledge about treatment agent classes. The former will be given by a weighted directed graph, called *sensitivity graph*, with a set of links $Y \subseteq T \times A$ from treatment nodes to causal agent nodes. The weight of a link (t_i, a_j) will be a value belonging to a scale V (e.g. [0, 1]) and denoted $pharm_adequacy(t_i, a_j)$. Actually, only links with a non-zero weight are considered as such. The latter will be simply described by a hierarchy of treatment agent classes, according to their common chemical or pharmacological properties.

The two graphs in KB_{pharm} can be coupled with a third graph putting into relation causal agents nodes of A with disorder nodes of D, corresponding to the third component mentioned in the previous section and that will call KB_{causal}. Since diagnosis is not the focus of the paper we will assume a very simple model, namely a one-to-many relationship. As a whole we have the following formalization.

Definition 2.1 *The knowledge bases KB_{pharm} and KB_{causal} are associated to graphs represented by the 5-tuple $< T, H, A, Y, K >$ and the 3-tuple $< A, D, Z >$ respectively, where:*

- *$Y \subseteq T \times A$ is a weighted binary relation such that each pair $< t_i, a_j > \in Y$ means that t_i can combat a_j with an adequacy degree $pharm_adequacy(t_i, a_j) \in V$.*
- *K is a hierarchy of class/subclass relationships where the leaf nodes are treatment agents in T, the rest of nodes are elements from $H \setminus T$ and such that a pair $< h_i, h_j > \in K$ means that h_i is a subclass of h_j.*
- *$Z \subseteq A \times D$ is a binary relation such that each pair $< a_j, d_k > \in Z$ means that agent a_j can cause disorder d_k.*

The third component KB_{contx} of the knowledge base refers to knowledge about risk (or incompatibility) of treatments with several contextual subject characteristics. Actually we will assume $KB_{contx} = \cup_{h \in H} KB^h_{contx}$, where each KB^h_{contx} contains the description of the adequacy of a treatment class h (in particular h may be a treatment agent of T) to different subject clinical characteristics. The working hypothesis, according to the clinical guidelines, is that the adequacy (or better, the incompatibility) of a class of treatments is *inherited* by each of the instances or subclasses. Each KB^h_{contx} will be under the form of fuzzy rule base where rules are of the form

$$\text{If } x_1 = \tilde{F}_1 \text{ and} \ldots \text{and } x_n = \tilde{F}_n \text{ then } contx_adequacy(h) = \tilde{B}$$

where the x_i's are context variables in P with $\tilde{F}_i$'s being fuzzy sets defined on the universe U_{x_i} of x_i, and $\tilde{B}$ is a fuzzy set defined on the scale V. Examples of the kind of rules about the adequacy of using a treatment belonging to a class h are :

$\mathbf{r}_1$: If $gender = female$ and $age = medium$ and $severity = low$ then $contx_adequacy(h) = high$

$\mathbf{r}_2$: If $age = old$ and $severity = high$ then $contx_adequacy(h) = low$

The inference mechanism in each fuzzy rule base KB^h_{contx} is the traditional one in *implicative* fuzzy rule systems, that is, opposed to Mamdani's inference model, a proper implication implication function is used to model fuzzy rules and the outputs are aggregated conjunctively.

3 Case information, abductive and deductive inference

A case (information about a particular subject) will be represented by a triple $C = \langle d, A', P' \rangle$ where $d \in D$ is the known disorder of the subject, $A' \subseteq A$ represents an imprecise diagnosis for d, meaning that the agent causing disorder d is in A'. P' contains the available clinical data of the subject under the form of a set $\{x_i = G_i\}_{i \in I}$ where $x_i \in P$ and G_i can be thought in general as a fuzzy set over the domain U_{x_i} of the variable x_i.

A solution for C will be a set of alternative therapeutic treatments, ranked by an *adequacy* degree. Each of these therapeutic treatments T' will be composed of a set of individual treatments, i.e. $T' \subseteq T$, satisfying a number of requirements, which can be basically enumerated as follows:

1. T' is a cover for A', i.e. the treatment agents in T' are effective for *all* the causal agents in A'.
2. T' is compatible with P', i.e. *all* treatment agents in T' are compatible with *all* patient's characteristics described in P'.

The notion of *cover* can be formalized as follows. Let us consider a knowledge base KB_{pharm} with a sensitivity graph given the tuple $< T, A, Y >$.

Definition 3.1 *We define:*

$indications(t) = \{a \in A \mid < t, a > \in Y\}$, $\forall \in T$
$indications(T') = \bigcup_{t \in T'} indications(t)$, $\forall T' \subseteq T$
$treatments(a) = \{t' \in T \mid < t, a' > \in Y\}$, $\forall a \in A$
$treatments(A') = \bigcup_{a \in A'} treatments(a)$, $\forall A' \subseteq A$.

The set $indications(t)$ (respec. $indications(T)$) contains all causal agents that may be combated, by treatment agent t (respec. the set of treatment agents $t \in T'$). The set $treatments(a)$ (respec. $treatments(A)$) contains all treatment agents that may combat causal agent a (respec. the set of causal agents $t \in T'$).

Let $A' \subseteq A$ be a set of causal agents in a given case. When A' is not a singleton, it might be the case that all the agents in A' are affecting the patient. In most cases, however, this situation occurs when there is no conclusive information about the identity of the (single but unknown) causal agent. In the case of infectious diseases, for instance, it is often the case that the physician can assert that the micro-organism causing the disease one of a set with 2 to 4 alternatives (see [1]). In this framework it is rarely the case that more than one the disease-related micro-organism is present in the same patient (even though the patient may be suffering from other diseases, including ones of infectionary nature as well). In any case, no matter whether A' is interpreted conjunctively or disjunctively, we shall treat them in the same manner.

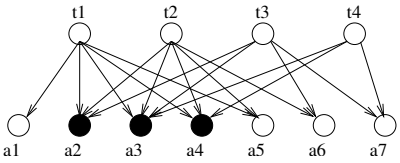


Figure 1: Illustration of a sensitivity graph Y and a case $A' = \{a_2, a_3, a_4\}$.

Definition 3.2 *Let $A' \subseteq A$ be a set of causal agents. A set of treatment agents $T' \subseteq T$ is a cover of A' iff $A' \subseteq indications(T')$.*

Covers can be classified according to different parsimony criteria [8].

Definition 3.3 *Let T' be cover for A'. We say that*

- *T' is* relevant *if it is a subset of $treatments(A')$;*
- *T' is* irredundant *if none of its proper subsets is also a cover of A', it is redundant otherwise;*
- *T' is* minimal *if its cardinality is smallest among all covers of A'.*

Thus, every minimal cover is irredundant and every irredundant cover is relevant. As an example, let us consider the sensitivity graph depicted in Figure 1, with $T = \{t_1, ..., t_4\}$ and $A = \{a_1, ..., a_7\}$, and the set of causal agents $A' = \{a_2, a_3, a_4\}$. In this case, all the covers of A' are relevant, and the set of irredundant and minimal covers are respectively $\{\{t_1\}, \{t_2\}, \{t_3, t_4\}\}$ and $\{\{t_1\}, \{t_2\}\}$.

Finally we come to inference. Given a knowledge base $KB = KB_{pharm} \cup KB_{causal} \cup KB_{contx}$, where $KB_{pharm} \cup KB_{causal}$ are described by graphs $< T, H, A, Y, K >$ and $< D, A, Z >$ (according to Definition 2.1), and a case $C = \langle d, A', P' \rangle$, the task is to provide a solution in terms of a set of alternative treatments, ranked by an adequacy degree.

The first step is to compute the set $Cov(A')$ of covers of A'. Then, given the adequacy degrees

$pharm_adequacy(t, a) \in [0, 1]$ of each treatment agent $t \in T$ for each causal agent $a \in A$ provided by KB_{pharm}, we compute the adequacy degree of each cover $T' \in Cov(A')$ to the case C as follows:

$$adequacy(T' \mid C) = \\ pharm_adequacy(T' \mid A') \bigotimes contx_adequacy(T' \mid P')$$

where $pharm_adequacy(T' \mid A')$ is the adequacy degree of the set of treatment agents T' for combating all the causal agents in A' and $contx_adequacy(T' \mid P')$ is the adequacy degree of all treatments T' to the contextual patient conditions P'. In our model these degrees are computed in turn as follows:

$$pharm_adequacy(T' \mid A') = \\ \bigotimes_{a \in A'} pharm_adequacy(T' \mid a) \ ,$$

$$contx_adequacy(T' \mid P') = \\ \bigotimes_{t \in T'} contx_adequacy(t \mid P') \ .$$

In these expressions, we use

$$pharm_adequacy(T' \mid a) = \\ \bigoplus_{t \in T'} pharm_adequacy(t, a),$$

and $contx_adequacy(t \mid P')$ is the (defuzzied if necessary) output of processing the fuzzy rule base KB^t_{contx} with the inputs P'. In all these above expressions, $\otimes$ stands for a conjunctive operator on V (t-norm like) while $\oplus$ stands for a disjunctive operator (t-conorm like) also in V.

Notice that in this description we have not been concerned with algorithmic efficiency issues. It is often the case that if KB_{contx} contains very powerful filters that may discard a number of treatment classes for some particular clinical contexts (patient data), then it is more efficient to compute first $contx_adequacy(T' \mid P')$ and for the computation of $pharm_adequacy(T' \mid A')$ take only into account those t which have not been discarded by P', i.e. those with a degree $contx_adequacy(t \mid P')$ not null.

Finally the solution to the case $C = (d, A', P')$ given the knowledge base $KB = KB_{pharm} \cup KB_{causal} \cup KB_{contx}$ will be the set of alternative graded covers

$$\{\langle T', adequacy(T' \mid C)\rangle \mid T' \in Cov(A')\},$$

from which a final choice by the user (a physician) can be made.

4 Case example of antibiotic treatment advice for a pneumonia disease

Pneumonia infection can be caused by a lot of microorganisms. The etiological diagnosis depends on the identification of the microorganism causing pneumonia. Nowadays, with the available diagnostic methodology, it is still very difficult to determine which of the microorganisms is the infecting agent in a particular pneumonia case. The research focused to determine which are the microorganisms causing pneumonia only succeeds in the 50% of the cases [3, 13]). Despite the uncertainty of the diagnosis, a treatment has to be speedily administrated. The goal of a recommending treatment system is to find a good cover for the all microorganisms that possible causes pneumonia in a concrete case. A treatment is an individual antibiotic or a combination of antibiotics, normally two.

Terap-IA [2] is a support system for recommending antibiotic treatment of pneumonia infection. Terap-IA contains two main sorts of knowledge: pharmacological and patient context knowledge. The former is the ontological knowledge about antibiotics and the adequacy of the antibiotics to treat different microorganisms. The patient context knowledge contains a lot of data about the patient—diagnosis, clinical history, physical examination, laboratory test and chest radiograph data, renal function, severity of illness, allergic reactions, genetics alterations and so on—and the adequacy of this data with the possible treatments. This systems starts considering a given previous diagnosis of pneumonia, that is, a set of microorganisms that possible cause the pneumonia of the patient.

In this section we deal with a simplified case of devising a treatment therapy for a pneumonia infection using our model. This example is based on the knowledge contained in Terap-IA. The concrete example is borrowed from the validation cases of Terap-IA (patient number 16 in [1]). The components of the knowledge used in this example are the following:

Ontological and sensitivity knowledge (KB_{pharm}): The ontological knowledge about antibiotics is specified by a semantic network K that represents a hierarchy of antibiotics groups that share important chemical properties. The sensitivity graph Y relates antibiotics with microorganisms. The links contain the degree of adequacy of an antibiotic to treat a microorganism only from the pharmacological point of view, without taking into account any information about clinical data of the patient.

Causal knowledge (KB_{causal}): In our example this component does not play a role since we are assuming the disease (pneumonia) is known.

Clinical Knowledge (KB_{contx}): It is described by a fuzzy rule base with knowledge about the risk of a particular antibiotic or group of antibiotics given the clinical data of the patient.

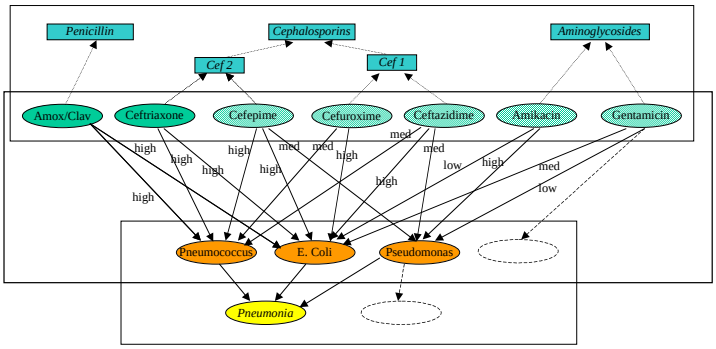


Figure 2: Example for pneumonia treatment.

In this example we consider the following sets of antibiotics T, antibiotic groups H, microorganisms A, diseases D, and clinical parameteres P:

$T = \{Amox/Clav, Gentamicin, Cefuroxime,$
$Cefepime, Ceftriaxone, Amikacin, Ceftazidime\}$
$H = T \cup \{Penicillin, Cephalosporins, Cef1,$
$Cef2, Aminoglycosides\}$
$A = \{Pneumococcus, E.Coli, Pseudomonas\}$
$D = \{Pneumonia\}$
$P = \{allergy_to_penicillin(yes/no), pregnancy(yes/no)\}$

Figure 2 presents a graph representing the ontological, sensitivity and causal knowledge of this example. The clinical knowledge is not considered here. Note that it also contains empty nodes representing—for generality—other diseases and other causal agents and their relations. The meaning of the labels in the links in the sensitivity graph—among antibiotics and microorganisms—are values from the set $V = \{null < low < medium < high\}$ that represent qualitative adequacy levels. For instance:

$$pharm_adequacy(Cefepime \mid Pneumococcus) = high$$

Let us suppose we have a case $C_1 = \{Pneumonia, A', P'_1\}$ of a patient diagnosed with pneumonia, caused by a microorganism in $A' = \{Pneumococcus, E.Coli, Pseudomonas\}$. Given this information and the knowledge base fragment for pneumonia described in Figure 2, let us consider the set $Cov_I(A')$ of irredundant covers (treatments) for A' and calculate their corresponding pharmacological adequacy degree according to the procedure in Section 3 (see Table 1 for the results). Here we have taken $\bigotimes = \min$ and $\bigoplus = \max$. Let us suppose that the patient is allergic to penicillin, $P'_1 = \{(allergy_to_penicillin = yes), (pregnancy = no)\}$. Then applying this information to rule base

$$KB_{Penicillin} = \{r_1 : \text{If } allergy_to_penicillin = yes \text{ then } contx_adequacy(Penicillin) = null\},$$

we derive that all antibiotics contained in class *Penicillin* are completely inadequate to treat the patient ($contx_adequacy(Penicillin \mid P') = null$). By the inheritance mechanism we derive that *Amox/Clav* has a null adequacy to treat the patient; all the other antibiotics have a high adequacy by default, since the rules in the other rule bases do not fire. In consequence, all treatments in $Cov_I(A')$ containing *Amox/Clav*, i.e. treatments T'_1 and T'_5 in Table 1, obtain the adequacy value *null*.

Let us now suppose that we have another case $C_2 = \{Pneumonia, A', P'_2\}$ of another patient, also suffering from pneumonia caused by a microorganism in A', but this time a pregnant woman, $P'_2 = \{(allergy_to_penicillin = no), (pregnant = yes)\}$. Applying this information to rule base

$KB_{Cef2} =$
$\{\ r_2$: If $pregnancy = yes$ then $contx_adequacy(Cef2) = null;$
r_3: If $pregnancy = yes$ then $contx_adequacy(Aminoglycosides) = low\}$

we derive that all antibiotics in Cef_2 (a subclass of Cephalosporins that is contraindicated during pregnancy) have a context adequacy equal to *null*, while all antibiotics in the class *Aminoglycosides* have a *low* context adequacy. Thus, all treatments in $Cov_I(A')$ containing antibiotics Cefepime and/or Ceftriaxone, i.e. treatments T'_2, T'_3 and T'_6 in Table 1, obtain the context adequacy value *null*.

From Table 1 we verify that the minimal covers (T'_1 and T'_2) are not the best treatments for case C_1, moreover T'_1 is completely inappropriate for case C_2. On the other hand, for case C_1 we have been able to find a completely suitable treatment that represents an irredundant cover (T'_4). However, for case C_2, none of covers is completely appropriate. In this particular case, the most appropriate treatment is the minimal cover $T'2$, with degree *medium*. The choice of par-

$Cov(A')$	treatment agents	$pharm_adequacy(T'_i \mid A')$	$adequacy(T'_i \mid C_1)$	$adequacy(T'_i \mid C_2)$
T'_1	Cefepime	medium	medium	null
T'_2	Ceftazidime	medium	medium	medium
T'_3	Amikacin, Amox/Clav	high	null	low
T'_4	Amikacin, Ceftriaxone	high	high	null
T'_5	Amikacin, Cefuroxime	medium	medium	low
T'_6	Amox/Clav, Gentamicin	low	null	low
T'_7	Ceftriaxone, Gentamicin	low	low	null
T'_8	Cefuroxime, Gentamicin	low	low	low

Tabla 1: Pharmacological and global adequacies for irredundant covers.

simony criterium depends on what is the goal of the decision maker; for instance, a doctor may prefer to minimize the number of medicaments taken by a patient, even if the treatment is not the most effective, or to maximize the efficiency of the treatment, even if that means increasing the number of medicaments. For that matter, other factors may also need to be taken into account, such as adverse effects, cost and rapidity of action.

5 Conclusion

In this work we have proposed a framework that, assuming a partially known diagnosis of a disease affecting a given patient, takes into account both pharmacological and patient characteristics to assign treatments. To order the treatments, the framework uses both deductive and abductive reasoning.

We have however not addressed certain important issues, such the possibly harmful interaction between treatment agents. Also, we have only used the irredundancy parsimony criterium, that aims at decreasing the size of the set of treatment agents used to combat a disease. However, other criteria can be used, for instance, when the goal is only to augment the overall efficiency of a treatment. For example, in case C_2 above, the redundant cover $\{Amox/Clav, Amikacin, Cefuroxime\}$ has an global adequacy degree *high*, thus it is more adequate than any irredundant cover. Moreover, considerations such as cost could direct the choice of other criteria.

Finally, note that although this work is focused in human infectious disorders, the framework can be also applied to other fields, such as control of phytopathogens attacking a plantation.

References

[1] P. Barrufet. *TERÀP-IA: Sistema expert d'ajuda al tractament de pneumonies.* PhD thesis, Universitat Autònoma de Barcelona, 2000.

[2] P. Barrufet, J. Puyol-Gruart, and C. Sierra. Terap-IA: A knowledge-based system for pneumonia treatment. In *Proc. of the Intl. ICSC Symp. on Engineering of Intelligent Systems, EIS'98*, vol. 1, pp. 176–182, La Laguna (Spain), 1998.

[3] J. Almirall et al. Epidemiology of community-adquired pneumonia in adults: a population-based study. *European Respiratory Journal*, 15:757–763, 2000.

[4] G.J. Kuperman, R.M. Gardner and T.A. Pryor. *HELP: A Dynamic Hospital Information System.* Springer-Verlag, New York, 1991.

[5] L. Leibovici, M. Fishman, HC. Schnheyder, C. Riekehr, B. Kristensen, and S. Andreassen. A causal probabilistic network for optimal treatment of bacterial infections. *IEEE Transactions on Knowledge and Data Engineering*, pp. 517- 528, 2000.

[6] S. Massruhá, S. Sandri, and J. Wainer. Fuzzy covering theory: an alternative approach for diagnostic problem-solving. In *Proc. 4th European Conf. on Information Technology in Agriculture, EFITA 2003*, pp. 768–775, Budapest (Hu), 2003.

[7] S. Massruhá. *Uma Teoria de Coberturas Nebulosas para Diagnóstico, Investigação e Tratamento.* PhD thesis, CAP/INPE, SJCampos (Br), 2003.

[8] Y. Peng and J. A Reggia. *Abductive Inference Models for Diagnostic Problem-Solving.* Springer-Verlag, 1990.

[9] J. Puyol-Gruart and C. Sierra. Milord II: a language description. *Mathware and Soft Computing*, 4(3):299–338, 1997.

[10] R. Schmidt, B. Pollwein and L. Gierl. Case-based reasoning for antibiotics therapy advice. In *Proc. of ICCBR-99*, pages 550–559. Springer-Verlag, 1999.

[11] R. Schmidt and L. Gierl. Case-based reasoning for antibiotics therapy advice: An investigation of retrieval algorithms and prototypes. *Artificial Intelligence in Medicine*, 23(2):171–186, 2001.

[12] E. H. Shortliffe. *Computer Based Medical Consultations: MYCIN.* American Elsevier, New York, 1976.

[13] A. Torres, J. Serra-Batlles, A. Ferrer, and P. Jimenez. Severe community-acquired pneumonia. Etiology prognosis and treatment. *Am Rev Respir Dis*, (144):312–318, 1991.

[14] S. Túnez , I.M. Águila and R. Marín. An expertise model for therapy planning using abductive reasoning. *Cybernetic and Systems*, (32):829–849, 2001.

CONTROLADOR FUZZY DE VELOCIDAD DE MOTORES ASINCRONOS CON DESLIZAMIENTO CONSTANTE

Francisco Plaza Pérez
Universidad de Valladolid
Dto: Tecnología Electrónica
fran@tele.uva.es

Luis Carlos Herrero de Lucas
Universidad de Valladolid
Dto: Tecnología Electrónica
lcherrer@tele.uva.es

Fernando Martínez Rodrigo
Universidad de Valladolid
Dto: Tecnología Electrónica
fer_mart@tele.uva.es

Resumen

De todos es sabido que los motores asíncronos son ampliamente utilizados en la industria por su construcción simple, robusta y bajo mantenimiento, pero que tienen por el contrario un control complejo debido a comportamientos difíciles de modelar para un controlador convencional.

En el presente artículo se va a describir un sistema que gracias a un controlador específico Fuzzy de STMicroelectronics (ST52E430) es capaz de realizar un control de velocidad de motores asíncronos de baja potencia y optimizar la tensión de alimentación de éste para minimizar las pérdidas en función del par aplicado.

El sistema está compuesto por dos tarjetas (figuras 1 y 2): una tarjeta de control que es independiente del tipo de motor (potencia, tensión, etc.) que controla el display, teclado y genera los pulsos PWM para el control del motor y otra tarjeta de potencia que incluye los drivers y el inversor (CC-CA) que se encarga de controlar el motor. En este artículo se describirá únicamente la tarjeta de control.

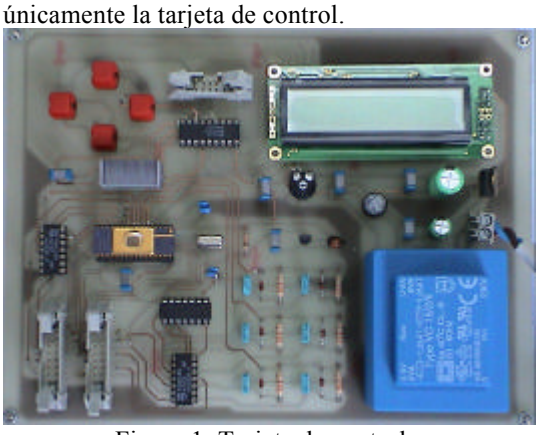


Figura 1: Tarjeta de control.

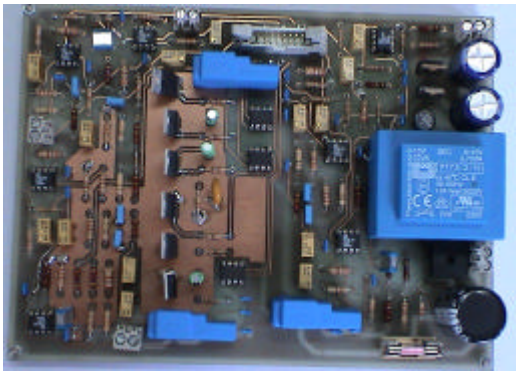


Figura 2: Tarjeta de potencia.

Palabras Clave : Control, Motores, ST52E430, Fuzzy.

1 CONTROL DE MOTORES TRIFÁSICOS ASÍNCRONOS

1.1. CARACTERÍSTICA PAR/VELOCIDAD DE LOS MOTORES ASINCRÓNICOS TRIFÁSICOS

En la figura 3 se representa la relación del Par con el Par nominal como función de la velocidad del rotor y la frecuencia del deslizamiento (fs)[1]. Para valores bajos de fs, el par varía linealmente con fs, véase la línea trazada en negrita en la zona estable (fig. 3). El par máximo que el motor puede producir está representado por el punto "Par máximo".

Si un motor de inducción se conecta con la fuente de alimentación y el esfuerzo de torsión de la carga es más bajo que el par de arranque, (con deslizamiento máximo s=1), entonces el motor podrá arrancar y entrar en la zona estable. Entonces la intersección de la característica del par del motor con el par de la carga, determina el punto de estado estacionario de trabajo del motor. Si el par de la carga alcanza el par máximo, el

[1] Diferencia entre la frecuencia del estator f y la frecuencia del rotor fr.

motor entra en el dominio "break down" (zona inestable) y se para.

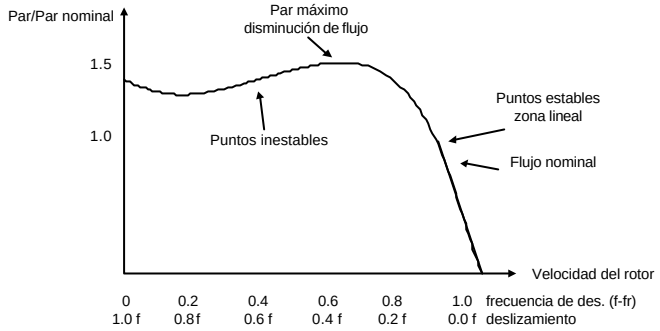


Figura 3: Diagrama par-velocidad con frecuencia y Tensión constantes.

1.2. CONTROL DE VELOCIDAD

La velocidad del rotor se puede controlar variando la frecuencia (f) del estator con el convertidor de energía. Para controlar la velocidad síncrona y por lo tanto la velocidad del motor, manteniendo el flujo constante, es necesario variar la tensión del estator de forma proporcional con las variaciones de la frecuencia del estator (fig. 4).

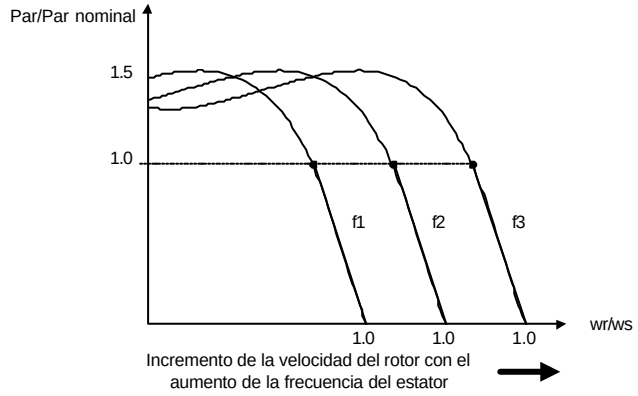


Figura 4: Control de velocidad variando la frecuencia del estator.

La característica del par con la velocidad varía horizontalmente, según lo mostrado en la figura 4 (se han representado varios valores de f). Notar que, en una carga-par constante, la frecuencia del deslizamiento (fs) es constante.

1.2.1. Control de velocidad con V/F constante.

El método más simple de control es mantener constante el flujo (V/F) con los convertidores de energía que varían la velocidad del motor. Esta regulación se llama control constante del par. Si cambiamos la frecuencia, también la tensión aplicada debe variar de una manera lineal para mantener constante V/F, es decir el flujo, ya que este cociente es dimensionalmente un flujo (fig.5).

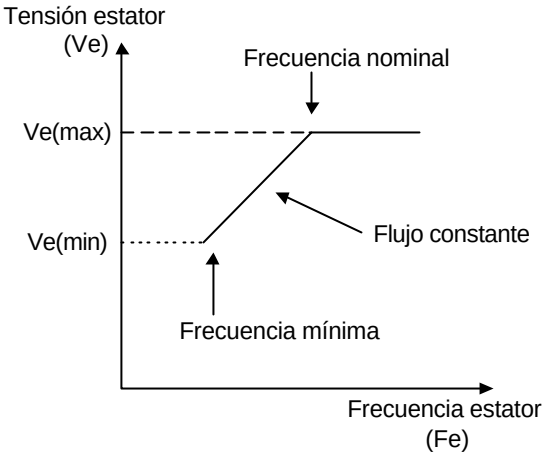


Figura 5: Relación Tensión frecuencia del estator.

Las características de V/F son las siguientes:

- El motor trabaja con flujo constante.
- El motor da siempre el par máximo.
- La eficacia no se optimiza.

En esta aplicación se ha realizado un control que minimiza el flujo, en vez del método de V/F constante. De esta manera, es posible alcanzar una buena eficiencia y regulación del par.

1.2.2. Optimización de la eficacia del motor con control del deslizamiento.

La eficacia del motor puede ser mejorada controlando la tensión del estator, manteniendo el deslizamiento constante con un flujo mínimo (minimización del flujo). Adaptando el flujo teniendo un deslizamiento superior al nominal, pero no lo suficientemente grande como para alcanzar el pull-out (zona inestable) que pararía el motor.

Las pérdidas de potencia del motor pueden ser minimizadas manteniendo constante el deslizamiento, y ajustando el par disponible (Fig.6).

Este método de control variando la tensión del motor ofrece posibilidades limitadas de regulación de la velocidad. Sin embargo, combinando el control de la tensión (flujo mínimo) y el control de la frecuencia, el motor se controla bien en un amplio rango de velocidades.

Este es el tipo de control que se ha aplicado en el sistema diseñado.

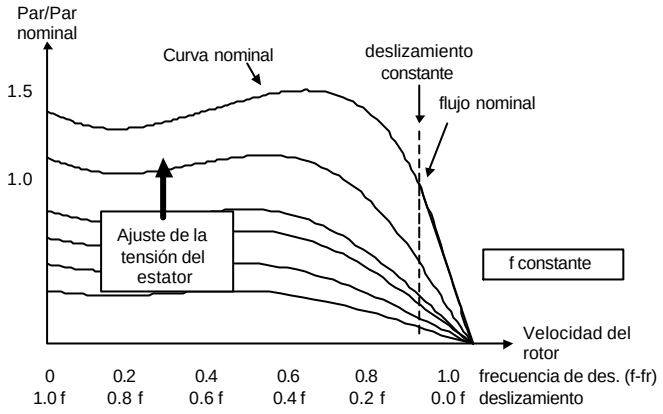


Figura 6: Características del motor variando la tensión del estator.

2 ESTRUCTURA DEL EQUIPO

2.1. TARJETA DE CONTROL

La tarjeta de control tiene 7 subsistemas (figura 7):

· **Teclado** formado por 4 pulsadores dispuestos en cruz con los que se puede aumentar/disminuir la velocidad del motor, ver el estado de la velocidad y la referencia fijadas y mostrar un mensaje informativo sobre los pulsadores a utilizar para variar la velocidad del motor.

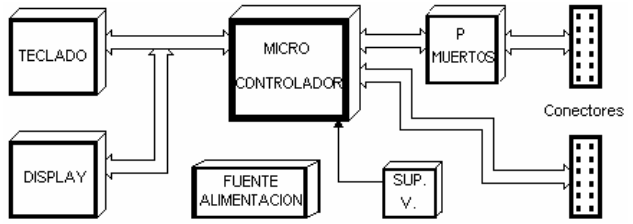


Figura 7: Diagrama de la tarjeta de control.

· **Microcontrolador ST52e430:** Este microcontrolador de STMicroelectronics, permite llevar a cabo de manera eficiente algoritmos boolean y fuzzy para alcanzar el mejor funcionamiento de las dos metodologías, es decir, permite describir un problema usando un modelo lingüístico en vez de un modelo matemático. De esta forma es muy útil y manejable modelizar y controlar sistemas complejos con una alta precisión. La flexible configuración de sus terminales de entrada y salida, permite el interface con un amplio rango de dispositivos externos. Para reducir las perturbaciones de ruido, la alimentación de la parte digital se mantiene separada de la alimentación de la parte analógica. Dispone de tres puertos PWM/Temporizadores que permiten manejar señales de control para dispositivos de potencia y señales de tiempo, implementado modos de operación distintos y control PMW (modulación de anchura de pulso) de alta frecuencia.

· **Codificador de teclado**: se necesitan cuatro pulsadores que se incorporan a la placa y a su vez se dispone de un conector de 10 vías para la posibilidad de utilizar un teclado estándar de 16 pulsadores. La atención del codificador de teclado es a través de interrupciones en el microcontrolador.

· **Display**: Debido al limitado número de terminales del microprocesador, se utiliza un interface con el procesador de 4 bits. El bus de datos utilizado por el display se comparte con el del codificador de teclado con lo cual se deberá tener especial cuidado a la hora de realizar la habilitación de ambos dispositivos.

· **Fuente de alimentación:** todos los dispositivos de la tarjeta utilizan una tensión de 5 v. La fuente de alimentación es lineal con un filtro en la salida, para eliminar los ruidos generados por el rectificador y estabilizar la tensión.

· **Supervisor de tensión**: es un dispositivo utilizado para inicializar el microcontrolador a través de la entrada 'Reset' del mismo en caso de que la tensión de alimentación esté por debajo de un determinado nivel. Incorpora un comparador con su salida en drenador abierto que se activa cuando la tensión de entrada es menor que una tensión umbral de 4.5 voltios.

· **Bloque generador de puntos muertos**: Este bloque genera las señales PWM inversas para cada una de las ramas del convertidor y no permite que las señales y su inversa estén activas simultáneamente, introduciendo un pequeño tiempo (tiempo muerto) durante el cual las dos señales están inactivas. Con esto se consigue que dos polos de potencia de la misma rama no puedan conducir en un mismo instante de tiempo y cortocircuitar la tensión de alimentación de entrada.

· **Conectores**: existen 2 conectores: un conector distribuye los 6 pulsos PWM necesarios para el control de un inversor trifásico. Se utilizan las tres salidas PWM específicas que tiene el microcontrolador y sus inversas generadas a través del bloque generador de puntos muertos. También dispone de una entrada analógica para poder recibir de la tarjeta de potencia una tensión de referencia proporcional a la velocidad actual del motor que se quiere controlar mediante la tarjeta diseñada. Se incluye a su vez otro conector del mismo tipo que a diferencia del primero, utiliza 6 terminales del microcontrolador para generar tanto las señales directas como las invertidas del PWM, de esta forma se pueden generar los puntos muertos por software, aumentando la flexibilidad del sistema.

2.2. SOFTWARE DE CONTROL.

La intención del control es llevar la velocidad del motor a la velocidad de la referencia, manteniendo el deslizamiento dentro de un cierto rango de valores, obtenidos de las medidas realizadas durante la fase del modelización del motor. Existen por lo tanto dos lazos borrosos (Fig.8):

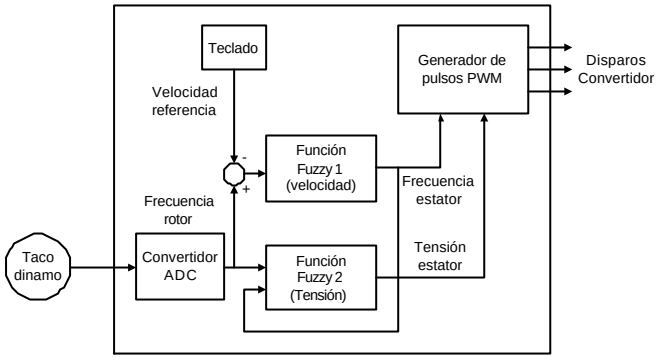


Figura 8: Estructura de control.

2.2.1. Lazo Fuzzy1 control de la frecuencia del estator.

El primer lazo borroso (FUZZY1) es de tipo incremental.

La entrada de este bloque borroso es el error de la velocidad, dado por la diferencia entre la velocidad de la referencia, que fija el usuario mediante el teclado y la velocidad del motor 'Frotor', medida con el convertidor A/D y proveniente de una dinamo colocada en el eje del motor. La salida del bloque ?f se suma algebraicamente a la frecuencia 'Festator' del estator para alcanzar la velocidad del motor a la referencia fijada.

En las figuras 9 y 10 aparece esta función Fuzzy1 definida en el programa de control del microcontrolador ST52e430 VisualFive.

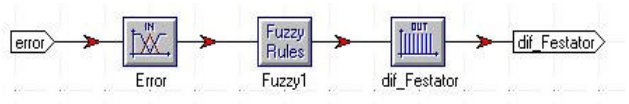


Figura 9: Función Fuzzy1 en VisualFive.

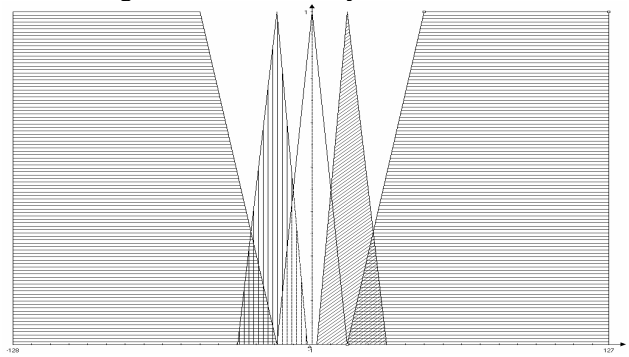


Figura 10: Función de pertenencia Fuzzy1.

Las reglas de inferencia son en este caso, para el motor utilizado en el laboratorio:

if Error is very_negative then dif_Festator is -20
if Error is negative then dif_Festator is -2
if Error is zero then dif_Festator is 0
if Error is positive then dif_Festator is 2
if Error is very_positive then dif_Festator is 20

La frecuencia f del estator es igual a:

$$f(k) = f(k \quad 1) \pm \quad f \qquad (1)$$

Donde ?f es el incremento o el decremento proporcionado por la salida del regulador borroso para ajustar la velocidad del rotor.

2.2.2. Lazo Fuzzy2 control de la tensión del estator.

El segundo lazo borroso (FUZZY2) recibe en la entrada la diferencia entre 'Festator' y 'Frotor' (deslizamiento, s), y ajusta el nivel de tensión para optimizar el flujo y para prevenir sobre intensidades de corriente en el motor (fig. 11, 12 y 13).

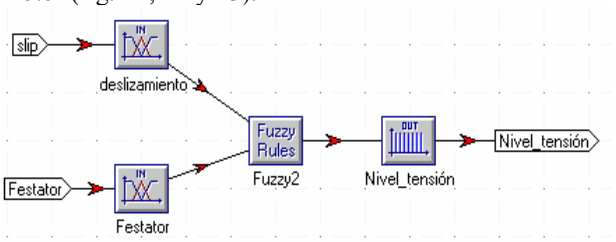


Figura 11: Función Fuzzy2.

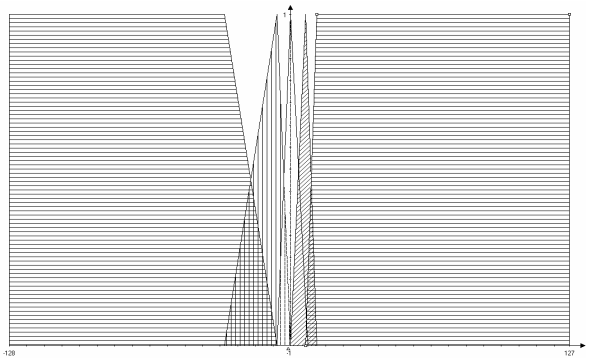


Figura 12: Función de pertenencia del deslizamiento.

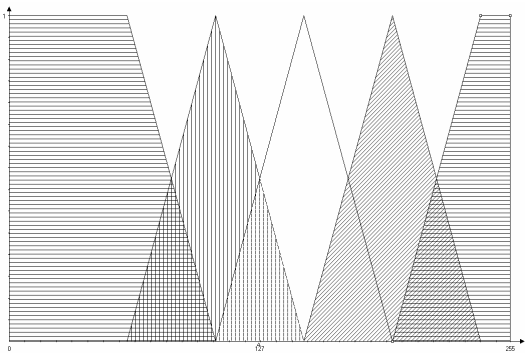


Figura 13: Función de pertenencia de la frecuencia del estator.

Conocido el comportamiento dinámico del motor del laboratorio, se establecen las siguientes reglas de inferencia, con las salidas correspondientes de la tensión del motor (expresadas en un byte):

Desliz./f.estator	Muy Bajo	Bajo	Medio	Alto	Muy Alto
Muy bajo	130	111	89	74	59
Bajo	121	98	81	65	51
Medio	111	89	74	59	42
Alto	98	81	65	51	34
Muy alto	89	74	59	42	24

La salida de este bloque Fuzzy2, es el índice de modulación del PWM utilizado para controlar el convertidor. A mayor anchura de pulso PWM, mayor tensión de salida.

Y finalmente según la frecuencia del estator y el nivel de tensión, el bloque generador de pulsos genera tres señales PWM para controlar el convertidor DC-AC y por lo tanto el motor.

En la figura 14 aparece el diagrama de flujo de todo el programa de control de velocidad con deslizamiento constante del motor trifásico.

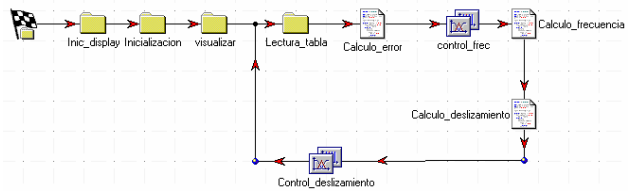


Figura 14: Software completo del control por deslizamiento del motor

La simplicidad de este programa, frente a un programa de similares características con un procesador que no implemente directamente las funciones Fuzzy es evidente.

También esto repercute en el coste, ya que un sistema completo se puede desarrollar con un procesador (de 8 bits) y con una reducida lógica externa (ver fig.1).

3 RESULTADOS DE LAS SIMULACIONES Y CONCLUSIONES

Se han realizado las simulaciones con el simulador incorporado en VisualFive, pero como en este simulador es imposible desde el punto de vista práctico realizar la simulación de la planta (motor), se ha optado por comunicar el simulador del procesador (VisualFive) con Matlab.

En la figura 15 se puede ver el diagrama en Simulink de Matlab, donde se ha incorporado el módulo que comunica VisualFive con Matlab (Subsystem) que tiene como entradas las velocidades (real y de referencia) y las salidas son la frecuencia del estator y el índice de modulación (tensión).

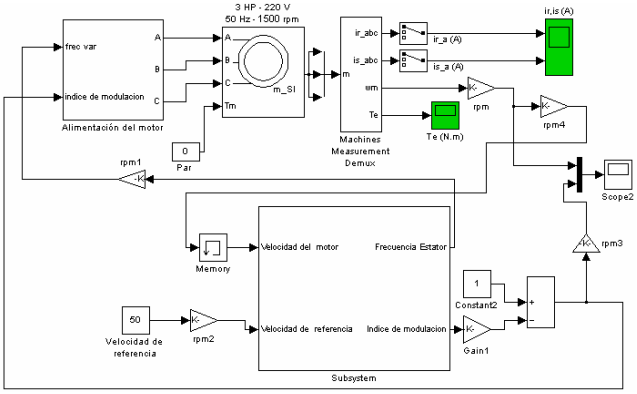


Figura 15: Esquema en Simulink del simulador del controlador.

Los parámetros del motor utilizado en Matlab son:

Potencia: 3 HP.
Tensión Nominal: 220 v.
Frecuencia Nominal: 50Hz.
Velocidad Nominal: 1500 r.p.m.
Resistencia e Inductancia del estator: 0.435 Ohm. 4mH.
Resistencia e Inductancia del rotor: 0.816 Ohm. 2mH.
Inductancia Mutua: 69mH.

En la figura 16 se ha representado el comportamiento del controlador cuando se modifica el par. El resultado es satisfactorio y comparable con un control vectorial por campo orientado utilizando sensores de velocidad.

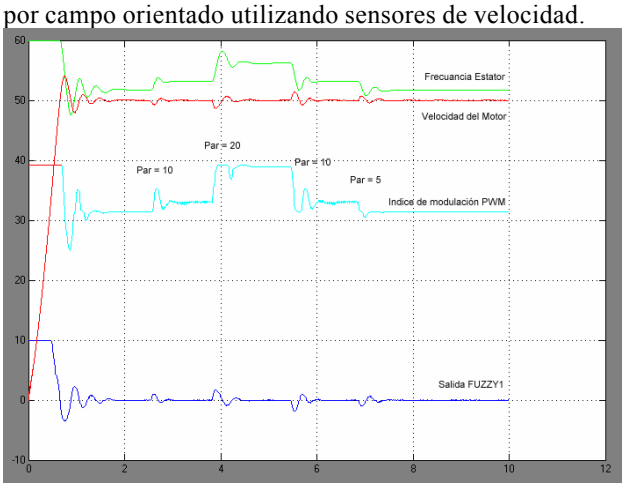


Figura 16: Salida del controlador: Frecuencia del estator, Velocidad del motor, Indice de Modulación y Salida FUZZY1.

Por supuesto aquí la ventaja es que un control vectorial necesita un procesador mucho más complejo, por ejemplo un DSP en coma flotante (Procesador Digital de Señal) y que evidentemente será más caro que el ST52e430 que estamos utilizando en este artículo.

Tenemos la experiencia que para trabajar holgadamente en control vectorial es necesario procesadores de 32 bits en coma flotante con rendimientos de 150 MFlops (Millones de operaciones en coma flotante por segundo). En nuestro caso utilizamos un procesador de 8 bits que trabaja en punto fijo, eso sí con un hardware específico para fuzzy.

Otras de las ventajas es que la estabilidad de los controladores vectoriales está muy ligada a los parámetros del motor (impedancias e inductancias del motor). De tal forma que modificando los parámetros de motor, obtenemos peores dinámicas del sistema en comparación con el controlador Fuzzy. Es decir el controlador Fuzzy es mucho más robusto y más fácil de implementar que el controlador vectorial.

Esta última ventaja es evidente si nos vamos a controladores sin sensores de velocidad, donde los controladores vectoriales son menos robustos todavía, dependiendo todavía más de los parámetros del motor.

Además los parámetros del motor en general son variables, ya que dependen de la temperatura y esta depende de la carga y de otros factores, por lo tanto interesa que el controlador sea lo más robusto posible.

Un controlador Fuzzy como el diseñado proporciona robustez con reducidos costes de implementación esta es la ventaja principal y además añade posibilidades en el control sin sensores (sensorless).

Agradecimientos

Tenemos que agradecer al Dto: de Tecnología Electrónica de la Universidad de Valladolid por los medios prestados, sin la financiación necesaria hubiese sido imposible realizar este Proyecto.

Gracias a STMicroelectronics por toda la documentación y programas desarrollados para sus microcontroladores ST Five que ha sido una ayuda indispensable para poder desarrollar este Proyecto.

Referencias

[1] B.K. Bose. *Power Electronic and A.C. drives*. Englewood Cliffs, NJ. Prentice Hall, 1997.

[2] C.C. Lee. *Fuzzy logic in control systems fuzzy logic controller* Part I e Part II IEEE Trans Syst. Man and Cibern, vol 21 nº 2 Part I pag. 404-418,1990 Part II pag. 419-436, 1990.

[3] M. Di Guardo, V. M. *Control of an asynchronous three-phase motor with st52x420*. STMicroelectronics, 2003.

[4] Peter Vas. *Artificial-Intelligence-Based Electrical Machines and Drives*. Oxford Science Publications, 1999.

[5] Peter Vas. *Sensorless vector and direct torque control*. Oxford Science Publications, 1996.

[6] Stamatios V. Kartalopoulos. *Understanding neural networks and Fuzzy Logic*. IEEE Press, 1996.

[7] STMicroelectronics. *Databook ST52E430*. STMicroelectronics, 2004.

DISEÑO DE SISTEMAS BORROSOS DE CONDUCCIÓN AUTOMÁTICA DE VEHÍCULOS

Ricardo García Rosa ricardo@iai.csic.es
Teresa de Pedro tere@iai.csic.es
Carlos González gonzalez@iai.csic
Eugenio Naranjo jnaranjo@iai.csic.es
Instituto de Automática Industrial, CSIC
La Poveda, Arganda del Rey
28500 Madrid

Resumen

El presente trabajo forma parte del programa AUTOPÍA que engloba un conjunto de proyectos de investigación del Instituto de Automática Industrial sobre guiado automático de vehículos. Los procedimientos de guiado consisten esencialmente en controladores borrosos que reciben información del entorno de conducción mediante sistemas de navegación por satélite, GPS y EGNOS. Este artículo se dedica fundamentalmente a describir una adaptación del procedimiento de concreción y a mostrar los resultados obtenidos de su aplicación a la conducción automática de vehículos.

Palabras clave: Control borroso, guiado automático

1 INTRODUCCIÓN

Actualmente la conducción completamente automatizada de vehículos no es un objetivo buscado por los fabricantes, ni es deseada por los conductores. Sin embargo el incremento del número de accidentes, la necesidad de hacer más eficaz el transporte por carretera y la creciente demanda de confort, entre otros motivos, hacen que se vayan instalando en los vehículos ayudas a la conducción. Las ayudas de conducción existentes actualmente tienen un nivel de automatización bajo, pero es un campo de investigación muy activo en el que se están diseñando y experimentando prototipos cada vez más sofisticados [8].

Los sistemas de ayuda a la conducción han de ser probados en simulación antes de implantarse en los vehículos de serie, entendiendo en este caso la simulación en sentido amplio, es decir no sólo la simulación numérica y gráfica, sino la realización de pruebas con vehículos reales en circuitos reales que reproduzcan los entornos habituales de tráfico.

En este artículo se presentan los métodos de automatización de guiado empleados, fundamentalmente simplificaciones de los sistemas clásicos de control borroso, para el control de vehículos y el seguimiento de trayectorias, así como los resultados obtenidos. Se han realizado muchos experimentos con vehículos de serie que se mueven en un circuito privado del IAI. Algunos de estos experimentos recogen las exhibiciones realizadas en vivo con motivo de las visitas técnicas programadas al Instituto de Automática Industrial durante el 10th World Congress on Intelligent Transportation Systems and Services, celebrado en Madrid en el mes de noviembre de 2003. Estos experimentos se describen someramente en el apartado de resultados.

2 MODELO DEL GUIADO: ESCENARIO Y VEHÍCULO

El proceso de guiado automático de un vehículo se puede descomponer en dos procedimientos, el control de la dirección, o lateral, y el control de la velocidad, o longitudinal. Ambos pueden ejecutarse por separado, si se quiere automatizar bien la dirección o bien la velocidad, o simultáneamente, si se desea el guiado sea completamente automático. A su vez el control de la velocidad puede hacerse controlando sólo el acelerador o controlando simultáneamente el acelerador y el freno. Los sistemas del primer tipo, que corresponden a los de control de crucero que existen en el mercado, no pueden usarse en situaciones que requieran una acción de frenado rápida y potente.

Los controladores del guiado contienen modelos del entorno, los escenarios de tráfico locales o globales, y modelos del vehículo que se utilizan en las simulaciones en computado, pero estos modelos no tienen sentido en los controladores reales de los coches.

Para el control de la dirección un modelo local sencillo pero suficiente, contiene la separación y la orientación del vehículo respecto de la trayectoria deseada. Estas variables son las entradas de control, cuyas particiones borrosas, o valores lingüísticos, son los siguientes:

Separación: { derecha, izquierda }
Orientación: { derecha, izquierda }

Para el control de la velocidad, el modelo consta de la distancia al primer obstáculo o hito situado en la trayectoria (bifurcación, señal de tráfico, vehículo próximo, etc), la velocidad y la aceleración del vehículo. Los términos borrosos definidos para estas entradas de control son:

Distancia: { nominal }
Velocidad: { nominal }
Aceleración: { acelera, frena }

Conviene observar que aunque solo se defina un valor lingüístico para algunas variables, nuestro sistema de control puede derivar nuevos valores lingüísticos a partir de los valores iniciales. En particular el valor *nominal*, de la distancia o de la velocidad, aparece en las reglas modificado por los operadores ***mayor que*** y ***menor que***.

Los términos anteriores son sólo un ejemplo, de hecho se utilizan otros muchos, tal como puede verse en las diversas secciones del apartado de resultados.

2.1. PERCEPCIÓN DEL ESCENARIO

Los valores instantáneos de las entradas se adquieren con los instrumentos propios del coche y con algunos sensores añadidos. El primer sensor utilizado, a veces el único, es un sistema de posicionamiento por satélite, aunque algunas maniobras necesitan otros sensores; por ejemplo ultrasonidos para aparcar, láser para detectar obstáculos, cámaras de vídeo para reconocer señales de tráfico, peatones, etc.



Foto 1: Rocinante con "lidar" y antenas

En los experimentos de cooperación entre varios vehículos, como adelantamientos o circulación en caravana, la información proporcionada por los sistemas de navegación global por satélite, GNSS, no basta para conocer el entorno. En estos casos cada vehículo comunica su posición a los restantes a través de una red inalámbrica.

2.1. MODELO DEL VEHÍCULO

El movimiento del vehículo se modela con las ecuaciones cinemáticas siguientes, que ligan las variables del modelo a las entradas y salidas de control [1]. Figura 1

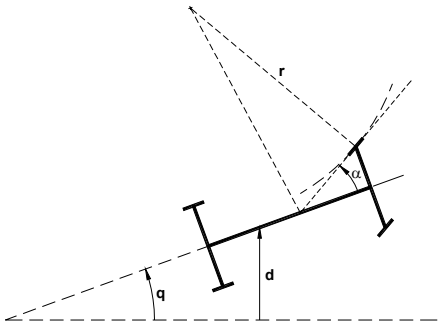


Figura.1. Modelo del vehículo:

d: separación entre el vehículo y la referencia
q: orientación
r: radio de curvatura
α: ángulo de giro de las ruedas directoras

$$dx/dt = v * \cos(q) \quad (1)$$

$$dy/dt = v * sen(q) \quad (2)$$

$$\omega = dq/dt \text{ , o bien } \omega = v/r = v * k \quad (3)$$

Si se considera que en el movimiento la curvatura *k* varía de forma continua. la ecuación (3) se puede sustituir por:

$$\omega = dq/dt = v * k \text{ y } dk/dt = \sigma \quad (4)$$

Nuestro sistema de control usa la ecuación intrínseca del movimiento, que descompone la aceleración en sus componentes tangencial y normal, según la ecuación (5)

$$\vec{a} = (dv/dt)\vec{\tau} + k * v^2 \vec{\eta} \quad (5)$$

siendo $\vec{\tau}$ y $\vec{\eta}$ vectores unitarios en la dirección tangencial y normal a la trayectoria, respectivamente.

Las salidas de control dan las acciones sobre los mandos del vehículo, el control lateral el giro del volante y el longitudinal la presión sobre el freno y el acelerador.

3 SISTEMA DE CONTROL

Para construir los controladores del vehículo se ha utilizado el denominado Controlador Borroso Universal (CBU) que es una adaptación de otro generador de controladores que realizamos con anterioridad [2].

En la ejecución del CBU se distinguen las tres etapas clásicas: difuminación, inferencia y concreción ("fuzzyfication", "inference" y "defuzzyfication" en palabras inglesas). La primera etapa es semejante a la de cualquier controlador borroso [7]. La segunda es más potente que la de otros sistemas pues trata conjunciones, disyunciones y predicados modificados en las condiciones de las reglas. Por último, veremos con más detalle la etapa de concreción, la más característica de nuestro controlador.

El procedimiento de concreción tiene dos fases: 1) Si en el conjunto de reglas hay varias con el mismo consecuente, por ejemplo *girar derecha*, se hace la unión de sus consecuentes, lo que equivale a elegir la regla más influyente en la acción de control. 2) El algoritmo de concreción usa una fórmula aproximada del centro de área, es decir, sustituye la expresión exacta:

$$y = \int xB^*(x)dx \,/\, \int B^*(x)dx \qquad (5)$$

por la aproximada:

$$y = \Sigma\, x_i B_i \,/ \Sigma\, B_i \qquad (6)$$

siendo $B^*(x) = \cup_i \omega_i B_i$ el área de la salida inferida por las reglas (Figura 2) y B_i el área de una partición de la variable de salida (Figura 3).

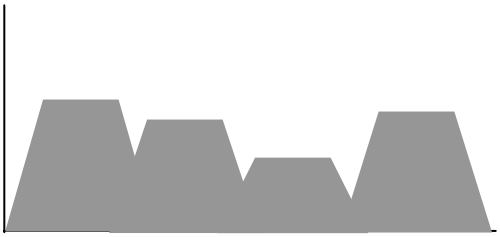


Fig. 2: Valor lingüístico inferido por el conjunto de reglas

La expresión (6) simplifica los cálculos y, sobre todo, permite hacer parte de ellos fuera del ciclo de control, pues la suma ΣB_i se puede calcular una vez definidas las particiones de las variables de salida. Aún más, el área de una partición se puede concentrar en su centro de masa, luego su función de pertenencia puede reducirse a un pico "singleton" definido en la coordenada del centro de área.

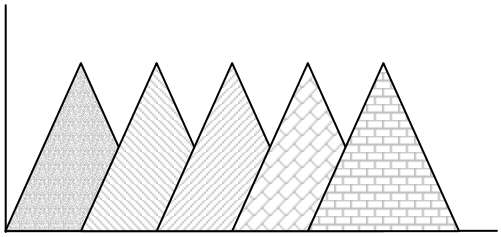


Fig. 3: Distribución típica de particiones de una variable borrosa

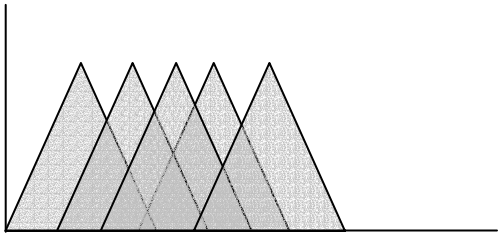


Fig. 4: Confusión de valores lingüísticos

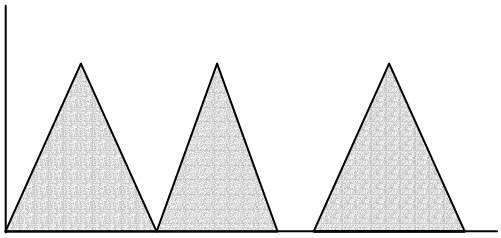


Fig. 5: Falta de conocimiento del sistema

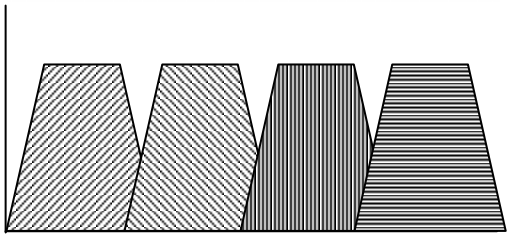


Fig. 6: Distribución típica de particiones trapezoidales

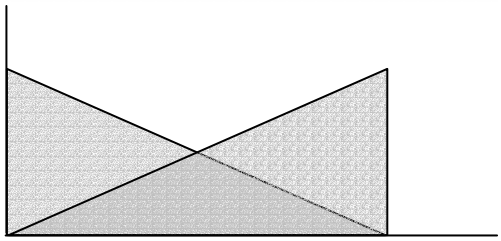


Fig.7. Particiones con funciones de pertenencia monótonas

En resumen, es posible simplificar la fase de concreción representando por picos los valores lingüísticos de las variables de salida sin pérdida de generalidad.

2.1. ERROR EN EL CÁLCULO DE LA SALIDA DE CONTROL

La simplificación anterior introduce un error en el cálculo de la acción de control, que se puede acotar. Empecemos por observar que en una definición cualquiera de una variable borrosa, si A es el área de la unión y A_i la de un elemento, (Figuras 3-7), ocurre que $A < \Sigma A_i$ si las particiones se solapan y $A = \Sigma A_i$ en caso contrario.

Interesa acotar el error para una distribución de particiones regular (Figura 3) pues es una distribución muy frecuente aunque no sea preceptiva. En cualquier caso es conveniente distribuir las particiones de forma que se eviten dos casos extremos: particiones muy solapadas (Figura 4) o disjuntas (Figura 5.). Cualquiera de los dos casos sería un mal modelo del sistema a controlar; el primero porque confundiría valores lingüísticos distintos y el segundo porque no tendría un conocimiento completo del sistema.

En el caso de una distribución regular (Figura 3), el error cometido al aproximar el centro de masa por la expresión (6) se puede calcular en función de los parámetros de las particiones. Si *b* es la base de una partición de altura unidad, la intersección entre dos particiones es un triángulo de base b/2, altura ½ y área b/8. Luego el error cometido es 1/8, si el número de particiones es par, y 1/6 si es impar. Si uno de los triángulos tiene una altura diferente, el área de la intersección es menor y por lo tanto también es menor el error cometido al despreciar la intersección.

Es evidente que en una distribución regular de particiones trapezoidales (Figura 6) el error sería menor, porque la suma de las áreas de las intersecciones tiene menor peso relativo en el área total. En el otro extremo el máximo error es ¼ y ocurre cuando dos particiones monótonas simétricas (Figuras 7) tienen el mismo peso en la acción de control.

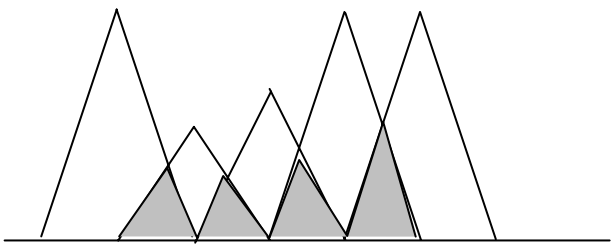


Figura 8: Intersección de particiones irregulares

Consideremos ahora que la expresión $B^* = \cup_i \omega_i B_i$ es la de la figura 8, donde los triángulos sombreados representan el error cometido al sustituir la unión por la agregación. El triángulo sombreado más a la derecha es el mayor y es la intersección de particiones de la misma altura.

4 RESULTADOS OBTENIDOS

Los sistemas de guiado automático coordinan diversos controladores laterales y longitudinales para seguir trayectorias en distintas situaciones de tráfico. A continuación citamos algunos casos.

4.1 Control de trayectorias rectas y curvas

El controlador de la dirección determina el giro del volante para seguir cualquier trayectoria recta o curva, tomar bifurcaciones o invertir el sentido de la marcha. Es independiente del control de la velocidad, de hecho se puede utilizar en solitario o combinar con cualquiera de los dos tipos de controladores de la velocidad que hemos implementado. Las reglas básicas son las expuestas por Sugeno [7] en sus primeras publicaciones y consisten en un simple control en posición. A modo de ejemplo una regla sería:

SI cabeceo nulo **Y** e_posición **MAYORQUE** cero
ENTONCES volante derecha
SI cabeceo **MENORQUE** nulo
ENTONCES volante izquierda

donde cabeceo designa el error en el la orientación, el ángulo entre la dirección del coche y la trayectoria y e_posición es el error de posición de un punto del coche, por ejemplo de su centro de gravedad (en la figura 1 las variables *q* y *d* respectivamente).

4.2 Control de la velocidad

El controlador de la velocidad más sencillo es el denominado control de crucero -que ya está incorporado en muchos vehículos comerciales. Su función es mantener o no superar una velocidad determinada. Unas reglas muy simples son las siguientes:

SI e_vel **MAYORQUE** cero **ENTONCES** pedal acelera
SI e_vel **MENORQUE** cero **ENTONCES** pedal frena

En la práctica ha sido necesario introducir también la aceleración, calculada por diferenciación

4.3 Control adaptativo de la velocidad

Un control más avanzado de la velocidad es el denominado control de crucero adaptativo. En este caso el valor de la velocidad se adapta en función de las características de la ruta y del tráfico. Por ejemplo, el control de crucero adaptativo acomoda la velocidad a la geometría de la trayectoria –baja en curvas, alta en rectas- mantiene la distancia de seguridad entre vehículos y puede respetar las señales de tráfico. Nuestro controlador longitudinal es del tipo de los denominados de *arranque y*

parada -stop and go- en la literatura especializada, pues puede parar completamente el vehículo y reiniciar la marcha. De hecho se han hecho experimentos que simulan casos de conducción en caravana y atascos.

Desde el punto de vista técnico hay una diferencia esencial entre el control de crucero y el control de crucero adaptativo, pues mientras el primero se puede limitar al control de un único actuador, el acelerador, obviando la existencia del freno, el último requiere hacer un control coordinado conjunto del freno y del acelerador. Reglas que utilizan dos actuadores, freno y acelerador, son como las siguientes:

SI separación **MENORQUE** nominal **O** curva cerca
ENTONCES freno frena

En [3] se comenta la acción de reglas en las que se introducen derivadas de las velocidades para estabilizar la marcha, en los ejemplos usan como sufijo "d_t_" para indicar derivada en el tiempo, son del tipo:

SI aceleración **MENORQUE** cero **Y** d_t_gap lejos
ENTONCES pedal acelera

SI d_t_gap cerca **Y** d_t_gap negativo
ENTONCES pedal frena

Conviene observar que, en el mundo del automóvil, la separación entre vehículos se mide en segundos, teniendo en cuenta que el dato interesante es el tiempo que tarda en reaccionar un conductor ante un imprevisto.

4.3 Adelantamientos

El guiado automático para maniobras de adelantamiento requiere encadenar y coordinar controladores laterales y longitudinales adaptativos para realizar cambios de carril a la vez que se acelera o decelera [4]. Se quiere destacar que la automatización de este tipo de maniobras casi no aparece referenciada en la literatura.

El cambio de carril se ha implementado de una manera muy simple, sólo se cambia la línea de referencia por una paralela a una distancia igual a la anchura del carril.

4.3 Experimentos realizados

Los experimentos se han realizado con dos vehículos eléctricos (Foto 2) que se mueven por un circuito privado y reproducen las anteriores situaciones de tráfico. En alguno de ellos, para reproducir con fidelidad casos reales, circulan a la vez un vehículo conducido manualmente y un vehículo conducido automáticamente.



Foto 2: Rocinante y Babieca, vehículos de serie

En la fotografía 2 se puede observar que uno de los vehículos, Babieca, lleva una cámara de visión utilizada por un equipo de la Universidad de Alcalá de Henares

4.3 Conclusiones y trabajo en curso

Los procedimientos de control borroso que hemos implementado para la automatización del guiado de vehículos derivan de otros más generales diseñados con anterioridad [2]. En esencia las procedimientos especializados para el guiado consisten en reducir a picos o "singletons" el tipo de las funciones de pertenencia de las variables de salida (en conversación particular el profesor Sugeno comentó que él hacía lo mismo y no dio ninguna razón). Esta limitación conduce a simplificar la fase de concreción y permite sacar parte de los cálculos del ciclo de control.

Este resultado es interesante desde el punto de vista práctico, porque permite reducir el tiempo de ejecución de los algoritmos de control. Por otra parte, los experimentos realizados muestran que el error cometido en el cálculo de la acción de control es aceptable a efectos prácticos.



Foto 3: Clavileño, mostrando la antena de GPS

Actualmente el programa AUTOPÍA continua con la instrumentación de dos nuevos vehículos propulsado por gasolina, uno de ellos (el Clavileño de la foto 3) será probado en un circuito público con fuertes curvas; Además estamos desarrollando técnicas que permitan derivar directamente las reglas de control de una descripción lingüística de la trayectoria.

Agradecimientos

El programa AUTOPÍA, que agrupa los trabajos descritos, está financiado por los proyectos ISAAC DPI 2002-04064-C05-02 del Plan Nacional, COPOS del MFOM y la colaboración de CITROËN ESPAÑA SA.

Referencias

1. Bishop R,Intelligent Vehicle Application Worldwide, IEEE Intelligent Systems Jan/Febr 2000, pp 78-81

2. García Rosa R, de Pedro T, "Modeling a fuzzy coprocessor and its programming language", Mathware and Soft Computing, nº 5, 167-174, 1998.

3. Naranjo J.E, Reviejo J, González C, García R, de Pedro T, Adaptive Fuzzy Control for Inter.-Vehicle Gap Keeping. IEEE T on Intelligent Transportation Systems, vol 4 n 3, September 2003, pp 132-141

4. Naranjo J.E, Reviejo J, González C, García R, de Pedro T, Overtaking Maneuver Experiments with Autonomous Vehicles. Proceedings ICAR 2003, Coimbra, pp 1699-1703

5. Patel A.V, Mohan B.M, Some numerical aspects of area defuzzification method, Fuzzy Sets and Systems 131, pp 401-409, 2002.

6. Sugeno M, An introductory survey of fuzzy control, Information Sciences, 36, 59-83, 1985.

7. Sugeno M, Fuzzy Algorithmic Control of a Model Car by Oral Instructions, IFSA´87

8. Vlacic L, Parent M, Harashima F, Intelligent Vehicle Technologies, SAE International 2001.

PLATAFORMA PARA EL DESARROLLO DE CONTROLADORES DIFUSOS COMO SISTEMAS EMPOTRADOS

A. Cabrera[1] S. Sánchez-Solano[2] A. Barriga[2] P. Brox[2] F. J. Moreno-Velo[2] I. Baturone[2]

[1] Facultad de Ingeniería Eléctrica, ISPJAE, Ciudad de la Habana, Cuba. alex@electrica.cujae.edu.cu
[2] Instituto de Microelectrónica de Sevilla, CNM-CSIC, Sevilla, España. santiago@imse.cnm.es

Resumen

En esta comunicación se describen los diferentes componentes físicos, lógicos y de procedimiento de una plataforma que permite implementar una estrategia de realización de controladores difusos empotrados, tanto híbridos hardware/software como puramente software. Dicha estrategia está basada en la utilización de módulos de Propiedad Intelectual para la realización del sistema de procesamiento y de una arquitectura hardware específica para la implementación de los módulos de inferencia difusos, de forma tal que facilita la realización de controladores difusos como sistemas empotrados sobre dispositivos programables.

Palabras Clave: Control difuso, VHDL, FPGA, Propiedad Intelectual, SoPC

1 INTRODUCCIÓN

La capacidad de la lógica difusa para describir el comportamiento de sistemas complejos mediante reglas lingüísticas simples ha motivado su aplicación a la realización de numerosos sistemas de control [1, 2].

Las diferentes alternativas que se han desarrollado para la implementación de sistemas de control difuso pueden ser agrupadas en realizaciones software, realizaciones hardware y realizaciones híbridas. Las realizaciones software han sido las soluciones más ampliamente adoptadas y se caracterizan por su elevado grado de flexibilidad, aunque su tiempo de respuesta viene limitado por la inherente ejecución secuencial de los programas. La tendencia actual en robótica y otras aplicaciones hacia el desarrollo de sistemas empotrados, que presentan restricciones en cuanto a peso, tamaño, consumo de potencia y costo, ha hecho que muchas de estas realizaciones utilicen microcontroladores [3]. Sin embargo, la utilización de microcontroladores comerciales puede resultar insuficiente para algunas aplicaciones o sobredimensionadas para otras.

Las realizaciones totalmente hardware del controlador difuso representan una alternativa totalmente opuesta a la anterior y son requeridas para aquellas aplicaciones en que la velocidad de inferencia sea un factor determinante [4]. Debido a la necesidad de reducir el área o el consumo de recursos sobre el circuito resultante, es preciso recurrir a estrategias de realización que limitan la flexibilidad de la solución, aunque la disponibilidad de arquitecturas configurables permite atenuar en alguna medida este inconveniente.

Las realizaciones híbridas basadas en técnicas de codiseño hardware/software (HW/SW) permiten obtener un compromiso adecuado entre las ventajas e inconvenientes de las alternativas anteriores [5]. La distribución entre la implementación hardware y la ejecución software de las tareas que debe realizar el sistema de control difuso permite obtener una solución que combine las ventajas de flexibilidad y velocidad. Una posible estrategia de particionado de tareas consiste en la implementación sobre hardware dedicado del módulo de inferencia difuso y la realización software de las restantes tareas de configuración y procesado convencional [6].

Estas realizaciones híbridas se ven favorecidas por el vertiginoso desarrollo de la industria microelectrónica y la aparición de novedosas técnicas de diseño digital, las cuales hacen posible actualmente la implementación de sistemas completos sobre un único circuito integrado. La disponibilidad de dispositivos lógicos programables como FPGAs (Field Programmable Gate Array) que incorporan una elevada cantidad de recursos de hardware, así como sus potentes entornos de diseño, facilita la implementación de dichos sistemas sobre un dispositivo programable (System on Programmable Chip, SoPC). La existencia de módulos de propiedad intelectual (Intellectual Property, IP) que proporcionan una gran diversidad de elementos hardware bien desarrollados y

[1] Este trabajo ha sido financiado por los proyectos TIC2001-1726 (CICYT) y DGAECE-58/02 (Junta de Andalucía).

depurados favorece las realizaciones SoPC. Muchos módulos IP pueden ser configurados por el diseñador, adaptándolos así a muy diversas aplicaciones [7].

En esta comunicación se presenta una plataforma que integra todos los componentes necesarios para el desarrollo de controladores difusos empotrados como SoPC, tanto híbridos como software, y que pueden ser configurados en función de la aplicación. Para ello en la sección 2 se expone la estrategia de realización y los componentes físicos y lógicos de la plataforma de desarrollo. La sección 3 presenta el procedimiento que permite interrelacionar los componentes anteriores con el objetivo de obtener el controlador difuso empotrado. Por último, la sección 4 muestra algunos resultados obtenidos en aplicaciones de robótica utilizando controladores difusos SoPC desarrollados íntegramente mediante la plataforma propuesta.

2 ESTRATEGIA DE REALIZACIÓN Y PLATAFORMA DE DESARROLLO

La estrategia seguida para la implementación de controladores difusos empotrados como SoPC y que permite tanto realizaciones híbridas HW/SW como puramente software de los mismos se ilustra en la Figura 1. Está basada en la utilización de un sistema de procesamiento configurable y disponible como módulo de propiedad intelectual, así como (para el caso de las realizaciones híbridas) en la utilización de una arquitectura también configurable para la implementación hardware del módulo de inferencia difuso (Fuzzy Inference Module, FIM). De esta forma y mediante las herramientas de CAD correspondientes, se procede a la integración de todo el sistema sobre una FPGA obteniéndose la realización SoPC.

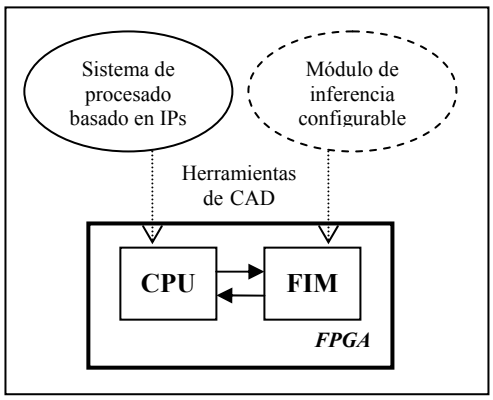


Figura 1: Estrategia de realización de controladores difusos empotrados como SoPC.

El sistema de procesamiento ejecuta todas las tareas del controlador que no requieran recursos específicos así como también la rutina de inferencia difusa en el caso de la realización totalmente software del controlador. Los recursos de este sistema (microprocesador, memorias, puertos de entrada/salida, temporizadores, etc.) son configurables y parametrizables, de forma tal que se puede conformar un sistema de procesamiento acorde a los requerimientos de la aplicación.

En el caso de requerirse una implementación híbrida, el módulo de inferencia se implementa sobre hardware utilizando los recursos del dispositivo programable. Este módulo de inferencia está basado en una arquitectura configurable que permite su adaptabilidad a muy variadas aplicaciones.

Para la materialización de esta estrategia se ha desarrollado una plataforma que incorpora los componentes físicos y lógicos así como el procedimiento de desarrollo del controlador. Los componentes físicos incluyen los módulos IP del sistema de procesamiento, la arquitectura configurable del FIM, así como la FPGA soporte de la implementación SoPC.

Como sistema de procesamiento se utilizan los módulos IP del procesador MicroBlaze de Xilinx y sus diversos periféricos [8]. Para la implementación hardware del módulo de inferencia difuso se utiliza la arquitectura mostrada en la Figura 2, basada en el procesado de reglas activas, la limitación a dos del grado de solapamiento de las funciones de pertenencia de las entradas y la utilización de métodos de defuzzificación simplificados, aspectos que contribuyen a disminuir el consumo de recursos del dispositivo programable [9].

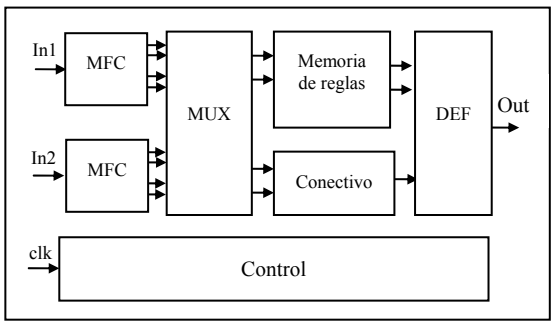


Figura 2: Arquitectura configurable del módulo de inferencia difuso.

Además, dispone de muy diversas opciones de implementación las cuales han sido ampliadas recientemente. De esta forma, los circuitos generadores de funciones de pertenencia (MFC) pueden fuzzificar las entradas mediante almacenamiento en memoria o mediante técnicas aritméticas; el bloque del conectivo de antecedentes puede implementar la función mínimo o el producto; la defuzzificación (DEF) puede realizarse mediante diversos métodos simplificados incluyendo métodos para toma de decisiones. Adicionalmente, permite el dimensionado de sus diferentes parámetros. Para la implementación final de la realización SoPC se debe disponer de una FPGA Spartan-IIE150 o superior.

Los recursos disponibles en esta familia de FPGAs permiten potenciar la realización tanto del sistema de procesamiento MicroBlaze como del módulo de inferencia. La disponibilidad de placas de desarrollo basadas en FPGAs de esta familia facilita la implementación SoPC.

El soporte lógico de la plataforma está formado por las herramientas de CAD correspondientes al sistema MicroBlaze, al módulo de inferencia difuso y a la FPGA. MicroBlaze dispone de entornos de desarrollo que facilitan tanto la síntesis hardware de los módulos IP del sistema de procesamiento como la compilación y depuración del programa de aplicación. De forma análoga, el entorno de desarrollo de sistemas difusos *Xfuzzy* permite la descripción, ajuste, simulación y síntesis, tanto software como hardware, del módulo de inferencia difuso [10]. Por último, las diferentes herramientas incorporadas en el entorno de desarrollo de FPGAs de Xilinx, permiten realizar la implementación SoPC de todo el controlador sobre la FPGA.

3 PROCEDIMIENTO DE DESARROLLO

La plataforma expuesta en el apartado anterior se completa con un procedimiento que permite combinar los diferentes componentes físicos y lógicos para la realización del controlador difuso empotrado.

La Figura 3 muestra el flujo de diseño del módulo de inferencia utilizando las diferentes herramientas del entorno de desarrollo *Xfuzzy*.

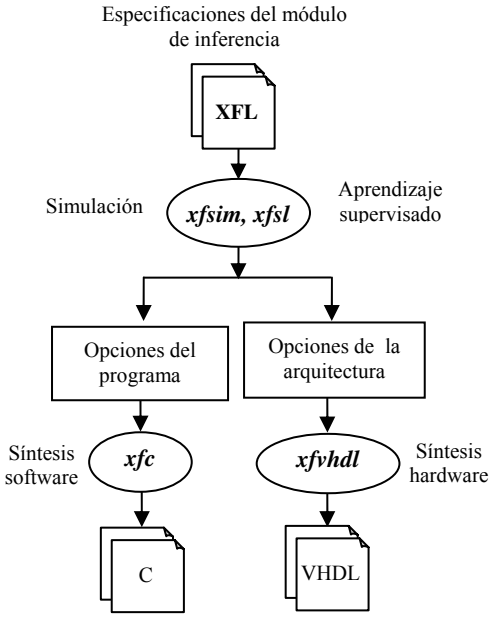


Figura 3: Flujo de diseño del módulo de inferencia.

Una vez descrito el sistema difuso (especificación .xfl), simulado su comportamiento y ajustada su base de conocimiento (herramientas *xfsl* y *xfsim*) se procede a realizar la síntesis del mismo. Para una realización totalmente software del controlador difuso se realiza la síntesis software del FIM mediante la herramienta *xfc*, la cual genera una rutina en código C que deberá ser compilada y enlazada con las rutinas de las restantes tareas del controlador y se ejecutará sobre el sistema de procesamiento empotrado.

De requerirse una implementación híbrida del controlador, se procede a la síntesis hardware del FIM mediante la herramienta *xfvhdl*. Esta herramienta permite dimensionar los bloques del sistema y seleccionar entre las distintas opciones de la arquitectura, generando una descripción en código VHDL que permite la realización hardware de los diferentes bloques de la arquitectura del FIM de la Figura 2 [11]. Los códigos VHDL correspondientes a la descripción hardware del FIM deben ser incorporados posteriormente (durante el proceso de síntesis e implementación de la FPGA) para su interconexión con el sistema de procesamiento.

Paralelamente a la síntesis del FIM puede desarrollarse el sistema de procesamiento MicroBlaze siguiendo el esquema de la Figura 4. A partir de las especificaciones de software (fichero .mss) la herramienta *libgen* compila las funciones de los diferentes periféricos, obteniéndose las bibliotecas y ficheros de cabecera que facilitarán la codificación de los programas de aplicación en C.

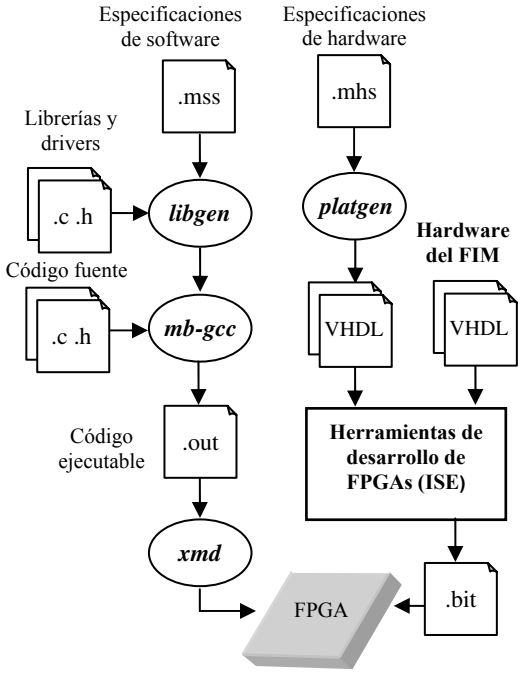


Figura 4: Flujo de desarrollo del sistema de procesamiento e integración del FIM.

De forma similar, los componentes del sistema se declaran en un fichero de especificaciones de hardware (.mhs). La herramienta *platgen* se encarga de integrar todo el hardware del sistema MicroBlaze, generándose una descripción VHDL que puede ser utilizada para crear un sistema jerárquicamente superior, tal como se requiere para la inclusión del módulo de inferencia (código generado por *xfvhdl*) en el caso de la realización híbrida del controlador. La implementación final del sistema completo se lleva a cabo utilizando las herramientas de desarrollo de FPGAs de Xilinx incluidas en el entorno ISE. Una vez obtenido el fichero de configuración (.bit) de la FPGA y programada la misma, se procede a compilar el programa de aplicación mediante la herramienta *mb-gcc*. Para el caso de una realización software del FIM, el código C generado por la herramienta de síntesis *xfc* debe ser compilado y enlazado conjuntamente con el resto del programa de aplicación. El proceso se completa con la herramienta de depuración *xmd* que facilita la carga y verificación del programa sobre el sistema MicroBlaze empotrado.

4 REALIZACIÓN DE CONTROLADORES DIFUSOS SoPC

La plataforma de desarrollo descrita anteriormente ha sido aplicada en la realización de controladores difusos, tanto software como híbridos, para el aparcamiento del vehículo autónomo Romeo4R (Figura 5), diseñado y construido en la Escuela Superior de Ingenieros de la Universidad de Sevilla [12].

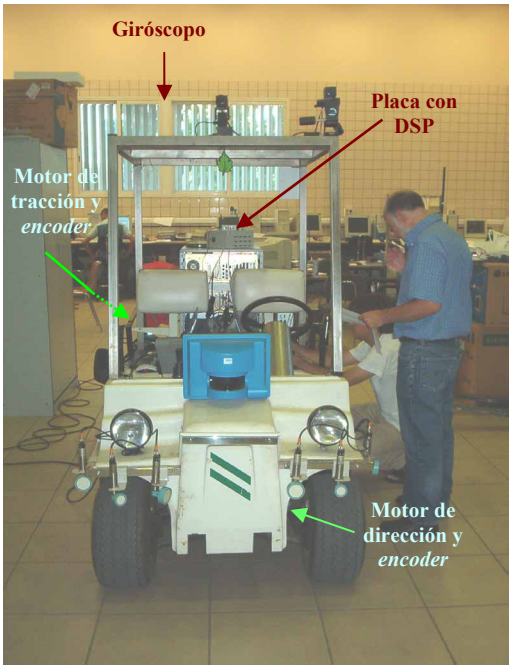


Figura 5: Fotografía de Romeo4R.

Romeo4R está equipado con motores de tracción y de dirección, y diferentes sensores que permiten determinar el estado del vehículo. El control a bajo nivel de los motores, así como la adquisición de la información procedente de los sensores y su procesado para determinar la velocidad (v), curvatura (γ), orientación (ϕ) y posición (x,y) del vehículo se realiza mediante un procesador digital de señal (DSP). De esta forma, el DSP puede interactuar con un controlador de alto nivel que se dedica a la ejecución del algoritmo de control de navegación del vehículo.

En los controladores difusos desarrollados, el procesador MicroBlaze realiza las tareas de interfaz con el controlador de bajo nivel, secuencialización de las diferentes etapas del sistema y, en función de que se realice una implementación software o híbrida, la ejecución de la rutina de inferencia o el intercambio de información con el módulo de inferencia difusa. Los controladores implementan en cada caso la estrategia de navegación del vehículo mediante un conjunto de reglas de actuación similares a las que emplearía cualquier conductor.

Siguiendo el procedimiento descrito en la Sección 3, se desarrollaron controladores difusos empotrados SoPC, tanto en variantes puramente software como híbridos, utilizando para ello una placa comercial de desarrollo de FPGAs [13]. Después de implementar los controladores se realizaron múltiples experimentos para registrar las trayectorias de Romeo4R y compararlas con los resultados de simulación del fichero de especificaciones XFL utilizando las herramientas del entorno *Xfuzzy*.

Para el caso simple de aparcamiento hacia atrás, el módulo de inferencia está compuesto por una base de conocimiento de dos entradas (la coordenada x y la orientación del vehículo ϕ) y una salida correspondiente a la curvatura de las ruedas (γ).

La Figura 6 muestra los resultados experimentales y de simulación de controladores difusos SoPC, tanto híbrido (Figura 6-a) como software (Figura 6-b), cuando el vehículo parte de una situación alejada de la posición objetivo. Las diferencias entre las trayectorias reales y simuladas se deben a las irregularidades del terreno, no contempladas en el modelo de simulación. Nótese no obstante cómo el controlador difuso corrige la desviación y logra aparcar correctamente en todos los casos.

Nótese también la gran similitud entre las trayectorias seguidas por el vehículo Romeo4R mediante ambos controladores. La diferencia fundamental entre ambos radica en el retraso del proceso de inferencia. Mientras la implementación híbrida es capaz de realizar una inferencia completa cada 2,16 microsegundos (con el FIM a 12,5 MHz), la realización software con el

procesador MicroBlaze a 50 MHz tarda 12,31 milisegundos.

Los resultados de implementación sobre una FPGA Spartan-IIE200 indican un consumo de recursos de 1566 slices (66%) para la realización híbrida y de 1281 slices (54%) para la realización software.

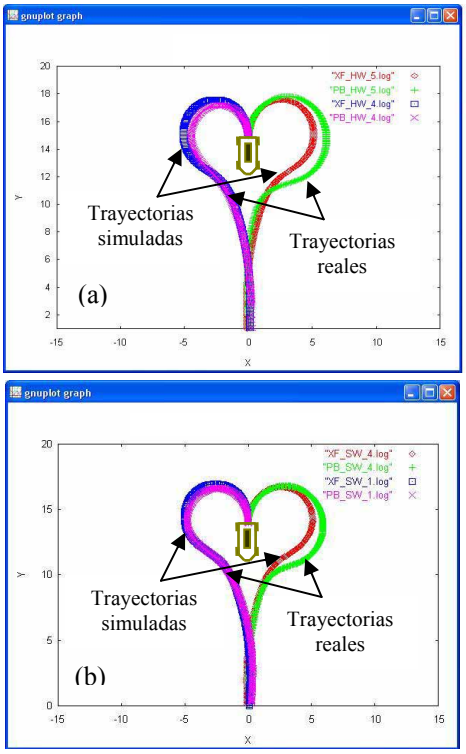


Figura 6: Trayectorias de aparcamiento mediante controladores SoPC híbridos (a) y software (b).

Para el caso más general y complejo del aparcamiento libre, el módulo de inferencia difuso está compuesto por las seis bases de conocimiento mostradas en la Figura 7.

Las bases de conocimiento *Position*, *Planning* y *Direction* corresponden a estructuras de toma de decisiones. *Position* decide si el vehículo se encuentra cerca o lejos de la posición objetivo en función del valor de las coordenadas (x,y). Su salida se combina con la orientación del vehículo (ϕ) en la base *Planning* para obtener la propuesta de sentido de circulación (*plan*). Sin embargo, dado que no se puede invertir bruscamente la polaridad del voltaje aplicado al motor de tracción, se combina esta salida con el valor anterior de la velocidad en la base *Direction*, cuya salida *way* indica el sentido que definitivamente debe seguir el vehículo. En la implementación hardware de estas bases de conocimiento se utilizan antecedentes basados en memoria y el método de defuzzyficación *MaxLabel*. En todos los casos se emplearon 5 bits para codificar las entradas y salidas y 3 bits para los grados de pertenencia.

Las restantes bases de conocimiento corresponden a sistemas con salidas interpoladas, utilizándose técnicas aritméticas para generar los antecedentes y el método de defuzzificación *FuzzyMean,* asi como una resolución de 8 bits en entradas y salidas y de 6 bits para codificar los grados de pertenencia en el caso de su realización harware.

La base de conocimiento *Celerity* determina el valor absoluto de la velocidad (*cel*), mientras que las bases *Backward* y *Forward* determinan la curvatura de Romeo4R en un sentido u otro (*bw*, *fw*). Dado que estas dos bases comparten las mismas entradas, es posible su unificación en un solo módulo.

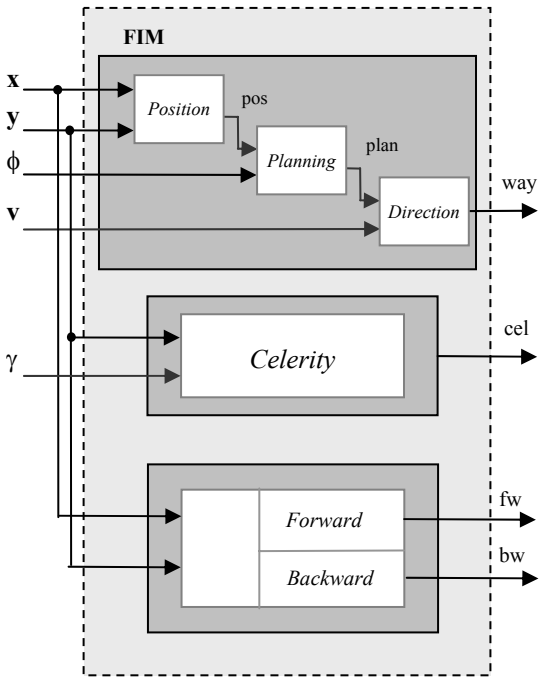


Figura 7: Estructura del FIM para el control de aparcamiento libre de Romeo4R.

La operación del controlador incluye la combinación de la salida que decide el sentido de la circulación (*way*) con las correspondientes al valor absoluto de la velocidad (*cel*) y las de curvatura de Romeo4R en un sentido u otro (*bw* y *fw*), obteniéndose los nuevos valores de velocidad y curvatura de acuerdo con las siguientes expresiones:

$$new\ v = way \,.\, cel$$

$$new\ \gamma = \begin{cases} fw, & if\ v > 0 \\ bw, & if\ v < 0 \\ 0, & if\ v = 0 \end{cases}$$

En este caso el controlador híbrido debe incorporar el sistema MicroBlaze y la estructura hardware del módulo de inferencia complejo de la Figura 7. Los resultados de implementación indican un consumo de recursos de 1.899 slices (80% de la Spartan-IIE200). La realización del

controlador software, al no requerir la implementación del FIM, es idéntica a la del caso anterior. La Figura 8 muestra la trayectoria real de la evolución de Romeo4R con el controlador híbrido, la cual difiere ligeramente de la obtenida mediante simulación por las irregularidades del terreno. Los resultados de trayectoria con el controlador software son similares.

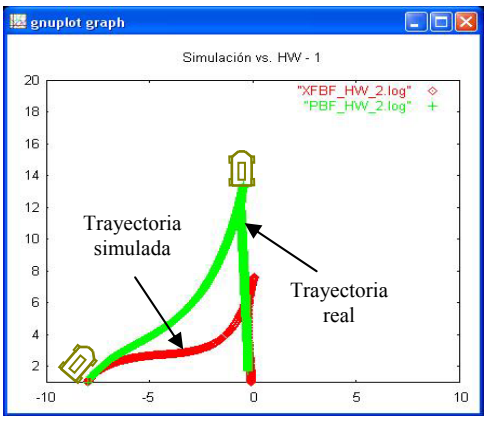


Figura 8: Trayectoria de aparcamiento libre de Romeo4R mediante el controlador difuso híbrido.

Nuevamente, la diferencia fundamental entre ambas realizaciones de controladores difusos SoPC radica en la velocidad de inferencia, siendo mucho más significativa en este caso: mientras el controlador híbrido produce una inferencia completa cada 2,88 microsegundos (con el FIM a 12,5 MHz), el controlador software (con el procesador MicroBlaze a 50 MHz) tarda 831 milisegundos.

5 CONCLUSIONES

Se ha descrito una plataforma reconfigurable basada en FPGAs que permite la implementación de diversas opciones de controladores difusos empotrados, tanto software como híbridos. Dicha plataforma se basa en la utilización del módulo IP MicroBlaze como procesador empotrado y del entorno de desarrollo *Xfuzzy* y sus herramientas de síntesis para la generación del módulo de inferencia difuso, existiendo diversas opciones de implementación hardware del mismo.

Tanto el sistema de procesado basado en MicroBlaze como el hardware del sistema de inferencia pueden ser fácilmente configurados para obtener una estructura del controlador difuso acorde a las necesidades del proceso a controlar.

La aplicación de la plataforma al control de un robot móvil utilizando una placa comercial de desarrollo de FPGAs permite validar su aplicabilidad. Los resultados obtenidos ponen de manifiesto la similitud existente entre el comportamiento de las implementaciones software y hardware del módulo de inferencia con diferencias significativas en el consumo de recursos y, fundamentalmente, en la velocidad de respuesta.

Referencias

[1] K. Passino, S. Yurkovich, *Fuzzy Control*, Addison-Wesley, 1998.

[2] J. Yen, R. Langari, L. Zadeh, Eds., *Industrial Applications of Fuzzy Logic and Intelligent Systems*, IEEE Press, 1995.

[3] C. Von Altrock, *Adapting existing Hardware for Fuzzy Computation*, in Handbook of Fuzzy Computation, Institute of Physics Publishing, 1998.

[4] I. Baturone, A. Barriga, S. Sánchez-Solano, C.J. Jiménez, D. López, *Microelectronic Design of Fuzzy Logic-Based Systems*, CRC Press, 2000.

[5] L. Reyneri, *Implementation issues of neuro-fuzzy hardware: going toward HW/SW codesign*, IEEE-Transactions-on-Neural-Networks, vol. 14, n.1, pp. 176-194, Jan. 2003.

[6] A. Cabrera, S. Sánchez-Solano, R. Senhadji, A. Barriga, C.J. Jiménez, *Hardware/software codesign methodology for fuzzy controllers implementation,* IEEE International Conference on Fuzzy Systems, pp. 464-469, Honolulu, May 2002.

[7] W. Savage, J. Chilton, R. Camposano, *IP reuse in the system on a chip era*, Proceedings 13th International Symposium on System Synthesis, pp. 2-7, Sep. 2000

[8] *MicroBlaze Reference Guide*, Xilinx Inc., 2002

[9] S. Sánchez-Solano, A. Barriga, C.J. Jiménez, J.L. Huertas, *Design and Applications of Digital Fuzzy Controllers*, IEEE International Conference on Fuzzy Systems, pp. 869-874, Barcelona, Jul. 1997.

[10] D.R. López, C.J. Jiménez, I. Baturone, A. Barriga, S. Sánchez-Solano, *Xfuzzy: A Design Environment for Fuzzy Systems,* IEEE International Conference on Fuzzy Systems, pp. 1060-1065, Anchorage, May 1998.

[11] E. Lago, C.J. Jiménez, D.R. López, S. Sánchez-Solano, A. Barriga, *Xfvhdl: A Tool for the Synthesis of Fuzzy Logic Controllers*, Proceedings of Design, Automation and Test in Europe, pp. 102-107, Paris, Feb. 1998.

[12] I. Baturone, F.J. Moreno-Velo, S. Sánchez-Solano, R. Martín de Agar, A. Ollero, *Automatic Design of Fuzzy Control Systems for Autonomous Mobile Robots*, Proc. 28th Annual Conference of the IEEE Industrial Electronics Society, pp. 2457-2462, Sevilla, Nov. 2002.

[13] *Digilab 2E Reference Manual*, Digilent Inc, 2002.

PROPUESTA DE UN MODELO DE CUANTIFICACION DEL RENDIMIENTO LABORAL MEDIANTE LOGICA FUZZY

Dr. Miguel Delgado Calvo
DECSAI Univ. De Granada,
Granada, España
mdelgado@ugr.es

Mg. Francisco Cofré Gajardo
Depto. Infor. y Comput. Univ. Tecnológica
Metropolitana, Santiago, Chile
fcofre@keko.com.ar
fcofre@utem.cl

Resumen

El rendimiento laboral de una persona es fundamentalmente función de sus cualidades y del puesto de trabajo que desempeñe. Si analizamos los modelos que se han propuesto para evaluar este rendimiento comprobamos que todos ellos admiten que la conducta humana en el trabajo se puede parametrizar por medio de un conjunto de variables predictoras (cualidades), si bien no hay acuerdo unánime de cuales son estas aunque todos los modelos actúan bajo una filosofía booleana considerando la presencia-ausencia de cada cualidad en cada persona y la necesidad o no necesidad de una cualidad para un trabajo.

Independientemente del modelo empleado es claro que tanto las cualidades como su necesidad o importancia para cada puesto de trabajo son factores de naturaleza imprecisa que se evalúan de forma más correcta considerando valores en [0,1] en lugar de en {0,1}.

En este trabajo proponemos un modelo de cuantificación del rendimiento laboral a través de lógica fuzzy, asignando las importancias relativas de los atributos por medio de operadores Owa; aún cuando se obtienen índices, estos son asimilables a etiquetas.

Palabras Clave: Rendimiento, Trabajo, Fuzzy, Owa.

1 INTRODUCCION.

Puesto que las organizaciones involucran el desempeño de cada uno de sus miembros humanos, se requiere disponer, al menos a nivel global, de un mecanismo que sea capaz de controlar los distintos factores, que influyen en el desempeño de una persona en la organización donde está vinculado; ello permitirá que la organización logre sus objetivos en forma más eficiente y que sus miembros realicen las tareas encomendadas en forma fluida y sin requerir de costosos mecanismos de presión y/o control, alcanzando altos niveles de satisfacción para ambas partes.

La gestión del recurso humano es una labor bastante antigua dentro de las organizaciones, alcanzando su máxima expresión en las empresas, al intentar lograr máximos niveles de rendimiento; sin embargo, a pesar de los aportes mayoritariamente cualitativos de las ciencias de cognición humana, esta labor suele desarrollarse en forma casi intuitiva; durante muchas décadas se han consignado como parámetros claves: la experiencia y la formación profesional, pero no hay acuerdos respecto de sus importancias relativas cuantitativas a la hora de "medir" el rendimiento laboral esperado y/o el más probable.

El estímulo más utilizado tiene la forma de recompensa monetaria, aún cuando se pueden utilizar diversas otras formas (permisos, ascensos, flexibilidad horaria, otros), pero su capacidad de inducir a mejores resultados tiene algún punto de agotamiento. Se ha comprobado que la curva de recompensa-producción tiene la forma que ilustra la siguiente gráfica:

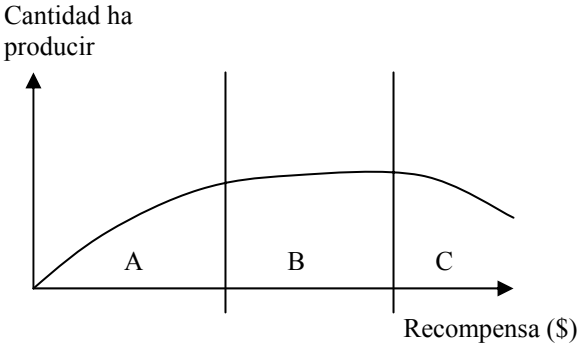


Fig.#1: Agotamiento del dinero como estímulo**.**

La zona "A" muestra el incremento en la productividad asociado a los incrementos en las recompensas, la zona "B" exhibe incrementos marginales positivos pequeños de productividad ante las variaciones positivas de recompensas, la zona "C" muestra los incrementos marginales negativos de productividad a pesar de continuar aumentando las recompensas.

2 MODELO CONCEPTUAL.

Integrando diversos modelos procedentes de las Ciencias Sociales, más específicamente en el ámbito de la Gestión de Recursos Humanos, proponemos el siguiente modelo conceptual que vincula los predictores genéricos con el rendimiento o desempeño laboral esperable:

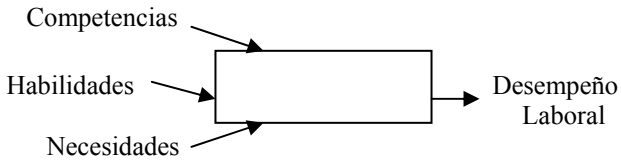


Fig.#2: Modelo de variables predictoras del rendimiento laboral.

Las variables (o mejor dicho macro-variables): Competencias, Habilidades y Necesidades constituyen los factores que gozan de mayor consenso en los diversos estudios disponibles, lo que resulta lógico si consideramos que:

Competencias; corresponden a los conocimientos adquiridos por un individuo para realizar una tarea, estos conocimientos pueden obtenerse ya sea en actividades de laboratorio (clases supervisadas) y/o en actividades experimentales propiamente tales (experiencia laboral).

Habilidades; corresponden a los talentos innatos de un individuo, los cuales se desarrollan durante las diversas experiencias de vida.

Necesidades; corresponden a los factores que determinan la búsqueda de los satisfactores asociados y por ende a la necesidad de "hacer" para alcanzarlos, estamos hablando de la motivación o ganas de hacer.

El análisis de las 3 macro-variables mencionadas como factores predictores, ha permitido constatar que detrás de ellas se encuentran varios factores o variables elementales. Es por ello que se las ha llamado "macro variables", mientras que los factores elementales que estas aglutinan pasaron a denominarse "factores o variables elementales".

Admitiremos que existen:

5 Competencias, caracterizadas por:

1. Básicas; lectura, comprensión de textos, relaciones matemáticas.
2. Técnicas; manejo de tecnologías específicas.
3. De Comportamiento; obediencia, compromiso, trabajar en equipo, adaptabilidad, escuchar, observar, comunicarse oralmente.
4. Intelectuales; análisis, síntesis, conclusión, pronósticos, concentración.
5. Orden y Seguridad; motricidad, seguir prescripciones para hacer una rutina de trabajo.

34 Habilidades, caracterizadas por:

1. Analítico; Desglosa los componentes, ve: estructuras, patrones y formatos.
2. Armónico; Evita las confrontaciones, no impone sus opiniones.
3. Auto-confiado; Confía en sus decisiones, es seguro de sí mismo.
4. Carismático; Se interesa por los otros.
5. Competitivo; Está pendiente del desempeño de los demás, le gustan las mediciones.
6. Comunicador; Le gusta: explicar, describir.
7. Conector; Supone que todo ocurre por alguna razón, respeta la diversidad.
8. Contextual; Mira el pasado para comprender el presente y predecir el futuro.
9. Creyente; Valora: la responsabilidad, el altruismo, la ética; apegado a los valores.
10. Desarrollador; Ve potencial en los demás, estimula a los otros hacia nuevos desafíos.
11. Disciplinado; Necesita un mundo: predecible, ordenado y planeado; es preciso y meticuloso.
12. Empático; Siente las emociones de quienes lo rodean.

13. Emprendedor; Necesita lograr algo tangible; tiene un hálito de descontento.
14. Enfocado; Necesita metas explícitas; hace que los demás no se desvíen del camino.
15. Equilibrado; Amante de la equidad; las normas son claras y se aplican para todos por igual.
16. Estratégico; Actúa después de anticipar dificultades y escoger alternativas.
17. Estudioso; Le interesa aprender sin importar el contenido.
18. Gusta de la excelencia; Valora a los expertos en algo específico y no en todo.
19. Flexible; Vive el momento; descubre su futuro en cada opción que se le presenta a diario.
20. Futurista; Disfruta evocando visiones del futuro; declara: "no sería genial si .....".
21. Ideador; Descubre dentro de lo complejo el porqué las cosas ocurren.
22. Inclusivo; Incluye a todos los demás sin discriminar para que se sientan parte.
23. Individualizador; Encuentra las cualidades únicas de cada persona; no le gusta generalizar.
24. Iniciador; Siempre está impaciente por actuar; primero experimenta a pesar de los riesgos.
25. Inquisitivo; Colecciona de todo ya que algún día servirá.
26. Intelectual; Disfruta del ejercicio intelectual; se pregunta e intenta responderse.
27. Mandatario; Le gusta imponer sus opiniones; se hace cargo de las cosas.
28. Organizador; Le gustan los rompecabezas.
29. Positivo; Elogia; siempre busca el lado bueno de una situación; ve el trabajo entretenido.
30. Prudente; Cuidadoso; siempre alerta; percibe los riesgos y los hace visibles para minimizarlos.
31. Relacionado; Prefiere estar con los amigos cercanos; acepta los riesgos de amistad genuina.
32. Responsable; Asume sus compromisos; asocia compromiso con reputación.
33. Restaurador; Anhela restaurar el orden perdido; le gusta resolver problemas.
34. Significativo; Necesita ser admirado por: creíble, profesional, exitoso; busca constantemente nuevas metas.

12 Necesidades, caracterizadas por:
1. Existencia; de tipo fisiológicas.
2. Afiliación; ser parte de una comunidad.
3. Poder; poder influir en otros.
4. Realización; preocupación por el futuro.
5. X; requiere el qué y el cómo para hacer.
6. Y; requiere el qué solamente para hacer.
7. Meta General; basta con una idea global de la meta.
8. Meta Específica; requiere una idea detallada de la meta.
9. Meta Implícita; basta con una idea borrosa por meta.
10. Meta Explícita; requiere una idea concreta por meta.
11. Meta Cercana; funciona bien con metas de corto plazo.
12. Meta Lejana; funciona bien con metas de largo plazo.

Por otro lado, aparecen los distintos roles (puestos de trabajo) que un individuo puede abordar en una empresa. Supondremos que para poder desempeñar de forma óptima un rol se requerirá la presencia en el individuo de algunas o de todas las variables en estudio, si bien con distinto nivel de importancia y, por lo tanto, un individuo será más o menos apto para asumir ciertos roles concretos.

3 FUZZIFICACION DE LOS DATOS.

Cuando se aborda la medición de los factores relevantes en un individuo, se observan dos características fundamentales: la aparición de un factor en un individuo es, no brusca, y por otro lado el valor de cada uno de los factores elementales identificados tiene una cuantía incierta, ya que los valores que adoptan no tienen causales concretas conocidas, algo propio si se trata de conductas parametrizadas de comportamientos humanos.

Estas dos características nos sugieren: la necesidad de formular estas valoraciones empleando la Lógica Difusa.

Veamos ahora como fuzzificamos cada uno de los atributos o variables elementales a ser consideradas en el modelo.

Los individuos de interés exhiben atributos con cierto grado de presencia, no necesariamente cada atributo está o no presente, por lo tanto nuestros datos los denotaremos por:

$$\left(individuo_i \, , \; grado-pertenencia_{i_{atributo\,j}} \right) \qquad (1)$$

Esta representación señala que el individuo i de la muestra, presenta el atributo j con un grado de pertenencia o cercanía, sujeto a cierta vaguedad; luego, al momento de adoptar esta notación también adoptamos la aplicabilidad de las propiedades y principios que la lógica difusa provee.

Sean los conjuntos: C: Competencias.
H: Habilidades.
N: Necesidades.
donde: C= { b, t, co, i, o }
H= { 1, 2, 3, … , 33, 34}
N= {e, a, p, r, x, y, mg, me, mi, mex, mc, ml}

Dado que cada individuo exhibe valores de: C(b), C(t), …, C(o), y considerando que el valor de C(b) presenta cierta incertidumbre, al igual que C(t) y C(o), su magnitud resulta vaga o borrosa; luego se hace necesario crear un

subconjunto fuzzy para C(b), otro para C(t) y así sucesivamente otro para C(o).

Sea X el conjunto de todos los individuos en estudio, y sea CB el subconjunto fuzzy de todos los individuos de X que cumplen C(b), con algún grado de pertenencia, basado en (L. A. Zadeh, 1965), se puede escribir:

Existe la función $\mu CB: X \rightarrow [0, 1]$

Luego: $CB = \{(xi, \alpha i), \forall xi \in X, \alpha i \in [0, 1]\}$ (2)

De igual forma se puede expresar CT como el subconjunto fuzzy de todos los X que cumplen C(t) con algún grado de pertenencia, como:

$$CT = \{(xi, \alpha i), \forall xi \in X, \alpha i \in [0, 1]\} \quad (3)$$

Lo mismo se puede expresar para: CCO que cumplen C(co), CI que cumplen C(i), CO que cumplen C(o); de la misma forma se puede proceder con H y N, ello genera los subconjuntos fuzzy de X: H1, H2, …, H33, H34, NE, NA, …, NMC, NML; respectivamente.

Veamos gráficamente algunos casos:

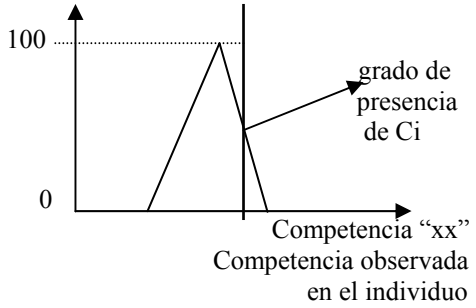


Fig.#3: Grado de presencia de la competencia observada.

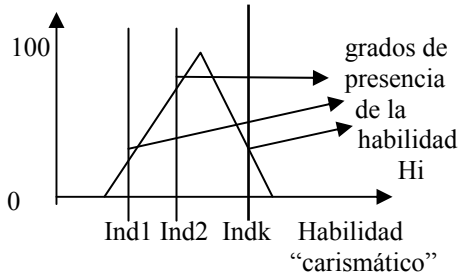


Fig.#4: Grado de presencia de la habilidad "carismático" en varios individuos.

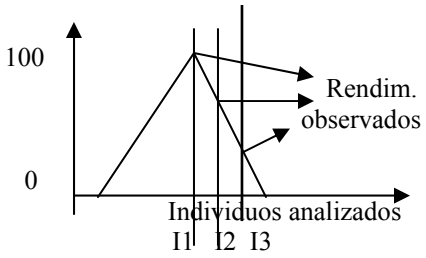


Fig.#5: Grado de presencia del rendimiento laboral para varios individuos.

Dado que los grados de presencia que cada individuo presenta, en cada uno de los atributos ligados a las variables en estudio, son distintos, la cuantificación del rendimiento laboral requerirá la integración de los atributos involucrados.

Para simplificar y sin pérdida de generalidad sólo consideraremos aquellos atributos cuyos grados de presencia sean a lo menos iguales a índices definidos como cotas mínimas.

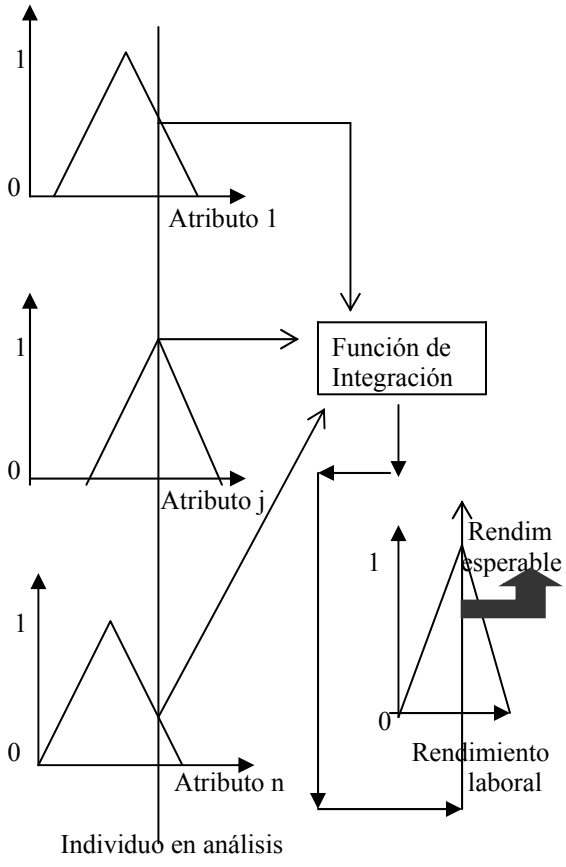


Fig.#6: Vinculación entre atributos presentes y el rendimiento laboral esperable.

La función de integración, como lo indica su nombre, deberá reunir todos los grados de presencia para todos los atributos medidos, de forma tal que genere un grado representativo del individuo, con respecto al rendimiento laboral más probable. A nivel intuitivo, se podría pensar en una función de integración tipo: sumatoria, unión, promedio, integral o alguna otra relación matemática.

4 MODELO PROPUESTO PARA CUANTIFICAR EL RENDIMIENTO LABORAL.

Con el propósito de alcanzar mayor conocimiento, a partir de los datos inicialmente disponibles (regla(s) básica(s) y algún(os) dato(s) aislado(s)), incorporaremos la utilización de operadores, ello permitirá agregar información a la ya conocida.

Del modelo conceptual quedó en evidencia que los atributos vinculados a los roles en los distintos niveles organizacionales demandan diferentes importancias relativas, ello nos llevó a contemplar la utilización de "ponderaciones".

La utilización de ponderación lleva asociado el hecho de que un atributo con grado de pertenencia bajo sea "premiado" en desmedro de otro atributo con grado de pertenencia alto, cuya ponderación sea baja; la lógica difusa provee algunos operadores que intentan superar esta deficiencia.

Como se busca agregar conocimiento a partir de ciertos datos inicialmente disponibles, un buen camino de exploración comienza a partir de los denominados "operadores de agregación", presentes en diversos desarrollos de sistemas inteligentes.

A través de la utilización de los operadores Owa es posible encontrar una forma de calcular la importancia relativa de cada uno de los atributos elementales asociados a las 3 macro variables, en este caso se trabaja con los atributos ligados directamente con el rol de interés.

Como se dispone de una muestra de individuos, donde cada uno de ellos exhibe algún grado de pertenencia con los atributos: h, c y n; aplicamos el operador Owa para ponderar y encontrar el **"peso más representativo"** de cada uno de los atributos.

Luego, genéricamente se tiene:

$FH(a1, a2, ..., an) = \sum_{j=1}^{n} wj\ bhj$, establece el peso promedio ordenado de la habilidad, $\forall$ xi $\in$ X. (4)

$FC(a1, a2, ..., an) = \sum_{j=1}^{n} wj\ bcj$, establece el peso promedio ordenado de la competencia, $\forall$ xi $\in$ X. (5)

$FN(a1, a2, ..., an) = \sum_{j=1}^{n} wj\ bnj$, establece el peso promedio ordenado de la necesidad, $\forall$ xi $\in$ X. (6)

Modelo Propuesto:

$$Rl_i = H_i * \left(\sum_{j=1}^{n} w_j b H_j\right) + C_i * \left(\sum_{j=1}^{n} w_j b C_j\right) + N_i * \left(\sum_{j=1}^{n} w_j b N_j\right); \quad \forall i = 1, \ldots, n \tag{7}$$

donde: Rli es el rendimiento laboral esperable del individuo xi $\in$X, del total de n individuos de la muestra bajo estudio; w_{jb} corresponde a los operadores Owa; H_j corresponde a los grados de pertenencia de las habilidades; C_j corresponde a los grados de pertenencia de las competencias; N_j corresponde a los grados de pertenencia de las necesidades.

Un tema que deberá ser abordado es el que establece que los pesos o importancias relativas de las macro variables serían distintos para diversas familias de rendimientos laborales asociados con los roles, en función del nivel organizacional en el cual está inserto cada rol.

A través de la utilización de los operadores Owa es posible encontrar una forma de calcular la importancia relativa de cada uno de los atributos elementales asociados a las 3 macro variables.

La aplicación del modelo de cuantificación del rendimiento laboral propuesto, genera valores numéricos, por ejemplo: 1.35, 1.09, 1.47; lo cual podría ser considerado irrelevante si le los mira en forma aislada; no obstante, cuando se aplica sobre una muestra de individuos, al comparar los rendimiento resultantes se puede seleccionar aquel miembro de la muestra que obtuvo el mayor valor.

Otra forma de utilizar el modelo de cuantificación propuesto, es diseñar etiquetas que estén correlacionadas con los resultados numéricos obtenidos como rendimiento laboral (no fuzzificar el índice resultante); ello permite establecer si, por ejemplo, el rendimiento es: alto, menos que alto, medio, menos que medio, más que bajo, bajo.

5 APLICACIÓN DEL MODELO PROPUESTO.

Sea X = {x1, x2, x3, x4, x5} y sean los subconjuntos fuzzy:

H = 0.2/x1, 0.6/x2, 0.4/x3, 0.3/x4, 0.7/x5
C = 0.1/x1, 0.4/x2, 0.7/x3, 0.2/x4, 0.3/x5
N = 0.0/x1, 0.3/x2, 0.5/x3, 0.6/x4, 0.1/x5

Sean las siguientes etiquetas de RL:
1.00 – 0.60: Alto rendimiento.
0.59 – 0.45: Buen rendimiento.
0.44 – 0.35: Bajo rendimiento.
0.34 – 0.00: Deficiente rendimiento.

Para nuestro caso consideraremos W = (0.05, 0.3, 0.1, 0.35, 0.2), luego aplicamos F para H, C y N respectivamente.

FH(0.2, 0.6, 0.4, 0.3, 0.7) = 0.05*0.7 + 0.3*0.6 + 0.1*0.4 + 0.35*0.3 + 0.2*0.2 = 0.4

FC(0.1, 0.4, 0.7, 0.2, 0.3) = 0.05*0.7 + 0.3*0.4 + 0.1*0.3 + 0.35*0.2 + 0.2*0.1 = 0.275

FN(0.0, 0.3, 0.5, 0.6, 0.1) = 0.05*0.6 + 0.3*0.5 + 0.1*0.3 + 0.35*0.1 + 0.2*0.0 = 0.245

RL1 = 0.1075; rendimiento deficiente.
RL2 = 0.4235; rendimiento bajo.
RL3 = 0.4750; rendimiento bueno.
RL4 = 0.3220; rendimiento deficiente.
RL5 = 0.3870; rendimiento bajo.

Luego, el individuo 3 es el único candidato aceptable.

Agradecimientos

Los autores quieren agradecer las reflexiones y críticas de: Dr. Mariano Muñoz, Mg. Liliana Calderón G., Mg. Homero Latorre A..

Referencias

[1]: Baker George, Jensen Michael, Murphy Kevin; "*Compensation And Incentives: Practice vs Theory*"; Harvard Business School & University Of Rochester; 1988 (Julio).

[2]: Buckingham Marcus & Clifton Donald O.; "*Ahora, Descubra Sus Fortalezas*"; Norma; 2001.

[3]: Cadenas José, Jiménez Fernando; "*Decisión Interactiva para un Problema General Fuzzy Multiobjetivo*"; Universidad de Murcia; 1994**.**

[4]: Castro J. Luis; "*Curso: Redes Neuronales y Sistemas Neuro-Difusos*"; Granada, España, 2001.

[5]: Cofré Francisco; "*Memoria Para Optar al Diploma de Estudios Avanzados dentro del Doctorado en Computación y Sistemas Inteligentes*"; Universidad de Granada, España; 2004.

[6]: Davis Keith & Newstrom John; "*Comportamiento Humano en el Trabajo*"; Mc Graw-Hill; 2002.

[7]: Espinosa Jairo and Vandewalle Joos; "*Constrained Predictive Control Using Fuzzy Models*"; Katholieke Universiteit Leuven; 1999.

[8]: Espinosa Jairo and Vandewalle Joos; "*Predictive Control Using Fuzzy Models*"; Katholieke Universiteit Leuven; 1998.

[9]: George J. Klir and Tina A. Folger; "*Fuzzy Sets, Uncertainty, and Information*"; Prentice Hall; 1988.

[10]: Gómez-Mejía Luis, Balkin David, Cardy Robert; "*Gestión De Recursos Humanos*"; Prentice Hall; 1998.

[11]: Katz Daniel &Kahm Robert; "*Psicología Social De Las Organizaciones*"; Trillas; 1998.

[12]: Marques Pereira Ricardo & Almeida Ribeiro Rita; "*Aggregation With Generalized Mixture Operators Using Weighting Functions*"; Universitá Di Trento; Lisboa; Portugal; 2002.

[13]: Mertens Leonard; "*La Gestión Por Competencia Laboral En La Empresa y La Formación Profesional; Normalización y Transferibilidad de la Competencia Laboral*"; OEI; 1997/8.

[14]: Modave F. & Kreinovich V.; "*Fuzzy Measures And Integrals As Aggregation Operators: Solving The Commensurability Problem*"; CS Dept. University Of Texas At El Paso; 2001.

[15]: Robbins Stephen; "*Comportamiento organizacional: Conceptos, Controversias y Aplicaciones*"; Prentice Hall; 1994.

[16]: Rosenblum Mark, Yacoob Yaser, Davis Larry; "*Human Emotion Recognition From Motion Using A Radial Basis Function Network Architecture*"; University Of Maryland; 1994.

[17]: Senge P.; "*La Quinta Disciplina*"; Editorial Granica; 1990.

[18]: Trillas Enric, Delgado Miguel, Vila M. Amparo, Castro J. Luis, Verdegay J. L., Moraga Claudio, Cuena José, Gutierrez Julio, Ruiz Antonio; "*Fundamentos e Introducción a la Ingeniería "Fuzzy"*"; Edit. Trillas; Omron Electronics S. A.; 1994.

[19]: www.bls.gov/; "*Occupational Classification System Manual*"; U.S. Department of Labor, Bureau of Labor Statistics; Washington, DC 20212-0001.

[20]: Yager Ronald; "*Induced Aggregation Operators*"; Iona College; New York; USA; 2002.

[21]: Zhao Feng; "*Intelligent Simulation in Designing Complex Dynamic Control Systems*"; The Ohio State University; 1995.

UN MODELO DIFUSO PARA FOCALIZACIÓN EN UN SISTEMA DE VISIÓN ACTIVA

Homero Latorre
Univ.Tecnológica Metropolitana
Utem,Chile
hlatorre@utyem.cl

Juan Luis Castro
Dep..C.C. e I.A.
Universidad de Granada
castro@decsai.ugr.es

José Manuel Benítez
Dep..C.C. e I.A.
Universidad de Granada
jmbs@decsai.ugr.es

Miguel García-Silvente
Dep..C.C. e I.A.
Universidad de Granada
mgs@decsai.ugr.es

Resumen

En este Trabajo se formula un modelo de visión que integre la Lógica Difusa y la Visión Activa Tradicional con el objetivo de desarrollar un sistema de Visión Activa basado en Lógica Difusa que permita realizar fijación visual, es decir, focalización sobre un objeto en el mundo visual real.

Palabras Clave: visión activa, visión difusa, modelo visual, focalización visual.

1 INTRODUCCIÓN

En los sistemas de visión activa, el problema principal radica en la formulación de un sistema de control cuya base principal sea la visión. Los cálculos de posición o ubicación relativa (error de posición) así como los cálculos de variación o de definición de las características propias de la imagen (iluminación, bordes, forma, características) deben realizarse sobre el plano visual, usando la relación de coordenadas Imagen versus coordenadas del mundo visual. Así mismo, la retroalimentación será dirigirá hacia los actuadores (plataforma, motores) que realizan el movimiento de las cámaras.

Se propone un modelo que contiene el análisis dinámico difuso de la visión y de la dinámica difusa del mundo visual, en relación con el análisis dinámico de los sistemas del movimiento difuso del sistema de control moto-ocular.

Se analizan los sistemas de visión activa binoculares en su formulación tradicional Crisp, además se analiza la Lógica Difusa y su aplicación al Procesamiento de imágenes en la modalidad estática, buscando las interacciones que permitan incorporar la Lógica Difusa a un Sistema de Visión Activa Binocular y Dinámica que sea capaz de percibir a los objetos en el mundo visual real y focalizar la atención sobre uno de ellos seleccionado previamente.

2 UBICACIÓN DIFUSA DE UN OBJETO

En un sistema de visión activa binocular, con capacidad de movimiento de vergencia, se debe detectar la posición de un objeto relativo a la ubicación de cada una de las cámaras. Al mismo tiempo, este movimiento de las cámaras, genera un desplazamiento de la imagen en relación al plano Imagen de cada cámara.

En el modelo de visión especificado, se debe considerar la existencia del plano visual (ubicado en el sensor) y la referencia al plano visión que corresponde al mundo real. Los sistemas de visión activa deben usar sensores de visión (cámaras) que tengan la cualidad de registrar sus variaciones en sus características visuales y en su posición.

Estos sensores necesitan tener un campo amplio de visión, que cubra todo la escena visual y a su vez una buena resolución visual, que permita que incluya una zona central de alta resolución (la fovea) y una zona circundante de baja resolución en las áreas periféricas al centro.

De este modo, se tiene una estructura en la cual su área central permitirá la ubicación, discriminación y reconocimiento de objetos. Esta está rodeada de una zona de baja resolución que contiene los objetos detectados y su ubicación en la imagen de manera que pueda ser analizada con posterioridad.

Se debe utilizar alguna estrategia para detectar la posición en la que se encuentra el objeto y, posteriormente, mover las cámaras para hacer coincidir el área de máxima resolución con la posición del objeto.

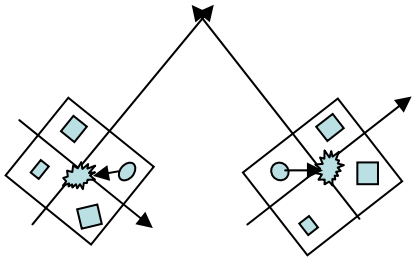


Figura 1: Plano imagen y centro focal.

En la figura 1 se puede comprobar cual es el sistema visual utilizado. Se usan cámaras con objetivos planos, de manera que tendremos un punto al que llamaremos centro focal (x_0, y_0) que pertenece al plano de visión (imagen), y que será el (X/2, Y/2).

El centro focal X_{OA}, X_{OB}, corresponde a los puntos extremos A y B, que definen el trazo o línea epipolar que define el "Plano de visión horizontal" para calibración cero.

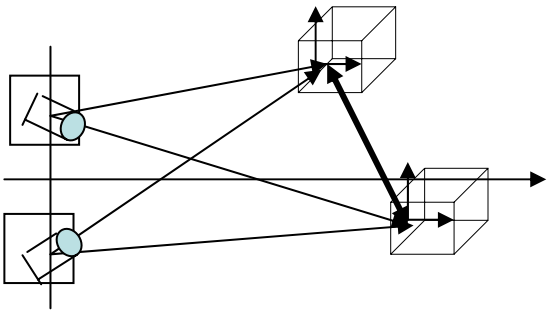


Figura 2: Plano visual y ubicación de objetos.

Cada objeto, tiene sus propias coordenadas exocéntricas, que reflejan su estado respecto al observador. Para este sistema, un objeto poseerá, lo que llamaremos, un área de alta resolución que corresponderá a la parte del objeto que se proyecta sobre el eje focal del plano de visión.

El área focal considerada para el reconocimiento de la ubicación de objetos en el mundo visual corresponderá al área definida por la "elipse mejor ajustada" al área del objeto que aparece en la imagen.

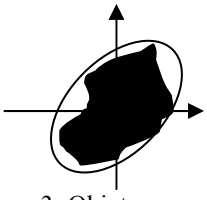


Figura 3: Objeto y contorno encerrado.

Para moverse desde la periferia hacia el centro visual será necesario establecer decisiones sobre varios parámetros visuales:

- Elegir un método para decidir qué objeto situado en la periferia se focalizará en el siguiente movimiento.
- Utilizar un mecanismo de movimiento rápido de desplazamiento Gaze, para focalizar sobre el objeto seleccionado en el mundo visual.

De este modo, existen dos cambios a considerar: de posición del objeto y de movimiento de las cámaras. De esta formas encontramos con los siguientes casos elementales que tienen que ser detectados: (a) un objeto estático y movimiento de las cámaras; (b) para una posición fija de las cámaras y un objeto en movimiento; (c) un objeto en movimiento y movimiento de las cámaras.

El sistema de visión activa estará descrito por el siguiente modelo:

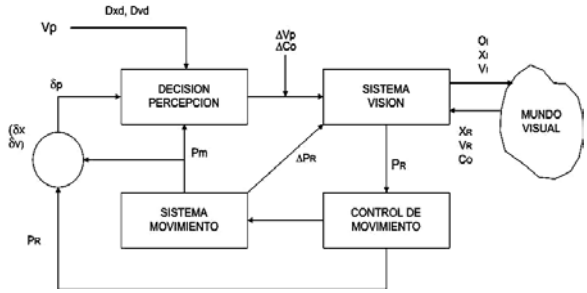


Figura 4: Modelo de sistema visión activa difuso

Este sistema de visión activa debe ser capaz de realizar todo el procesamiento visual de las imágenes captadas del mundo real, antes y después de un movimiento.

El problema a formular es: Dados varios objetos en el mundo visual real, se deberá conocer la ubicación (posición y orientación) de cada uno de ellos y controlar la ejecución de los movimientos para la focalización, y que de esta forma, fijen la visión sobre un objeto seleccionado según la percepción de este objeto en relación a toda la escena visual.

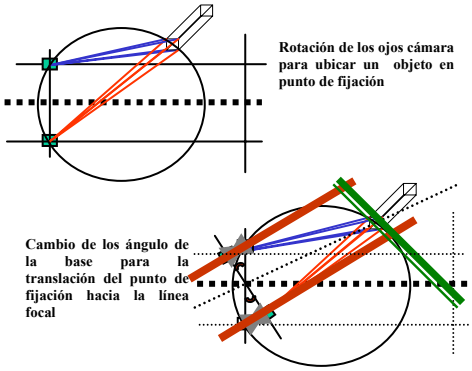


Figura 5: Movimientos para focalización.

A continuación, se debe decidir en qué grado el sistema está enfocado sobre el objeto propuesto y cómo debería moverse el sistema para focalizar sobre otro objeto seleccionado en el mundo real.

Este sistema, así descrito, se puede especificar de la siguiente forma:

$$\Pr : \begin{pmatrix} Ubicacion\ Mundo \\ \text{Re}al - Visual \end{pmatrix} \rightarrow \begin{Bmatrix} Xr\ \ Posicion \\ Co\ \ Orientacion \\ Vr\ \ Velocidad \end{Bmatrix} \quad (1)$$

Pr = {Xr, Vr, Cor}

Pv: Ubicación en el plano visual

$$Pv = \left\{ O_{Bi} / O_{Bi} = \left\{ O_{Bij} \right\}_{j=1}^{N} \exists O_B \in \left[X, V, C \right] \right\} \quad (2)$$

$$\Delta Pm : \begin{pmatrix} Variacion \\ Movimiento \end{pmatrix} \rightarrow \begin{Bmatrix} Ubicacion\ Posicion \\ Orientacion\ Caracteristicas \end{Bmatrix} \quad (3)$$

Δ Pm = {Pm / Pr - Pv /}

El problema se puede formular desde una visión del mundo real. Correspondería a conocer la ubicación (posición y orientación) de un objeto y poder seleccionar otros objetos y determinar sus respectivas ubicaciones.

La formulación matemática será:

$$\xi_{Perp} = \sum_{j=1}^{N} \left\| \overrightarrow{\mathrm{P}_{\mathrm{r}\,j}} - \overrightarrow{P_{Mj}} \right\| \leq \left\| \overrightarrow{P_{Vj}} \right\| \quad (4)$$

Donde: P_{rj}: Vector ubicación en el mundo visual,

P_{Mj}: Vector de movimiento en el plano Imagen,

P_{Vj}: Vector ubicación del objeto percibido en los planos Imagen **(x, y)** y plano visual **(x, y, z)**.

Los vectores ubicación tienen cuatro componentes:

$$P = [P(x, y, z), V(\overrightarrow{x, y, z}), O(x, y, z), I(x, y, z)] \quad (5)$$

Donde P es la posición, V la velocidad, O la orientación e I la segmentación, y además:

$$\vec{P} \subset \mathbb{R}^3, \ \vec{X} = (x, y, z)^T \quad (6)$$

La ecuación representa las características de la percepción visual necesarias para controlar un proceso de ubicación de los objetos en el mundo visual. Teniendo en cuenta las restricciones de la visión y de una segmentación factible que separe objetos que resultan claramente distinguibles en el mundo visual.

Debemos tener en cuenta las restricciones impuestas por la visión, la plataforma, los objetos y el mundo visual:

1) Los movimientos moto-oculares son discretos y los desplazamientos angulares son paso a paso. Además, la aceleración es proporcional a la suma la longitud de la secuencia discreta del movimiento.

$||\mathrm{Wm}|| \leq$ Step-paso θ motor posición

$||\alpha \mathrm{m}|| = \Sigma ||\mathrm{Wm}||$ velocidad angular.

2) Si la velocidad angular de movimiento es mayor a la capacidad visual de captura de imágenes en movimiento, la definición de la ubicación y posición del objeto será indistinguible y tenderán a un valor nulo.

$$\text{Si } ||\alpha \mathrm{m}|| > \text{Ips} \Rightarrow \text{Pr} \rightarrow 0$$

3) El movimiento de la plataforma de visión siempre cambiará el campo visual efectivo, incorporando o sacando objetos desde las áreas laterales de la visión.

$$\forall \Delta P_M \Rightarrow \exists P_R^N \therefore \left\{ O_B^N = (O_{B_{IN}} - O_{B_{OUT}}) \right\} \quad (7)$$

4) El sentido del movimiento tiene relación directa con la incorporación de objetos en el área visual lateral hacia el movimiento y con sacar objetos desde el área visual lateral opuesta al movimiento.

5) La plataforma de visión cubre un área visual específica que dependerá de la profundidad visual y del ángulo de movimiento y comprende además los cuatro cuadrantes visuales para un punto de focalización específico.

$$\forall P_F \ \exists P_R \therefore A_V \subseteq M_V, Z_V \subseteq M_V ; A_V \rightarrow (V_I, V_{II}, V_{III}, V_{IV}) \quad (8)$$

6) Los movimientos de vergencia (convergencia y divergencia) mantienen el área y los puntos de fijación y conservan la cantidad de objetos dentro del área visual. Son movimientos conservativos.

7) Los movimientos de seguimiento, Poursuit o tracking; desplazan el área visual el punto de fijación con una suave afiliación y desafiliación de objetos (objetos que se incorporan o salen del área visual pero que alcanzan a ser distinguibles en ubicación, posición y orientación) desde el mundo visual hacia el plano visual.

8) Los movimientos de Saccadic (cambios bruscos de fijación) trasladan el plano visual desde un área visual hacia otra, sin percibir las áreas visuales intermedias, creando toda una nueva área visual obtenida desde una nueva área del mundo visual real.

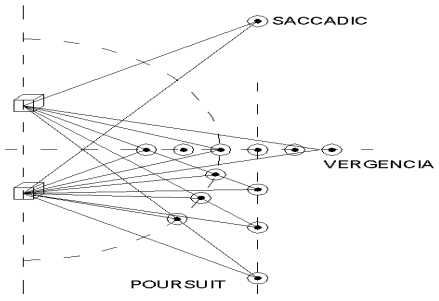


Figura 6: Movimientos visuales

La información obtenida a partir de la imagen se debe procesar e introducir en el sistema de visión para controlar el sistema de movimiento moto-ocular.

Una opción sería utilizar un sistema de control difuso similar a los propuestos en la literatura y utilizados para controlar plataformas o manipuladores robóticos. Pero en este caso, orientado al control moto-ocular basado en el paradigma de control de visión utilizamos retroalimentación de movimientos. Esto proporciona un adecuado movimiento en la vecindad del punto-objeto visual, con lo cual el sistema de percepción determina qué tipos de movimientos de deberán realizar.

En el modo de control tenemos que:

$$\delta P(\Delta x, \Delta y) = \delta Pp - \delta Pm \qquad \text{Donde:}$$

$$\delta X = Xp - Xm = (X_0, Y_0, Z_0) - (X_m, Y_m, Z_m)$$

$$\delta V = Vp - Vm = (Vx, Vy, Zy) - (Vmx, Vmy, Vzm)$$

Las variaciones de posición **δ(x, y, z)** indican la cercanía e influencia de los puntos (objetos) vecinos al punto focal del objeto seleccionado; y la variable de velocidad **(Vx, Vy, Vz)** se asocia a los movimientos de Poursuit y Saccadic para alcanzar puntos lejanos, a partir del punto focal.

3 CUESTIONES DE CONTROL

Una fase importante y crucial de la visión activa es generar y operar el control Gaze, el cual es la colección diferentes mecanismos de control para mantener la fovea, área de alta resolución, centrada sobre un objeto espacial dado.

Este sistema de control Gaze produce un cambio en los métodos de cálculo computacional aplicados a imágenes y visión dado que el proceso visual y el proceso de control están entrelazados para obtener un conocimiento y un procesamiento integral de la visión.

El control de parámetros gaze [7] involucra dos cuestiones a resolver: (1) Cambio: la detección autónoma y la identificación dinámica de los parámetros requeridos por la tarea visual y la acomodación de los requerimientos para la manipulación activa de los parámetros internos del sensor. (2) Estabilización: corresponde al problema de control de los parámetros óptico-visuales, con relación al modelo de control y sus problemas derivados.

En los sistemas de visión activa binoculares, con las cámaras y sus ejes ópticos en el mismo plano, el movimiento de orientación para fijar un objeto se llama Gaze Holding. Está basado en el ajuste del ángulo panorámico (Pan Azimut), de tal modo que ambos sensores (cámaras) miren y enfoquen sobre el mismo punto del espacio visual.

Una segunda cuestión de interés visual del Saccade está relacionada con el control de este movimiento. El redireccionamiento de la orientación de los sensores visuales (gaze), debido al movimiento Saccade, involucra un considerable número de decisiones de control en varios niveles.

4 MODELO DE CONTROL

En un sistema de visión activa binocular con capacidad de movimiento de vergencia se debe detectar la posición de un objeto relativo a la ubicación de cada una de las cámaras (ojos) y, a al mismo tiempo, este movimiento de las cámaras genera un desplazamiento de la imagen en relación al plano Imagen de cada cámara.

El modelo de control a utilizar se describe en la estructura de bloques funcionales siguiente:

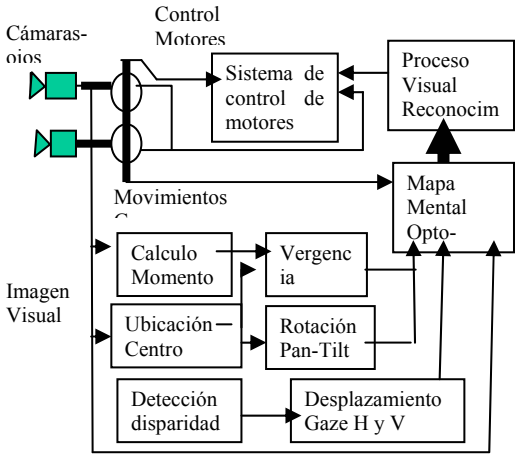


Figura 7: Sistema de Control Visual

De este modo, existen dos cambios, de posición y de movimiento, que deben ser detectados por el sistema para los casos elementales. Estos casos elementaltes son: (a) un objeto estático y cámaras en movimiento; (b) para una posición fija de las cámaras y un objeto en movimiento; (c) un objeto en movimiento y las cámaras también en movimiento.

En los sistemas de visión activa, el problema principal radica en la formulación de un sistema de control cuya planta principal sea la visión y los cálculos de posición o ubicación relativa (error de posición) así como los cálculos de variación o de definición de las características propias de la imagen (iluminación, bordes, forma, características). Esta formulación debe realizarse sobre el plano visual, usando la relación de coordenadas Imagen frente a las coordenadas del mundo visual y la retroalimentación será dirigida hacia los actuadores (plataforma, motores) que realizan el movimiento de las cámaras.

Esto difiere del clásico problema visual para el control de un robot en lazo cerrado, puesto que ahora se utiliza un "Sistema de Visión en Lazo Cerrado con Retroalimentación de Movimiento".

Para los sistemas de visión activa, se deberá centrar el desarrollo en un modelo que contenga el análisis dinámico de la visión y de la dinámica del mundo visual,

en relación con el análisis dinámico de los sistemas del movimiento moto-ocular.

De este modo, un sistema de visión activa quedará descrito por el modelo indicado en la Fig. 4. En este sistema de visión se realiza todo el procesamiento visual de las imágenes vistas del mundo real, antes y después de un movimiento.

5.- Algoritmo Difuso para Visión Activa

El sistema de visión activa, usado para realizar focalización sobre un objeto, utiliza una caracterización difusa del punto central de contacto visual entre el objeto puntual y los vectores visuales de enfoque o focalización sobre el objeto.

En el plano visual 2-D de la cámara de visión tendremos una representación difusa de la forma:

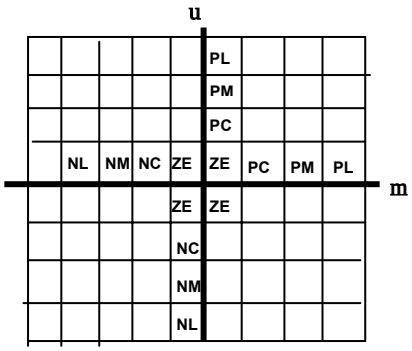


Figura 8: Plano imagen difuso

En estos dos ejes principales de enfoque tenemos las variaciones angulares de elevación (Tilt) y de rotación (Pan). Las etiquetas lingüísticas usadas son:

N = Negativo; P = Positivo; C = Cercano; M = Medio; L = Lejano.

Para la función de pertenencia consideraremos sólo la rotación sobre el plano horizontal **μ(x).**

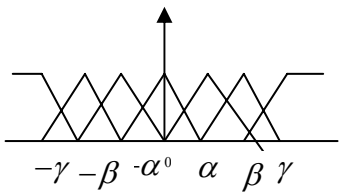


Figura 9: Pertenencia difusa de focalización

El sistema de control deberá fijar o focalizar la visión sobre el punto definido Ze, como centro del objeto dentro de la zona de visión central, para un objeto determinado detectado en la visión periférica y conocido su centro de masa como punto central del área del objeto.

El primer movimiento a realizar se llamará "Amplio", que involucra una secuencia de más de tres pasos angulares de movimiento. Una vez que nos encontramos cerca del punto central, el movimiento debe ser más fino, teniendo dos acciones básicas en tres pasos angulares: negativo - centro – positivo. De este modo, la función de pertenencia deberá relacionar la variable difusa cercana al centro de visión Ze con un movimiento fino de control. Cuando la zona de movimiento esté en la periferia visual, lejos del punto central, los movimientos deberán ser más amplios.

El rango de pertenencia a utilizar será: **(C-β) ≥ (δ -α) ≥ β,** donde **(C-β) , (δ - α)** y **β** son los límites de los movimientos posibles para el objeto a fijar la visión sobre el centro del objeto, según la función de pertenencia utilizada.

Como en la función de pertenencia las variables difusas están sobrepuestas, el motor de inferencia deberá determinar cuál será la salida de control a utilizar.

Además, ya que la función de pertenencia es compleja, pues existen más de dos conjuntos difusos posibles de posición y movimiento, una regla genérica de control y decisión difusa puede ser de la forma siguiente:

$$Si\ \overline{P}\ (posicion) \rightarrow Ai(lejania)\ y\ \overline{M}\ (movimiento) \rightarrow Bi(amplio)\ entonces\ \overline{R}\ (accion) \rightarrow Ci(mueve\ centro) \quad (9)$$

Las variables difusas de entrada son la posición **P** y el movimiento **M,** y la variable de salida será la acción de cómo moverse para cambiar la posición más cercana al centro del objeto.

Consideremos el sistema de visión activa formado por dos cámaras binoculares unidas a una distancia **d** sobre una línea base común; y cada una de ellas con movimientos de rotación $\theta i \ \ y \ \ \theta d$ para fijar la visión en el espacio visual:

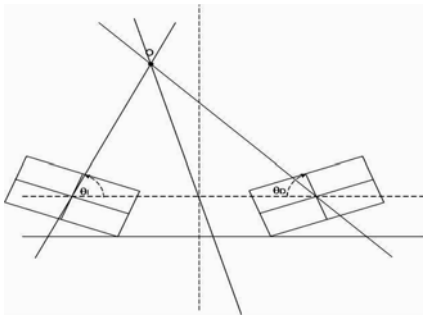


Figura 10: Sistema de Cámaras Binoculares

En este sistema, cada cámara tiene su propio plano visual de la imagen del objeto capturado. Así, las reglas de control serán

$$Regla \quad Ci \quad cámara \ izquierda$$
$$Si \quad \overline{Pi} \rightarrow A_{Ii} \quad y \quad \overline{M\theta i} \rightarrow B_{Ii} \qquad (10)$$
$$: entonces \quad R_{Ci} \rightarrow C_{Ii}$$

Donde: Regla cámara izquierda, **RCi**, deberá partir de una Posición **Pi** y un valor angular de movimiento **Mqi** y obtener un valor verdadero al llegar a la posición **Pio** de la función de pertenencia **m Ci** (P_o^I) .

$$Regla \quad Cd \quad camara \ derecha$$
$$Si \quad \overline{Pd} \rightarrow A_{Di} \quad y \quad \overline{M_{\theta d}} \rightarrow B_{Di} \qquad (11)$$
$$: entonces \quad R_{Cd} \rightarrow C_{Di}$$

Donde: Regla cámara derecha **RCd**, deberá partir de una Posición **Pd** y un valor angular de movimiento **Mqd** y obtener un valor verdadero al llegar a la posición **Pdo** de la función de pertenencia **m Cd** (P_o^D)

En ambas cámaras existen dos conjuntos difusos para validar la no certeza del objeto y fijar la visión sobre el centro del objeto. Esto depende de los conjuntos difusos de posición **m**A(P) y movimiento **m**B(M)**.** Por tanto, el peso de las acciones en cada cámara será:

$$cámara \ izquierda$$
$$\mu_{CI} = \mu_{AI}(Po) \wedge \mu_{BI}(Mo) \qquad (12)$$
$$cámara \ derecha$$
$$\mu_{CD} = \mu_{AD}(Po) \wedge \mu_{BD}(Mo) \qquad (13)$$

La decisión final del movimiento de ambas cámaras dependerá del valor de acción/reacción difusas de cada una de las cámaras:

$$\mu_{CID}(R_{ID}) = \mu_{CI}(R_I) \vee \mu_{CD}(R_D) \qquad (14)$$

Luego:

$$\mu_{CID}(R) = \left[\mu_{AI}(Po) \wedge \mu_{BI}(Mo)\right] \vee \left[\mu_{AD}(Po) \wedge \mu_{BD}(Mo)\right] \qquad (15)$$

De este modo, el movimiento en conjunto a realizar por ambas cámaras será $\mu_{cid}(R)$, que será el movimiento de la plataforma de visión obtenida como una combinación de μ_{ci} y μ_{cd}.

5 CONCLUSIONES

Se presenta un modelo de sistema de visión activa que utiliza teoría de Conjuntos Difusos para la focalización sobre un objeto específico. De entre todos los existentes en el mundo visual, se analiza una técnica difusa factible, el método de los clusters difusos para realizar segmentación y separación del objeto seleccionado desde la imagen completa del mundo visual.

En el modelo del mundo difuso visual se analiza la opción de describir los objetos utilizado el centro de masa en su modalidad difusa , como punto de fijación de los vectores visuales de ambas cámaras y el contorno encerrado por la elipse mas posibles , dado que se esta solo fijando el objeto y no describiendo su forma o reconociendo la forma del objeto .

Para la ubicación del objeto en el mundo visual dinámico se propone un modelo difuso que describa la presencia y desplazamiento de los objetos en el mundo visual, el cual considera la detección del objeto usando dos cámaras del modelo de visión binocular difusa y activa, utilizando correlación difusa cruzada para obtener la disparidad cero entre las imágenes del objeto entre ambas dos cámaras.

Referencias

[1] J.C Bezdek, S. K. Pal Fuzzy Models for Pattern Recognition IEEE Pres New York 1992

[2] J. M. Keller Fuzzy set theory in computer vision : A prospectus. Fuzzy Sets and Systems 90 (1997) 177-182

[3] J. Keller, R. Krishnapuram Fuzzy Set methods in computer vision. Kluwer Academic Publishers 1992

[4] J. Keller R. Krishnapuram P. Gader, Y. Choi. Fuzzy rule-based models in computer vision. Kluwer Academic Publishers 1996

[5] H. Latorre. Prototipo de Sistema de Visión Activa Informe Interno Usach Chile

[6] H. Latorre, J. Castro, J.M.Benitez, M.García-Silvente. Modelo de Vision Activa Difuso. Reporte Interno Doctorado Computación y Sistemas Inteligentes Decsai Utem Ugr 2004

[7] D. Marr. Vision. Freeman San Francisco. 1982

[8] E.L. Walker. A Fuzzy Approach to Pose Determination in Object Recognition. Proc. Conference of the north American Fuzzy Society.Syracuse NY 1997

DISEÑO DE UN CONTROLADOR DIFUSO PARA AJUSTAR LA FRECUENCIA EN LA SINCRONIZACIÓN DE UN GENERADOR ELÉCTRICO

Juan Luis Castro
Depto ciencias de la computación
Universidad de Granada
castro@decsai.ugr.es

Carlos J. Mantas Ruiz
Depto Ciencias de la Computación
Universidad de Granada
cmantas@decsai.ugr.es

Luis H. Escobar Arrue
Depto electricidad
Universidad Tecnológica Metropolitana
escobar466@hotmail.com

Resumen

En este tema presentamos el concepto de calidad de servicio de un sistema eléctrico, los problemas de un sistema de puesta en paralelo con operador y las ventajas de los controladores difusos. La presentación de estas cuestiones nos lleva a la motivación de este trabajo de investigación: implementar un controlador difuso para resolver los problemas que plantea la puesta en paralelo de varias máquinas con el fin de mejorar la calidad de servicio de un sistema eléctrico.

Palabras claves: lógica difusa, puesta en paralelo de generadores eléctricos, control.

1.- Introducción.

La calidad de servicio de un sistema eléctrico puede cuantificarse a través de varios parámetros relacionados con: la continuidad de servicio, las fluctuaciones de voltaje, el contenido armónico de las formas de onda de voltaje y de corriente, variaciones de frecuencia, y la regulación.

El concepto de calidad de servicio, como se puede apreciar es bastante amplio, de manera que no es posible sintetizarlo en un solo parámetro o índice. Por una parte, para las fluctuaciones lentas y rápidas de voltaje, existen diversos cuantificadores que dan cuenta de la presencia de tales anomalías e indican la necesidad de tomar medidas correctivas, dado que las fuentes de estos problemas son conocidas. Por otra parte, los cortes de suministro de energía eléctrica, que afectan a todos los usuarios produciendo graves distorsiones tanto en el desarrollo habitual de cualquier actividad como en el confort de las personas, se cuantifican midiendo su frecuencia de aparición y su duración.

Una manera de evaluar la confiabilidad del servicio eléctrico es a través de la disponibilidad.

En Chile la calidad de servicio está adquiriendo más importancia, dada la presencia de una mayor cantidad de cargas sensibles tanto a las variaciones de voltaje y frecuencia como a los cortes de suministro.

Como una forma de subsanar estos inconvenientes, es hacer trabajar máquinas en paralelo, teniendo presente su rendimiento óptimo y su buena calidad de servicio.

Para ello es necesario agregar/retirar máquinas al servicio de acuerdo a las necesidades de demanda, por ello es pertinente un sistema de sincronización (puesta en paralelo) automática de acuerdo con los requerimientos del cliente en cuanto a calidad y seguridad del servicio.

Actualmente controlar el sistema Chileno de energía es complicado puesto que:
Se han desarrollado procedimientos automáticos para controlar en forma aislada partes del sistema.

Generalmente se utiliza SCADA para tener la información y control de los distintos sistemas que forman el sistema intercomunicado de Chile.

En distintas partes del sistema eléctrico, se han desarrollado aplicaciones particulares para resolver el problema, basándose normalmente en el uso de métodos numéricos para el cálculo de los parámetros eléctricos**.[1]**

Ahora bien para desarrollar el sistema eléctrico se requiere que los generadores que entregarán la potencia al consumo, deben en primer lugar conectarse al sistema interconectado, para lo cual deben sincronizarse con la línea o sincronizarse con otros generadores, este proceso es casi manual excepto en las centrales eléctricas.

En esta parte del trabajo se dan las bases para reemplazar al operador humano en el sincronismo de generadores.

1.1.-Inconvenientes de operador humano frente a ventajas de controlador difuso

- Sistema gestionado por un operador humano (Inconvenientes)

1. Debido al instrumental utilizado se producen errores en las medidas que no se detectan.

2. Ante la variación de consumo de carga eléctrica, es necesario sincronizar el funcionamiento paralelo de varios generadores eléctricos:

   a) Depende de la concentración del operador.
   b) Depende de la reacción del operador.
   c) El operador debe ser personal conocedor del tema y con experiencia

   Estas características implican:
   - Menor confiabilidad.
   - Oscilaciones de carga en los puntos problema.

- Sistema gestionado por un controlador difuso (Solución de los inconvenientes anteriores)

1. Debido a la propia definición de conjunto difuso, un controlador difuso es robusto frente al manejo de información imprecisa.

2. Ante la variación de consumo de carga eléctrica, es necesario sincronizar el funcionamiento paralelo de varios generadores eléctricos:

   a) El controlador difuso siempre está disponible.
   b) Tiene un tiempo de respuesta bajo.
   c) El sistema informático se basa en reglas especificadas por una persona conocedora del tema y con experiencia.

   Estas características implican:
   - Mayor confiabilidad en el sistema.
   - Estabilidad en el reparto de carga eléctrica incluso en los puntos problema.

"Esto nos lleva a implementar un controlador difuso como sistema de sincronización automático para mejorar la calidad de servicio".

Generadores en paralelo

La mayoría de los generadores síncronos se utilizan en grandes redes de energía y, por consiguiente, se emplean en tándem con otros generadores, sin embargo, un generador síncrono puede servir como un único generador de una red de energía aislada. [2] [14]

El generador posee dos alimentaciones:
- acelerador del sistema de accionamiento mecánico
- corriente de campo de cc (Corriente continua).

Esta red de energía tiene las siguientes características:

- La frecuencia está determinada por la velocidad del sistema de accionamiento mecánico. La frecuencia eléctrica es $(\frac{P}{2})\ \omega_e$, donde ω_e es la velocidad mecánica del rotor. Por consiguiente se requiere un estricto control de la velocidad si se necesita que la frecuencia del sistema sea constante.

- El voltaje de la red está controlado en parte por la corriente de campo. Un aumento en la corriente de campo produce un incremento en la tensión generada.

2.-Estructura del Controlador Difuso

En la figura Nº 1 se muestra el diagrama que utilizaremos, con la nomenclatura siguiente [11],[13], [15] :

f_{ref} será la frecuencia de la red .
f_{act} será la frecuencia medida en esta muestra.
$e = f_{ref} - f_{act}$ y $de = e(k) - e(k-1)$

donde e: medida del error y de: derivada del error.

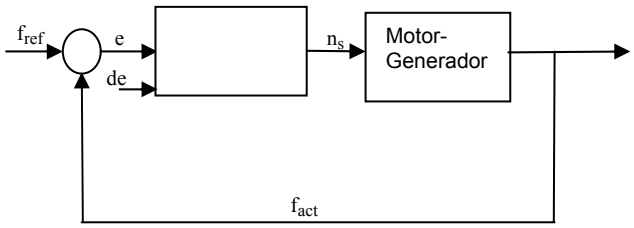


figura Nº 1

2.1.-Elección del Universo de Discurso de las Variables de Entrada

Es usual escoger universos de discurso estandarizados, p.e. [-1,+1] y luego aplicar factores de escala para cada variable según la planta a controlar [3], [4],[6]. En nuestro caso hemos supuesto que las referencias a seguir por la planta estarán comprendidas entre [-1,+1]:

1. El Error variará normalmente entre [-1,+1].
2. La Derivada lo hace entre [-1,+1].

Recordemos que para sincronizar el generador a la línea se deben controlar fundamentalmente su frecuencia y el voltaje de salida de la máquina, consideraremos ahora el controlar la frecuencia del generador.

Esta frecuencia debe ser la frecuencia de la red que en nuestro caso es de 50 Hz ± 1%, y que nosotros llamaremos frecuencia de referencia f_{ref}

Como variable de entrada consideramos $e = f_{ref} - f_{act}$, que corresponderá a la diferencia entre esas frecuencias y, que llamaremos error.

La otra variable de entrada a considerar será la derivada del error "de/dt" y, que corresponderá a la diferencia entre dos períodos del error, error actual e(k) y error en la lectura anterior e(k-1)

$$de = e(k) - e(k-1)$$

al considerar el rango de la frecuencia de la red de 50 ± 1% Hz, el rango permitido de funcionamiento es de 49,5 a 50,5 Hz que es considerado normal.

De acuerdo con lo anterior, consideraremos que los límites de universo de discurso está entre U = [49, 51] Hz,.
Lo anterior es debido que no es aceptable frecuencias muy distintas al rango propuesto, pues los generadores actuarían como motores, y en tal caso consumirían energía en vez de entregar energía.

Partición y definición difusa de las variables de entrada

Esta elección es idéntica para ambas variables de entrada, Error y Derivada. Hemos particionado el universo de discurso de las variables de entrada en 5 conjuntos borrosos, definidos cada uno con **funciones de pertenencia** triangulares. A cada conjunto borroso le asociamos una **etiqueta lingüística (NG: negativo grande, NP negativo pequeño, Z: cero, PP positivo pequeño, PG: positivo grande)**

El rango de valores para las variables lingüísticas error y la derivada del error, se muestran en la Figura Nº2:

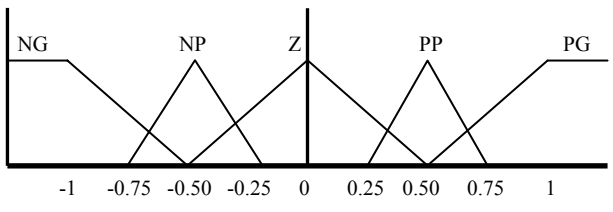


Figura Nº 2

2.2.-El Decisor o Máquina de inferencia

El bloque llamado "Decisor" o "Motor de Inferencias" es donde se almacena el banco de reglas de control, [7],[13],[16], con reglas de la forma:

***SI** (Error es NP) **Y** (Derivada es Z) **ENTONCES** (Salida es PP)*.

Para programarlo habrá que especificar:

1. Definición de la variable borrosa de control.
2. Definición del mecanismo de operación de todas las reglas simultáneamente.

Partición y Definición Difusa del Espacio de Salida

Para ajustar la frecuencia de la planta se debe actuar sobre el número de revoluciones que la máquina motriz debe entregar, para que esto suceda debemos actuar sobre la corriente de excitación de la máquina, para nuestro caso usaremos directamente n_s. (velocidad del sistema de accionamiento mecánico)

Para las máquinas de dos polos mantener una frecuencia de 50 Hz, significa que se le deben entregar 3000 rpm (revoluciones por minuto) que correspoderá a la referencia, lo que fija un rango de variación entre [2940, 3060] rpm, nuevamente consideramos un universo de discurso U=[-1, 1], y la variable de salida será n_s .

De manera análoga a como se hizo antes para el espacio de entrada, escogemos los siguientes conjuntos borrosos para la variable de control (salida del sistema), con las etiquetas (**NG: negativo grande, NM: negativo mediano, NP: negativo pequeño, Z: cero, PP: positivo pequeño, PM: positivo mediano, PG: positivo grande)**
En la figura Nº3, se muestra el rango de valores asumidas por la variable de control(salida del sistema).

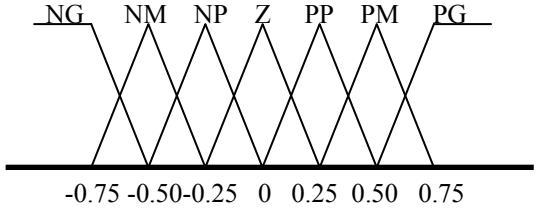
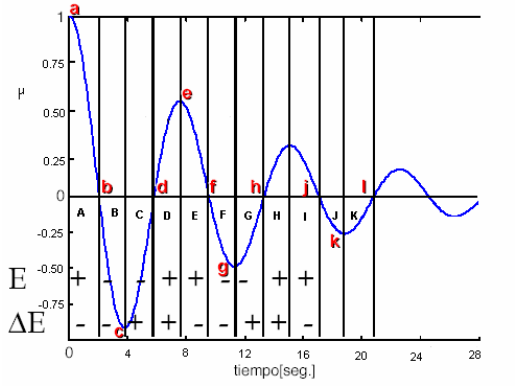


Figura Nº 3

A la luz de la Figura se entiende que hemos escogido particionar el intervalo [-1,+1] en 7 zonas, de forma triangular y con etiquetas lingüísticas:

2.3.-MECANISMO DE OPERACIÓN DE MAQUINA DE INFERENCIA

Las reglas de control las deduciremos con un experto de la forma siguiente:
Consideremos el plano temporal de respuesta de la planta.



Dividamos este plano en zonas que hemos clasificado en dos clases de atención:

1.- zonas de puntos críticos: que corresponde a los puntos donde el error o su derivada cruzan el eje del tiempo.

Por ejemplo,

en puntos a, e, i	e>0 y de = 0 ,	salida >0
b, f, j	e=0 y de < 0 ,	salida <0
c,g,k	e<0 y de = 0,	salida < 0
d, h, l	e=0 y de >0 ,	salida >0
valor deseado	e=0 y de=0,	salida=0

2.- zonas entre puntos críticos:

Por ejemplo en zonas

A, E	e>0	y de <0,	salida >0
B, F	e<0	y de<0,	salida <0
C, G	e<0	y de>0,	salida<0
D, H	e>0	y de>0,	salida>0
I	e>0	y de<0,	salida=0
K	e<0	y de>0,	salida=0

Esta forma de división en zonas tiene como objetivos: Reducción del pick y Tiempo de subida.

Para establecer este resultado (reducción del pick y tiempo de subida), cuando la derivada del error tienda a disminuir el error, se deberá acelerar en el mismo sentido para disminuir el tiempo de subida, cuando la derivada tienda a aumentarlo deberá propiciar un cambio de esta pendiente para evitar los picks, y así sucesivamente para así obtener las reglas en forma natural.

Lo que da origen a las siguientes reglas
Si se hubieran elegido tres funciones de pertenencia para las entradas del error y derror, con etiquetas **N: negativo, Z: cero y P: positiva,** se producen inconsistencias en las reglas Nº 6, 8 10 y 11, como se muestra en el recuadro siguiente:

1.- Si error =P y de=Z entonces salida P, reducir tiempo de subida, a, e, i
2.- Si error =Z y de=N entonces salida N , reducir pick b, f, j
3.- Si error =N y de=Z entonces salida N , reducir pick c, g, k
4.-Si error =Z y de=P entonces salida P , no tener osciilación d, h, l
5.- Si error =Z y de=Z entonces salida Z , bueno
6.- Si error =P y de=N entonces salida P ,reducir tiempo de subida A, E
7.- Si error =N y de=N entonces salida N ,reducir pick B, F
8.- Si error =N y de=P entonces salida N ,reducir pick C,G
9.-Si error =P y de=P entonces salida P ,no tener oscilación D, H
10.-Si error =P y de=N entonces salida Z ,bueno I
11.-Si error =N y de=P entonces salida Z , bueno. K

Lo que justifica que las variables de entrada se han dividido en cinco funciones de pertenencia a fin de evitar oscilaciones alrededor de cero (que sería un ciclo límite), cuando la granularidad no es muy fina.

De esta forma obtenemos las siguientes reglas:

1.- Si error =PG y de=Z entonces salida PG
2.- Si error =PP y de=Z entonces salida PP
3.- Si error =Z y de=NG entonces salida NG
4.- Si error =NG y de=Z entonces salida NG
5.- Si error =NP y de=Z entonces salida NP
6.- Si error =Z y de=PG entonces salida PG
7.- Si error =Z y de=PP entonces salida PP
8.- Si error =Z y de=Z entonces salida Z
9.- Si error =PG y de=NP entonces salida PM
10.-Si error =PP y de=NG entonces salida NM
11.-Si error =NG y de=PP entonces salida NM
12.- Si error =NP y de=PG entonces salida PM
13.- Si error =PP y de=NP entonces salida Z
14.- Si error =NP y de=PP entonces salida Z

A continuación se plantea la tabla Nº 1, con todas las reglas de inferencia

De\e	NG	NP	Z	PP	PG
NG	NM	NM	NP	PM	Z
NP	NM	NP	NP	Z	NP
Z	NP	NP	Z	PP	PP
PP	NP	Z	PP	PP	PM
PG	Z	PP	PP	PM	PM

Tabla Nº 1

Desfuzzificador

La señal crisp que se debe entregar al sistema es un cambio en las rpm de la máquina motriz, que actúa según las reglas propuestas.

A continuación en las figuras Nº 4 y 5 se muestran las salidas del sistema para distintos valores para la variable derror.



Figura Nº 4:Salida desfuzificada para derror=0.5

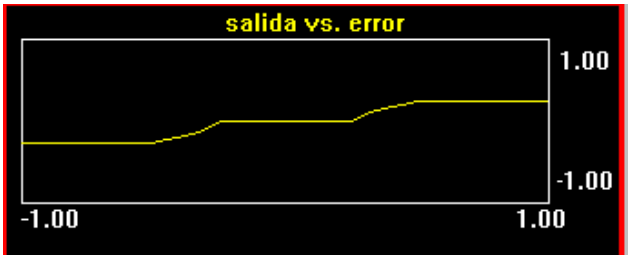


Figura Nº 5:Salida desfuzificada para derror=0

Finalmente se muestra la superficie en la figura Nº 6

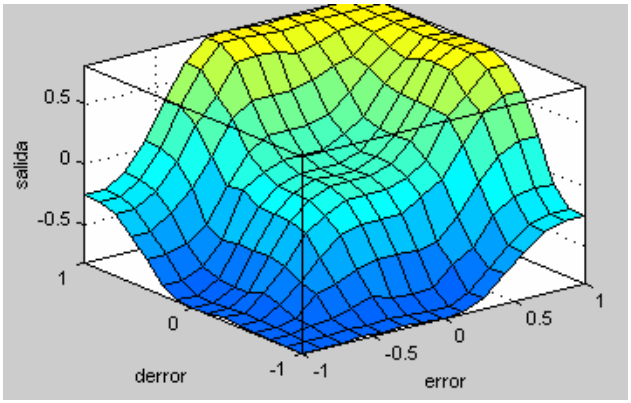


Figura Nº 6

Resultados

El controlador difuso diseñado se ha implementado y probado obteniendo algunos resultados como:

- La partición en cinco funciones de pertenecia de las variables de entrada error y derror, pues como se pudo establecer al utilizar sólo tres funciones de pertenencia se produjeron oscilaciones alrededor de cero, el cual sería un ciclo límite, como se puede apreciar en el recuadro anterior.
- La partición en siete funciones de pertenencia de la salida del controlador, nos permite hacer un ajuste más fino de la variable de control.
- Se deben mejorar los límites de las particiones para un mejor control de la variable. [12],[13],[17].

Después de las pruebas se concluye un aumento de la confiabilidad y rapidez del sistema a controlar

Referencias

[1].-D.Cámac y Rudnick, un algoritmo branch%bound para el despacho óptimo de carga. Pontificia Universidad Católica de Chile, proyecto Fondecyt 1930708.

[2].-J. R. Cogdell, Fundamentos de Máquinas Eléctricas, Prentice Hall, ISBN 970-26-0143-6, 2002

[3].-O. duarte V, aplicaciones de la lógica difusa, en Ingeniería e Investigación, Nº 45, 2000.

[4].-J. Jantsen; Fuzzy Supervisory Control, Technical University of Denmarck, November 1998.

[5].-J. Jantzen, Neural and Neurofuzzy Control, Tech. report Nº 99-11-999, nov 2003.

[6].-J. Jantzen, Design of Fuzzy Controllers, Tech. report Nº 98-E-864, august,1998.

[7].-J. Jantzen, Tunning of Fuzzy PID Controllers, Tech. report Nº 98-N-871, sept. 1998

[8].-D. Jenkins, K. Passino, An introduction to nonlinear analysis of Fuzzy Control System , Journal of intelligent and Fuzzy system, vol 7, Nº1, pp 75-103,1999.

[9].-J.F.López, Comparación entre sistemas difusos para el control de una banda transportadora dentro de un

proceso de manufactura flexible., U. Javeriana, Cali, 1995.

[10].-N. Nilsson, Inteligencia Artificial, Mc Graw Hill, ISBN 1-55860-467-7, 2001.

[11].- E. Mateos, Diseño de un regulador de voltaje controlado borrosamente, Centro de Instrumentos UNAM, 1997.

[12].-Muñoz M, A. Pardo G, nuevos diseños de controladores por lógica fuzzy, rev. Colombiana de computación, volumen 2, pág 35-42

[13].-M. Olivares y R.Rojas, Modelado y control difuso, UTFSM, 2002.

[14].-R. Palma, Flujo de potencia óptimo con programación cuadrática secuencial, UCH, Fondecyt 1000966, 2000

[15].- M. Rachid y otros Supervisión difusa de un regulador pid de velocidad, , Universidad Politécnica de Cataluña, 1996

[16].-M. Sánchez P, L. Baharnaca F., Realización de controladores lógicos difusos para el control de procesos en tiempo real, especificación y diseño.

[17].-R. Toxqui, Redes neuronales difusas dinámicas para identificación y control adaptable,2002.

t-SUBNORMAS DISCRETAS SUAVES

M. Mas
Dpt. de Matemàtiques i Infor.
Universitat de les Illes Balears
07122 Palma de Mallorca
dmimmg0@uib.es

M. Monserrat
Dpt. de Matemàtiques i Infor.
Universitat de les Illes Balears
07122 Palma de Mallorca
dmimma0@uib.es

J. Torrens
Dpt. de Matemàtiques i Infor.
Universitat de les Illes Balears
07122 Palma de Mallorca
dmijts0@uib.es

Resumen

En este trabajo se estudian las t-subnormas suaves en una cadena finita, obteniendo una clasificación de las mismas similar al caso [0,1], en forma de sumas ordinales de una t-norma suave y una t-subnorma arquimediana. Respecto a las t-subnormas arquimedianas, distinguimos diversos casos en función de su región cero. En unos las clasificamos todas, mientras que otros caracterizamos algunas familias concretas.

Palabras Clave: t-norma, t-subnorma, negación, dualidad, suavidad, cadena finita.

1 INTRODUCCIÓN

La importancia de las t-normas tanto en la lógica difusa como en el campo de los espacios métricos probabilísticos, es sobradamente conocida. Recientemente, se han introducido en [6], generalizaciones de estos operadores llamadas t-subnormas a las que, en lugar de tener 1 como elemento neutro, se les requiere que tomen valores por debajo del mínimo. Las t-subnormas se han mostrado especialmente útiles en la construcción de t-normas continuas por la izquierda (ver [7]) y pueden ser vistas como un caso particular de semigrupos ordenados o de semigrupos topológicos en el caso de t-subnormas continuas (ver [2]). Sin embargo, su estructura es compleja y, ni siquiera en el caso continuo, han sido todavía caracterizadas. Un estudio, que contiene su estructura como sumas ordinales y diversas formas de generar t-subnormas arquimedianas, puede hallarse en [12].

Por otra parte, cada vez es más frecuente (ver [3], [4], [8], [9], [10], [11]) el estudio de operadores definidos sobre una cadena finita, que es interpretada habitualmente como un conjunto de términos lingüísticos o etiquetas. Trabajar directamente con este tipo de operadores, y no con operadores definidos sobre el intervalo [0,1], permite evitar la interpretación numérica de dichas etiquetas y por tanto los problemas que ello conlleva. En este sentido, muchos autores han trasladado diversas familias de operadores sobre [0,1] al caso de una cadena finita L obteniendo comportamientos más limitados pero similares. En particular, en [11] se caracterizan las t-normas suaves y las t-conormas suaves sobre L, en [8] se caracterizan sobre dicho conjunto L las uninormas y los t-operadores y versiones no conmutativas de estos operadores se caracterizan en [3] y en [9].

En este trabajo queremos estudiar las t-subnormas suaves sobre un conjunto finito y totalmente ordenado L. En particular, vemos que toda t-subnorma propia suave es suma ordinal de una t-norma suave y una t-subnorma arquimediana suave. Esto reduce el problema al estudio de las t-subnormas arquimedianas suaves. Estas últimas las caracterizamos todas en determinados casos mientras que en otros presentamos sendas familias de t-subnormas arquimedianas.

2 PRELIMINARES

Es conocido que toda t-norma (al igual que toda t-subnorma) sobre una cadena finita cualquiera $C = \{x_0 < x_1 < \ldots < x_n\}$ puede ser trasladada a otra sobre $L = \{0, 1, 2, \ldots, n\}$ vía la biyección $\varphi : C \to L$ dada por $\varphi(x_i) = i$, y viceversa. De esta forma el estudio de las t-normas (t-subnormas) sobre una cadena finita de longitud $n + 1$ será el mismo para cualquiera de ellas. Por este motivo trabajaremos con la más sencilla y así, a lo largo de todo el trabajo, L representará la cadena finita

$$L = \{0, 1, 2, \ldots, n\}$$

mientras que $[a, b]$ denotará la cadena finita dada por todos los $x \in L$ tales que $a \leq x \leq b$.

La siguiente definición es original de [4] y es frecuentemente utilizada como un tipo de "continuidad" sobre L.

Definición 1 *Sea F un operador binario sobre L. Diremos que F es un operador suave si verifica la siguiente condición: Si $F(x,y) = z$ entonces, siempre que $x - 1 \geq 0$, se tiene*

$$F(x-1, y) \in \{z-1, z\}$$

y, análogamente, siempre que $y - 1 \geq 0$ se tiene

$$F(x, y-1) \in \{z-1, z\}.$$

Nota: Notemos que de la propia definición se deduce que todo operador suave es necesariamente creciente respecto de cada variable.

Los conceptos de t-norma, t-conorma y negación fuerte pueden extenderse a ámbitos más amplios que el intervalo [0,1]. Por ejemplo, en [1] se dan dichas definiciones sobre un conjunto parcialmente ordenado cualquiera, por lo que podemos considerarlas dadas sobre nuestro conjunto L. En este sentido tenemos los siguientes resultados.

Proposición 1 *La única negación fuerte posible sobre L es la dada por*

$$N(x) = n - x \quad \textit{para todo } x \in L$$

En cambio, el número de t-normas y t-conormas posibles sobre L es enorme, aunque si añadimos la condición de suavidad obtenemos una caracterización completa de las mismas que se puede hallar en [11] bajo el nombre de álgebras dirigidas.

Proposición 2 (Ver [11]). *Existe una única t-norma suave arquimediana sobre L que viene dada por*

$$T(x,y) = \max(0, x + y - n) \tag{1}$$

Además, dado cualquier subconjunto J de L con $0, n \in J$, existe una única t-norma suave sobre L cuyos elementos idempotentes son exactamente los de J. Concretamente, si J es el conjunto

$$J = \{0 = a_0 < a_1 < \ldots < a_{m-1} < a_m = n\}$$

entonces dicha t-norma suave viene dada por:

$$T(x,y) = \max(a_k, x + y - a_{k+1})$$

si existe un idempotente $a_k \in J$ tal que $a_k \leq x, y \leq a_{k+1}$; y viene dada por

$$T(x,y) = \min(x,y)$$

en cualquier otro caso.

Nota: La t-norma dada por (1) es la llamada t-norma de Łukasiewicz por lo que el teorema anterior puede leerse como que toda t-norma suave es suma ordinal de t-normas de Łukasiewicz y mínimo.

Las t-conormas suaves quedan igualmente caracterizadas por dualidad a través de la única negación fuerte N existente sobre L. De la proposición anterior se deduce inmediatamente el siguiente resultado

Nota: La condición de suavidad dada en la definición 1 es equivalente para t-normas a la condición de divisibilidad ($x \leq y$ si y sólo si existe $z \in L$ tal que $T(y,z) = x$) que deriva en la estructura que se conoce con el nombre de BL-cadena (ver [5]).

Proposición 3 (Ver [11]). *Existen exactamente 2^{n-1} t-normas suaves diferentes sobre L.*

3 t-SUBNORMAS SUAVES

Definición 2 *Sea T un operador binario sobre L. Diremos que T es una t-subnorma si es associativo, conmutativo, creciente respecto de cada variable y $T(x,y) \leq \min(x,y)$ para cualesquiera $x, y \in L$.*

Obviamente, toda t-norma sobre L es también t-subnorma pero no al revés. Por ejemplo, la menor t-subnorma posible sobre L es la t-subnorma nula ($T(x,y) = 0$ para todo $x, y \in L$) que claramente no es una t-norma. Notemos que para el caso $n = 1$, ésta y la t-norma mínimo son las únicas t-subnormas sobre $L = \{0, 1\}$, por eso de ahora en adelante supondremos $n \geq 2$.

Una clase importante de t-subnormas que constituye la base para su caracterización es la siguiente.

Definición 3 *Una t-subnorma T se dice que es arquimediana si para cualesquiera $x, y \in L - \{0, n\}$ existe $m \in \mathbb{Z}^+$ tal que $x_T^{(m)} < y$ donde $x_T^{(m)}$ viene definido recursivamente por $x_T^{(1)} = x$ y*

$$x_T^{(m+1)} = T(x, x_T^{(m)}) \qquad \textit{para todo} \qquad m \geq 1.$$

Dada una t-subnorma T, un *elemento idempotente* para T es un $a \in L$ tal que $T(a,a) = a$. Obviamente 0 es siempre un elemento idempotente para cualquier t-subnorma. Por otra parte, tenemos el siguiente resultado.

Proposición 4 *Sea T una t-subnorma sobre L tal que $T(n,n) < n$. Son equivalentes:*

i) T es arquimediana.

ii) 0 es el único idempotente de T.

iii) $T(x,y) < \min(x,y)$ *para cualesquiera* $x, y \in L - \{0\}$.

iv) $T(x,n) < x$ *para todo* $x \in L - \{0\}$.

Demostración
i) $\Rightarrow$ *ii)*. Supongamos $x \neq 0$ un idempotente de T. Entonces, se ve fácilmente por inducción que $x_T^{(m)} = x$ para todo $m > 0$ lo cual contradice la propiedad arquimediana.

ii) $\Rightarrow$ *iii)*. Supongamos $0 < x \leq y$ tales que $T(x,y) = x$ y consideremos el conjunto

$$A = \{z \in L \mid T(x,z) = x\}.$$

A no es vacío y podemos tomar $a = \min A$. Entonces:

$$T(T(a,a),x) = T(a,T(a,x)) = T(a,x) = x$$

por lo que, dada la minimaliddad de a, tenemos $T(a,a) \geq a$ y por tanto $T(a,a) = a$. Por hipótesis $a = 0$, pero entonces

$$T(a,x) = T(0,x) = 0 \neq x$$

llegando a contradicción.

iii) $\Rightarrow$ *iv)*. Obvio.

iv) $\Rightarrow$ *i)*. Consideremos $x, y \in L - \{0\}$, de *iv)* se deduce trivialmente que T no tiene idempotentes distintos de 0 y por tanto $x_T^{(2)} < x$. Si $T(x,x) = 0$ entonces $x_T^{(2)} = 0 < y$ y sino, de forma análoga, se ve que $x_T^{(4)} < x_T^{(2)} < x$ y así sucesivamente. Esto demuestra que existe un $m \in \mathbb{Z}^+$ tal que $x_T^{(m)} < y$ y por tanto T es arquimediana. ∎

Durante el resto del trabajo trataremos exclusivamente con t-subnormas suaves que es el caso equivalente al caso continuo en [0,1], sin otra referencia a este hecho. Por eso, de ahora en adelante por t-subnorma entenderemos t-subnorma suave. En este sentido, tenemos el siguiente resultado idéntico al caso [0,1].

Proposición 5 *Sea T una t-subnorma sobre L. Entonces, T es una t-norma si y sólo si $T(n,n) = n$.*

Demostración
Si T es una t-norma, claramente $T(n,n) = n$. Recíprocamente, si $T(n,n) = n$, como $T(n,0) = 0$, por suavidad tenemos que para cualquier $y \in L$ existe $x \in L$ tal que $T(n,x) = y$. Pero entonces,

$$T(n,y) = T(n,T(n,x)) = T(T(n,n),x) = T(n,x) = y$$

obteniendo que n es neutro para T y por tanto T es una t-norma ∎

Llamaremos t-subnorma *propia* a toda t-subnorma T que no sea t-norma. En vista de la proposición anterior, ésto es equivalente a decir $T(n,n) < n$.

Proposición 6 *Sea T una t-subnorma sobre L y $a \in L$ un elemento idempotente para T con $a \neq 0$. Entonces*

i) $T(x,a) = \min(x,a)$ *para todo* $x \in L$

ii) $T(x,y) = \min(x,y)$ *para cualesquiera* $x, y \in L$ *tales que* $\min(x,y) \leq a \leq \max(x,y)$.

iii) La restricción de T a $[0,a]^2$ es una t-norma suave sobre $[0,a]$.

Demostración
Las demostraciones de *i)* y *ii)* son similares al caso de t-normas suaves, mientras que al ser a un elemento idempotente, la restricción de T a $[0,a]^2$ es una t-norma aplicando la Proposición 5. ∎

Proposición 7 *Sea T una t-subnorma sobre L. Existen un elemento idempotente $a \in L$ de T, una t-norma suave T_0 sobre $[0,a]$ y una t-subnorma arquimediana T_A sobre [a,n] tales que T es suma ordinal de T_0 y T_A. De esta forma, T se escribe como:*

$$T(x,y) = \begin{cases} T_0(x,y) & \text{si } x,y \in [0,a] \\ T_A(x,y) & \text{si } x,y \in [a,n] \\ \min(x,y) & \text{en cualquier otro caso} \end{cases} \quad (2)$$

Recíprocamente, todo operador definido de esta forma es una t-subnorma sobre L.

Demostración
Consideremos a el máximo elemento idempotente de T. Entonces, claramente la restricción de T a $[a,n]^2$ es una t-subnorma arquimediana sobre $[a,n]$ mientras que, al ser a idempotente aplicando la proposición anterior, la restricción de T a $[0,a]^2$ es una t-norma suave sobre $[0,a]$. Por último, de nuevo la proposición anterior nos asegura que la t-subnorma T vale el mínimo en los demás casos. El recíproco es una mera comprobación. ∎

Nota: Observemos que en la proposición anterior, el caso $a = 0$ correspondería a las t-subnormas arquimedianas, mientras que el caso $a = n$ correspondería a las t-normas suaves.

En la figura 1 se puede ver la estructura general de las t-subnormas dada en la Proposición 7.

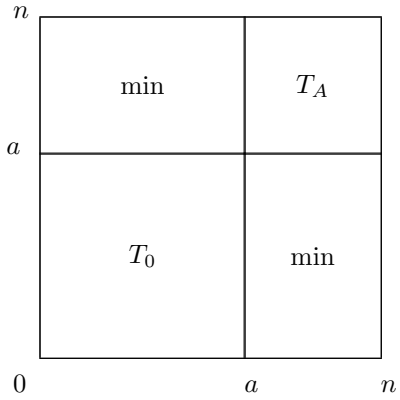


Figura 1: Estructura de una t-subnorma (suave).

4 t-SUBNORMAS SUAVES ARQUIMEDIANAS

En esta sección supondremos todas las t-subnormas no sólo suaves sino también propias. El objetivo ahora es estudiar más detenidamente las t-subnormas arquimedianas.

Proposición 8 *Sea T una t-subnorma arquimediana sobre L. Sea $\alpha = \max\{x \in L \mid T(x,n) = 0\}$. Entonces:*

i) $\alpha \geq 1$

ii) $T(n,n) \leq n - \alpha$.

Demostración
De la Proposición 4 se deduce inmediatamente que $T(n,1) = 0$ y por tanto la parte i). Respecto a ii), como $T(n,\alpha) = 0$ por suavidad obtenemos $T(n,\alpha+1) \leq 1$ y recursivamente tenemos en general $T(n,\alpha+k) \leq k$. De esta manera

$$T(n,n) = T(n,\alpha+(n-\alpha)) \leq n-\alpha.$$

■

En función del valor de α con respecto a n podemos clasificar las t-subnormas arquimedianas de forma diferente.

Proposición 9 *Sea T un operador binario sobre L con $T(n,0) = 0$ y sea $\alpha = \max\{x \in L \mid T(x,n) = 0\}$. Entonces, si $\alpha \geq n/2$, T es una t-subnorma arquimediana si y sólo si T es conmutativo y suave.*

Demostración
La implicación directa es por definición mientras que, para el recíproco, notemos que al ser $T(\alpha+1,n) = 1$ obtenemos $T(\alpha+2,n) \leq 2$ y en general

$$T(x,x) \leq T(x,n) \leq x - \alpha$$

para todo $x \geq \alpha$, mientras que $T(x,x) = 0$ para todo $x \leq \alpha$ lo cual demuestra la propiedad arquimediana. Por tanto, basta ver que en este caso la propiedad associativa se verifica siempre. Ésto es cierto ya que los dos términos que aparecen en dicha propiedad son siempre menores o iguales al valor:

$$T(n,T(n,n)) \leq T(n,n-\alpha) = 0$$

donde la última igualdad es debida a que $n - \alpha \leq \alpha$ por ser $\alpha \geq n/2$. Consecuentemente ambos términos en la propiedad asociativa se anulan. ■

Ejemplos de t-subnormas arquimedianas en el caso $\alpha \geq n/2$ pueden verse más adelante en la nota posterior a la Proposición 11.

Proposición 10 *Sea T un operador binario sobre L con $T(n,0) = 0$ y sea $\alpha = \max\{x \in L \mid T(x,n) = 0\}$. Entonces, si $0 < \alpha < n/2$ y $T(n,n) \leq \alpha$, T es una t-subnorma arquimediana si y sólo si T es conmutativo y suave.*

Demostración
Similar a la demostración de la proposición anterior. ■

Nos queda por tanto estudiar solamente el caso de t-subnormas ar quimedianas con $\alpha < n/2$ y $\alpha < T(n,n) \leq n - \alpha$.

Proposición 11 *Sea T un operador binario sobre L con $T(n,0) = 0$ y sea $\alpha = \max\{x \in L \mid T(x,n) = 0\}$. Entonces, si $0 < \alpha < n/2$ y $T(n,n) = n - \alpha$, T es una t-subnorma arquimediana si y sólo si T viene dada por*

$$T(x,y) = \max(0, x+y-n-\alpha) \qquad (3)$$

para cualesquiera $x, y \in L$.

Demostración
Si T viene dada por la ecuación (3) se ve fácilmente que es una t-subnorma arquimediana. Para el recíproco, sólo daremos la línea de la demostración que podemos dividir en varios pasos:

- Primero se ve fácilmente por suavidad que si $T(n,n) = n - \alpha$, entonces

  $$T(n,x) = \max(0, n-\alpha-(n-x))$$

  para todo $x \in L$.

- A continuación se tiene:

  $$T(n-1,n-\alpha) = T(n-1,T(n,n)) =$$

  $$T(T(n-1,n),n) = T(n-\alpha-1,n) = n-2\alpha-1$$

donde la última igualdad se debe a que $\alpha < n/2$ y por lo tanto $n - 2\alpha - 1 \geq 0$. De este resultado se deduce de nuevo por suavidad que

$$T(n-1, n-\alpha+k) = n - 2\alpha + k - 1$$

para todo $k \in [0, \alpha]$.

- Análogamente se ve que

$$T(n-1, n-\alpha-k) = n - 2\alpha - k - 1$$

para todo $k \in [0, n-2\alpha-1]$ y en particular $T(n-1, \alpha+1) = 0$.

- Así, obtenemos que la ecuación (3) es válida para todos los puntos (n, y) y $(n-1, y)$ con $y \in L$. Podemos iterar el razonamiento hasta demostrar que la ecuación es válida para todos los puntos $(x, y) \in L^2$. ∎

Nota: La ecuación (3) da una t-subnorma sobre L incluso cuando $\alpha \geq n/2$, pero entonces no es la única con $T(n, n) = n - \alpha$, como se desprende de la Proposición 9. Por ejemplo, cuando $\alpha \geq n/2$, la expresión:

$$T(x, y) = \max(0, \min(x, y) - \alpha) \qquad (4)$$

para todo $x, y \in L$, es una t-subnorma con $T(n, n) = n - \alpha$.

La estructura de las t-subnormas arquimedianas dadas por las ecuaciones (3) y (4) pueden verse en las figuras 2 y 3 respectivamente.

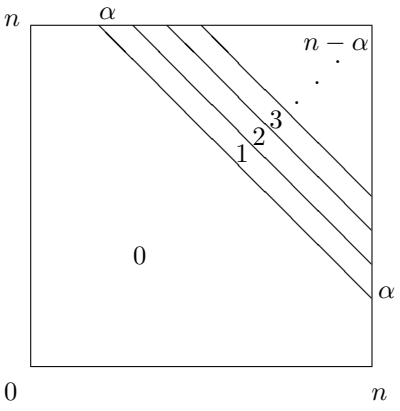


Figura 2: Estructura de la t-subnorma arquimediana dada por la ecuación (3).

Para el caso general, $0 < \alpha < n/2$ y $\alpha < T(n, n) < n - \alpha$, tenemos el siguiente resultado:

Proposición 12 *Sea T una t-subnorma arquimediana sobre L con $T(n, n) = m$ y sea $\alpha = \max\{x \in L \mid T(x, n) = 0\}$. Entonces, si $\alpha < n/2$ y $\alpha < m < n - \alpha$, se verifica:*

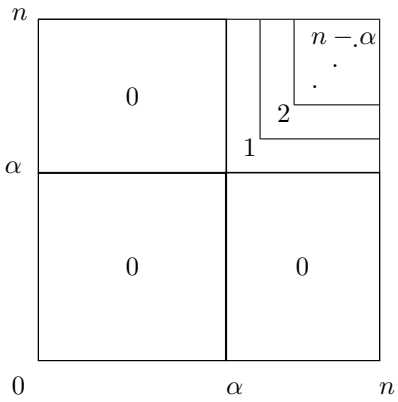


Figura 3: Estructura de la t-subnorma arquimediana dada por la ecuación (4).

i) $T(\alpha + 1, x) = 0$ *para todo* $x \leq m$.

ii) Si $T(n, x)$ es constante en un intervalo $[a, b]$ entonces $T(m, x)$ también lo es en ese mismo intervalo.

Demostración
Para ver *i)* notemos que

$$T(\alpha+1, m) = T(\alpha+1, T(n, n)) =$$

$$T(T(\alpha+1, n), n) = T(1, n) = 0.$$

Por otra parte, un razonamiento análogo demuestra la parte *ii)*. ∎

En vista de la proposición anterior, tenemos la siguiente familia de t-subnormas arquimedianas en el caso $\alpha < T(n, n) < n - \alpha$.

Proposición 13 *Fijado n, para cada α tal que $0 < \alpha < n/2$ y para cada m tal que $\alpha < m < n - \alpha$, el operador binario:*

$$T(x, y) = \begin{cases} 0 & \text{si } x + y - n - \alpha \leq 0 \\ x + y - 2n + m & \text{si } x + y - 2n + m > 1 \\ 1 & \text{en cualquier otro caso} \end{cases}$$

es una t-subnorma arquimediana.

La familia de t-subnormas arquimedianas dadas en la proposición anterior pueden verse en la figura 4.

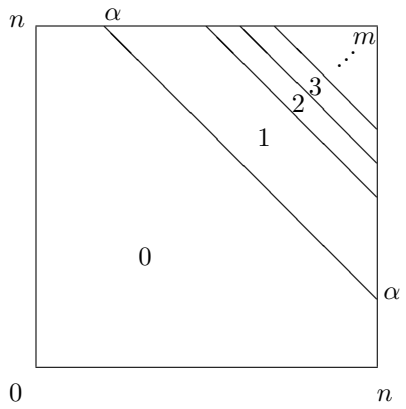


Figura 4: Estructura de las t-subnormas arquimedianas dadas en la Proposición 13.

Nota: Hemos estudiado aquí las t-subnormas suaves sobre L, pero es obvio que resultados análogos pueden establecerse para t-superconormas de forma dual vía la única negación fuerte $N(x) = n - x$ existente sobre L. Recordemos que una t-superconorma sobre L es un operador binario $S : L^2 \to L$ asociativo, conmutativo, creciente respecto de cada variable y tal que $S(x, y) \geq \max(x, y)$ para todos $x, y \in L$.

5 Conclusiones y trabajo futuro

En este trabajo hemos iniciado el estudio de las t-subnormas suaves sobre una cadena finita L. Hemos visto que todas ellas se pueden formar como una suma ordinal de una t-norma suave sobre la subcadena $[0, a]$ y una t-subnorma arquimediana suave sobre $[a, 1]$ donde a es el último elemento idempotente de la t-subnorma. Las t-subnormas arquimedianas han sido caracterizadas en todos los casos excepto cuando $\alpha < T(n, n) < n - \alpha$ siendo α el máximo elemento tal que $T(n, \alpha) = 0$.

Queda por tanto, caracterizar este último caso y a partir de ahí, calcular el número total de t-subnormas arquimedianas sobre $L = \{0, 1, \ldots, n\}$. Notemos que entonces, el número total de t-subnormas sobre L sería, en virtud de las Proposiciones 7 y 3:

$$\sum_{i=1}^{n-1} 2^{i-1} N(n-i) + N(n)$$

donde $N(a)$ indica el número de t-subnormas arquimedianas sobre la cadena finita de $a + 1$ elementos $[0, a]$.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el proyecto del Govern de les Illes Balears número: PDIB-2002GC3-19

Referencias

[1] B. De Baets and R. Mesiar, Triangular norms on product lattices, Fuzzy Sets and Systems, 104 (1999) 61-75.

[2] A.H. Clifford, Naturally ordered commutative semigroups, Amer. J. Math. 76 (1954) 631-646.

[3] J.C. Fodor, Smooth associative operations on finite ordinal scales, IEEE Trans. on Fuzzy Systems, 8 (2000) 791-795.

[4] L. Godo and C. Sierra, A new approach to connective generation in the framework of expert systems using fuzzy logic, Proc. of the XVIIIth International Symposium on Multiple Valued Logic (Palma de Mallorca, Spain), 1988, pp. 157-162.

[5] P. Hájek, Basic fuzzy logic and BL-algebras, Soft Computing, 2 (1998) 124-128.

[6] S. Jenei, Fibred triangular norms, Fuzzy Sets and Systems, 103 (1999) 67-82.

[7] S. Jenei, A note on the ordinal sum theorem and its consequence for the construction of triangular norms, Fuzzy Sets and Systems, 126 (2002) 199-205.

[8] M. Mas, G. Mayor and J. Torrens, $t-$operators and uninorms on a finite totally ordered set, Int. J. of Intelligent Systems 14, N. 9, Special Issue: The Mathematics of Fuzzy Sets (1999) 909-922.

[9] M. Mas, M. Monserrat and J. Torrens, On left and right uninorms on a finite chain, Fuzzy Sets and Systems, article in press, (2004).

[10] M. Mas, J. Torrens, T. Calvo and M. Carbonell, Idempotent operators on a finite chain, Mathware and Soft Computing 6 (2-3) (1999) 235-247.

[11] G. Mayor and J. Torrens, On a class of operators for expert systems, Int. J. of Intelligent Systems 8, N. 7 (1993) 771-778.

[12] A. Mesiarová, Continuous triangular subnorms, Fuzzy Sets and Systems, 142 (2004) 75-83.

Caracterización de Implicaciones Residuadas en $\mathcal{J}[0,1]$

Cristina Alcalde
Dpt. Matemática Aplicada
Escuela Universitaria Politécnica
Universidad del País Vasco
Plaza de Europa 1
20018 San Sebastián, España
mapalvac@sp.ehu.es

Ana Burusco
Dpt. Automática y Computación.
Universidad Pública de Navarra
Campus de Arrosadía
31006 Pamplona, España
burusco@si.unavarra.es

Ramón Fuentes-González
Dpt. Automática y Computación.
Universidad Pública de Navarra
Campus de Arrosadía
31006 Pamplona, España
rfuentes@si.unavarra.es

Resumen

En este trabajo se muestra una nueva caracterización para las implicaciones residuadas entre intervalos que nos permitirá trabajar con ellas de una manera simple y eficaz, puesto que el cálculo de los valores que toma una implicación intervalo-valorada aplicada a dos intervalos se reduce al estudio de una implicación fuzzy aplicada a los extremos de dichos intervalos.

Se ilustra este resultado con algunos ejemplos y para finalizar se demuestran algunas propiedades.

Palabras Clave: Operador de implicación fuzzy, Operador de implicación fuzzy intervalo-valorado, R-implicación fuzzy.

1. Introducción

Para poder abordar el estudio de las R-implicaciones fuzzy definidas entre intervalos, debemos recordar previamente los conceptos de operador de implicación fuzzy y de R-implicación fuzzy. Recordaremos así mismo en esta sección las definiciones de t-norma y de operador de implicación cuando trabajamos sobre intervalos

1.1. Operadores de implicación fuzzy

Una *implicación fuzzy* es una función de la forma:

$$I\colon [0,1]\times[0,1]\longrightarrow[0,1]$$

la cual, dados los valores de verdad a y b de dos proposiciones dadas p y q respectivamente, define el valor de verdad $I(a,b)$ de la proposición $p\Rightarrow q$.

Manejaremos implicaciones Fuzzy que sean una extensión de la implicación clásica $p\Rightarrow q$ en el dominio $\{0,1\}$, esto es, deben obtenerse los valores:

$$\begin{cases} I(0,0)=I(0,1)=I(1,1)=1\\ I(1,0)=0\end{cases}$$

En la literatura sobre el tema aparecen una serie de axiomas que pueden cumplir las implicaciones Fuzzy [5, 7].

i.1. $x\le z\Longrightarrow I(x,y)\ge I(z,y)\quad\forall y\in[0,1]$

i.2. $y\le t\Longrightarrow I(x,y)\le I(x,t)\quad\forall x\in[0,1]$

i.3. $I(0,y)=1\quad\forall y\in[0,1]$

i.4. $I(x,1)=1\quad\forall x\in[0,1]$

i.5. $I(1,x)=x\quad\forall x\in[0,1]$

i.6. $I(x,x)=1\quad\forall x\in[0,1]$

i.7. $I(x,I(y,z))=I(y,I(x,z))\quad\forall x,y,z\in[0,1]$

i.8. $I(x,y)=1\quad\Longleftrightarrow x\le y$

i.9. $I(x,y)\ge y\quad\forall x,y\in[0,1]$

i.10. $I(x,y)=I(n(y),n(x))\quad\forall x,y\in[0,1]$ siendo n una negación fuerte

i.11. *Continuidad* de la función I en $[0,1]\times[0,1]$

Entre los distintos tipos de implicaciones que aparecen en la bibliografía [5, 7], centraremos nuestro estudio en las R-implicaciones (implicaciones definidas a través de una t-norma).

Definición 1. Se llama *R-implicación (o implicación residuada) asociada a la t-norma* T [7, 5] a la implicación definida de la forma:

$$I(a,b)=\sup\{x\in[0,1]/T(a,x)\le b\}\qquad\forall a,b\in[0,1]$$

siendo T una t-norma.

Se comprueba que todas las implicaciones residuadas verifican las propiedades de las implicaciones i.1, i.2, i.3, i.4, i.5 e i.6.

1.2. Operaciones entre intervalos

Sea $\mathcal{J}[0,1]$ el conjunto de intervalos cerrados en $[0,1]$, esto es $\mathcal{J}[0,1] = \{[a,b]/a,b \in [0,1]\}$.

Consideraremos en el conjunto de intervalos $\mathcal{J}[0,1]$ la relación de orden definida por

$$[a,b] \leq [c,d] \iff a \leq c \text{ y } b \leq d$$

En la literatura se definen, entre otros, los operadores en $\mathcal{J}[0,1]$ siguientes:

1.2.1. t-normas en $\mathcal{J}[0,1]$

Definición 2. [4] Una *t-norma* definida en el conjunto de intervalos $\mathcal{J}[0,1]$ es una función de la forma

$$\mathcal{T}\colon \mathcal{J}[0,1] \times \mathcal{J}[0,1] \longrightarrow \mathcal{J}[0,1]$$

que cumple las siguientes propiedades:

t.1. *Condición de frontera:*

$$\mathcal{T}([a,b],[1,1]) = [a,b] \quad \forall [a,b] \in \mathcal{J}[0,1]$$

t.2. *Conmutatividad:*

$$\mathcal{T}([a,b],[c,d]) = \mathcal{T}([c,d],[a,b]) \quad \forall [a,b],[c,d] \in \mathcal{J}[0,1]$$

t.3. *Monotonía:*

$$[c,d] \leq [e,f] \Longrightarrow \mathcal{T}([a,b],[c,d]) \leq \mathcal{T}([a,b],[e,f]) \quad \forall [a,b] \in \mathcal{J}[0,1]$$

t.4. *Asociatividad:*

$$\mathcal{T}([a,b],\mathcal{T}([c,d],[e,f])) = \mathcal{T}(\mathcal{T}([a,b],[c,d]),[e,f]) \quad \forall [a,b],[c,d],[e,f] \in \mathcal{J}[0,1]$$

Sea T una t-norma definida en el intervalo $[0,1]$.

Definición 3. Se define en [6] la *t-norma asociada* $\boldsymbol{T}$ en el conjunto de intervalos $\mathcal{J}[0,1]$ de la forma:

$$\boldsymbol{T}\colon \mathcal{J}[0,1] \times \mathcal{J}[0,1] \longrightarrow \mathcal{J}[0,1]$$

donde

$$\boldsymbol{T}([a,b],[c,d]) = [T(a,c), T(b,d)]$$

Es inmediato comprobar que la operación así definida verifica las propiedades de las t-normas en $\mathcal{J}[0,1]$. Además es una extensión de la t-norma en $[0,1]$, ya que si se representan los números reales mediante intervalos del tipo $[a,a]$,

$$\boldsymbol{T}([a,a],[b,b]) = [T(a,b), T(a,b)]$$

el resultado también será un número real.

1.2.2. Implicaciones en $\mathcal{J}[0,1]$

Cornelis y Deschrijver [2] definen el operador de implicación fuzzy intervalo-valorado como una extensión del operador implicación fuzzy al conjunto de intervalos $[2,1]$: Se llama *implicación fuzzy intervalo-valorada* a toda función $\boldsymbol{I}$ definida en el conjunto de intervalos $\mathcal{J}[0,1]$:

$$\boldsymbol{I}\colon \mathcal{J}[0,1] \times \mathcal{J}[0,1] \longrightarrow \mathcal{J}[0,1]$$

que cumple las siguientes condiciones iniciales:

- $\boldsymbol{I}([0,0],[0,0]) = [1,1]$
- $\boldsymbol{I}([0,0],[1,1]) = [1,1]$
- $\boldsymbol{I}([1,1],[1,1]) = [1,1]$
- $\boldsymbol{I}([1,1],[0,0]) = [0,0]$

Además, como aparece en la literatura [2, 1, 6], en general exigiremos a toda implicación fuzzy entre intervalos $\boldsymbol{I}$ que verifique las siguientes propiedades:

I.0 Sea una extensión de la implicación fuzzy, esto es:

$$\text{Si} \quad \boldsymbol{I}([a,a],[b,b]) = [x,y], \text{ entonces} \quad x = y$$

I.1 Si $[a,b] \leq [a_1,b_1]$ entonces $\boldsymbol{I}([a,b],[c,d]) \geq \boldsymbol{I}([a_1,b_1],[c,d]) \quad \forall [c,d] \in \mathcal{J}[0,1]$

I.2 Si $[c,d] \leq [c_1,d_1]$ entonces $\boldsymbol{I}([a,b],[c,d]) \leq \boldsymbol{I}([a,b],[c_1,d_1]) \quad \forall [a,b] \in \mathcal{J}[0,1]$

Las implicaciones fuzzy intervalo-valoradas pueden verificar además de las anteriores otra serie de propiedades que los autores dividen en dos bloques [1, 6]:

• Extensión de las propiedades de las implicaciones fuzzy al caso intervalo-valorado:

I.3 $\boldsymbol{I}([0,0],[a,b]) = [1,1] \quad \forall [a,b] \in \mathcal{J}[0,1]$

I.4 $\boldsymbol{I}([a,b],[1,1]) = [1,1] \quad \forall [a,b] \in \mathcal{J}[0,1]$

I.5 $\boldsymbol{I}([1,1],[a,b]) = [a,b] \quad \forall [a,b] \in \mathcal{J}[0,1]$

I.6 $\boldsymbol{I}([a,b],[a,b]) = [1,1] \quad \forall [a,b] \in \mathcal{J}[0,1]$

I.7 $\boldsymbol{I}([a,b],\boldsymbol{I}([c,d],[e,f])) = \boldsymbol{I}([c,d],\boldsymbol{I}([a,b],[e,f]))$ $\forall [a,b],[c,d],[e,f] \in \mathcal{J}[0,1]$

I.8 $\boldsymbol{I}([a,b],[c,d]) = [1,1]$ si y sólo si $[a,b] \leq [c,d]$ $\forall [a,b],[c,d] \in \mathcal{J}[0,1]$

I.9 $\boldsymbol{I}([a,b],[c,d]) \geq [c,d] \quad \forall [a,b],[c,d] \in \mathcal{J}[0,1]$

I.10 $\boldsymbol{I}([a,b],[c,d]) = \boldsymbol{I}([n(d),n(c)],[n(b),n(a)])$ $\forall [a,b],[c,d] \in \mathcal{J}[0,1]$, siendo n una negación fuerte.

• Propiedades propias de los conjuntos fuzzy intervalo-valorados:

Denotamos con $W([a,b])$ a la amplitud del intervalo $[a,b]$, esto es, $W([a,b]) = b - a$.

I.11 $W(\boldsymbol{I}([a,b],[c,d])) \leq \bigvee(n(a),n(b),n(c),n(d)) = \bigvee(n(a),n(c))$ siendo n una negación fuerte.

I.12 Si $[a,b] = [c,d]$ entonces $W(\boldsymbol{I}([a,b],[c,d])) = W([a,b])$

I.13 Si $W[a,b] = W[c,d]$ entonces $W(\boldsymbol{I}([a,b],[c,d])) = W([a,b])$

A partir de este momento centraremos nuestro estudio en las implicaciones residuadas.

2. Implicaciones Residuadas en $\mathcal{J}[0,1]$

Definición 4. Dada una t-norma T en $[0,1]$, definimos la *R-implicación* asociada a $\mathbf{T}$ en $\mathcal{J}[0,1]$ como:

$$\boldsymbol{I_{R_T}}([a,b],[c,d]) = \sup\{[x,y]/\boldsymbol{T}([a,b],[x,y]) \leq [c,d]\}$$

donde $\mathbf{T}$ es la extensión de T a $\mathcal{J}[0,1]$ definida en el apartado 1.2.1.

En el siguiente teorema vemos que toda implicación residuada entre intervalos se puede caracterizar a partir de una R-implicación fuzzy.

Teorema 1. *Toda implicación residuada entre intervalos $\boldsymbol{I_{R_T}}$ asociada a la t-norma $\boldsymbol{T}$ tiene la forma:*

$$\boldsymbol{I_{R_T}}([a,b],[c,d]) = [I_{R_T}(a,c) \wedge I_{R_T}(b,d), I_{R_T}(b,d)] \tag{1}$$

donde I_{R_T} es la implicación residuada asociada a la t-norma T en $[0,1]$ que genera $\boldsymbol{T}$ en $\mathcal{J}[0,1]$.

Demostración.

En general, toda implicación residuada entre intervalos tiene la forma:

$$\begin{aligned}\boldsymbol{I_{R_T}}([a,b],[c,d]) &= \sup\{[x,y]/\boldsymbol{T}([a,b],[x,y]) \leq [c,d]\} \\ &= \sup\{[x,y]/T(a,x) \leq c \quad \& \quad T(b,y) \leq d\} \quad (2)\end{aligned}$$

Vamos a comprobar que las expresiones (1) y (2) son iguales.

Llamamos M al conjunto:

$$M = \{[x,y]/ \quad T(a,x) \leq c \quad \& \quad T(b,y) \leq d\}$$

Si recordamos la definición de implicación residuada con valores en $[0,1]$:

$$I_{R_T}(a,c) = \sup\{x/T(a,x) \leq c\}$$

se obtiene que

$$\begin{aligned}\forall [x,y] \in M \quad T(a,x) \leq c \quad \& \quad T(b,y) \leq d \Longrightarrow \\ I_{R_T}(a,c) \geq x \quad \& \quad I_{R_T}(b,d) \geq y \geq x\end{aligned}$$

Por lo tanto, tomando el mínimo:

$$[I_{R_T}(a,c) \wedge I_{R_T}(b,d), I_{R_T}(b,d)] \geq [x,y] \quad \forall [x,y] \in M$$

Con lo que queda demostrado que el segundo miembro de la igualdad (1) es una cota superior de M:

$$[I_{R_T}(a,c) \wedge I_{R_T}(b,d), I_{R_T}(b,d)] \geq \boldsymbol{I_{R_T}}([a,b],[c,d]) \tag{3}$$

Veamos que es la menor de las cotas superiores de M.

Supongamos que no lo es. Sea $[\alpha,\beta]$ otra cota superior de M que no es mayor que la anterior, entonces:

$$[\alpha,\beta] \geq [x,y] \quad \forall [x,y] \in M \tag{4}$$

$$[I_{R_T}(a,c) \wedge I_{R_T}(b,d), I_{R_T}(b,d)] \not\leq [\alpha,\beta]$$

se deduce entonces uno de estos tres casos:

a)
$$\left.\begin{array}{lcl}\alpha & < & I_{R_T}(a,c) \wedge I_{R_T}(b,d) \\ \beta & < & I_{R_T}(b,d)\end{array}\right\}$$

b)
$$\left.\begin{array}{lcl}\alpha & < & I_{R_T}(a,c) \wedge I_{R_T}(b,d) \\ \beta & \geq & I_{R_T}(b,d)\end{array}\right\}$$

c)
$$\left.\begin{array}{lcl}\alpha & \geq & I_{R_T}(a,c) \wedge I_{R_T}(b,d) \\ \beta & < & I_{R_T}(b,d)\end{array}\right\}$$

Veamos qué ocurre en cada caso:

a) Si $\left.\begin{array}{lcl}\alpha & < & I_{R_T}(a,c) \wedge I_{R_T}(b,d) \\ \beta & < & I_{R_T}(b,d)\end{array}\right\}$ entonces

$$\left.\begin{array}{lcl}\alpha & < & I_{R_T}(a,c) \\ \alpha & < & I_{R_T}(b,d) \\ \beta & < & I_{R_T}(b,d)\end{array}\right\}$$

y aplicando la definición de implicación residuada

$$\left.\begin{array}{l}\alpha < \sup\{x/T(a,x) \le c\}\\ \alpha < \sup\{x/T(b,x) \le d\}\\ \beta < \sup\{y/T(b,y) \le d\}\end{array}\right\}$$

de donde se deduce que

$$\left.\begin{array}{l}\exists z_1 > \alpha/T(a,z_1) \le c\\ \exists z_2 > \alpha/T(b,z_2) \le d\\ \exists z_3 > \beta \ge \alpha/T(b,z_3) \le d\end{array}\right\}$$

Sean $r = z_1 \wedge z_2$ y $s = z_2 \vee z_3$, se verifica entonces que

$$r \le z_2 \le s \Rightarrow r \le s$$

Tenemos que, por una parte

- Si $z_2 < z_3$ entonces $s = z_3$ y por lo tanto

$$T(b,s) = T(b,z_3) \le d$$

- Si $z_2 \ge z_3$ entonces $s = z_2$ y en este caso

$$T(b,s) = T(b,z_2) \le d$$

y por otra:

- Si $z_1 < z_2$ entonces $r = z_1$ y se verifica

$$T(a,r) = T(a,z_1) \le c$$

- Si $z_1 \ge z_2$ entonces $r = z_2$ y por la monotonía de la t-norma,

$$T(a,r) = T(a,z_2) \le T(a,z_1) \le c$$

Por lo tanto, $[r,s] \in M$ y verifica $[r,s] \not\leq [\alpha,\beta]$, lo cual contradice (4)

b) Si $\left.\begin{array}{lll}\alpha & < & I_{R_T}(a,c) \wedge I_{R_T}(b,d)\\ \beta & \ge & I_{R_T}(b,d)\end{array}\right\}$

Tenemos, entonces, que:

$$\alpha < I_{R_T}(a,c) \wedge I_{R_T}(b,d) \Longrightarrow \left.\begin{array}{lll}\alpha & < & I_{R_T}(a,c)\\ \alpha & < & I_{R_T}(b,d)\end{array}\right\}$$

y por la definición de R-implicación

$$\left.\begin{array}{l}\alpha < \sup\{x/T(a,x) \le c\}\\ \alpha < \sup\{x/T(b,x) \le d\}\end{array}\right\}$$

y por lo tanto

$$\left.\begin{array}{l}\exists z_1 > \alpha/T(a,z_1) \le c\\ \exists z_2 > \alpha/T(b,z_2) \le d\end{array}\right\}$$

Sea $r = z_1 \wedge z_2$. Se verifica que:

- Si $z_1 < z_2$, entonces $r = z_1$ y por lo tanto
$\left\{\begin{array}{l}T(a,r) = T(a,z_1) \le c\\ T(b,r) = T(b,z_1) \le T(b,z_2) \le d\end{array}\right.$
- Si $z_1 \ge z_2$, entonces $r = z_2$ luego
$\left\{\begin{array}{l}T(b,r) = T(b,z_2) \le d\\ T(a,r) = T(a,z_2) \le T(a,z_1) \le c\end{array}\right.$

En consecuencia, $[r,r] \in M$ y verifica $[r,r] \not\leq [\alpha,\beta]$, lo cual contradice (4)

c) Si $\left.\begin{array}{lll}\alpha & \ge & I_{R_T}(a,c) \wedge I_{R_T}(b,d)\\ \beta & < & I_{R_T}(b,d)\end{array}\right\}$

Tenemos en primer lugar que $\alpha \le \beta$ y $\beta < I_{R_T}(b,d)$, por lo tanto $\alpha < I_{R_T}(b,d)$.

Entonces:

$$\left.\begin{array}{lll}\alpha & < & I_{R_T}(b,d)\\ \alpha & \ge & I_{R_T}(a,c) \wedge I_{R_T}(b,d)\end{array}\right\} \Longrightarrow \alpha \ge I_{R_T}(a,c)$$

y como I_{R_T} es una implicación residuada,

$$\alpha \ge x \quad \forall x/T(a,x) \le c$$

Fijamos cualquier punto r tal que $T(a,r) \le c$, entonces $\alpha \ge r$.

Por otra parte:

$$\beta < I_{R_T}(b,d) \Longrightarrow \beta < \sup\{y/T(b,y) \le d\} \Longrightarrow$$

$$\exists z > \beta \ge \alpha/T(b,z) \le d$$

Se obtiene entonces que $r \le \alpha \le \beta < z$ y además
$\left\{\begin{array}{l}T(a,r) \le c\\ T(b,z) \le d\end{array}\right.$

Por lo tanto, $[r,z] \in M$ y verifica $[r,z] \not\leq [\alpha,\beta]$, lo cual contradice (4)

Tenemos demostrado que

$$[I_{R_T}(a,c) \wedge I_{R_T}(b,d), I_{R_T}(b,d)] = \\ = \sup\{[x,y]/ \quad T([a,b],[x,y]) \le [c,d]\}$$

esto es:

$$[I_{R_T}(a,c) \wedge I_{R_T}(b,d), I_{R_T}(b,d)] = \boldsymbol{I_{R_T}}([a,b],[c,d]) \quad \square$$

Ilustraremos el teorema anterior con algunos ejemplos notables.

Ejemplos.

1. Si utilizamos la t-norma del mínimo $T(x,y) = \text{mín}(x,y)$, se obtiene la implicación $\boldsymbol{I_{R_B}}$, extensión de la implicación de *Brouwer-Gödel*:

   $$\boldsymbol{I_{R_B}}([a,b],[c,d]) = [I_{R_B}(a,c) \wedge I_{R_B}(b,d), I_{R_B}(b,d)]$$

   siendo:

   $$I_{R_B}(a,b) = \sup\{x / \text{mín}(a,x) \leq b\} = \begin{cases} 1 & \text{si } a \leq b \\ b & \text{si } a > b \end{cases}$$

   de donde se obtiene:

   $$\boldsymbol{I_{R_B}}([a,b],[c,d]) = \begin{cases} [c,d] & \text{si } a > c \text{ y } b > d \\ [c,1] & \text{si } a > c \text{ y } b \leq d \\ [1,1] & \text{si } a \leq c \text{ y } b \leq d \\ [d,d] & \text{si } a \leq c \text{ y } b > d \end{cases}$$

2. Si utilizamos la t-norma $T(x,y) = \text{máx}(0, x+y-1)$, se obtiene la implicación $\boldsymbol{I_{R_L}}$, extensión de la implicación de *Lukasiewicz*:

   $$\boldsymbol{I_{R_L}}([a,b],[c,d]) = [I_{R_L}(a,c) \wedge I_{R_L}(b,d), I_{R_L}(b,d)]$$

   siendo:

   $$I_{R_L}(a,b) = \text{mín}(1, 1-a+b)$$

   con lo cual,

   $$\boldsymbol{I_{R_L}}([a,b],[c,d]) = \begin{cases} [(1-a+c) \wedge (1-b+d) & , (1-b+d)] \\ & \text{si } a > c \text{ y } b > d \\ [1-a+c, 1] & \text{si } a > c \text{ y } b \leq d \\ [1,1] & \text{si } a \leq c \text{ y } b \leq d \\ [1-b+d, 1-b+d] & \text{si } a \leq c \text{ y } b > d \end{cases}$$

Podemos ver en [3] una versión intuicionista de estas mismas implicaciones.

La R-implicación $\boldsymbol{I_{R_T}}$ verifica las propiedades de las implicaciones fuzzy intervalo-valoradas que son extensión de las propiedades que verifica la R-implicación I_{R_T}, es decir, verifica las propiedades: *I.1, I.2, I.3, I.4, I.5* e *I.6*, como se muestra en la siguiente

Proposición 1.

I.1 *Si* $[a,b] \leq [a_1,b_1]$ *entonces* $\boldsymbol{I_{R_T}}([a,b],[c,d]) \geq \boldsymbol{I_{R_T}}([a_1,b_1],[c,d]) \quad \forall [c,d] \in \mathcal{J}[0,1]$.

I.2 *Si* $[c,d] \leq [c_1,d_1]$ *entonces* $\boldsymbol{I_{R_T}}([a,b],[c,d]) \leq \boldsymbol{I_{R_T}}([a,b],[c_1,d_1]) \quad \forall [a,b] \in \mathcal{J}[0,1]$.

I.3 $\boldsymbol{I_{R_T}}([0,0],[a,b]) = [1,1] \quad \forall [a,b] \in \mathcal{J}[0,1]$.

I.4 $\boldsymbol{I_{R_T}}([a,b],[1,1]) = [1,1] \quad \forall [a,b] \in \mathcal{J}[0,1]$.

I.5 $\boldsymbol{I_{R_T}}([1,1],[a,b]) = [a,b] \quad \forall [a,b] \in \mathcal{J}[0,1]$.

I.6 $\boldsymbol{I_{R_T}}([a,b],[a,b]) = [1,1] \quad \forall [a,b] \in \mathcal{J}[0,1]$.

Demostración.

I.1 $[a,b] \leq [a_1,b_1] \Longleftrightarrow \begin{cases} a \leq a_1 \\ b \leq b_1 \end{cases}$

Y por la monotonía de la implicación I_{R_T} se obtiene

$$\begin{cases} I_{R_T}(a,c) \geq I_{R_T}(a_1,c) \\ I_{R_T}(b,d) \geq I_{R_T}(b_1,d) \end{cases} \Longrightarrow$$

$$[I_{R_T}(a,c) \wedge I_{R_T}(b,d), I_{R_T}(b,d)] \geq [I_{R_T}(a_1,c) \wedge I_{R_T}(b_1,d), I_{R_T}(b_1,d)] \Longrightarrow$$

$$\Longrightarrow \boldsymbol{I_{R_T}}([a,b],[c,d]) \geq \boldsymbol{I_{R_T}}([a_1,b_1],[c,d])$$

I.2 La demostración es similar a la anterior teniendo en cuenta que la implicación I_{R_T} es creciente en el segundo argumento.

I.3 La demostración es obvia teniendo en cuenta que $I_{R_T}(0,y) = 1 \quad \forall y \in [0,1]$.

I.4 Es evidente puesto que $I_{R_T}(x,1) = 1 \quad \forall x \in [0,1]$.

I.5 A partir de la igualdad $I_{R_T}(1,y) = y \quad \forall y \in [0,1]$, se obtiene que

$$\boldsymbol{I_{R_T}}([1,1],[a,b]) = [I_{R_T}(1,a) \wedge I_{R_T}(1,b), I_{R_T}(1,b)] = [a \wedge b, b] = [a,b]$$

I.6 La demostración es trivial teniendo en cuenta que $I_{R_T}(x,x) = 1 \quad \forall x \in [0,1]$.

3. Líneas futuras

Este artículo es parte de un trabajo más amplio sobre el estudio de la información obtenida a partir de contextos L-fuzzy intervalo-valorados. Estos contextos se representan mediante relaciones fuzzy, cuyos valores son intervalos, a partir de las cuales extraemos información relevante mediante la teoría de conceptos L-fuzzy desarrollada por nosotros, y utilizando implicaciones entre intervalos. Para ello, necesitamos utilizar algunas operaciones entre intervalos como las definidas aquí.

Agradecimientos
Los autores quisieran agradecer a los referees sus valiosos comentarios.

Referencias

[1] H.Bustince, E.Barrenechea, V.Mohedano. Operador de Implicación Intervalo-Valorado Fuzzy. Una expresión y primeras propiedades. *ESTYLF 2002.* León (2002), 309-314.

[2] C.Cornelis, G.Deschrijver. The Compositional Rule of Inference in an Intuitionistic Fuzzy Logic Setting. *Proccedings of the Sixth ESSLLI Students Session.* Kristina Striegnitz (editor), 2001.

[3] C.Cornelis, G.Deschrijver, E.E.Kerre. Implication in intuitionistic fuzzy and interval-valued fuzzy set theory: construction, classification, application. *International Journal of Approximate Reasoning.* 35 No.1 (2004), 55-95.

[4] G.Deschrijver, C.Cornelis, E.E.Kerre. On the Representation of Intuitionistic Fuzzy t-Norms and t-Conorms. *IEEE Transactions on Fuzzy Systems.* 12 No.1 (2004), 45-61. Kristina Striegnitz (editor), 2001.

[5] J.Fodor, M.Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support.* Theory and Decision Library (Kluwer Academic Publishers), Dordrecht/Boston/London, 1994.

[6] S.Jenei. A more efficient method for defining fuzzy connectives. *Fuzzy sets and Systems.* 90 (1997), 25-35.

[7] G.J.Klir, B.Yuan. *Fuzzy sets and fuzzy logic. Theory and Applications.* Prentice Hall PTR, New Jersey, 1995.

GENERADORES ADITIVOS DE NORMAS TRIANGULARES DISCRETAS

Gaspar Mayor
Dep. de Matemáticas e Informática
Universitat de les Illes Balears
gmayor@uib.es

Jaume Monreal
Dep. de Matemáticas e Informática
Universitat de les Illes Balears
jaumemonreal@mallorcaweb.net

Resumen

En este trabajo se define el concepto de generador aditivo de una t-norma discreta y se demuestra la existencia de generadores aditivos enteros para aquellas t-normas que admiten generación aditiva. La existencia de generadores aditivos enteros minimales hace posible una representación reducida para una t-norma. Resultado a destacar en todo el trabajo es el que permite obtener un generador aditivo de una suma ordinal de t-normas a partir de generadores aditivos de cada t-norma. En particular se demuestra que toda t-norma divisible admite generadores aditivos, y, en este sentido, se da un procedimiento constructivo para hallar un generador mínimo de una t-norma divisible.

Palabras Clave: cadena finita, operación binaria, t-norma discreta, t-norma divisible, pseudo-inversa de una función, representación aditiva, generador aditivo, suma ordinal, minimal.

1 INTRODUCCIÓN

Las normas triangulares (t-normas) son una herramienta importante a la hora de interpretar la conjunción en lógica borrosa y consecuentemente la intersección de conjuntos borrosos. Las t-normas tal como se conocen hoy en día fueron introducidas por primera vez en el contexto de los espacios métricos probabilísticos ([5]) pero actualmente también juegan un papel importante en toma de decisiones, en estadística, en teorías de medidas no aditivas, etc. Bajo un punto de vista algebraico, una t-norma T es una operación binaria sobre el intervalo real $[0,1]$ de forma que $([0,1],T,\leq)$ es un semigrupo abeliano totalmente ordenado con elemento neutro 1.

Motivado por el hecho que en la mayoría de situaciones prácticas es necesario discretizar el intervalo unidad, se hace necesaria la introducción y estudio de t-normas definidas no sobre $[0,1]$ sino sobre una cadena finita, digamos $\{0,1,...,n\}$, manteniendo los mismos axiomas que definen las t-normas ordinarias.

Información completa sobre t-normas definidas en $[0,1]$ puede encontrarse en [1], en esta monografía se dedica también parte de un capítulo a las t-normas discretas. Un estudio exhaustivo sobre t-normas discretas podrá consultarse próximamente en [4].

El trabajo se presenta a continuación en dos secciones: la primera se dedica a preliminares y en la segunda se exponen los conceptos y resultados destacables del trabajo. Después de una breve lista de referencias se sitúan dos anexos que pueden resultar ilustrativos de parte del trabajo llevado a cabo en el caso de una cadena de 5 elementos ($n=4$) y en el de una de 17 elementos ($n=16$).

2 PRELIMINARES

Sea $L = \{0,1,2,...,n\}$, $n \geq 1$. A continuación se define el concepto de norma triangular discreta.

Definición 1 *Una norma triangular (t-norma para abreviar) sobre L es una operación binaria T sobre L, es decir $T : L \times L \to L$, tal que $\forall x,y,z \in L$ se satisfacen los axiomas de conmutatividad, asociatividad, monotonía y elemento neutro n. Es decir:*

$(T1)$ $T(x,y) = T(y,x)$
$(T2)$ $T(x,T(y,z)) = T(T(x,y),z)$
$(T3)$ $T(x,y) \leq T(x,z)$ *siempre que* $y \leq z$
$(T4)$ $T(x,n) = x$

De la definición se deduce inmediatamente que cualquier t-norma T verifica $T(x,y) \leq min(x,y)$

$\forall x, y \in L$. En particular, $T(x, 0) = 0\ \forall x \in L$.

Ejemplos 1: Las siguientes son las tres t-normas básicas, llamadas respectivamente t-norma mínimo, t-norma de Łukasiewicz y t-norma drástica:

1. $T_M(x, y) = min(x, y)$
2. $T_{\text{Ł}}(x, y) = max(x + y - n, 0)$
3. $T_D(x, y) = \begin{cases} 0 & si\ max(x, y) < n \\ min(x, y) & si\ max(x, y) = n \end{cases}$

Observar que para cualquier t-norma T se verifica $T_D \leq T \leq T_M$ (orden puntual). Por otra parte, T_M es la única t-norma que verifica $T(x, x) = x$ $\forall x \in L$; asimismo, T_D es la única t-norma que satisface $T(x, x) = 0\ \forall x \in L,\ x \neq n$.

El equivalente a la continuidad de una t-norma sobre el intervalo [0,1] en el marco de las t-normas discretas es la *divisibilidad*.

Definición 2 *Sea T una t-norma sobre L. Diremos que T es divisible si $\forall x, y \in L$ tal que $x \leq y$ existe $z \in L$ de forma que $x = T(y, z)$.*

La divisibilidad de una t-norma puede caracterizarse de varias formas, una de ellas es a través de la condición de Lipschitz: $T(x, y) - T(z, y) \leq x - z$ siempre que $x \geq z$. De ahí que otros autores refieran esta propiedad como una propiedad de *suavidad*.

De los ejemplos anteriores, T_M y $T_{\text{Ł}}$ son t-normas divisibles. T_D no lo es (excepto si $n \leq 2$). Por otra parte, $T_{\text{Ł}}$ es la única t-norma divisible que verifica $T(x, n - x) = 0\ \forall x \in L$.

Un método que permite construir una t-norma a partir de otras t-normas es el conocido como suma ordinal. Damos a continuación la definición para dos t-normas, pero puede extenderse sin ningún problema a cualquier colección finita de ellas.

Definición 3 *Sea T_1 una t-norma sobre $L_1 = \{0, 1, ..., m\}$ y sea T_2 una t-norma sobre $L_2 = \{0, 1, ..., n\}$, con $m, n \geq 1$. La suma ordinal de T_1 y T_2 es la t-norma sobre $L = \{0, 1, ..., m, m + 1, ..., m + n\}$ definida por*

$T(x, y) =$

$$\begin{cases} T_1(x, y) & si\ (x, y) \in L_1^2 \\ m + T_2(x - m, y - m) & si\ (x, y) \in \{m, ..., m + n\}^2 \\ min(x, y) & en\ otro\ caso. \end{cases}$$

Es importante observar que mediante la suma ordinal la asociatividad de T_1 y T_2 se transmite a la operación T, así como las restantes propiedades que permiten afirmar que T es una t-norma. Digamos también que

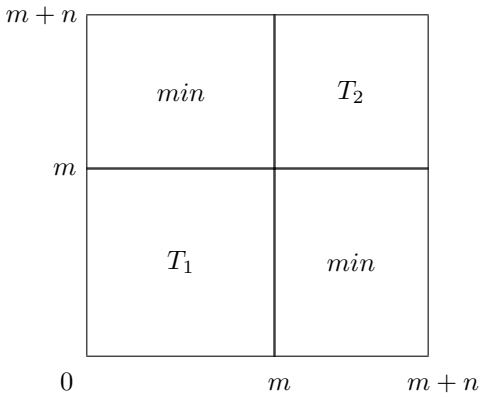


Figura 1: suma ordinal de dos t-normas.

la propiedad de divisibilidad de T_1 y T_2 implica la divisibilidad de su suma ordinal.

El siguiente resultado describe totalmente la estructura de las t-normas divisibles ([3],[1]). Este teorema de representación es análogo al teorema de representación de t-normas (sobre $[0, 1]$) continuas ([5])

Teorema 1 *Si T es una t-norma divisible sobre $L = \{0, 1, ..., n\}$ entonces $T = T_{\text{Ł}}$, $T = T_M$, o bien, T es una suma ordinal no trivial de t-normas de Łukasiewicz.*

Una consecuencia directa del resultado anterior es que cada t-norma divisible está únicamente determinada por sus elementos idempotentes ($a \in L$ es *idempotente* si $T(a, a) = a$). Consiguientemente, existen exactamente 2^{n-1} t-normas divisibles diferentes sobre $L = \{0, 1, ..., n\}$.

Además de poder construir t-normas mediante sumas ordinales de otras t-normas, tratamos a continuación otra vía que permite generar t-normas pero esta vez a partir de funciones de una variable. Esto se consigue haciendo uso de la pseudo-inversa de una función monótona, de manera similar a cómo se hace en el caso continuo ([5], [6]). Este método nos retrotrae al viejo problema (N.H. Abel, 1826) de si existen construcciones que involucren sólo una función de una variable real y la suma ordinaria (o multiplicación) y que definan funciones reales de dos variables (operaciones en la recta real) que posean propiedades algebraicas interesantes, en particular, la asociatividad. Con posterioridad, los trabajos de Aczél (1949), Schweizer & Sklar (1961 y 1963), Ling (1965) han sido importantes para el tratamiento de este problema.

3 GENERADOR ADITIVO DE UNA T-NORMA DISCRETA

Como acabamos de comentar, la idea de construir t-normas con la ayuda de funciones de una variable utilizando la suma ordinaria está contenida en un resultado de Abel en que da una condición suficiente para que operaciones sobre la recta real sean asociativas (una versión actualizada de este resultado puede verse en [1], Theorem 3.22). Para t-normas sobre el intervalo real [0,1] son conocidos los teoremas que caracterizan las t-normas que tienen generador aditivo continuo. También es sabido que existen generadores aditivos (no continuos) para la t-norma drástica y para otras t-normas no continuas, sin embargo esto no es cierto para la t-norma mínimo y, en general, para cualquier t-norma que posea idempotentes no triviales. En el caso discreto, la situación en algunos aspectos es diferente tal como se verá a continuación

Si una función $f : L \to [0, +\infty)$ es estrictamente decreciente y $f(n) = 0$, podemos construir a partir de ella una operación binaria T sobre L, $T : L \times L \to L$, de la manera siguiente: $T(x, y) = f^*(f(x)+f(y))\ \forall x, y \in L$, donde f^* es la función llamada pseudo-inversa de f que se define por $f^*(t) = min\{x \in L : f(x) \leq t\}$, $t \in [0, +\infty)$. Es inmediato demostrar que T es conmutativa, creciente en cada variable, tiene n como neutro y 0 es absorbente ($T(x, n) = x$, $T(x, 0) = 0\ \forall x \in L$).

Definición 4 *Diremos que una operación (binaria) T sobre L está generada aditivamente si existe una función f, en las condiciones anteriores, tal que $T(x, y) = f^*(f(x) + f(y))\ \forall x, y \in L$. En este caso, diremos también que f es un generador aditivo de T.*

Una notación cómoda para indicar que T tiene generador aditivo f es $T = \langle f\rangle$ o también $T = \langle(a_0, a_1, ..., a_n)\rangle$ donde $a_x = f(x)$, $x \in L$. Por supuesto, $a_0 > a_1 > ... > a_{n-1} > a_n = 0$.

Naturalmente, hay operaciones T sobre L (con las propiedades citadas anteriormente) que no tienen generador aditivo. Así, por ejemplo, en el caso $L = \{0, 1, 2, 3, 4\}$, la operación T definida por $T(1, 1) = T(1, 2) = T(1, 3) = 0$, $T(2, 2) = T(2, 3) = 1$, $T(3, 3) = 3$ no posee generador aditivo. En efecto, si $f = (d > c > b > a > 0)$ fuera un generador aditivo de T tendría que ser $2a < b$, $c \leq a+b < 2b < d \leq a+c$, por tanto, $b < \frac{a+c}{2}$. Por otra parte, $2b > 2a+b \geq a+c$ luego $b > \frac{a+c}{2}$, llegando una contradicción.

Proposición 1 *Sean $f = (a_0, a_1, ..., a_n)$ y $g = (b_0, b_1, ..., b_n)$ funciones sobre $L = \{0, 1, ..., n\}$ estrictamente decrecientes con $a_n = b_n = 0$. Entonces, $\langle f\rangle = \langle g\rangle$ si y sólo si se verifican $\forall x, y, z \in L$, $z \neq 0$:*

1) $a_x + a_y \geq a_0 \Longrightarrow b_x + b_y \geq b_0$
2) $a_z \leq a_x + a_y < a_{z-1} \Longrightarrow b_z \leq b_x + b_y < b_{z-1}$

Demostración: Evidente teniendo en cuenta la Definición 4. ♢

Proposición 2 *Si $f : L \to [0, +\infty)$ es estrictamente decreciente con $f(n) = 0$, entonces $\langle f\rangle = \langle kf\rangle\ \forall k > 0$.*

Demostración: Inmediata al aplicar la Proposición 1. ♢

Proposición 3 *Sea $T = \langle f\rangle$, entonces existe un generador aditivo g de T tal que Rang $g \subset Z^+$.*

Demostración Sea $f = (a_0 > a_1 > ... > a_{n-1} > a_n = 0)$ un generador aditivo de T. Por la proposición 2, bastará comprobar que existe un generador aditivo de T, $g = (b_0, b_1, ..., b_n = 0)$ tal que $b_i \in Q$, $\forall i = 0, 1, ..., n$.

Sean $\delta_t = a_t - max\{a_x + a_y : x, y \in L, a_x + a_y < a_t\}$, $t = 0, 1, ..., n-1$. Sea ahora $0 < \delta < min\{\delta_0, \delta_1, ..., \delta_{n-1}\}$.

Sean ahora $\epsilon_t = \frac{\delta}{4^{t+1}}$, $t = 0, 1, ..., n-1$, y sean $b_t \in (a_t - 4\epsilon_t, a_t - 2\epsilon_t) \cap Q$. Veamos que $g = (b_0, b_1, .., b_{n-1}, b_n = 0)$ es un generador aditivo de T.

Por la proposición 1 bastará comprobar que
1) $a_x + a_y \geq a_0 \Longrightarrow b_x + b_y \geq b_0$
2) $a_z \leq a_x + a_y < a_{z-1} \Longrightarrow b_z \leq b_x + b_y < b_{z-1}$

Veamos 1: Al suponer $0 < x \leq y$ (si $0 = x \leq y$ entonces es obvio que $b_0 \leq b_0 + b_y$) tenemos:

$$\begin{aligned} b_x + b_y &> a_x + a_y - 4(\epsilon_x + \epsilon_y) \\ &\geq a_0 - 2\epsilon_{x-1} \\ &\geq a_0 - 2\epsilon_0 \\ &> b_0 \end{aligned}$$

Veamos 2: Por una parte, al suponer de nuevo que $z < x \leq y$ (si $z = x \leq y$ entonces es obvio que $b_z \leq b_z + b_y$) tenemos:

$$\begin{aligned} b_x + b_y &> a_x + a_y - 4(\epsilon_x + \epsilon_y) \\ &\geq a_z - 2\epsilon_{x-1} \\ &\geq a_z - 2\epsilon_z \\ &> b_z \end{aligned}$$

y por otra parte,

$$\begin{aligned} b_x + b_y &< a_x + a_y \\ &< a_{z-1} - \delta \\ &< a_{z-1} - 4\epsilon_{z-1} \\ &< b_{z-1} \end{aligned}$$

♢

Si $T = \langle f\rangle$ podemos considerar el conjunto de generadores aditivos de T con rango incluido en $Z^+ = \{0, 1, 2, 3, ...\}$ (lo que podríamos llamar generadores aditivos "enteros" de T). De esta manera, con-

siderando este conjunto ordenado de manera habitual (orden producto), podemos hablar de generadores (enteros) minimales de T. Así, $f = (a_0, a_1, ..., a_n = 0)$ generador aditivo de T es minimal si y sólo si $g = (b_0, b_1, , b_n = 0)$ con $b_x \leq a_x \ \forall x \in L$ y $b_z < a_z$ para algún $z \in L$, no es generador aditivo de T.

Ejemplos 2: Las t-normas básicas citadas en Ejemplos 1 tienen generador aditivo. Así,

1. $T_M = \langle (2^n - 1, 2^{n-1} - 1, ..., 7, 3, 1, 0) \rangle$.
2. $T_{\text{Ł}} = \langle (n, n-1, ..., 1, 0) \rangle$.
3. $T_D = \langle (2n-2, 2n-3, ..., n, n-1, 0) \rangle$

Estos generadores son obviamente no sólo minimales sino mínimos para cada t-norma.

Por otra parte, $T = \langle f \rangle$ no tiene porqué ser asociativa. Puede probarse fácilmente que la condición $Rang\ f + Rang\ f \subset Rang\ f \cup [f(0), +\infty)$ es suficiente para asegurar la asociatividad de T, pero evidentemente esta condición no es necesaria (ver, por ejemplo, el generador para $n = 4$ de $T_M : (15, 7, 3, 1, 0)$). El problema de hallar condiciones necesarias y suficientes (sobre f) que aseguren la asociatividad de T queda de momento pendiente. Sin embargo, en este trabajo estamos interesados principalmente en el estudio de t-normas sobre L que tengan generador aditivo. Para un reciente estudio de este problema pero para t-normas definidas sobre $[0, 1]$ puede verse ([6]).

En lo que sigue, generador significará generador aditivo entero.

Proposición 4 *Si $f_1 = (a_0, a_1, ..., a_m)$ es un generador de una t-norma T_1 y $f_2 = (b_0, b_1, ..., b_n)$ es un generador de una t-norma T_2, entonces $f = (b_0(2a_0 + 1), .., b_0(2a_m + 1) = b_0, b_1, ..., b_n)$ es un generador de la t-norma T suma ordinal de las t-normas T_1 y T_2.*

Demostración. De acuerdo con la Definición 3, se trata de probar:

1. Si $0 \leq x, y \leq m$, entonces $T(x, y) = T_1(x, y)$. En efecto,
$$\begin{aligned} T(x,y) &= f^*(f(x) + f(y)) \\ &= f^*(b_0(2a_x + 1) + b_0(2a_y + 1)) \\ &= f^*(2b_0(a_x + a_y + 1)) \\ &= f_1^*(a_x + a_y) \\ &= f_1^*(f_1(x) + f_1(y)) \\ &= T_1(x, y) \end{aligned}$$
2. Si $m \leq x, y \leq m + n$, entonces $T(x, y) = m + T_2(x - m, y - m)$. En efecto,
$$\begin{aligned} T(x,y) &= f^*(f(x) + f(y)) \\ &= f^*(b_{x-m} + b_{y-m}) \\ &= m + f_2^*(b_{x-m} + b_{y-m}) \\ &= m + f_2^*(f_2(x - m) + f_2(y - m)) \\ &= m + T_2(x - m, y - m) \end{aligned}$$
3. Si $0 \leq x \leq m \leq y \leq m + n$, entonces $T(x, y) = x = min(x, y)$. En efecto,
$$\begin{aligned} T(x,y) &= f^*(f(x) + f(y)) \\ &= f^*(b_0(2a_x + 1) + b_{y-m}) \\ &= x \end{aligned}$$

♢

Consecuencia inmediata del resultado anterior, del Teorema 1 y del hecho que la t-norma de Łukasiewicz es representable de forma aditiva, es la siguiente

Proposición 5 *Cualquier t-norma divisible tiene generador.*

Vale la pena observar que el generador construido en la Proposición 4 a partir de generadores de T_1 y T_2 no tiene porqué ser minimal. A continuación, y para t-normas divisibles, vemos cómo se puede construir un generador mínimo a partir de los generadores mínimos de cada una de las t-normas sumandos de la t-norma divisible.

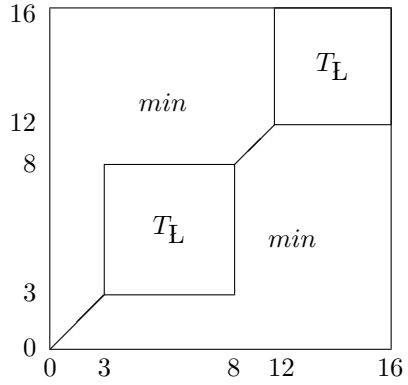


Figura 2: una t-norma divisible.

Sea T una t-norma divisible sobre $L = \{0, 1, ..., n\}$, y sean $r, s : 0 \leq r < s \leq n$ elementos idempotentes de T consecutivos (es decir, siendo $r \leq x \leq s$, $T(x, x) = x$ equivale a $x = r$ o $x = s$).

Por el teorema 1 sabemos que la restricción de T a $[r, s] = \{r, r + 1, ..., s - 1, s\}$ "es" la t-norma de Lukasiewicz sobre $\{0, 1, ..., s - r\}$ que tiene generador mínimo $(s-r > s-r-1 > ... > 1 > 0)$, así que vamos a construir el generador mínimo f de T "pegando" trozos de la forma $((s-r+1)a_s+(s-r) > ... > 3a_s+2 > 2a_s + 1 > a_s)$, es decir $f(s) = a_s$, $f(s - 1) = 2a_s + 1$, $f(s - 2) = 3a_s + 2$, ..., $f(r) = (s - r + 1)a_s + (s - r)$.

Sabiendo que tiene que ser $f(n) = 0$, esto permite la construcción paso a paso del generador.

Veamos a continuación un ejemplo de cómo se construye el generador mínimo de la t-norma T de la figura 2 (ver anexo 2). Los idempotentes de esta t-norma son: $0, 1, 2, 3, 8, 9, 10, 11, 12, 16$ y empezamos por definir f en el intervalo $[12, 16] : (4 > 3 > 2 > 1 > 0)$, luego en $[11, 12] : (9 > 4)$, en $[10, 11] : (19 > 9)$, en $[9, 10] : (39 > 19)$, en $[8, 9] : (79 > 39)$, en $[3, 8] : (479 > 399 > 319 > 239 > 159 > 79)$, en $[2, 3] : (959 > 479)$, en $[1, 2] : (1919 > 479)$ y, finalmente, en $[0, 1] : (3839 > 1919)$.

Así, el generador mínimo de T es:

$f = (3839, 1919, 959, 479, 399, 319, 239, 159, 79, 39, 19, 9, 4, 3, 2, 1, 0)$.

Observación Una conorma triangular (t-conorma para abreviar) sobre $L = \{0, 1, 2, ..., n\}$ es una operación binaria S sobre L tal que se satisfacen los tres primeros axiomas de t-norma (T1 a T3) sustituyendo el cuarto por $S(x, 0) = x \; \forall x \in L$. A cada t-norma T se le puede asociar una t-conorma S (la dual de T) de la manera siguiente: $S(x, y) = n - T(n - x, n - y) \; x, y \in L$, y recíprocamente. Esta correspondencia define una biyección (dualidad) entre t-normas y t-conormas sobre L, que permite trasladar resultados establecidos en una de las clases a la otra clase. En particular, es evidente que si una t-norma T tiene generador aditivo f entonces la t-conorma dual S tiene generador aditivo $g(x) = f(n - x)$, $x \in L$; entendiendo por generador aditivo g de una t-conorma S una función estrictamente creciente entre L y $[0, +\infty)$ con $g(0) = 0$ y tal que $S(x, y) = g^*(g(x) + g(y)) \; x, y \in L$ siendo $g^*(t) = max\{x \in L; g(x) \leq t\}$, $t \in [0, +\infty)$.

Conclusiones En este trabajo se ha abordado el problema de la representación aditiva de t-normas definidas en una cadena finita, obteniéndose resultados que permiten afirmar, por ejemplo, que toda t-norma divisible tiene generador aditivo. A destacar el resultado de la Proposición 4 que pone de manifiesto una clara diferencia con respecto al caso continuo. Quedan pendientes varias cuestiones, una de ellas es la de hallar condiciones necesarias y suficientes para que una operación generada aditivamente sea asociativa. Otro problema a resolver es el de estudiar si todas las t-normas tienen generador aditivo (ver anexo 1). Estos dos problemas, que en el caso real siguen sin ser resueltos en su totalidad, no parecen de fácil solución, y no hay que descartar la ayuda que pueden proporcionarnos las herramientas a nivel de computación de que disponemos.

Agredecimientos Este trabajo ha sido financiado por los proyectos PRDIB-2002GC-19 (Gobierno Balear) y BFM2003-05308 (MCYT).

Referencias

[1] E.P. Klement, R.Mesiar, E. Pap. (2000). Triangular Norms. Kluwer Academic Publishers.

[2] C. M. Ling. (1965) *Representation of associative functions*, Publ. Math. Debrecen 12, 189-212

[3] G. Mayor, J. Torrens. (1993) *On a class of operators for expert systems*, Int. J. Intell. Syst. 8, 771-778.

[4] G. Mayor, J. Torrens (2004) *Triangular Norms on Discrete Settings* en Triangular Norms and Related Operations. Theory and Applications. Physica-Varlag, edited by E.P. Klement, R. Mesiar, E. Pap. Pendiente de publicación.

[5] B. Schweizer, A. Sklar (1983). Probabilistic Metric Spaces. North Holland.

[6] P. Viceník. *Additive Generators Of Non-Continuous Triangular Norms*, Topological and Algebraic Structures in Fuzzy Sets. Edited by S.E. Rodabaugh and E.P.Klement, Chapter 18 (2003), Kluwer Academic Publishers.

Anexo 1 Generadores aditivos minimales para las 22 t-normas sobre $L = \{0, 1, 2, 3, 4\}$:

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	2
3	0	0	1	2	3
4	0	1	2	3	4

$T_{\mathbf{L}} = \langle (4, 3, 2, 1, 0) \rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	1	1
2	0	1	1	1	2
3	0	1	1	2	3
4	0	1	2	3	4

$T = \langle (7, 3, 2, 1, 0) \rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	1	1	1
2	0	1	2	2	2
3	0	1	2	2	3
4	0	1	2	3	4

$T = \langle (8, 5, 2, 1, 0) \rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	2	2
3	0	1	2	3	3
4	0	1	2	3	4

$T = \langle (7, 5, 3, 1, 0) \rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	1	1
2	0	1	2	2	2
3	0	1	2	2	3
4	0	1	2	3	4

$T = \langle (11, 5, 2, 1, 0) \rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	1	1
2	0	1	1	2	2
3	0	1	2	3	3
4	0	1	2	3	4

$T = \langle (11, 5, 3, 1, 0) \rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	1	1	1
2	0	1	2	2	2
3	0	1	2	3	3
4	0	1	2	3	4

$T = \langle(11, 7, 3, 1, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	1	1
2	0	1	2	2	2
3	0	1	2	3	3
4	0	1	2	3	4

$T_M = \langle(15, 7, 3, 1, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	1	1
2	0	1	1	1	2
3	0	1	1	3	3
4	0	1	2	3	4

$T = \langle(9, 4, 3, 1, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	0	2	2
3	0	1	2	3	3
4	0	1	2	3	4

$T = \langle(8, 6, 4, 1, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	0	2
3	0	0	0	0	3
4	0	1	2	3	4

$T_D = \langle(6, 5, 4, 3, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	0	2
3	0	0	0	1	3
4	0	1	2	3	4

$T = \langle(5, 4, 3, 2, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	2	2	2
3	0	1	2	3	3
4	0	1	2	3	4

$T = \langle(9, 7, 3, 1, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	0	2
3	0	0	0	3	3
4	0	1	2	3	4

$T = \langle(7, 6, 5, 2, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	2
3	0	0	1	1	3
4	0	1	2	3	4

$T = \langle(6, 4, 3, 2, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	0	2
3	0	0	0	2	3
4	0	1	2	3	4

$T = \langle(6, 5, 4, 2, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	2	2
3	0	0	2	3	3
4	0	1	2	3	4

$T = \langle(6, 5, 3, 1, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	2	2	2
3	0	0	2	3	3
4	0	1	2	3	4

$T = \langle(6, 5, 2, 1, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	1	1	2
3	0	0	1	1	3
4	0	1	2	3	4

$T = \langle(9, 6, 4, 3, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	1	1
2	0	1	1	1	2
3	0	1	1	1	3
4	0	1	2	3	4

$T = \langle(9, 4, 3, 2, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	2	2	2
3	0	0	2	3	3
4	0	1	2	3	4

$T = \langle(8, 7, 3, 1, 0)\rangle$

T	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	0	1	2
3	0	1	1	3	3
4	0	1	2	3	4

$T = \langle(6, 4, 3, 1, 0)\rangle$

Anexo 2 Tabla de la t-norma de la figura 2.

T	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	**1**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	1	**2**	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	0	1	2	**3**	3	3	3	3	3	3	3	3	3	3	3	3	3
4	0	1	2	3	3	3	3	3	4	4	4	4	4	4	4	4	4
5	0	1	2	3	3	3	3	4	5	5	5	5	5	5	5	5	5
6	0	1	2	3	3	3	4	5	6	6	6	6	6	6	6	6	6
7	0	1	2	3	3	4	5	6	7	7	7	7	7	7	7	7	7
8	0	1	2	3	4	5	6	7	**8**	8	8	8	8	8	8	8	8
9	0	1	2	3	4	5	6	7	8	**9**	9	9	9	9	9	9	9
10	0	1	2	3	4	5	6	7	8	9	**10**	10	10	10	10	10	10
11	0	1	2	3	4	5	6	7	8	9	10	**11**	11	11	11	11	11
12	0	1	2	3	4	5	6	7	8	9	10	11	**12**	12	12	12	12
13	0	1	2	3	4	5	6	7	8	9	10	11	12	12	12	12	13
14	0	1	2	3	4	5	6	7	8	9	10	11	12	12	12	13	14
15	0	1	2	3	4	5	6	7	8	9	10	11	12	12	13	14	15
16	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	**16**

MODELO DIFUSO PARA LA RECUPERACIÓN DE SENSORES EN PRESENCIA DE FALLAS CON INCERTIDUMBRE: APLICACIÓN A LA VIGILANCIA EN TIEMPO REAL DE UN TURBO-REACTOR DE AVIACIÓN

Joseph Aguilar-Martin aguilar@laas.fr
Michel Combacau combacau@laas.fr
Eduard Diez-Lledó ediezlle@laas.fr

LAAS-CNRS, 7, Avenue du Colonel Roche, 31077 TOULOUSE (Francia)

Resumen

El problema tratado se incluye en la detección y localización en tiempo real de fallos en los sensores de un motor de avión, así como en ofrecer al sistema de control el mejor valor reconstituido de la variable correspondiente.

Se efectúa una comparación de cada uno de los valores dados por los sensores redundantes instalados con una predicción producida por un filtro de Kalman. Este procedimiento permite decidir el mejor valor recuperado pero deja en suspenso el caso en que la decisión no puede ser fiable debido a la poca diferencia entre las distancias de cada sensor al predictor.

Se atribuye a cada situación una variable "caso". Proponemos en este trabajo resolver el problema de decisión que corresponde al caso de "incertidumbre" mediante la definición de funciones de pertenencia que permitan realizar una interpolación. La caracterización de estos subconjuntos difusos depende de un parámetro que se puede condicionar por las observaciones y decisiones atribuidas a la variable en cuestión. Se establece entonces una base de reglas cuyos antecedentes toman en consideración los fallos detectados y el porcentaje de errores observados, así como la fiabilidad a priori de cada vía de los sensores.

Una implementación en tiempo real sobre un simulador industrial del motor ha permitido evaluar la proposición descrita aquí.

***Palabras clave**: subconjunto difuso, interpolación, zona de incertidumbre, simulador en tiempo real, turborreactor de avión*

1 INTRODUCCIÓN. DIAGNÓSTICO EN LÍNEA DE UN MOTOR DE AVIÓN

El problema que se plantea es el de detectar la presencia de un fallo en los sensores, localizar cual es el sensor que presenta el fallo y ofrecer al sistema de control el mejor valor reconstituido de la variable correspondiente, o en su lugar un valor aceptable y sin riesgo.
Las hipótesis de base son: cada variable posee dos líneas de medida redundantes materialmente, sólo se trata en este trabajo un fallo en un solo sensor. Se ha realizado un detector-localizador de fallos, orientado hacia el hecho de que la redundancia física no permite efectuar una decisión a pesar de detectar claramente en fallo. Se obtiene entonces un tercer sensor virtual basado en una predicción de la familia del filtro de Kalman sobre un horizonte deslizante. Esta predicción calcula para cada variable su nuevo valor en el instante siguiente en función linear de sus propios valores en un intervalo finito de tiempo anterior, y los valores de las variables termodinámica o físicamente relacionadas con la variable de la cual se hace el cálculo de predicción en el mismo intervalo temporal. A ese intervalo se le llama '*ventana temporal'*, y del hecho que sea finito y avance con el intervalo de tiempo real de simulación estudiado, '*deslizante'*.

$$y(t) = \sum_{i=1}^{N} a_i \cdot y(t-i) + \sum_{k=1}^{K}\sum_{i=1}^{N} b_{ki} \cdot u_k(t-i) \quad (1)$$

N = longitud de la ventana temporal
K = número de variables
a_i = coeficiente linear de auto-regresión para la variable estimada en el instante t-i
b_{ki} = coeficiente linear para las variables que influyen en la variable estimada en t-i

La comparación de cada uno de los valores dados por los dos sensores redundantes con esa predicción permite decidir cual de los dos obtiene el mejor valor. Sin embargo este procedimiento deja en suspenso el caso en que la decisión no puede ser fiable debido a la poca diferencia entre las distancias de cada sensor hacia el predictor.

2 LÓGICA DE DECISIÓN

Dentro de las diversas situaciones en las cuales puede encontrarse el estimador de una variable y los valores ofrecidos por los sensores se realiza una lógica basada, por una parte, en la situación relativa de las dos vías de medida, por la otra parte en la distancia entre cada uno de los valores y la estimación.

A cada instante ***t*** las dos vías del sensor ofrecen los valores $y_1(t)$ e $y_2(t)$, el estimador es $y_E(t)$, y el valor recuperado es $y_R(t)$, donde es este último que será utilizado por el sistema de control del motor. Se considera también un umbral de detección S_1. Las distancias entre valores son:

$$d(1,2) = |y_1(t) - y_2(t)| \quad (2)$$
$$d(1,E) = |y_1(t) - y_E(t)| \quad (3)$$
$$d(2,E) = |y_2(t) - y_E(t)| \quad (4)$$

En el marco del sistema lógico de decisión PEPI, distinguimos cuatro estados, llamados variables "caso", en función de dos parámetros: la distancia entre las dos vías de detección o sensores, **d(1,2)**, y la diferencia entre las distancias de cada una de las vías al predictor, notada como **d(1,E)-d(2,E)**. Teniendo en cuenta estas definiciones, la tabla 1 nos resume los estados mencionados:

Tabla 1: Variables caso asignadas en función de las relaciones entre las distancias

	Dist vías	**Dist. estimador**	**Valor recuperado**
1	$d(1,2) \leq S_1$	No hay detección	$y_R(t) = ½ [y_1(t) + y_2(t)]$
2	$d(1,2) > S_1$	$d(2,E) - d(1,E) > S_2$	$y_R(t) = y_1(t)$
3	$d(1,2) > S_1$	$d(1,E) - d(2,E) > S_2$	$y_R(t) = y_2(t)$
4	$d(1,2) > S_1$	$\| d(1,E) - d(2,E) \| \leq S_2$	***INCERTIDUMBRE***

La lógica de base se completa con las decisiones que hay que tomar en función de las situaciones de validez de las medidas en relación a sus rangos respectivos conocidos a priori. Los otros cinco casos posibles vienen dados en la tabla 2 a continuación y corresponden a estas situaciones. Como se puede apreciar en los dos últimos casos es necesario conocer un "valor de seguridad", que en el caso del motor estudiado corresponde a valores de variables que reducen la velocidad de rotación del turbopropulsor y avisan al piloto de tal perdida de potencia en dicho motor, concepto clave para el funcionamiento óptimo del avión.

Tabla 2: Variables caso asignadas para comprobar la validez de los valores obtenidos

	Validez medidas	**Validez estimador**	**Valor recuperado**
5	$y_1(t)$ único no válido	$y_E(t)$ válido o no	$y_R(t) = y_2(t)$
6	$y_2(t)$ único no válido	$y_E(t)$ válido o no	$y_R(t) = y_1(t)$
7	$y_1(t)$ y $y_2(t)$ no válidos	$y_E(t)$ válido	$y_R(t) = y_E(t)$
8	$y_1(t)$ y $y_2(t)$ no válidos	$y_E(t)$ no válido	$y_R(t)$ = valor de seguridad
9	$d(1,2) > S_1$	$y_E(t)$ no válido	$y_R(t)$ = valor de seguridad

3 SOLUCIÓN DE LA ZONA DE INCERTIDUMBRE

3.1. SOLUCIÓN INICIAL

Proponemos resolver el problema de decisión correspondiente al caso de incertidumbre mediante la definición de funciones de pertenencia que permitan realizar una interpolación lineal del tipo,

$$y_R(t) = \alpha \cdot y_2(t) + (1-\alpha) \cdot y_1(t) \quad (5)$$

completando así la zona de incertidumbre antes comentada, la tabla 3 modifica el caso cuatro descrito anteriormente en la tabla 2:

Tabla 3: Nuevo estado '4' con valor reconstituido combinación linear delas dos vías

	Dist. vías	Dist. Estimador	Valor recuperado
4	d(1,2)> S_1	|d(1,E)-d(2,E)| ≤ S_2	$y_R(t)= \alpha \cdot y_2(t) + (1-\alpha) \cdot y_1(t)$

El estado cuatro prevé una zona de incertidumbre dónde la diferencia entre las distancias de las vías al valor estimado difiere menos de un umbral de decisión $\mathbf{S_2}$, llamado también umbral difuso ($\mathbf{S_F}$). En este estado, el valor reconstituido por la lógica de decisión PEPI es un valor interpolado linealmente entre dos valores determinados a partir de subconjuntos de pertenencia, según el principio de la lógica difusa.
Los subconjuntos de pertenencia difusas nos permiten definir un parámetro alfa (α) el cual pondera la importancia y el peso de cada uno de los sensores en función de la variable definida previamente **d(1,E)-d(2,E).** De esta manera, el valor alfa se describe como una recta definida en tres intervalos:

$$\alpha = f[d(1,E)-d(2,E)] = \begin{cases} d(1,E)-d(2,E) < -S_2 & \alpha = 0 \\ -S_2 < d(1,E)-d(2,E) < S_2 & \\ d(1,E)-d(2,E) > S_2 & \alpha = 1 \end{cases}$$

con α en el intervalo $[-S_F, S_F]$:

$$\alpha = \frac{1}{2 \cdot S_2} \cdot [d(1,E) - d(2,E)] + 0.5 \qquad (6)$$

La representación gráfica de esta recta se muestra a continuación:

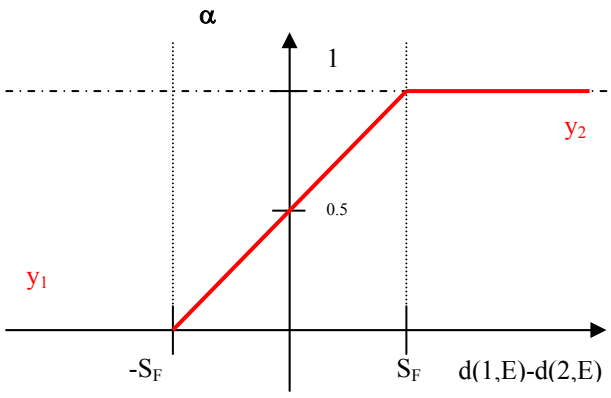


Figura 1: Parámetro alfa en función de la relación de distancias

El objetivo de esta modelización es realizar el cambio entre los dos valores posibles de los dos sensores de forma progresiva y gradual de manera que, en el caso donde la estimación no sea totalmente apropiada, el valor reconstituido $\mathbf{y_R(t)}$ no se distancie demasiado de la media entre los dos valores mencionados, permitiendo así filtrar de manera parcial los posibles errores de estimación, así como otras aproximaciones precedentes, teniendo en cuenta el estrecho margen en el que se llevan a cabo los cálculos y las medidas.
Las dos vías son tratadas de forma dependiente, pues una se considera complementaria de la otra en el cálculo del valor reconstituido, suponiendo que cada una es capaz de suministrar un valor fiable. Este razonamiento no es completamente apropiado a la lógica de detección de fallas, pues no tiene en cuenta la posibilidad que dentro del intervalo de incertidumbre una de los dos sensores sea más fiable uno que el otro, como así es tenido en cuenta en los otros estados definidos (caso 2 y caso 3).

3.2. INTRODUCCIÓN DE PESO EN CADA SENSOR

La idea de esta modificación es ponderar la influencia de las vías sobre el valor reconstituido con parámetros que reflejen, por ejemplo, la fiabilidad de cada una de las vías, las antiguas fallas detectadas o así como el porcentaje de errores detectados en una y otra vía. En este método las vías son tratadas de forma independiente, asignando a cada una de ellas un parámetro alfa (α_1, α_2 respectivamente) como en el método inicial, pero esta vez modificados por los nuevos parámetros definidos, β_1 y β_2, los cuales le dan a cada vía una ponderación independiente y permiten un mejor ajuste al peso específico real de cada una de las medidas en el valor reconstituido. Por consiguiente, definimos dos funciones de pertenencia aplicando los mismos principios de la lógica difusa que para el caso inicial, modificadas esta vez

por los parámetros que acabamos de definir. Las funciones de pertenencia en función de los valores α_i y β_i se definen como rectas partidas en tres intervalos en función de la variable de distancia **d(1,E)-d(2,E)**:

$$\alpha_1 = f[d(1,E)-d(2,E)] = \begin{cases} d(1,E)-d(2,E) < -S_F & \alpha = 0 \\ -S_F < d(1,E)-d(2,E) < S_F & \\ d(1,E)-d(2,E) > S_F & \alpha = \beta_1 \end{cases}$$

siendo la expresión de la recta para el intervalo $[-S_F , S_F]$:

$$\alpha_1 = \frac{\beta_1}{2 \cdot S_F} \cdot [d(1,E) - d(2,E)] + 0.5 \cdot \beta_1 \qquad (7)$$

Su correspondiente representación gráfica de la función de pertenencia:

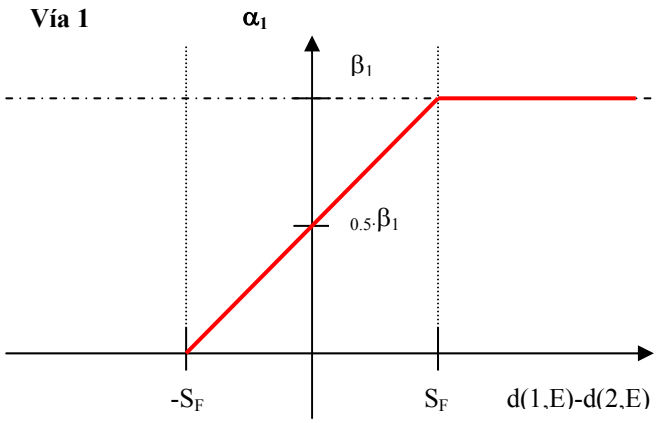


Figura 2: Parámetro α modificado por su correspondiente β, función de las propiedades de la vía 1

Para la vía 2, la definición es homóloga a la anterior:

$$\alpha_2 = f[d(1,E)-d(2,E)] = \begin{cases} d(1,E)-d(2,E) < -S_F & \alpha = \beta_2 \\ -S_F < d(1,E)-d(2,E) < S_F & \\ d(1,E)-d(2,E) > S_F & \alpha = 0 \end{cases}$$

Igualmente, para la expresión del intervalo de incertidumbre $[-S_F , S_F]$, tenemos:

$$\alpha_2 = \frac{\beta_2}{2 \cdot S_F} \cdot [d(1,E) - d(2,E)] + 0.5 \cdot \beta_2 \qquad (8)$$

con la correspondiente gráfica de pertenencia:

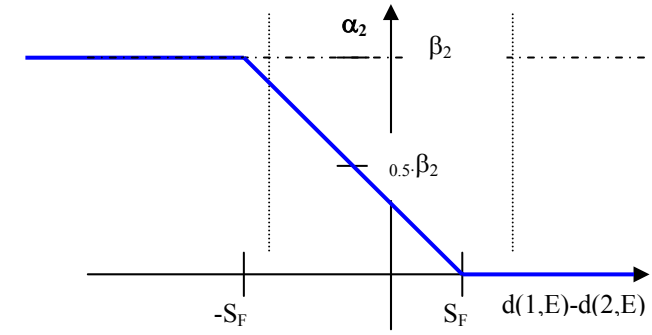


Figura 3 : Parámetro α modificado por su correspondiente β, función de las propiedades de la vía 2

Notar la complementariedad de los dos subconjuntos de pertenencia excepto que la suma de los dos parámetros no es igual a la unidad como en el caso anterior. Debido a esa particularidad, se debe introducir el concepto de 'defuzzification', entendido como el método de obtener un valor real de la combinación de varios subconjuntos difusos (dos en el caso presente). Aplicamos el método de 'defuzzification' ponderado, la expresión matemática del cual se expresa como:

$$y_R(t) = \frac{\alpha_1 \cdot y_1(t) + \alpha_2 \cdot y_2(t)}{\alpha_1 + \alpha_2} \qquad (9)$$

En caso de salir de la zona de incertidumbre, los valore reconstituidos que devuelve el sistema son las respectivas vías, con lo cual se demuestra la continuidad y concordancia en el método global de decisión.

El estudio y cálculo de los parámetros beta **β** introducidos, los cuales definen el peso final de cada vía en el valor reconstituido y caracterizan dichas vías, son importantes para el estudio y desarrollo del método. Estos dos parámetros pueden ser inicializados en valor igual a la unidad, recuperando así la definición inicial del problema.

4 RESULTADOS

4.1. INICIALIZACIÓN DE LA SIMULACIÓN

Las simulaciones se realizan para los diferentes tipos de fallas introducidas de forma artificial en el sistema, en los diferentes escenarios definidos en situaciones de vuelo tales como el despegue, el crucero o el aterrizaje[1]. Los resultados obtenidos son las evoluciones en el tiempo de simulación de las variables suministradas por los sensores, con las fallas introducidas, la detección y consecuente respuesta del regulador PEPI a dicha detección. Se evaluará, por tanto, la capacidad de detección del sistema, así como su fiabilidad a no emitir falsas alarmas de avería ante variaciones (las fallas introducidas) de las medidas de los captores, el cual es el objetivo final en los sistemas de fiabilidad y seguridad en aviones.

4.2. CRITERIOS DE SIMULACIÓN

La evaluación de la fiabilidad de este método se compara con el método inicial, basándose en las variables que más problemas mostraba su respuesta, y a otras variables intermedias en cuanto a respuesta óptima se refiere. De esta manera, aseguramos una mejora en el método con la comparación de variables problemáticas, así como la certificación de que no empeora la respuesta de otras.
Mayormente, los problemas se encuentran cuando se introducen las fallas de bruit anormal (ruido anormal) y mésure figée (medida fija) en cualquiera de los captores. Otro caso de menos problemática se encuentra al aplicar falla sobre el captor de la presión en la cámara de combustión.

4.3. EJEMPLOS DE RESULTADOS

Los resultados aquí mostrados, a modo de ejemplo, corresponden a las medidas de potencia y velocidad de rotación de la turbina, siendo estos los parámetros más críticos en un turbo-reactor. Las curvas mostradas en las figura 4 corresponde a los parámetros mencionados, a la izquierda la potencia del motor y a la derecha la velocidad de rotación N1 del eje principal, sin incluir el método de interpolación fuzzy.

[1] A su vez estos escenarios se convierten en siete tipos más, matizando variables como diferentes alturas en despegue y aterrizaje, o diferentes velocidades en el régimen de crucero.

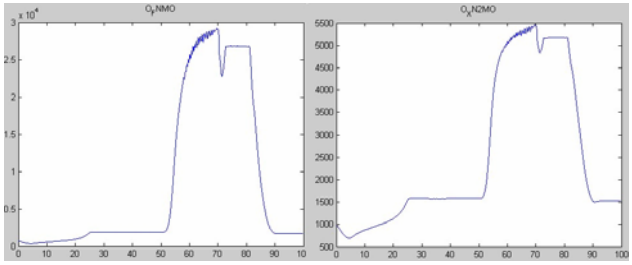


Figura 4: curvas representativas del método inicial

A su vez, se muestra en la figura 5 las curvas correspondientes a las variables comentadas anteriormente, y en el mismo orden para su comparación, con el método fuzzy incorporado en la simulación del turbo-reactor.

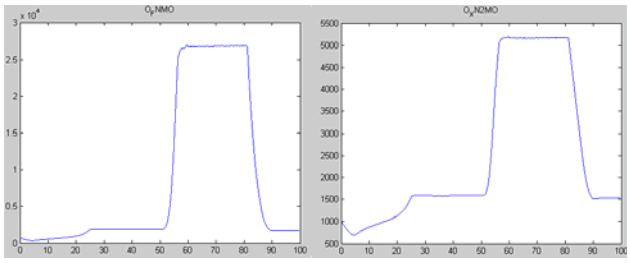


Figura 5: curvas representativas del método fuzzy

Observamos en la zona de pico de potencia un mejor seguimiento y estabilidad de su valor estimado ante la presencia de fallas, evitando las fluctuaciones de las variables que se encontraban antes, debido a los cambios de valor que sufrían de forma radical en el intervalo de incertidumbre.

6 CONCLUSIÓN

En este trabajo, aun en su forma preliminar, se demuestra el interés de introducir conceptos de la Lógica Fuzzy para atenuar los efectos de conmutaciones y decisiones poco fiables, volviéndolas mas robustas. También se puede apreciar la importancia de suavizar las medidas de los sensores cuando estas están incluidas en un control en anillo cerrado.

Los siguientes pasos al respecto de este proyecto deben darse en el campo del estudio de la influencia de la anchura del intervalo difuso en la respuesta obtenida; así como en la adaptación de decisiones inteligentes, empezando por la modelización de los parámetros beta que dotan de variabilidad independiente a las dos vías en función de su rendimiento.

Agradecimientos

Este trabajo está incluido parcialmente en el marco del convenio de investigación CNRS 141 3004.00 conjuntamente con la empresa HISPANO-SUIZA 950 000 407 (GRUPO SNECMA)

Referencias

[1] J. Aguilar-Martin, M. Combacau, A. Doncescu. *Méthode PEPI de détection-localisation des défaillances de capteurs sur le modèle SNECMA REACTEUR 2.6.* Febrero 2004.

[2] S. Kanev. *Robust-fault tolerant control.* Chapitre 1, Febrero 2004

[3] Yager R.R. *An introduction to fuzzy logic applications in intelligent systems*. 1992

Aplicación del Control Difuso en el Proceso de Secado de Tabaco

Ihosvany Alvarez López[a], Orestes Llanes Santiago[b], José Luis Verdegay[a]

[a] Departamento de Ciencias de la Computación e I.A. Universidad de Granada. E-18071 Granada

[b] Departamento de Automática y Computación. Facultad Eléctrica, ISPJAE. La Habana (Cuba)

Resumen

En este articulo se presentan los primeros resultados de un trabajo dirigido al control con lógica difusa del secado de tabaco en hoja en Cuba. Se describe el diseño del controlador desarrollado, y se presentan algunos de los resultados obtenidos que, aunque preliminares, muestran ya las ventajas de este tipo de técnica, en la que el conocimiento de los expertos es fundamental.

Palabras Claves: Control difuso

1 Introducción

La cura controlada del tabaco viene realizándose en Cuba desde hace varias décadas, resultando un producto que se obtiene en menor tiempo que el natural, que arroja un mayor porcentaje de capas para la exportación que aquel, y que requiere un menor número de secaderos, aunque puede que no sean de tipo tradicional. Para lograr las condiciones necesarias en la cura controlada hay que garantizar la temperatura y la humedad relativa adecuadas dentro del secadero durante los pasos de la curación [2], [4]. La tecnología que hasta el momento se ha utilizado para controlar ese proceso de secado [7], no permite introducir cambios que, sin afectar el proceso, consigan aumentar la calidad del producto final y obtener menores consumos de combustible y energía, lo que indudablemente supone una limitación para el curado de tabaco en las actuales condiciones de aquel país.

En este contexto se sitúa el presente trabajo. Sin ignorar toda la experiencia que existe sobre el tema, lo que aquí proponemos es un cambio de tecnología, apoyándonos en técnicas de Soft Computing, en particular en las basadas en lógica difusa, para adquirir el "conocimiento" de los expertos de manera que se logren mejores prestaciones en el proceso de secado de la hoja para, en la mejor tradición del Control Difuso [3,6,7,8], diseñar e implementar un controlador difuso que permita superar los problemas comentados en el proceso del secado del tabaco.

Los resultados que aquí se presentan se refieren a la primera parte de un proyecto de investigación que se está desarrollando entre el ISPJAE (Cuba) y la Universidad de Granada (España) y, aunque preliminares por haberlos obtenido en base a un prototipo experimental, son muy alentadores. Por otro lado, es muy destacable que hasta donde hemos podido saber, la propuesta que en lo que sigue desarrollaremos es innovadora y original, no habiendo encontrado aplicaciones análogas en la literatura especializada en el tema.

El resto del articulo está organizado como sigue. En la sección 2 se expone de manera resumida el proceso tecnológico de cura controlada del tabaco. En la sección 3 se presenta el diseño del sistema de control. En la sección 4 se realiza un análisis de los resultados obtenidos, y finalmente se presentan las conclusiones y los futuros trabajos a realizar

2 Proceso de cura controlada del Tabaco

A finales de la década de los 80 y fundamentalmente en la de los 90, se introduce en Cuba, a escala de producción, la tecnología del curado controlado del tabaco para capas, con objeto de elevar la calidad de la materia prima en la producción de tabacos, y aprovechar las capacidades de curación con respecto al proceso tradicional, ya que de manera controlada el proceso dura 21 días y de manera natural 60 [4].

El sistema de cura controlada permite que el proceso transcurra con control efectivo de dos de los elementos climáticos mas fundamentales que intervienen en el proceso, como son temperatura y humedad, existiendo un tercer elemento, el flujo de aire en el interior de la casa de cura, que se mantiene constante durante el proceso.

A partir de estudios realizados por parte del Instituto de Investigaciones del Tabaco de Cuba [4], acerca de las distintas etapas por las que pasa la hoja de tabaco en el proceso de curado natural, se establecieron cuatro etapas en el proceso de curado, y las condiciones de humedad y temperatura que debían obtenerse en cada una de ellas para realizar este proceso de manera controlada. Las etapas establecidas en el proceso de cura controlada son

1. Amarilleo

2. Desarrollo del color
3. Secado de paños y venas, y
4. Secado de la vena central

y cada una de ellas se subdividió en otras, de modo que finalmente pudimos lograr variaciones suaves en la referencia de la humedad y la temperatura. La humedad en este proceso es crítica, y no debe experimentar cambios bruscos

En el proceso se controlan dos variables, la temperatura y la humedad relativa. La temperatura se controla calentando el aire que se hace circular por el interior de la cámara, mientras que la humedad se controla mediante la apertura de unas ventanillas situadas al paso de la circulación del aire (Fig.1) para intercambiar aire frío del exterior, menos cargado de humedad, con aire caliente del interior de la cámara, cargado de humedad. Todo este proceso es controlado por un dispositivo que garantiza el seguimiento de un programa de curación previamente seleccionado. La Figura 1 muestra gráficamente el proceso tecnológico de la cura controlada del tabaco para capas.

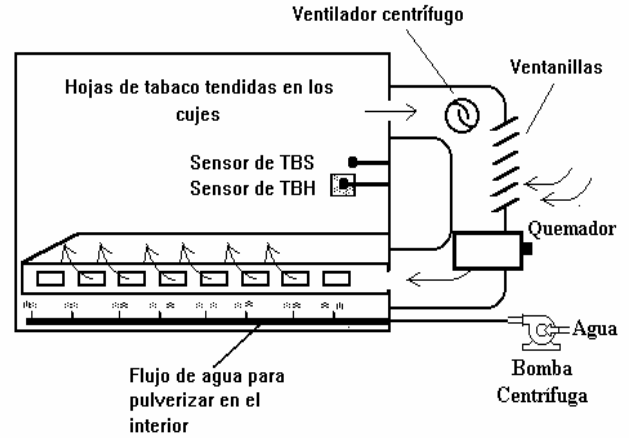


Fig.1 Esquema tecnológico del proceso (TBS y TBH son sensores de humedad y sequedad)

Mediante dos ventiladores centrífugos se hace circular aire caliente por el interior de la casa. Este flujo se mantiene invariable durante toda la cura. El aire se introduce por la parte inferior de la cámara, por un canal que lo reparte de manera uniforme por el interior del recinto, y se extrae por la parte superior una vez que batió sobre las hojas de tabaco tendidas en los cujes por toda la casa, homogenizando el ambiente en la misma. Antes de iniciarse el proceso, el aire pasa por un intercambiador de calor. Mediante el encendido y apagado del quemador, se controla la temperatura en el interior de la cámara. Al elevar la temperatura del aire que circula por la cámara, se está aumentando su capacidad de extraerle la humedad a la hoja del tabaco, debido al aumento de la presión de saturación de la mezcla de aire húmedo.

La humedad se controla en la medida que se abren más o menos las ventanillas. Estas ventanillas se encuentran al paso de la circulación del aire. Con el grado de apertura de las ventanillas se produce más o menos intercambio del aire caliente, que está circulando por el interior de la cámara, con el aire frío del exterior. Este choque con la masa de aire frío del exterior provoca que parte del aire caliente cargado de humedad salga de la circulación, y entre en circulación aire frío del exterior con un contenido de humedad menor.

Con la puesta en marcha de la bomba, que permite la entrada de agua para ser pulverizada en el interior del recinto, se realiza un proceso de humidificación cuando la humedad disminuye por debajo de los límites establecidos. Esto se realiza en los sistemas instalados de forma manual dependiendo del criterio del curador.

3 Diseño del controlador difuso

En esta sección describiremos el diseño del controlador difuso de la humedad relativa, las variables de entrada y salida que se han considerado, sus respectivas funciones de pertenencia, y la correspondiente base de reglas para el control de la humedad.

3.1 La humedad relativa.

Frecuentemente se utiliza el concepto de humedad relativa para caracterizar al aire húmedo. Se llama humedad relativa, y se nota φ (Hr), a la razón existente entre la presión parcial del vapor de agua contenido en una mezcla de aire húmedo y la presión de saturación del vapor de agua a la temperatura dada, es decir a la presión parcial máxima posible del vapor de agua a esta temperatura.[9]

Siguiendo [1], la expresión que se utilizó para el cálculo de la humedad relativa fue

$$\varphi = Hr = [pvs(Tw) / pvs(T)] - Aw\,[p / pvs(T)]\,(T - Tw)$$

donde Tw es una temperatura de saturación de humedad especifica de la zona y el problema, T la temperatura de la mezcla, p la presión de la mezcla (presión atmosférica), pvs la presión de vapor saturado y Aw una constante psicrométrica. Esa constante psicrométrica Aw depende de muchos factores, entre los que destacan la velocidad del aire o la conductividad térmica del mismo. Por las condiciones del lugar donde se instalaron nuestros sensores, y por la experiencias previa [4], se tomó

$$Aw = 6.53\times10^{-4}\ (°C)^{-1}$$

3.2 Variables de entrada y de salida

Como variables de entrada al controlador se escogieron el error,

$$e(k) = RefHr - Hr$$

y su derivada

$$de = [e(k) - e(k-1)]/T$$

donde T es el tiempo de muestreo que se tomó para el error, e(k) el error en el instante de muestreo actual, y e(k-1) el error en el instante de muestreo anterior. En lo que sigue T se toma como unidad, por lo que no aparecerá en lo sucesivo.

La variable de salida del controlador es obvio que será la señal de control U que se aplica al servomecanismo que realiza la apertura o no de las ventanillas que actúan sobre la dinámica de la humedad relativa en el proceso.

De forma compacta, el esquema del controlador difuso desarrollado para la humedad es el que se recoge en la Figura 2.

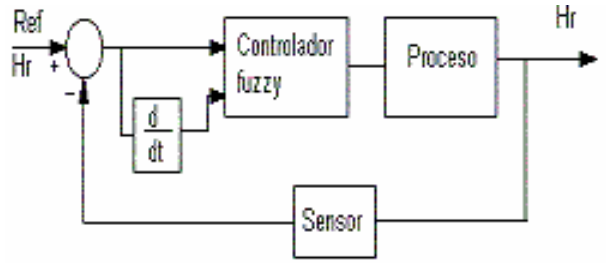


Fig. 2.- Diagrama en bloques del controlador difuso

3.3 Funciones de pertenencia de la variable error

Sobre el universo de variación de la variable error se establecieron 7 conjuntos difusos. Cada una de las variables lingüísticas en las que se subdivide el universo de variación del error, fueron creadas a través de entrevistas con operadores del proceso. De acuerdo a las características de la planta, solo tiene interés analizar errores que no sean menores que -5% ni mayores que el 1 o 2%, es decir, para errores que se encuentren fuera de este rango, el experto genera siempre una señal de control U que no varía.

Así, las funciones de pertenencia que se consideraron para el error (e = RefHr – Hr) son las representadas en la Figura 3, que se corresponden con las expresiones lingüísticas de los expertos, es decir Ng representaría que "un error grande es que la humedad sea del 4 o 5% o mayor que la referencia de la etapa, Nm3 "un error negativo alrededor del 4%", Nm2 "un error negativo pero alrededor del 3%", Nm1 "un error en la humedad relativa que sea pequeño pero no muy pequeño, más o menos alrededor del -2%", Np "un error pequeño, alrededor del 1% pero negativo", Z "un error que esté alrededor de cero, que la humedad esté más o menos un 1% por encima o por debajo del valor de la referencia", y P "un error que sea positivo, que la humedad sea menor que la referencia fijada por más del 1%".

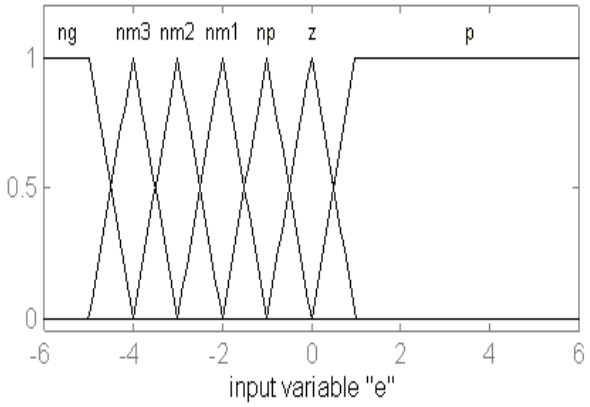


Fig. 3 Funciones de pertenencia de la variable error.

En la definición de los conjuntos difusos sobre el universo de discurso de la variable error se observa que para valores de error debajo del -5% hay una saturación. Esta saturación del conjunto difuso Ng se debe a que para valores de error que sean menores del -5% la acción de control a ejecutar es apertura total de las ventanillas. Es decir la señal de salida del controlador no varía ante nuevas variaciones del error que aumenten negativamente. Igual sucede con los errores positivos mayores que el 1% de Hr que tienen total pertenencia al conjunto P. Frente a errores mayores que el 1% de Hr la decisión del operador sería cierre total de las ventanillas, y no habrá más alternativa, existe una saturación.

3.4. Funciones de pertenencia de la derivada del error

Cuando se instaló el equipamiento, se realizó una identificación del sistema para calcular la constante de tiempo de respuesta, que permitiera establecer el intervalo de muestreo para la derivada del error. Las pruebas se hicieron para el peor de los casos, cuando la casa de tabaco está vacía. En esta situación la dinámica del sistema es la mayor. Al no haber hojas de tabaco en su interior, que constituyen un punto de acumulación de humedad que determinan la dinámica del sistema conjuntamente con otros factores, se acotó la variación del error entre -5% de Hr y +5% de Hr estableciéndose el tiempo de muestreo adecuado para esto.

Cuando la cámara de curado de tabaco esté llena es lógico pensar que el error no va a experimentar cambios superiores al 4 o 5% respecto al intervalo anterior de muestreo. En la operación normal del proceso las variaciones del error, es decir su derivada, estarán sobre el 2 o 3% de Hr que serían las condiciones nominales del proceso.

Como es evidente las funciones de pertenencia para los distintos valores de la variable derivada del error (de =

e(k) - e(k-1)) también fueron seleccionadas a partir del criterio de los expertos. Se representan en la Figura 4.

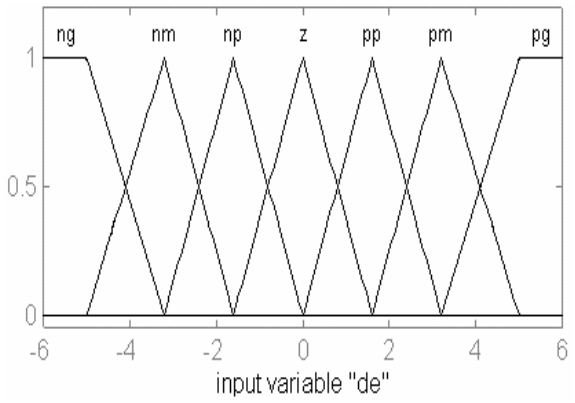


Fig. 4.- Funciones de pertenencia de la derivada del error

En lo que se refiere a su significado,

- Ng se corresponde con un error que vaya disminuyendo a razón de 5% o mayor en humedad relativa por cada instante de muestreo,
- Nm supone un error que vaya disminuyendo con una razón de alrededor del 3,2% en humedad relativa por cada instante de muestreo,
- Np se asocia a un error que vaya disminuyendo con una razón de alrededor del 1,6% en humedad relativa aproximadamente por cada instante de muestreo,
- Z corresponde a una razón de cambio del error que esté alrededor de 0 en porcentaje de humedad relativa con respecto al instante anterior de muestreo,
- Pp se refiere a un error que vaya aumentando con una razón de alrededor del 1,6% en humedad relativa aproximadamente por cada instante de muestreo,
- Pm se asocia con un error que vaya aumentando con una razón de alrededor del 3,2% en humedad relativa por cada instante de muestreo, y finalmente
- Pg se corresponde con un error que vaya aumentando a razón de 5% o mayor en humedad relativa por cada instante de muestreo.

3.5. Funciones de pertenencia de la señal de salida U

El operador experto considera todo el rango de apertura de las ventanillas, y no tiene una zona dentro de la gama de aperturas en la que tome un mayor número de decisiones. A partir de los datos obtenidos y del estudio del proceso, se decidió tomar las siguientes funciones de pertenencia para la señal de salida U del controlador. Las funciones de pertenencia de la salida del controlador son de tipo "singleton". Cada una de esas etiquetas se interpreta como el valor de máxima pertenencia de ese valor de la variable al conjunto difuso, y se muestran en la figura:

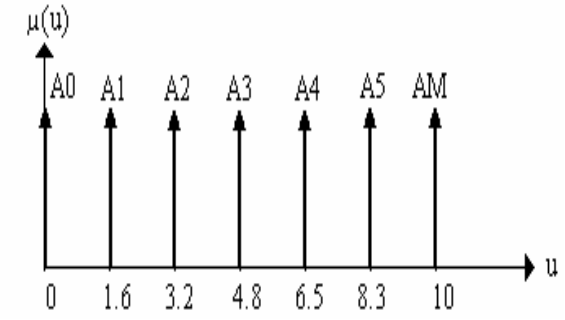


Fig. 6 Funciones de pertenencia de la señal de salida U

con los siguientes significados:

A0 (Apertura 0) cierre total de las ventanillas,
A1 (Apertura 1) 16% de la apertura máxima posible
A2 (Apertura 2) 32% de la apertura máxima posible,
A3 (Apertura 3) 48% de la apertura máxima posible,
A4 (Apertura 4) 65% de la apertura máxima posible,
A5 (Apertura 5) 83% de la apertura máxima posible, y
AM (Apertura Máxima) 100% de apertura de ventanillas.

Una vez que se obtiene el grado de cumplimiento de un predicado de una regla, este se aplica directamente a la salida "singleton" como un factor de escala. Las reglas o proposiciones se expresan de igual forma que si fueran otro tipo de funciones de pertenencia en el espacio de salida. Cuando se han aplicado las reglas, la altura de cada "singleton" ya no será 1, sino el valor de cumplimiento de cada una de ellas.

Para la desfuzzificación se hace una interpolación lineal entre los puntos que se corresponden con la altura resultante de cada "singleton", para construir así la representación del conjunto difuso de salida. Se consigue así una sustancial simplificación del método de desfuzzificación por el centro de gravedad, que finamente se calcula como

$$u^* = \Sigma_{i=1..n} d_i S_i / \Sigma_{i=1..n} S_i$$

donde S_i es la altura resultante del singleton i, correspondiente a la i-esima regla, d_i es el valor del dominio que corresponde al singleton, y n es el numero de reglas, es decir $i \in \{1, 2, \ldots, n\}$.

3.6 Base de reglas para la humedad

Siguiendo la metodología usual, después de diseñar las funciones de pertenencia de cada una de las variables que maneja el controlador difuso, procedimos a la generación de las reglas o base de conocimientos, lo que nos llevó a la siguiente tabla:

e / de	Ng	Nm3	Nm2	Nm1	Np	Z	P
Ng	[1]Am	[2]Am	[3]Am	[4]Am	[5]A5	[6]A4	[7]A3
Nm	[8]Am	[9]Am	[10]Am	[11]A5	[12]A4	[13]A3	[14]A2
Np	[15]Am	[16]Am	[17]A5	[18]A4	[19]A3	[20]A2	[21]A1
Z	[22]Am	[23]A5	[24]A4	[25]A3	[26]A2	[27]A1	[28]A0
Pp	[29]A5	[30]A4	[31]A3	[32]A2	[33]A1	[34]A0	[35]A0
Pm	[36]A4	[37]A3	[38]A2	[39]A1	[40]A0	[41]A0	[42]A0
Pg	[43]A3	[44]A2	[45]A1	[46]A0	[47]A0	[48]A0	[49]A0

Donde evidentemente todas las reglas están expresadas como sentencias del tipo

"si e es X1 y de es X2 entonces U es Ax".

Se debe destacar que para errores positivos o casi cero la salida del controlador difuso es A0, o pequeños valores que indicarían aperturas que estarán a menos del 50% de la apertura total de las ventanillas. Esto se ve en las reglas aplicadas cuando se rompe la secuencia de que al mantener una variable en un conjunto difuso determinado y aumentar la otra en un conjunto difuso, la salida del controlador aumenta en un conjunto difuso también.

4 Análisis de los resultados obtenidos

Una vez implementado el controlador, el sistema completo se adaptó y se probó en una casa de la batería de curación "Hoyo de Mena". Para un mejor análisis se realizó la comparación de los resultados con los obtenidos en una casa testigo sin el sistema y sometida a las mismas condiciones que la casa de prueba.

Para el desarrollo de las pruebas se instalaron tanques para la medición del consumo de combustible, tanto en la casa de prueba como en la testigo.

Es necesario precisar que la temperatura se controló en un rango de 30ºC a 34ºC y la Humedad Relativa del 90% al 75%, según documento emitido por el Instituto de Investigaciones del Tabaco.

Se realizaron tres ciclos de curación con el sistema, donde el tabaco curado fue de muy buena calidad, lo que ha permitido que los resultados reflejen con mayor veracidad las bondades del sistema.

Se pudo disponer de una casa testigo con tabaco de igual campo, corte y productor.

Prueba 1:

Parámetros Comparados	Prueba	Testigo
Duración del Ciclo (días)	19	25
Variedad	Habana 2000	Habana 2000
Corte	Centro Fino	Centro Fino
# Cujes	3 543	3 500
Temp. Ambiente Promedio	14 ºC	14 ºC
Temp. Mínima	11 ºC	11 ºC
Combustible Consumido (lts)	2 046	2 905
Producción (x100)	24.89	20.57
% Exportación	47.4	28.53
% Consumo	45	59.4

A partir de estos datos determinamos que el ahorro de portadores energéticos en este ciclo fue de 30% de combustible.

La prueba realizada no es suficiente como para afirmar en que porcentaje puede mejorar la calidad de la hoja con el uso del sistema, pero si se puede decir que esta no es afectada y se aprecia mejoría en algunos parámetros, otros no se encuentran en el rango deseado pero aún así superan la casa testigo. El uso generalizado del sistema en la batería permitirá profundizar en este estudio para llegar a conclusiones definitivas.

Prueba 2:

El tabaco curado en este ciclo fue de excelente calidad y se curó el mismo tabaco tanto en la casa de prueba como en la testigo.

Parámetros Comparados	Prueba	Testigo
Duración del Ciclo (días)	25	30
Variedad	Criollo 98	Criollo 98
Corte	1y2 Ligero	1y2 Ligero
# Cujes	2 900	3 010
Temp. Ambiente Promedio	17 ºC	17 ºC
Temp. Mínima	16 ºC	16 ºC
Combustible Consumido (lts)	2 151	3 210
Producción (x100)	29.49	-
% Exportación	26	20
% Consumo	63.6	-

El ahorro de portadores energéticos en este ciclo fue del 33% de combustible.

Prueba 3:

En esta prueba no contamos con una casa testigo al no disponer de la cantidad requerida de este tabaco para la misma, por lo que la comparación se hizo respecto a los índices promedio de una casa de esta batería en la zafra.

Parámetros Comparados	**Casa en Prueba**	**Casa Testigo**
Productor	UBP-3 S.Luis	-
Duración del Ciclo (días)	24	25
Variedad	Criollo 98	-
Corte	Ligero y fino	-
# Cujes	3 050	3 335
Consumo (lts)	1 041	1 968

Como resultado se obtuvo que el ahorro de portadores energéticos en este ciclo fue de 47% de combustible.

De los resultados anteriores se aprecia un ahorro considerable de combustible con la utilización del control difuso. Con respecto a la calidad de la hoja, de las pruebas realizadas no se puede concluir que se obtenga un porcentaje mayor de hojas con calidad de capa, pero si se puede afirmar que el porcentaje de hojas con calidad de capa no disminuye.

Finalmente, es preciso indicar que por cuestiones de limitación de la longitud de este articulo, no podemos detallar aspectos tan interesantes y necesarios como los referentes a la ley de inferencia empleada, la manera de elegir los conjuntos de la variable derivada del error, el desempeño de los controladores o la comparación con las instalaciones estándar para este tipo de procesos. En este sentido, el lector interesado puede obtener de los autores una versión mas extensa de este trabajo que actualmente esta en prensa [1].

Conclusiones y trabajos futuros

En el ámbito de los procesos que se siguen para el secado del tabaco en hoja en Cuba, se ha mostrado como la introducción de un sistema de control difuso que simula un control PD ha proporcionado mejores resultados que los algoritmos de control actualmente establecidos, ya que según nuestro criterio este tipo de control logró reflejar de una mejor forma los criterios de los expertos en cuanto a la acción de control a desarrollar en dependencia del estado de las variables.

Aunque pueden obtenerse resultados similares refinando los algoritmos de control actuales, es muy destacable la importante reducción de consumo energético que conlleva el uso del controlador difuso, lo que justifica su empleo frente a los convencionales.

En la actualidad se cuenta con una gran base de datos que refleja las acciones de control que tomó el sistema, los valores de las variables controladas y las acciones de control que impusieron los curadores según su experiencia mientras chequeaban el estado del proceso en cada uno de los pasos. En esta información está implícito el conocimiento y las experiencias de los curadores. Se precisa entonces, con esta información, desarrollar un algoritmo de control fuzzy que nos permita ir adaptando las funciones de pertenencia para incorporar este conocimiento en las mismas y obtener cada vez mejores resultados, lo que constituye el objetivo de la continuación de este trabajo.

Agradecimientos

Este trabajo ha sido realizado al amparo del proyecto JA00/69, financiado por la Consejería de Presidencia de la Junta de Andalucía (España).

Bibliografía

[1] Alvarez López, I, O. Llanes Santiago y J.L. Verdegay: Drying process of tobacco leaves by using a fuzzy controller. Fuzzy Sets and Systems (en prensa).

[2] Ares, D.M., H. García, S. Naranjo e I. Peláez.: Caracterización parcial de las fracciones proteicas extraídas de las hojas de tabaco. Cubatabaco 1(1): 55-61 (1999).

[3] Driankov, D., H. Hellendoorn and M. Reinfrank.: An Introduction to Fuzzy Control. Springer-Verlag (1993).

[4] Instituto de Investigaciones del Tabaco, Ministerio de la Agricultura de Cuba: Instructivo Técnico para el cultivo del Tabaco Negro Curado (2001)

[5] Kalfrisa. Manual de usuario. (1988).

[6] Kosko B.: Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machinary Intelligence. Prentice Hall Inc. (1992).

[7] Wang, X.L.: A Couse in Fuzzy Systems and Control. Prentice Hall Inc. (1997).

[8] Wang, X.L.: Adaptive Fuzzy Systems and Control: Design and Stability Analysis. Prentice Hall Inc. (1994).

[9] Wexler, A.: Humidity and Moisture. Reinhold Pub. Co. EUA. (1965).

COOPERACION DE REGLAS DIFUSAS BASADA EN LA ADAPTACION EVOLUTIVA DE LOS OPERADORES DE CONJUNCIÓN Y LOS DEFUZZIFICADORES

F. Herrera
Dpto. Ciencias de la Computación e I.A.
Universidad de Granada
18071-Granada
herrera@decsai.ugr.es

F. A. Márquez
Depto. Ingeniería Electrónica Sistemas Informáticos y A.
Universidad de Huelva
21071-Huelva
alfredo.marquez@diesia.uhu.es

A. Peregrín
Depto. Ingeniería Electrónica, Sistemas Informáticos y A.
Universidad de Huelva
21071-Huelva
peregrin@diesia.uhu.es

Resumen

En este trabajo[1] se estudia el modelado lingüístico adaptativo cuando el operador de conjunción y el defuzzificador son adaptados simultáneamente, es decir, el Sistema de Inferencia y la Interfaz de Defuzzificación al mismo tiempo. Los parámetros de ambos componentes serán aprendidos empleando algoritmos evolutivos. El objetivo de estos modelos es avanzar en el equilibrio entre precisión e interpretabilidad de los Sistemas Difusos.

Palabras Clave: Modelado Difuso, Métodos de Defuzzificación Adaptativos, Sistema de Inferencia Adaptativo.

1 INTRODUCCIÓN

Los dos requerimientos generalmente contradictorios en el diseño de Modelos Difusos (MD) son la interpretabilidad y la precisión. La interpretabilidad es la capacidad de expresar el comportamiento del sistema real de una forma entendible. La precisión es la capacidad de representar fielmente el comportamiento del sistema real. En la práctica, una de estas dos propiedades prevalece sobre la otra, es decir, los sistemas más interpretables son menos precisos y los más precisos son menos interpretables. Los diseñadores tratan de buscar un balance entre precisión e interpretabilidad, lo que ha producido creciente interés en su estudio [2, 3] y sobre el que versa este trabajo.

El aumento de la precisión de los MD lingüísticos manteniendo el equilibrio con la interpretabilidad se puede llevar a cabo modificando la forma de las funciones de pertenencia, integrando el diseño de la Base de Conocimiento y el sistema de inferencia, reduciendo la base de reglas, cambiando la estructura del modelo (modificadores lingüísticos, reglas con doble consecuente, bases de conocimiento jerárquicas o con pesos) o parametrizando los componentes del modelo difuso, que es la línea de trabajo en la que se sitúa este estudio.

La parametrización de los MD en la mayoría de los casos se ha realizado introduciendo parámetros en la Interfaz de Defuzzificación [11, 15, 16, 17], y ajustándolos mediante algoritmos evolutivos [9, 18, 19]. Algunos trabajos recurren a la parametrización del Sistema de Inferencia [1, 14, 21], si bien el interés en este procedimiento de adaptación ha sido generalmente inferior entre los autores.

El presente trabajo se ocupa de estudiar el uso de expresiones parametrizadas tanto en el Sistema de Inferencia como en la Interfaz de Defuzzificación simultáneamente, y de su ajuste mediante algoritmos evolutivos. El objetivo de estos modelos es avanzar en el equilibrio entre precisión e interpretabilidad. En [14] se mostró que en el ajuste del Sistema de Inferencia, la mayor influencia corresponde al operador de conjunción frente al de implicación, de modo que en esta ocasión se adapta el Sistema de Inferencia sólo a través del primero.

En la Sección 2 se describen los MD Adaptativos Evolutivos. La Sección 3 desarrolla la experimentación realizada y por último la Sección 4 muestra las conclusiones del estudio.

2 MODELOS DIFUSOS ADAPTATIVOS EVOLUTIVOS

En esta Sección se estudia la parametrización conjunta de los componentes de los MD. Se describirá en primer lugar la correspondiente a la Interfaz de Defuzzificación, seguida del Sistema de Inferencia.

[1] Soportado por el MCyT bajo el Proyecto TIC2002-04036-C05-01

2.1. LA INTERFAZ DE DEFUZZIFICION ADAPTATIVA

La expresión formal (1) se emplea generalmente para generar los métodos de defuzzificación adaptativos,

$$y_0 = \frac{\sum_{i}^{N} f(h_i) \times V_i}{\sum_{i}^{N} f(h_i)}, \qquad (1)$$

donde h_i es el grado de emparejamiento, $f(h_i)$ es un término funcional del grado de emparejamiento y V_i es el máximo valor (MV_i) o el centro de gravedad (CG_i). Se trata de la expresión de un método de defuzzificación que actúa en Modo B, es decir, primero defuzzifica la aportación individual de la inferencia obtenida con cada regla y posteriormente se computa una suma ponderada.

El término funcional puede utilizar un solo parámetro , o bien un parámetro por cada regla de la Base de Conocimiento, $_i$, que es la opción utilizada en este trabajo por obtenerse mejores resultados [9]. Por otro lado el término funcional puede ser de tipo producto o potencia, como se muestra en las expresiones (2) y (3).

$$f(h_i) = h_i \times {}_i, \quad f(h_i) = h_i{}^{i}. \qquad (2), (3)$$

La combinación de estos dos funcionales con el MV o el CG genera cuatro métodos de defuzzificación (D_3 a D_6), expresiones (6) a (9). Los métodos D_1 y D_2 son los métodos clásicos MV y CG ponderados el grado de emparejamiento a los que equivalen los parametrizados cuando el valor de los parámetros es igual a 1 (expresiones (4) y (5)).

D_1, MV ponderado por h_i:

$$y_0 = \frac{\sum_{i}^{N} h_i \times MV_i}{\sum_{i}^{N} h_i}, \qquad (4)$$

D_2 , CG ponderado por h_i:

$$y_0 = \frac{\sum_{i}^{N} h_i \times CG_i}{\sum_{i}^{N} h_i}, \qquad (5)$$

D_3 , CG Preciso:

$$y_0 = \frac{\sum_{i}^{N} h_i{}^{i} \times CG_i}{\sum_{i}^{N} h_i{}^{i}}, \qquad (6)$$

D_4:

$$y_0 = \frac{\sum_{i}^{N} h_i{}^{i} \times MV_i}{\sum_{i}^{N} h_i{}^{i}}, \qquad (7)$$

D_5 :

$$y_0 = \frac{\sum_{i}^{N} h_i \times {}_i \times CG_i}{\sum_{i}^{N} h_i \times {}_i}, \qquad (8)$$

D_6 :

$$y_0 = \frac{\sum_{i}^{N} h_i \times {}_i \times MV_i}{\sum_{i}^{N} h_i \times {}_i}. \qquad (9)$$

El funcional de tipo producto tiene un efecto equivalente al empleo de reglas con pesos [4], es decir, equivale a un valor que acompaña a cada regla de la Base de Conocimiento y cuyo significado indica qué grado de importancia debe tener esa regla en el proceso de inferencia. El siguiente ejemplo presenta un conjunto de reglas con pesos, donde éstos son denominados por w_i:

Si X_{11} es A_{11} y… y X_{1n} es A_{1n} entonces Y es B_1 con w_1
Si X_{21} es A_{21} y… y X_{2n} es A_{2n} entonces Y es B_2 con w_2
…
Si X_{n1} es A_{n1} y… y X_{nn} es A_{nn} entonces Y es B_n con w_n

El empleo del funcional de tipo potencia mediante un parámetro para cada regla es equivalente a una de las técnicas empleadas para realizar la modificación lingüística de la estructura de la regla [5, 20]. La modificación lingüística de la regla consiste en relajar la estructura de dicha regla cambiando ligeramente el significado de las etiquetas involucradas. Concretamente, la equivalencia se produce cuando la modificación lingüística utiliza la potencia para cambiar la forma de la función de pertenencia del conjunto difuso asociado a la etiqueta lingüística de los antecedentes. La modificación con potencia es una modificación no lineal que mantiene inalterado el soporte del conjunto difuso. Cuando el conjunto difuso es modificado por un valor en la potencia mayor que 1, la función de pertenencia se contrae. Cuando el conjunto difuso se modifica con valores menores que 1, la función de pertenencia se expande.

2.2. EL SISTEMA DE INFERENCIA ADAPTATIVO

La expresión de la Regla Composicional de Inferencia en el modelado difuso con fuzzificación puntual es mostrada en la expresión (10),

$$_{B'}(y) = I(C(_{A1}(x_1), \ldots, _{An}(x_n)), _{B}(y)), \qquad (10)$$

donde $_{B'}$ es la función de pertenencia del consecuente inferido, I es el operador de implicación, C el operador de conjunción, $_{Ai}$ al tratarse de fuzzificación puntual son los puntos de corte de las entradas discretas $(x_1,\ldots,x_n)$ del sistema con las funciones de pertenencia de los antecedentes de las reglas, y $_B$ el consecuente de la regla.

El sistema de inferencia pues, emplea dos operadores: el de conjunción, C(.) para calcular el grado de emparejamiento, y el de implicación, I(.), lo cual podría expresarse formalmente en (11) para el caso de n variables de entrada y un consecuente.

$$I(C(x_1,\ldots,x_n),y) \qquad (11)$$

Con objeto de obtener un Sistema de Inferencia adaptativo, las alternativas que se presentan consisten en emplear para cualquiera de los dos operadores, una selección que tenga parámetros en su expresión, donde y son los parámetros de I y C respectivamente, tal y como muestran (12) y (13). Trabajos previos [14] demostraron la mayor ganancia de precisión cuando se emplea un ó para cada regla de la Base de Conocimiento, por lo que aquí sólo se empleará así.

$$I_{\ i}(C(x_1,\ldots,x_n),y) \qquad (12)$$

$$I(C_{\ i}(x_1,\ldots,x_n),y) \qquad (13)$$

El aprendizaje de dichos parámetros puede llevarse a cabo mediante algoritmos evolutivos. Este proceso puede ser considerado como una alternativa o complemento a otros métodos que proporcionan precisión [8]. El efecto obtenido empleando un parámetro individual para cada regla es el de adaptar localmente la inferencia para cada regla de la Base de Conocimiento.

Las T-normas son una familia de operadores generalmente usados en el diseño de MD, modelando tanto el operador de conjunción como el de implicación [12] y ofreciendo buen comportamiento [7]. En este trabajo utilizaremos la t-norma del mínimo y tres t-normas parametrizadas recogidas en [13] cuyas expresiones son (14), (15) y (16).

Frank (>0),(1) $$T_{Frank}(x,y) = \log \quad 1 + \frac{(\quad ^{x} \quad 1)(\quad ^{y} \quad 1)}{1 \quad} \qquad (14)$$

Dombi (>0) $$T_{Dombi}(x,y) = \frac{1}{1 + \sqrt{\frac{1 \quad x}{x} \div + \frac{1 \quad y}{y} \div}} \qquad (15)$$

Dubois (0 1) $$T_{Dubois}(x,y) = \frac{x \rangle y}{Max(x,y, \quad)} \qquad (16)$$

En [14] se estudiaron las diferentes formas de parametrización del Sistema de Inferencia observándose menor influencia en el uso de operadores de implicación adaptativos frente al de los de conjunción cuando los conjuntos difusos son simétricos y se emplean métodos de defuzzificación basados en CG y MV (los más frecuentes). Por este motivo, en este trabajo se utiliza la parametrización del Sistema de Inferencia exclusivamente mediante el operador de conjunción.

Los operadores de conjunción adaptativos modulan la influencia del grado de emparejamiento atenuándolo o acentuándolo. Este cambio no es lineal debido a que las expresiones de las t-normas usadas no lo son. Su efecto se asemeja a un conocido mecanismo para modificar el significado lingüístico de la estructura de las reglas [5, 20] que consiste en relajar la estructura de la regla cambiando el significado de las etiquetas involucradas. Por tanto, un operador de conjunción adaptativo actúa similarmente a un modificador de las funciones de pertenencia asociadas a los antecedentes de las reglas. Las t-normas parametrizadas como operadores de conjunción sólo producen efectos de concentración.

2.3. CONJUNCIÓN Y DEFUZZIFICACIÓN ADAPTATIVAS

En esta sección se estudia la construcción de MD con el Sistema de Inferencia y la Interfaz de Defuzzificación adaptativos. Para ello se van a emplear los componentes parametrizados descritos en las secciones 2.1 y 2.2, y se va a estudiar el efecto conjunto que producen.

La Tabla 1 resume el efecto que se consigue tanto con la Inferencia como con la Interfaz de Defuzzificación Adaptativas descritas en las secciones anteriores.

Tabla 1: Efectos del uso de los operadores adaptativos

	Inferencia Adaptativa (Conjunción)	**Defuzzificación Adaptativa**	
		Potencia	**Producto**
Añade Pesos o Niveles de Confianza			X
Modificación Lingüística Concentra Antecedentes	X	X	
Modificación Lingüística Dilata Antecedentes		X	

Resumiendo, las combinaciones de Inferencia Adaptativa con el operador de conjunción parametrizado y el Interfaz de Defuzzificación Adaptativo basado en un método que trabaja en Modo B – FITA con funcional de tipo:

Producto: introduce un peso para cada regla, y permite modificar los antecedentes de las reglas ajustando un grado de concentración de los mismos.

Potencia: permite modificar lingüísticamente el significado de las funciones de pertenencia de la regla tanto dilatando como concentrando los antecedentes.

En ambos casos la adaptación de los parámetros del sistema resultará en el conjunto de reglas de la Base de Conocimiento que mejor cooperan para producir el mejor comportamiento del sistema difuso.

3. ESTUDIO EXPERIMENTAL

En esta Sección se ha planteado una experimentación que tiene como objetivo observar el comportamiento de los MD adaptativos basados en el empleo de operadores parametrizados tanto en el Sistema de Inferencia como en la Interfaz de Defuzzificación según se describió en la sección previa. Para ello, se construirán varios MD y se comprobará su comportamiento en un problema práctico.

3.1. DESCRIPCIÓN DE LA APLICACIÓN

La aplicación utilizada para estudiar el comportamiento práctico de los diferentes MD corresponde a una de las utilizadas en [6], la cual consiste en un problema de distribución del sistema eléctrico español: Se trata del cálculo de costes óptimos teóricos de la línea de media tensión en núcleos urbanos.

Se dispuso de un conjunto de datos de 1059 ciudades que constan de cuatro variables de entrada y una de salida. El dominio de las variables de entrada y salida fue dividido en cinco etiquetas {MP, P, M, G, MG}.

Las bases de reglas, de 65 reglas de tipo Mamdani, se generaron mediante el Método de Wang y Mendel [22] con los datos de entrenamiento, que son el 80% de los datos disponibles, es decir, 847 de las 1059 ciudades,

tomados de modo aleatorio. La evaluación de los modelos se realizó con el 20% restante del conjunto inicial, es decir, las 212 ciudades restantes.

3.2. EL ALGORITMO EVOLUTIVO CHC

Se ha empleado el algoritmo evolutivo CHC [10], el cual es considerado como un modelo evolutivo con un buen balance entre diversidad y convergencia en problemas con grandes espacios de búsqueda.

Aunque el algoritmo CHC fue concebido para cromosomas con codificación binaria, existen versiones para su uso con cromosomas con codificación real como la empleada en este trabajo, la cual utiliza el BLX- como operador de cruce. El cálculo de la distancia de Hamming se realiza convirtiendo los genes reales en una representación en cadenas y hallando la distancia de Hamming teniendo en cuenta si difiere o no cada uno de los caracteres. Sólo aquellas cadenas con una distancia (mayor del umbral) son combinados. El umbral se inicializa a L/4 siendo L la longitud de la cadena o cromosoma. Cuando ningún descendiente es insertado en la nueva población el umbral se reduce a 1.

En la fase de recombinación no se aplica ningún proceso de mutación. En su lugar, cuando la población converge o el proceso de búsqueda deja de progresar adecuadamente (el umbral de cruce llega a 0 y no se generan nuevos descendientes), la población se reinicia. El cromosoma que representa la mejor solución hasta ese momento se utiliza como patrón para generar la nueva población.

En este caso se ha ejecutado con 40000 evaluaciones. La población empleada fue de 50 (inicializados aleatoriamente con excepción del primer cromosoma que fue inicializado a 0 para el caso de la t-norma de Dubois (equivale a la t-norma clásica del mínimo), a 1 en caso de la de Dombi (equivale a la t-norma de Hamacher) y a 0.5 en caso de la de Frank, y a 1 para los métodos de defuzzificación). El valor en el operador de cruce es 0,5. El intervalo de búsqueda fue establecido en el rango [0,1] para la t-norma de Dubois, (0,10] para la de Dombi, y (0,100] para la de Frank. En el caso de los métodos de defuzzificación adaptativos el intervalo es [0,10].

3.3 METODOLOGÍA DE COMPARACIÓN

Se han construido varios modelos combinando el Sistema de Inferencia Adaptativo junto con la Interfaz de Defuzzificación Adaptativa. Como operador de conjunción se han empleado las tres t-normas parametrizadas citadas en la Sección 2.2, manteniendo siempre como operador de implicación la t-norma del mínimo. En cuanto a la Interfaz de Defuzzificación, con objeto de reducir la extensión del trabajo, se han empleado los métodos adaptativos D_3 y D_5 (ambos basados en CG). Como modelos no adaptativos de referencia se ha tomado la t-norma del mínimo para modelar tanto el operador de conjunción como el de implicación y el método de defuzzificación elegido es D_2.

La medida de comparación de los resultados de los modelos elegida es el Error Cuadrático Medio (ECM(.)),

$$ECM\ (S[i,j]) = \frac{\frac{1}{2}\sum_{k=1}^{P} (y_k - S[i,j](x_k))^2}{N} \qquad (17)$$

donde S[i,j] es el modelo difuso en el que el Sistema de Inferencia se construye utilizando el operador de conjunción C_i, y la Interfaz de Defuzzificación que emplea el método de defuzzificación D_j. Esta medida precisa del empleo de un conjunto de datos de evaluación constituido por P pares de datos numéricos $Z_k=(x_k, y_k)$, k=1,...,P, siendo x_k los valores de las variables de entrada e y_k los correspondientes a sus salidas asociadas.

En realidad el ECM(.) empleado es la media aritmética de los ECM(.) obtenidos realizando 30 procesos de evolución con cada modelo: 6 semillas diferentes para el generador de números aleatorios del algoritmo evolutivo por 5 particiones de datos con validación cruzada de orden 5.

Con objeto de mostrar con claridad el incremento en la precisión de los modelos adaptativos, se ha empleado el porcentaje de mejora PM(.) (18) presentado por el ECM,

$$PM(i,j) = 100\ \left(1 - \frac{ECM(S(i,j))}{ECM(S_{NA}(i))}\right) \qquad (18)$$

donde $S_{NA}(.)$ es el modelo de referencia no adaptativo con el operador de conjunción C_i, con sus parámetros inicializados al valor comentado con anterioridad.

3.4 RESULTADOS EXPERIMENTALES Y ANÁLISIS

En esta sección se muestran los resultados y análisis de la experimentación realizada. En primer lugar se muestran los resultados obtenidos para la conjunción y la defuzzificación adaptativas, y posteriormente, los de utilizar esta técnica combinada con una técnica muy conocida para aumentar la precisión de los MD lingüísticos: el ajuste de las funciones de pertenencia.

3.4.1 CONJUNCIÓN Y DEFUZZIFICACIÓN ADAPTATIVAS

Las siguientes tablas muestran los resultados de la experimentación:

Tabla 2: Errores cuadráticos medios de los modelos iniciales con los parámetros inicializados a los valores descritos en la Sección 3.3.

Tabla 3: Errores cuadráticos medios obtenidos por los modelos evolucionados.

Tabla 4: Porcentajes de mejora tras el proceso evolutivo, es decir, de la Tabla 3 respecto de la 2.

Tabla 2: Error Cuadrático Medio inicial

ECM inicial	Entrenamiento	Prueba
Dubois	56135,74	56359,42
Dombi	55418,16	55531,72
Frank	80441,20	80597,13

Tabla 3: Error Cuadrático Medio obtenido

ECM	Entrenamiento		Prueba	
	D_3	D_5	D_3	D_5
Dubois	21757,02	21985,91	25545,22	24974,17
Dombi	22334,75	25915,42	25264,46	29228,98
Frank	28673,98	23145,21	33535,45	26887,47

Tabla 4: Porcentaje de Mejora de los modelos adaptativos

PM	Entrenamiento		Prueba	
	D_3	D_5	D_3	D_5
Dubois	61,24	60,83	54,68	55,69
Dombi	59,70	53,24	54,51	47,38
Frank	64,35	71,23	58,39	66,64

Analizando las tablas 3 y 4, los modelos que utilizan el sistema de inferencia y la interfaz de defuzzificación adaptativos mejoran en todos los casos, tanto en entrenamiento como en prueba la precisión con respecto al modelo de partida no adaptativo. Los mejores resultados corresponden al caso de la t-norma parametrizada de Dubois, seguramente debido al menor espacio de búsqueda que se precisa utilizar para ella.

Los porcentajes de mejora que se observan en la Tabla 4 muestran un importante progreso en los modelos adaptativos, siendo los mejores aquellos que inicialmente presentaba mayores errores: t-norma de Frank.

En relación al método de defuzzificación, en el caso de la t-norma de Dubois no se aprecia diferencia entre el empleo del basado en funcional potencia (D_3) y producto (D_5). Sin embargo, el Sistema de Inferencia Adaptativo basado en Dombi obtiene las mejores cifras con D_3 mientras que el de Frank lo hace con D_5.

Es posible afirmar que la t-norma de Dubois se ha mostrado como la más adecuada ya que es la más sencilla computacionalmente, el intervalo de búsqueda que debe emplear el algoritmo evolutivo es el más pequeño, y obtiene los mejores resultados independientemente de emplearla combinada con un método de defuzzificación adaptativo basado en funcional producto o potencia.

3.4.2 COMBINACIÓN DE LA CONJUNCIÓN Y DEFUZZIFICACIÓN ADAPTATIVAS CON EL AJUSTE DE LAS FUNCIONES DE PERTENENCIA

En esta sección se presentan los resultados de combinar la técnica de la conjunción y defuzzificación adaptativas estudiada en este trabajo, con el empleo de una de las técnicas más conocidas y que ofrecen mejores resultados para mejorar la precisión de los MD lingüísticos: la adaptación de las funciones de pertenencia de los conjuntos difusos de las particiones de las variables implicadas, contenidas en la base de datos.

Para combinar ambas técnicas, el cromosoma del algoritmo evolutivo CHC ha sido ampliado con los genes correspondientes a la descripción de los conjuntos difusos de las particiones de las variables, de modo que la evolución de la base de datos y los parámetros del sistema se ha realizado conjuntamente. El número de evaluaciones empleadas en el algoritmo evolutivo también se ha elevado hasta 60000.

Las siguientes tablas muestran los resultados obtenidos en la experimentación:

Tabla 5: Errores cuadráticos medios obtenidos adaptando sólo las funciones de pertenencia. Para ello, el parámetro de las t-normas se ha fijado a los valores de inicialización descritos en la Sección 3.3. El algoritmo CHC ha empleado 30000 evaluaciones, y se inicializó un cromosoma con los valores correspondientes a la base de datos inicial, y el resto de modo aleatorio.

Tabla 6: Errores cuadráticos medios obtenidos por los modelos evolutivos conjuntos.

Tabla 7: Porcentajes de mejora del modelo evolutivo conjunto: mejora de la Tabla 6 respecto de la Tabla 2.

Tabla 5: Error Cuadrático medio obtenido empleando sólo la adaptación de las funciones de pertenencia

	Entrenamiento	Prueba
Dubois	14320,62	17460,91
Dombi	14004,49	17933,84
Frank	15304,25	17921,22

Tabla 6: Error Cuadrático Medio adaptando al mismo tiempo conjunción, defuzz. y funciones de pertenencia

ECM	Entrenamiento		Prueba	
	D_3	D_5	D_3	D_5
Dubois	7927,37	6149,11	11367,18	9580,60
Dombi	7060,52	7247,97	10156,85	10497,60
Frank	8001,53	7292,56	11255,34	10405,02

Tabla 7: Porcentajes de Mejora del modelo adap. conjunto

PM	Entrenamiento		Prueba	
	D_3	D_5	D_3	D_5
Dubois	85,88	89,05	79,83	83,00
Dombi	87,26	86,92	81,71	81,10
Frank	90,05	90,93	86,04	87,09

Observando la Tabla 5, y en comparación con la Tabla 3, se aprecia que la tradicional adaptación de las funciones de pertenencia es un método que aporta mayor incremento en la precisión que la adaptación de los operadores. Sin embargo, la combinación de ambas técnicas, Tabla 6, ofrece unos resultados notablemente mejores aún, siendo el error cuadrático medio en entrenamiento

aproximadamente la mitad que empleando sólo la adaptación de las funciones de pertenencia. Los porcentajes de mejora, Tabla 7, con respecto a la base de conocimiento original obtenida por el método de Wang y Mendel son de un orden de magnitud muy importante.

Analizando estos resultados podemos afirmar que la adaptación de los operadores de conjunción y defuzzificación no es un método que compita en precisión con otros métodos sino que se trata de un complemento que puede aportar mejoras finales muy importantes.

4. CONCLUSIONES

En el ámbito del compromiso entre precisión e interpretabilidad de los sistemas difusos lingüísticos, este trabajo trata sobre el empleo de modelos difusos con dos de sus componentes parametrizadas: el Sistema de Inferencia y la Interfaz de Defuzzificación, empleando inferencia y defuzzificación locales, es decir, sendos parámetros para cada regla.

El resultado se ha mostrado como beneficioso, concluyéndose pues que este método constituye una alternativa que mejora a otras similares que sólo parametrizaban una componente. Estas ventajas no compiten con otros métodos tradicionales como la adaptación de las funciones de pertenencia sino que se complementan como se ha mostrado en la experimentación llevada a cabo.

Referencias

[1] I. Batyrshin, O. Kaynak. Fuzzy Modeling Based on Generalized Conjunction Operators. *IEEE Transaction on Fuzzy Systems,* Vol. 10, nº 5, Págs 678-683, 2003.

[2] J. Casillas, O. Cordón, F. Herrera, L. Magdalena. (Eds.) Interpretability issues in fuzzy modelling, *Srpinger-Verlag*. 2003.

[3] J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.) Accuary improvements in linguistic fuzzy modeling, *Springer-Verlag*. 2003

[4] J.S. Cho, D.J. Park. Novel fuzzy logic control based on weighting of partially inconsistent rules using neural network. *Journal of Intelligent and Fuzzy Systems,* Vol. 8, Pág. 99-100, 2000.

[5] O. Cordón , M.J. del Jesus, F. Herrera. Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods. *International Journal of Intelligent Systems*, Vol. 13, Pág. 1025-1053, 1998.

[6] O. Cordón, F. Herrera, A. Peregrín. Characterisation of Implication Operators in Fuzzy Rule Based Systems from Basic Properties. *Proc. of the 9th Congr. sobre Tecnologías y Lógica Fuzzy,* Pág. 163-166, 1999.

[7] O. Cordón, F. Herrera. A. Peregrín. Searching for Basic Properties Obtaining Robust Implication Operators in Fuzzy Control. *Fuzzy Sets and* Systems**,** Vol. 111, Pág. 237-251, 2000.

[8] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena. Genetic Fuzzy Systems: evolutionary tuning and learning of knowledge bases, *World Scientific*. 2001.

[9] O. Cordón, F. Herrera, F. Márquez, A. Peregrín. A Study on the Evolutionary Adaptive Defuzzification Methods in Fuzzy Modelling. Internacional *Journal of Hybrid Intelligent Systems*,Vol.1,nº1, Pág. 36-48, 2004.

[10] L.J. Eshelman. The CHC Adaptive search algorithm: How to have safe search when engaging in Nontraditional genetic recombination. *Foundations of Genetic Algorithms-1*, Rawlins,G.J.E.(*Eds*), Pág.265-283, 1991.

[11] A.O. Esogbue, Q. Song, W.E. Hearnes II. Defuzzification filters and applications to power system stabilization problems**.** *Journal of Mathematical Analysis and Applications*,Vol.251, Pág. 406-432, 2000

[12] M. Gupta, J. Qi. Theory of T-norms and Fuzzy Inference Methods. *Fuzzy Sets and Systems*, Vol. 40, Pág. 431-450, 1991.

[13] F. Herrera, M. Lozano J.L. Verdegay. Dynamic and Heuristic Fuzzy Connectives-Based Crossover Operators for Controlling the Diversity and Convergence of Real-Coded Genetic Algoritms. *International Journal of Intelligent Systems*, Vol. 11, Pág. 1013-1040, 1996.

[14] F. Herrera, F. Márquez, A. Peregrín. Genetic Adaptation of Rule Connectives and Conjunction Operators in Fuzzy Rule Based System: An Experimental Comparative. *Third International Conference of the European Society for Fuzzy Logic and Technology Study*. Pág. 100-104, 2003.

[15] T. Jiang, Y. Li. Generalized defuzzification strategies and their parameter learning procedures. *IEEE Transactions on Fuzzy Systems*, Vol. 4, nº 1, Pág. 64-71, 1996.

[16] A. Kandel, M. Friedman. Defuzzification using most typical values. *IEEE Transactions on Systems, Man, and Cybernetcis*, Vol. 28, nº 6, Pág. 901-906, 1998.

[17] H. Kiendl, Non - translation - invariant defuzzification. *Proceedings of the 6th FUZZ-IEEE'97*, Pág. 737-741, 1997.

[18] D. Kim. An Accurate COG Defuzzifier Design by the Co-adaptation of Learning and Evolution, *Proc. 9th IEEE International Conf. on Fuzzy Systems,* Pág. 741-747, 2000.

[19] D. Kim, Y. Choi, S. Lee. An acuarate COG defuzzifier design using Lamarckian coadaptation of learning and evolution. *Fuzzy Sets and Systems*, Vol. 130, nº 2, Pág. 207-225, 2002.

[20] B.D. Liu, C.Y. Chen C-Y., J.Y. Tsao. Design of adaptative fuzzy logic controller based on linguistic-hedge concepts and genetic algorithtms. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 31, nº 1, Pág. 32-53, 2001.

[21] T. Whalen. Parametrized R-implications. *Fuzzy Sets and Systems*, Vol. 134, Pág. 231-281, 2003.

[22] L.X. Wang, J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 22, nº 6, Pág. 1414-1427, 1992.

CONTROLADOR JERÁRQUICO EN SISTEMAS FOTOVOLTAICOS AUTÓNOMOS

S. G. Galán, J. C. Bago, J. Aguilera
Investigación y desarrollo en Energía Solar y Automática (IDEA).
Dpto. de Electrónica.
Universidad de Jaén.
E.U.P. de Linares Alfonso X El Sabio, 28.
23700 Linares (Jaén) Spain
E-mail: {sgalan, jcbago, aguilera}@ujaen.es

Juan Ramón Velasco Pérez
Dpto. de Automática
Universidad de Alcalá.
Escuela Politécnica.
Crta. N-II, Km 31,600
28871 Alcalá de Henares
Madrid Spain
E-mail: juanra@aut.uah.es

Luis Magdalena Layos
Dpto. Matemática Aplicada a las Tecnologías de la Información
Universidad Politécnica de Madrid.
E.T.S.I. de Telecomunicación.
Ciudad Universitaria s/n, 28040
Madrid Spain.
E-mail: llayos@mat.upm.es

Resumen

Este documento presenta una aplicación práctica de control borroso en el campo de los sistemas fotovoltaicos autónomos. Previamente se realizará una introducción a la problemática existente en estos sistemas en cuanto al funcionamiento de los reguladores de carga. Posteriormente, se presenta una implementación de un regulador borroso utilizado. Dado que dicho controlador está basado en una estructura jerárquica, este documento presenta cada una de las bases de conocimiento utilizadas. Para finalizar, se muestra los resultados de operación obtenidos con dicho controlador en una instalación real, comparando los resultados con el funcionamiento de un regulador comercial de alta gama utilizado en otra instalación real de similares características a la anterior.

Palabras Clave: Aplicación, Regulador de carga, Sistemas fotovoltaicos autónomos.

1 INTRODUCCIÓN

La operación de los Sistemas Fotovoltaicos Autónomos depende no sólo de la calidad individual de los elementos que los componen, sino también de la interacción entre los mismos dentro del sistema [1]. En este sentido el controlador de carga adquiere una importancia especial. La funcionalidad del controlador de carga se centra fundamentalmente en la protección del sistema de acumulación (constituido por baterías de plomo-ácido).

A pesar de una continua evolución en las estrategias de control de carga de las baterías sufrida por el funcionamiento de los reguladores de carga, el funcionamiento de los mismos en la actualidad, en la práctica presenta un comportamiento manifiestamente mejorable.

Según estudios anteriores [2][4][6][8], se ha podido comprobar como la utilización de un controlador borroso [3][7], permite adaptar las funciones del controlador de carga al funcionamiento de las baterías utilizadas en el sistema de acumulación.

En este documento, se describe un controlador borroso de carga. Dicho controlador se ha desarrollado a partir de otros trabajos [2][4][6][8] previos realizados con el objeto de verificar la viabilidad del uso de un controlador de carga borroso en sistemas fotovoltaicos autónomos.

Para ello se ha implementado un regulador borroso mejorado, que incorpora tres bases de conocimiento dispuestas de forma jerárquica, con el objetivo de permitir una estructuración del conocimiento existente sobre el comportamiento de los sistemas de acumulación y facilitar, por lo tanto, la incorporación de reglas borrosas en las distintas bases de conocimiento que incluidas en el regulador. Éste regulador será utilizado en posteriores trabajos como punto de partida en sistemas evolutivos con el objeto de obtener otro regulador mediante aprendizaje automático.

Este trabajo se estructura de la siguiente forma: En el apartado 2, se realizará una breve introducción a los sistemas fotovoltaicos autónomos, en la cual se presentará al controlador de carga como la parte más crítica en el funcionamiento global del sistema. En el apartado 3, se presentará la estructura del controlador borroso que está siendo utilizado en la actualidad y cuyo funcionamiento se pretende mejorar. En el apartado 4, se presentan resultados de operación obtenidos al utilizar el regulador presentado en una instalación real, y una comparación con

los obtenidos por un regulador comercial de gama alta en una instalación de similares características. Finalmente el apartado 5, se presentan algunas conclusiones y acciones futuras derivadas del presente trabajo.

2 SISTEMAS FOTOVOLTAICOS AUTÓNOMOS.

Las posibles aplicaciones de la energía solar fotovoltaica son muy numerosas, pero en general estas aplicaciones pueden englobarse en dos tipos de sistemas fotovoltaicos, los sistemas conectados a la red eléctrica y los sistemas autónomos [10]. En los primeros, la energía generada y no consumida por la instalación es inyectada a la línea de distribución eléctrica existente, quedando disponible para otros consumidores.

Cuando se utiliza un sistema fotovoltaico autónomo, la energía que no se consume, puede ser almacenada por un sistema de acumulación, para ser consumida con posterioridad por la propia instalación.

Un sistema fotovoltaico autónomo genérico, puede estar constituido por los siguientes elementos, ver figura 1:

- **Generador Fotovoltaico:** Produce energía eléctrica a partir de la radiación solar incidente.

- **Sistema de acumulación de energía:** La inserción de baterías en los SFA se realiza para adaptar los diferentes ritmos de producción y demanda de energía eléctrica.

- **Controlador de carga:** Su función consiste en proteger al sistema de acumulación y evitar situaciones de comportamiento extremo que puedan dañar las baterías.

- **Inversor:** Estos elementos se utilizan cuando se quiere utilizar cargas que requieran corriente alterna.

- **Carga:** Elementos a los que se pretende alimentar con energía eléctrica.

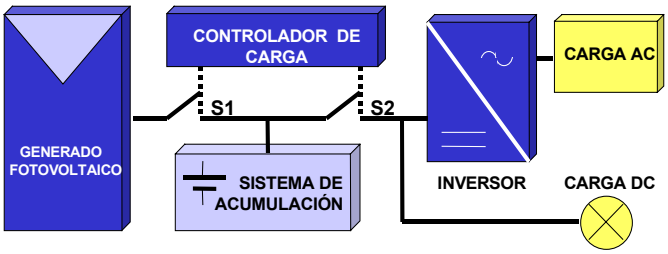


Figura 1: Sistema Fotovoltaico Autónomo.

En las instalaciones de tipo autónomo, la energía eléctrica generada por los módulos fotovoltaicos no sólo alimentará a la carga eléctrica, sino que además, con el exceso de energía generada, se cargarán los acumuladores. Por lo tanto, el sistema controlador de carga debe, por un lado, evitar sobrecargas en el acumulador (debido a una tensión excesiva proporcionada por el generador), y por otro lado, impedir que las baterías se sobredescarguen (debido a una alimentación excesiva a la carga, por parte del sistema de acumulación).

En definitiva, el sistema de control de carga debe proteger al acumulador de situaciones extremas de funcionamiento, independientemente del tamaño de la instalación, del diseño, y de los posibles cambios estacionales en el perfil de consumo, en la radiación solar y en la temperatura de operación. Todo ello buscando como objetivo final alargar la vida útil del sistema acumulador.

Tal y como se ha mencionado con anterioridad, el controlador de carga se presenta como la parte más crítica de toda la instalación. Sin embargo, y a pesar de la aparente simplicidad en el funcionamiento del controlador de carga, los controladores comerciales en existentes en la actualidad no muestran unos requerimientos necesarios de calidad [9].

3 REGULADOR DE CARGA JERÁRQUICO

Resultados anteriores [2] [4] [6] [8], permiten concluir que la utilización de un controlador borroso puede conducir a una mejora en el comportamiento del controlador de carga. Hasta el momento se ha presentado un regulador de carga borroso destinado a proteger a los sistemas de acumulación de sistemas fotovoltaicos autónomos tanto de sobrecargas como de sobredescargas.

En este apartado se presenta un nuevo regulador de carga borroso. Éste nuevo regulador de carga borroso, denominado jerárquico, estará dedicado solamente a la protección del sistema de acumulación frente a sobrecargas.

A pesar de la flexibilidad presentada por un regulador borroso, no es fácil incorporar nuevo conocimiento a través de reglas, ya que las relaciones existentes entre los distintos factores que intervienen no pueden estructurarse normalmente en la base de conocimiento existente..

El regulador de carga borroso propuesto, gracias a su estructura jerárquica, dispone de una gran facilidad para incorporar conocimiento experto. Es, por lo tanto, este controlador de carga borroso, el que será utilizado

posteriormente para intentar conseguir, mediante técnicas de aprendizaje automático [3][5], optimizar su comportamiento. La estructura del controlador de carga borroso jerárquico propuesto puede apreciarse en la figura 2.

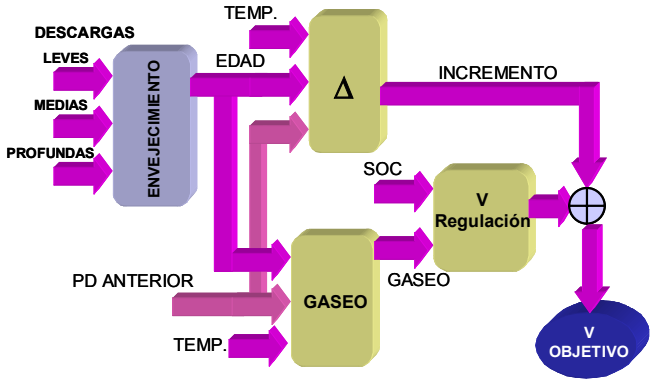


Figura 2: Estructura del controlador de carga borroso Jerárquico.

Este controlador de carga dispone de cuatro módulos, cada uno de ellos con distinta funcionalidad. Así, el módulo de envejecimiento se encarga de estimar la edad del sistema de acumulación en función del número de descargas leves, medias y profundas que haya sufrido la batería. El resto de los módulos, son sistemas borrosos con la siguiente funcionalidad:

Gaseo: Se encarga de estimar el tiempo que debe estar gaseando la batería. Éste tiempo se obtiene en función de la edad de la batería, de la temperatura y de la profundidad de descarga anterior.

Incremento: Se encarga de calcular un incremento en voltios, que debe aplicarse a la tensión objetivo con la que se debe de cargar la batería. Éste incremento se obtiene en función de la edad de la batería, de la temperatura y de la profundidad de descarga anterior.

Regulación: Éste módulo se encarga de establecer el modo de carga de la batería, bien en gaseo (incluyendo plena carga), bien en flotación. Éste modo de carga se obtiene en función del tiempo de gaseo y del estado de carga de la batería.

Como puede apreciarse en la figura 2, la tensión objetivo del sistema de acumulación se obtiene como la suma de la tensión de regulación y el incremento.

A continuación se describirán las tres bases de conocimiento que constituyen el regulador jerárquico.

3.1. Base de conocimiento de gaseo.

Los antecedentes considerados en esta base de conocimiento son las siguientes:

- **Temperatura de la batería** (TEMP), al no poder medir la temperatura interna de la batería se utiliza la temperatura ambiente del recinto donde se ubica el sistema de acumulación. El universo de discurso utilizado es de el rango [0,1] que representa dominio de 0º a 50º.

- **Edad de la batería** (EDAD), calculada en el módulo de envejecimiento según la siguiente expresión:

$$EDAD = \frac{\text{Leves}}{1100} + \frac{Medias}{400} + \frac{\text{Pr}ofundas}{200}$$

  donde Leves/Medias/Profundas representa el número de descargas leves/medias/profundas que ha sufrido el sistema de acumulación.

- **Profundidad de descarga anterior** (PDDANTER), calculada haciendo un balance de carga mediante estimaciones realizadas de la capacidad suministrada y almacenada.

- El consecuente perseguido con esta base de conocimiento es el **tiempo de gaseo** (TPGASEO) objetivo, es decir el tiempo que se estima para que el sistema de acumulación esté en la fase de gaseo. El universo de discurso utilizado es de el rango [0,1] que representa dominio de 5 a 60 minutos.

Las funciones de pertenencia que se han utilizado para cada uno de los antecedentes y para el consecuente se representan la figura 3.

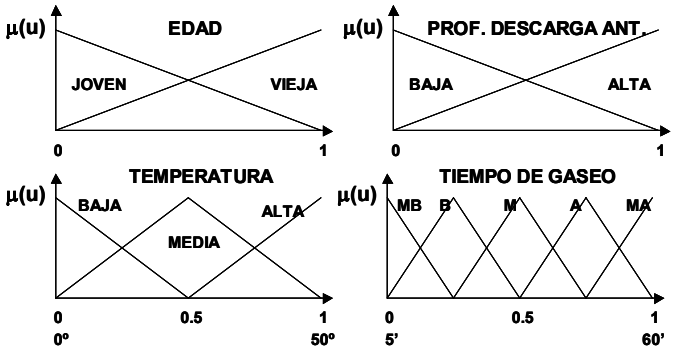


Figura 3: Funciones de pertenencia para las variables de entrada y salida utilizadas en la base de conocimiento de gaseo.

Las reglas utilizadas en la base de conocimiento de gaseo quedan representadas en la tabla 1.

Tabla 1: Reglas de la base de conocimiento de gaseo.

BASE DE CONOCIMIENTO DE GASEO		
	ANTECEDENTES	**TIEMPO GASEO**
G1	**IF** "TEMP=BAJA" **and** "EDAD=JOVEN" **and** "PDDANTER=BAJA"	MEDIO
G2	**IF** "TEMP=BAJA" **and** "EDAD=JOVEN" **and** "PDDANTER=ALTA"	ALTO
G3	**IF** "TEMP=BAJA" **and** "EDAD=VIEJA" **and** "PDDANTER=BAJA"	ALTO
G4	**IF** "TEMP=BAJA" **and** "EDAD=VIEJA" **and** "PDDANTER= ALTA"	MUY ALTO
G5	**IF** "TEMP=MEDIA" **and** "EDAD=JOVEN" **and** "PDDANTER=BAJA"	BAJO
G6	**IF** "TEMP=MEDIA" **and** "EDAD=JOVEN" **and** "PDDANTER= ALTA"	MEDIO
G7	**IF** "TEMP=MEDIA" **and** "EDAD=VIEJA" **and** "PDDANTER=BAJA"	MEDIO
G8	**IF** "TEMP=MEDIA" **and** "EDAD=VIEJA" **and** "PDDANTER= ALTA"	ALTO
G9	**IF** "TEMP=ALTA" **and** "EDAD=JOVEN" **and** "PDDANTER=BAJA"	MUY BAJO
G10	**IF** "TEMP=ALTA" **and** "EDAD=JOVEN" **and** "PDDANTER= ALTA"	BAJO
G11	**IF** "TEMP=ALTA" **and** "EDAD=VIEJA" **and** "PDDANTER=BAJA"	BAJO
G12	**IF** "TEMP=ALTA" **and** "EDAD=VIEJA" **and** "PDDANTER= ALTA"	MEDIO

3.2. Base de conocimiento de incremento.

Para la base de conocimiento se han utilizado los mismos antecedentes que para la base de conocimiento de Gaseo. Lógicamente el consecuente es distinto, en esta ocasión se pretende obtener un valor para el **Incremento** (INCRE). Como se ha comentado anteriormente, este incremento pretende ser una cantidad en voltios que hay que sumarle a la tensión de regulación calculada. El universo de discurso utilizado para el consecuente es de el rango [0,1] que representa dominio de [-0,6, 0,6] voltios.

Las funciones de pertenencia que se han utilizado para cada uno de los antecedentes y para el consecuente se representan la figura 4.

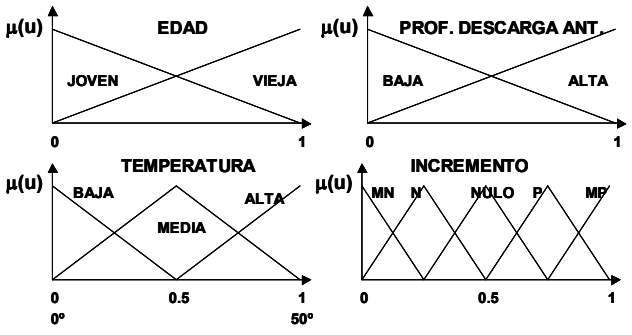


Figura 4: Funciones de pertenencia para las variables de entrada y salida utilizadas en la base de conocimiento de incremento.

En esta ocasión, las reglas utilizadas en la base de conocimiento de Gaseo quedan representadas en la tabla 2.

Tabla 2: Reglas de la base de conocimiento de incremento.

BASE DE CONOCIMIENTO DE INCREMENTO		
	ANTECEDENTES	**INCRE.**
I1	**IF** "TEMP=BAJA" **and** "EDAD=JOVEN" **and** "PDDANTER=BAJA"	POSITIVO
I2	**IF** "TEMP=BAJA" **and** "EDAD=JOVEN" **and** "PDDANTER=ALTA"	MUY POSITIVO
I3	**IF** "TEMP=BAJA" **and** "EDAD=VIEJA" **and** "PDDANTER=BAJA"	MUY POSITIVO
I4	**IF** "TEMP=BAJA" **and** "EDAD=VIEJA" **and** "PDDANTER= ALTA"	MUY POSITIVO
I5	**IF** "TEMP=MEDIA" **and** "EDAD=JOVEN" **and** "PDDANTER=BAJA"	NULO
I6	**IF** "TEMP=MEDIA" **and** "EDAD=JOVEN" **and** "PDDANTER= ALTA"	POSITIVO
I7	**IF** "TEMP=MEDIA" **and** "EDAD=VIEJA" **and** "PDDANTER=BAJA"	POSITIVO
I8	**IF** "TEMP=MEDIA" **and** "EDAD=VIEJA" **and** "PDDANTER= ALTA"	MUY POSITIVO
I9	**IF** "TEMP=ALTA" **and** "EDAD=JOVEN" **and** "PDDANTER=BAJA"	NEGATIVO
I10	**IF** "TEMP=ALTA" **and** "EDAD=JOVEN" **and** "PDDANTER= ALTA"	NULO
I11	**IF** "TEMP=ALTA" **and** "EDAD=VIEJA" **and** "PDDANTER=BAJA"	NULO
I12	**IF** "TEMP=ALTA" **and** "EDAD=VIEJA" **and** "PDDANTER= ALTA"	POSITIVO

3.3. Base de conocimiento de regulación.

Ahora las variables de entrada son el estado de carga del sistema de acumulación (SOC) y el tiempo de gaseo que lleva el sistema de acumulación (GASEO). El consecuente que se utiliza en esta base de conocimiento es el estado en el que debe de trabajar el sistema de acumulación, en gaseo o en flotación (VREG). El universo de discurso utilizado para los antecedentes y el consecuente es de el rango [0,1].

Las funciones de pertenencia que se han utilizado para cada uno de los antecedentes y para el consecuente se representan la figura 5.

Las reglas utilizadas en la base de conocimiento de regulación quedan representadas en la tabla 3.

Tabla 3: Reglas de la base de conocimiento de regulación.

BASE DE CONOCIMIENTO DE REGULACIÓN		
	ANTECEDENTES	**VREG**
R1	**IF** "SOC=BAJO" **and** "GASEO=BAJO"	GASEO
R2	**IF** "SOC=BAJO" **and** "GASEO=ALTO"	FLOTACIÓN
R3	**IF** "SOC=ALTO" **and** "GASEO=BAJO"	FLOTACIÓN
R4	**IF** "SOC=ALTO" **and** "GASEO=ALTO"	FLOTACIÓN

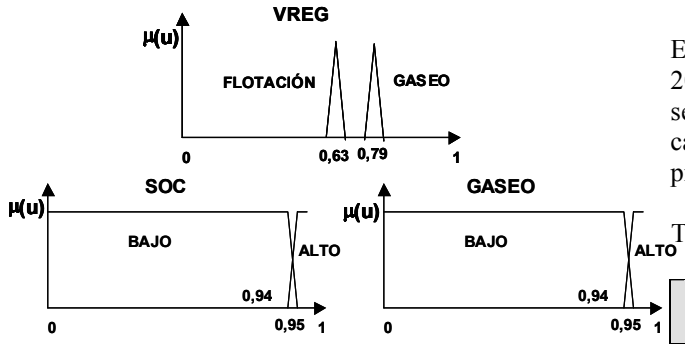


Figura 5: Funciones de pertenencia para las variables de entrada y salida utilizadas en la base de conocimiento de regulación.

4. RESULTADOS DE OPERACIÓN.

Para poder valorar el comportamiento del controlador de carga presentado, se han instalado dos sistemas fotovoltaicos autónomos con las mismas características, uno con el controlador diseñado y otra con un controlador de carga comercial de gama alta. Las características principales de dichas instalaciones son las siguientes:

- **Generador fotovoltaico:** Está formado por un panel fotovoltaico de la marca ISOFOTON modelo I-106. Este módulo esta constituido por 36 células en serie y 2 en paralelo, siendo las siguientes las características más destacables:
  - Potencia nominal de 106 W±5%
  - medidos a 1000 W/m^2 y 25ºC.
  - Isc =6,54 A
  - Voc = 21,6 V
  - 36 células en serie y 2 en paralelo
  - TONC=47ºC, a 800 W/m^2; 1,5 MA y 20ºC
- **Sistema de acumulación:** el sistema de acumulación utilizado consiste una batería de plomo-ácido de 72Ah y de 12 voltios de tensión nominal.
- **Carga:** la carga consiste en una lámpara halógena de 3Ah de consumo.

El periodo de pruebas consiste en someter a los sistemas de acumulación a un continuo ciclado. Así, durante periodos de ausencia de radiación solar, se conectarán las cargas (halógenos) de forma que se producirá una descarga de los sistemas de acumulación. Por otra parte, durante los periodos de existencia de radiación solar y en ausencia de cargas conectadas se producirá el cargado de los sistemas de acumulación. Evidentemente, estos periodos de carga de estarán supervisados por los controladores de carga de cada uno de los sistemas fotovoltaicos autónomos.

El proceso de ciclado comenzó el día 4 de diciembre de 2003, y finalizó el día 7 de enero de 2004. A continuación se presenta una tabla con un resumen de los datos de capacidad medidos al inicial y al final de periodo de pruebas

Tabla 4: Capacidades iniciales y finales de los sistemas de acumulación utilizados.

SISTEMA	Capacidad (AH)		% Capacidad perdida
	Inicial	Final	
JERARQUICO	50,82	26,94	46,98
COMERCIAL	55,08	26,40	52,07

5. CONCLUSIONES Y ACCIONES FUTURAS.

En este documento se ha presentado un controlador de carga borroso para sistemas fotovoltaicos autónomos. Dicho controlador presenta una estructura jerárquica que facilita la incorporación de nuevo conocimiento, ya sea a través de un experto o de un sistema de aprendizaje automática. El controlador presentado presenta un comportamiento similar al de un controlador de carga comercial de gama alta.

El controlador de carga borroso presentado ha sido utilizado como punto de partida de un sistema de aprendizaje automático con el objetivo de optimizar el comportamiento del mismo.

Referencias

[1] R. Kaiser, A. Armbruster, G. Bopp, M. Rehm, J. Zuckschwerdt. "Control Strategies for Photovoltaic Systems". EUROSUN-96, pp 732-738. (1996).

[2] J.C. Bago, S.G. Galán, J. Aguilera, J.R. Velasco, L. Magdalena. "Regulador borroso para sistemas fotovoltaicos autónomos. Estructura del regulador." XI Congreso Ibérico e VI Iberoamericano de Energía Solar. Algarve. Portugal. Septiembre de 2002.

[3] Cordón, O., Herrera, F., Hoffmann, F., Magdalena, L.."Genetic Fuzzy Systems. Ed. World Scientific. 2001.

[4] Bago, J.C. Galán, S.G. Aguilera, J. Velasco, J. R. Magdalena, L. "Aplicación de la lógica borrosa a los reguladores de carga en sistemas fotovoltaicos autónomos". ESTYLF2002, pp 65-70, Septiembre, 2002.

[5] Goldberg, D.E. "Genetic algorithms in search, optimization, and machine learning". Adisson Wesley, 1989.

[6] Galán, S.G. Bago, J.C. Aguilera, J. Magdalena, L. "Fuzzy charge controller for stand-alone photovoltaic systems". EUSFLAT 01, pp160-163, 2001.

[7] Pedrycz, W. “Fuzzy Control and Fuzzy Systems”. Wiley, 1989.
[8] J.C. Bago, S.G. Galán, J. Aguilera, J.R. Velasco, L. Magdalena. “Fuzzy Charge Applications in Stand-Alone Photovoltaic Systems” Mathware & Soft Computing. Vol .IX . n. 1 (2002). Pp 85-105.
[9] Preiser, K. J. Kuhmann, O. Parodi. “*Quality Issues for Solar Home Systems”,* 13th European Photovoltaic Solar Energy Conference. Nice, France, pp 510-513. (1995).
[10] Lorenzo, E. “Electricidad solar”. Ingeniería de los sistemas fotovoltaicos. Editorial Progensa. Sevilla 1994.

CONTROLADOR BORROSO GENÉTICO EN SISTEMAS FOTOVOLTAICOS AUTÓNOMOS

Joaquin Cañada Bago
Jorge Aguilera Tejero
Investigación y desarrollo en Energía Solar y Automática (IDEA).
Departamento de Electrónica
C/ Alfonso X El Sabio s/n
23700 Linares (Jaén) Spain
{jcbago,aguilera}@ujaen.es

Juan Ramón Velasco Pérez
Dpto. de Automática
Universidad de Alcalá.
Escuela Politécnica.
Crta. N-II, Km 31,600
28871 Alcalá de Henares
Madrid Spain
juanra@aut.uah.es

Luis Magdalena Layos
Dpto. Matemática Aplicada a las Tecnologías de la Información
Universidad Politécnica de Madrid.
E.T.S.I. de Telecomunicación.
Ciudad Universitaria s/n, 28040
Madrid Spain.
llayos@mat.upm.es

Resumen

Este trabajo presenta la aplicación de un sistema de evaluación de conocimiento, utilizando un sistema borroso genético, al campo de las energías renovables, en concreto a los sistemas fotovoltaicos autónomos. Partiendo de un controlador borroso jerárquico [Gala-04], se ha diseñado un controlador borroso genético donde la evaluación se realiza mediante un sistema basado en conocimiento. El conocimiento obtenido en diversas pruebas ha sido introducido en instalaciones fotovoltaicas autónomas reales, donde se han comparado los comportamientos y los resultados de diversos controladores.

Palabras Clave: Evaluación de conocimiento, sistemas borroso jerárquicos, sistemas evolutivos, aplicación en sistemas fotovoltaicos autónomos reales.

1 INTRODUCCIÓN

Los sistemas fotovoltaicos autónomos SFA [Lore-94] son sistemas de generación de energía eléctrica a partir de la energía solar incidente sin disponer de conexión a la red de distribución eléctrica. El control de los SFA no es nada preciso, por lo que la lógica borrosa se adapta perfectamente al control de los procesos, como son los de control de carga y descarga de los acumuladores de plomo ácido frecuentemente utilizados en estos sistemas.

Adaptado a los SFA, se ha diseñado un sistema borroso, diseñado con estructura jerárquica[Gala-04], con los objetivos de incorporar el conocimiento actual y que los sistemas borrosos sean sencillos, con pocas variables y reglas, que permita en un futuro su incorporación en un sistema embebido, diseñado con un microcontrolador con recursos y capacidad de proceso de información muy limitados.

A partir del sistema jerárquico, este trabajo presenta un sistema evolutivo, utilizando algoritmos genéticos, que integra un sistema de evaluación basado en conocimiento, que ha permitdo adquirir nuevo conocimiento de control adaptado a los SFA.

Posteriormente se ha analizado un conjunto de individuos generados, conocimiento adquirido con el sistema borroso genético, y se ha seleccionado un individuo para controlar un sistema fotovoltaico real durante un periodo de tiempo significativo.

Finalmente se han comparado las prestaciones del individuo generado con las de otros tres controladores en las instalaciones reales existentes.

2 SISTEMA BORROSO GENÉTICO

A partir del controlador jerárquico descrito en [Gala-04], se ha diseñado un controlador borroso genético, mostrado en la figura 1, que integra un sistema evolutivo basado en algoritmos genéticos y un sistema de evaluación basado en conocimiento. De esta forma el esquema evolutivo propondrá modificaciones a las bases de conocimiento, que serán incorporadas al sistema simulado durante un periodo de tiempo y evaluadas mediante el sistema de evaluación.

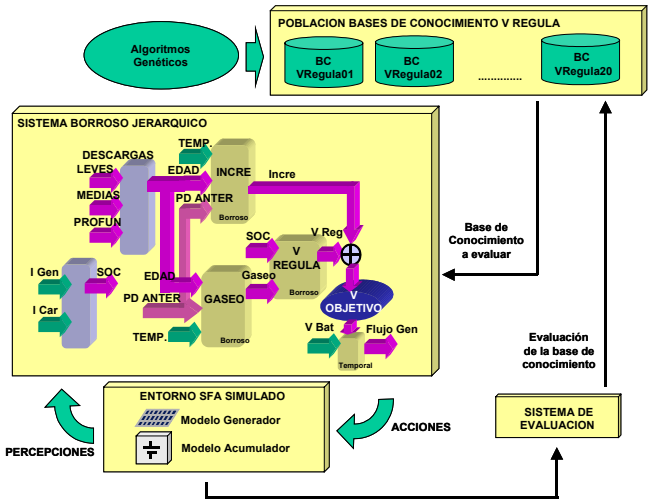


Figura 1. Esquema del sistema borroso genético.

El esquema evolutivo puede generar conocimiento que se adapte correctamente al entorno, al sistema que controlan. Sin embargo, es factible que se genere conocimiento mal adaptado al sistema controlado. Es este caso se producirá un control defectuoso que puede llegar a dañar irreparablemente los acumuladores mediante situaciones de sobrecarga o sobredescarga excesiva. Por este motivo es necesario evaluar el nuevo conocimiento en un sistema simulado que permita verificar, mediante su evaluación, su grado de bondad. En una segunda fase, el conocimiento mejor adaptado, o mejor evaluado, se incorporará a los sistemas reales con objeto de verificar su grado de bondad.

La figura 1 muestra el esquema del controlador borroso genético basado en el enfoque de Pittsburg, donde se pueden apreciar los siguientes bloques:

- El sistema a controlar: sistema fotovoltaico autónomo, [Gala-04] simulado por medio de radiación solar y temperatura de año meteorológico típico, modelo del generador fotovoltaico, modelo del acumulador y simulación de cargas de consumo.
- Controlador lógico borroso que consta del controlador lógico borroso jerárquico de generación y del controlador lógico borroso de consumo. El controlador jerárquico está basado en los sistemas borrosos de tensión de regulación, incremento de tensión y de tiempo de gaseo.
- Población de bases de conocimiento compuesta por un conjunto de individuos de bases de conocimiento de tensión de regulación.
- Esquema evolutivo que permite evolucionar el conocimiento incorporado en la población de individuos, mediante la utilización de diversos operadores genéticos. Las principales características del esquema utilizado se presentan en el punto 2.1.
- Sistema de evaluación de los individuos generados por la evolución de la población de bases de conocimiento. La evaluación utilizada se encuentra a su vez basada en conocimiento, estando descrita en el punto 2.2.

2.1. ESQUEMA EVOLUTIVO

El esquema evolutivo utilizado en el controlador es algoritmos genéticos aplicados sobre el sistema borroso de tensión de regulación, en el que cada individuo representa una base de conocimiento completa según el enfoque de Pittsburg. Para la evolución del conocimiento se dispone de un conjunto de individuos que representa la población, sobre la que se aplican los operadores genéticos. Los algoritmos genéticos siguen el esquema básico definido en [Gold-89] [Cord-01] basado en los procesos de selección de padres, entrecruzamiento de los genes de los padres, mutación de los genes de los descendientes y sustitución de individuos de la población por los individuos generados.

La selección de los individuos padres se realiza de forma proporcional a la bondad de los individuos de la población, teniendo mayor probabilidad de ser seleccionado como padre aquellos individuos que poseen una evaluación mayor. No obstante es factible que sean seleccionados individuos con menor bondad.

El entrecruzamiento de los individuos padres, bases de conocimiento completas de tensión de regulación, se realiza mediante el cruce de dos conjuntos de reglas procedentes de los padres. El proceso de selección de un conjunto de reglas lleva implícito la inclusión de las variables utilizadas en las reglas así como la definición de los conjuntos borrosos asociados a cada variable. De esta forma, la base de conocimiento descendente se forma con un conjunto de reglas de un ascendente y otro conjunto de reglas correspondiente al otro ascendente.

A los individuos descendientes obtenidos en el proceso de cruzamiento, se le aplican las siguientes mutaciones:

- Mutación de puntos que definen los conjuntos borrosos. Los conjuntos borrosos utilizados son trapecios definidos por cuatro puntos. Este operador modifica la posición de un conjunto de puntos dentro de la base de conocimiento.
- Mutación de conjuntos borrosos en regla. Este operador muta los conjuntos borrosos utilizados en los antecedentes y consecuente de las reglas.
- Mutación de variable en el antecedente de reglas. Mediante esta mutación se altera las variables utilizadas en los antecedentes de las reglas de actuación. Este operador permite que la base de conocimiento generada utilice variables diferentes a las del antecesor.

La sustitución de los individuos de la población por los individuos generados por el esquema evolutivo se realiza mediante elitismo, por el cual son eliminados de la población aquellos individuos con menor bondad.

Las características principales de los algoritmos genéticos utilizados son los siguientes:

Variables Número de variables 12
Individuo Base de conocimiento regulación de tensión.

Codificación del individuo:
Variable: número entero
Punto de conjunto borroso: número real
Conjunto borroso: número entero
Regla: número entero.

Población 20 individuos.

Algoritmos genéticos

Selección

Proporcional a la bondad del padre

Tasa de selección 0,3

Cruzamiento

Cruce de bases de conocimiento: variables, conjuntos borrosos y reglas.

Tasa de cruzamiento de bases 0,5

Mutación

Mutación de puntos de un conjunto borroso

Tasa mutación puntos 0,05

Mutación de conjuntos borrosos en regla

Tasa mutación conjunto borroso 0,05

Mutación de variable antecedente de reglas

Tasa mutación antecedente 0,05

Evaluación

Sistema de evaluación basada en conocimiento, descrito en el punto 2.2.

Sustitución

Elitismo: son eliminados los individuos con menor evaluación

2.2. SISTEMA DE EVALUACIÓN

El sistema de evaluación utilizado por el sistema evolutivo es un sistema basado en conocimiento [Bago-99], que utiliza el conocimiento actual de los procesos de carga y descarga de los acumuladores de plomo ácido y la experiencia de los miembros del grupo *Investigación y desarrollo en energía solar* de la Universidad de Jaén.

Las variables utilizadas en la base de conocimiento de evaluación, cuyas funciones de pertenencia se muestran en la figura 2, son las siguientes:

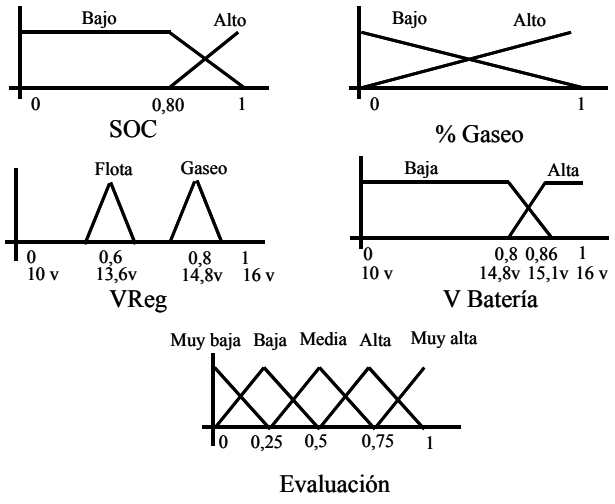


Figura 2 Funciones de pertenencia de las variables utilizadas en el conocimiento de evaluación

SOC Estado de carga de la batería, donde se definen dos conjuntos borrosos, bajo y alto, que permiten representar las descargas leves en la transición entre los dos conjuntos y los estados de descarga superior al 20% de la capacidad del acumulador.

%GASEO Grado de realización de la fase de gaseo, con dos conjuntos borrosos, bajo y alto.

Vbatería Tensión de la batería, con dos conjuntos borrosos que delimitan la máxima tensión admisible en el proceso de carga de la batería.

VReg Tensión objetivo de regulación de la batería, con dos conjuntos borrosos centrados en las tensiones típicas de gaseo y flotación.

Evaluación Evaluación inferida por el sistema, con cinco conjuntos definidos.

Las reglas de actuación de la base de conocimiento de evaluación de individuos generados son las siguientes:

GRUPO EVAL REGLAS 7

EV1 (SOC Bajo) (VRegula Flota) (Eval Baja)
EV2 (SOC Bajo) (VRegula Gaseo) (Eval MuyAlta)
EV3 (SOC Alto) (%Gaseo Bajo) (VRegula Flota) (Eval Baja)
EV4 (SOC Alto) (%Gaseo Bajo) (VRegula Gaseo) (Eval MuyAlta)
EV5 (SOC Alto) (%Gaseo Alto) (VRegula Flota) (Eval MuyAlta)
EV6 (SOC Alto) (%Gaseo Alto) (VRegula Gaseo) (Eval Baja)
EV7 (VBAT Alta) (Eval MuyBaja)

La regla EV1 interviene en la evaluación cuando el estado de carga es bajo, debido a una profundidad de descarga superior al 20% de la capacidad de la batería, y se infiere por parte del individuo a evaluar, una tensión objetivo de flotación. Esta situación es errónea y se evalúa como muy baja.

La regla EV2 evalúa muy positivamente la decisión de inferir una tensión objetivo de gaseo cuando el estado de carga sea bajo (profundidad de descarga superior al 20%).

La regla EV3 penaliza la situación definida por el estado de carga del acumulador alto, porcentaje de tiempo de gaseo bajo e inferencia de tensión objetivo de flotación.

La regla EV4 premia la situación definida por el estado de carga del acumulador alto, porcentaje de tiempo de gaseo bajo e inferencia de tensión objetivo de gaseo.

Cuando el estado de carga es alto y el porcentaje de tiempo en gaseo también es alto, el controlador debe inferir la tensión de flotación. Esta situación es la que refleja la regla EV5 evaluándola muy positivamente.

La regla EV6 evalúa negativamente las situaciones de sobrecarga excesiva en donde una vez terminada completamente la fase de gaseo y el estado de carga de la

batería es alto, el individuo infiere continuar en la fase de gaseo.

La regla EV7 añade una protección adicional penalizando gravemente las tensiones superiores a los 14,8v, ya que provocarían una sobrecarga excesiva que dañaría el acumulador.

La base de conocimiento de evaluación permite inferir la evaluación instantánea del individuo a partir del estado del sistema. No obstante, es necesario evaluar la bondad del individuo en un periodo determinado de tiempo, obteniendo una calificación del individuo sobre el sistema fotovoltaico simulado en este periodo. En esta aplicación se ha utilizado periodos de evaluación de un año, con objeto de considerar las diferencias estacionales existentes en los años meteorológicos, en concreto se ha utilizado el año meteorológico típico de Jaén.

Las evaluaciones anuales son realizadas bajo unas determinadas condiciones iniciales del sistema, que constituyen el punto de partida, así como con determinados condicionantes durante el periodo de evaluación. De esta forma, se evalúa al individuo un número determinado de veces, con diferentes condiciones cada vez, premiando a aquellos que consiguen una evaluación mínima en todas sus evaluaciones, es decir, a los individuos que dispongan de completitud.

En esta aplicación se han utilizado tres variables para modificar las condiciones de cada evaluación: el estado de carga del acumulador con dos valores iniciales (60% y 90%); la edad del acumulador, con dos valores iniciales (0% y 20%) y el patrón de consumo con dos patrones diferentes durante todo el periodo de la evaluación (ambos de consumo constante nocturno de valores 10Ah y 15Ah).

De esta forma la evaluación de cada individuo se realiza 8 veces con diferentes condiciones de estado de carga, edad y patrón cada vez, y por un periodo de un año de duración cada una, 8 años de simulación en total. Para que la base de conocimiento evaluada se considere que dispone de completitud, lo que conlleva la evaluación adicional, es necesario que todas sus evaluaciones diarias sean superiores a 0,65.

La figura 3 presenta la evaluación de un individuo con completitud, mostrando las evaluaciones de cada mes para las ocho situaciones diferentes.

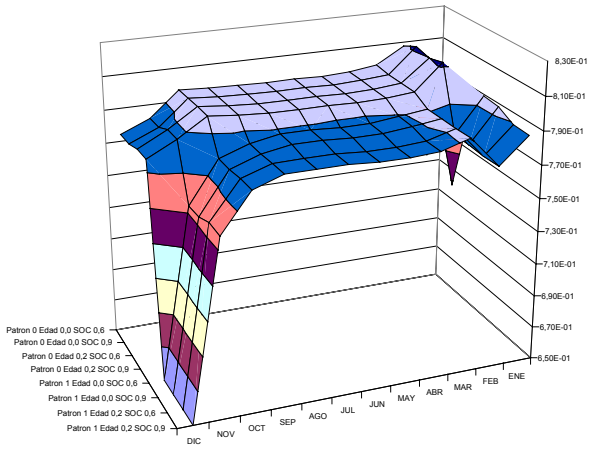


Figura 3 Evaluación de un individuo con 8 condiciones diferentes.

La evaluación mostrada en la gráfica 3 puede considerarse típica, puesto que se presentan algunos de los efectos que se han repetido en un número considerable de individuos generados:

- Los individuos presentan mejor evaluación cuando se está utilizando un patrón de consumo de menor intensidad, que permite que los ciclos de carga de la batería se completen en todas sus fases.
- Mejor evaluación en los meses con mayor irradiancia, que al igual que en caso anterior permite que los procesos de carga se completen.
- Muy ligera menor evaluación de los acumuladores de menor edad inicial, lo que indica que el individuo se encuentra adaptado a la evolución de la edad de la batería, obteniendo evaluaciones similares con diferentes edades.
- No existe variación de evaluación significativa con la variación del SOC inicial de acumulador, indicando que se trata de una condición inicial que no influye en la evaluación final del individuo (para las dos condiciones consideradas SOC 60% y 90%).

3 ADQUISICIÓN DE CONOCIMIENTO

Partiendo de una población inicial de 20 individuos basados en el conocimiento actual, se ha obtenido, mediante el sistema borroso genético, un grupo de más de 80 individuos generados con mejor evaluación que los originales. Para su consecución se han realizado, utilizando el sistema borroso genético, 10 pruebas completas, donde cada prueba consta de la evolución de la población durante 30 generaciones, y la selección de los individuos con puntuación superior a 0,8. Para cada generación los individuos son evaluados con cuatro simulaciones en diferentes condiciones de edad inicial y patrón de consumo energético. La reducción de simulaciones es debida al alto coste computacional de

cada generación y a que las diferentes condiciones de estado de carga inicial SOC no afectan prácticamente a la evaluación final.

Como primer resultado se ha obtenido un conjunto de 81 individuos diferentes con evaluación superior a 0,8. A partir del análisis del conjunto de individuos se pueden resaltar los siguientes puntos:

- Se ha incrementado el número de variables y de reglas de actuación que utilizaba el individuo original[Gala-04], sin embargo sigue siendo un número pequeño que permite su posterior integración en un sistema embebido.
- Las variables utilizadas por el conjunto de individuos generados se presentan en la tabla 1.

Variable		**Número y porcentaje De individuos que la utilizan**	
Tensión de regulación	Vreg	81	100 %
Estado de carga	SOC	81	100 %
Porcentaje de gaseo	%GASEO	64	79 %
Tensión de batería	Vbat	38	46 %
Edad de batería	Edad	18	22 %
Temperatura ambiente	Temp	12	15 %
Profundidad descarga anterior	PDDAnter	12	15 %
Incremento de tensión	Incre	2	2,5 %
Tiempo de gaseo	TPGaseo	2	2,5 %

Tabla 1. Variables utilizadas en el grupo de individuos generado.

- La tensión de regulación es utilizada por todos los individuos debido a que representa la variable de salida del sistema borroso y el sistema evolutivo no actúa sobre ella.
- Es de destacar que existe un grupo de variables, entre las que se encuentran el estado de carga (SOC) y el porcentaje de gaseo (%GASEO), que son utilizadas ampliamente por el conjunto de individuos generados
- La variable tensión de la batería, que no se utiliza en el individuo original, es utilizada por el 46% de los individuos generados.
- Las variables edad de batería, temperatura ambiente y profundidad de descarga anterior son utilizadas en menor grado (22%, 15% y 15% respectivamente), mientras que el incremento de tensión y el tiempo de gaseo solamente se utilizan en el 2,5% de los individuos.

Con respecto a las reglas de actuación de los individuos generados, es difícil realizar un análisis comparativo debido a que el sistema borroso genético evoluciona los conjuntos borrosos asociados a las variables, por lo que la misma regla puede realizar un acción diferente en cada individuo. No obstante, a modo indicativo, el conjunto de individuos utiliza un total de 352 reglas de las que existen 62 reglas diferentes (el individuo original dispone de 4 reglas). A partir del conjunto de reglas de actuación utilizado por los individuos generados, se pueden resaltar los siguientes puntos:

- Existe un grupo de 15 reglas que son ampliamente utilizadas en el conjunto de individuos generados. El resto de reglas tiene escasa presencia.
- Dentro de las primeras 20 reglas de actuación, 14 son generadas mientras que 6 se consideran originales.
- Dentro del grupo de reglas, se han incorporado las variables tensión de batería Vbat, temperatura Temp, edad del acumulador y profundidad de descarga anterior PDDAnter.
- La variable tensión de batería es utilizada para realizar una limitación de la tensión del acumulador en el gaseo del acumulador.
- La temperatura es utilizada para permitir la fase del gaseo en las temperaturas bajas.
- La variable de salida utiliza una tensión intermedia entre el gaseo y la flotación. Las tensiones superiores al gaseo se utilizan en muy pocas ocasiones por individuos con muy poca presencia y las inferiores a la flotación nunca.

La frecuencia de aparición de las reglas en los diferentes individuos generados no implica necesariamente una medida del grado de bondad de cada regla, siendo necesario un estudio individualizado de cada regla de actuación con objeto de realizar su evaluación. En este aspecto se están realizado trabajos como el propuesto en [Gala-99] que permite asignar credibilidad a cada regla, mediante el estudio de la cooperación entre las reglas, consistente en la supresión temporal de la regla, evaluación de la base de conocimiento resultante y comparación con la evaluación de la base original.

4 RESULTADOS EN SISTEMAS REALES

A partir del análisis de los resultados obtenidos y del conocimiento adquirido, se ha seleccionado un individuo generado con objeto de incorporarlo al control de un sistema fotovoltaico real.

Las siguientes pruebas realizadas han tenido como objetivo verificar que el individuo obtenido puede mejorar el rendimiento energético y el tiempo de vida de

un acumulador en un sistema fotovoltaico real. En este sentido se han comparado los siguientes controladores en los sistemas autónomos reales:

- Controlador borroso jerárquico basado en el diseño original descrito en [Gala-04], en adelante sistema original.
- Controlador borroso jerárquico basado en el individuo generado en las pruebas anteriores, en adelante sistema generado.
- Controlador básico que realiza la carga del acumulador exclusivamente la en la fase de flotación, sin realizar en ningún momento la fase de gaseo, en adelante sistema básico.
- Controlador comercial de alta gama, en adelante sistema comercial.

El comportamiento típico del controlador generado mediante el sistema borroso genético ha presentado diferencias importantes con respecto a los controladores de los sistemas original y comercial, por un lado la fase de gaseo es de menor intensidad y duración, y por otro lado la fase de flotación se define en una tensión ligeramente inferior.

Con respecto a las capacidades de los acumuladores, el sistema básico, con el conocimiento peor evaluado, caracterizado por la carga exclusivamente en flotación, ha presentado el mayor deterioro en el acumulador, mientras que el controlador generado, con el conocimiento obtenido a través del sistema borroso genético, presenta el menor deterioro de la batería. Los controladores jerárquico y comercial presentan unas capacidades similares, siendo ligeramente inferior las perdidas de capacidad en el controlador jerárquico.

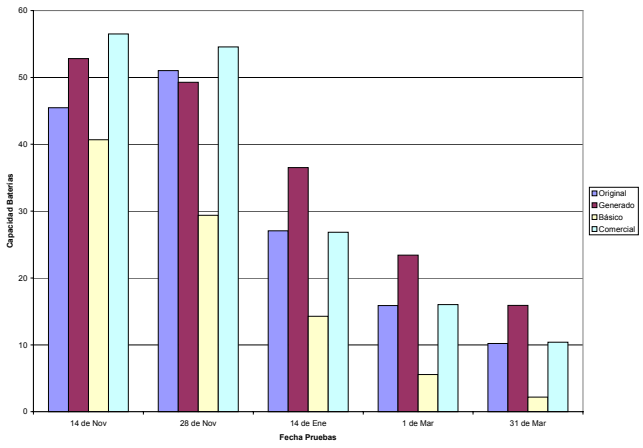


Figura 4. Evolución de las capacidades de las baterías.

El rendimiento energético, relación entre generación y consumo, presenta valores muy similares para todos los controladores. La tendencia del rendimiento durante las pruebas es descendente, lo cual ratifica la necesidad del aumento de aporte energético con el incremento de edad del acumulador.

5 CONCLUSIONES Y TRABAJO FUTURO.

Como resumen del análisis del conocimiento generado en el grupo de individuos obtenidos, es importante incorporar la variable Vbat con objeto de limitar las tensiones máximas de sobrecarga, realizar un gaseo de menor duración e intensidad y finalmente realizar la fase de flotación en una tensión ligeramente inferior.

El individuo generado por el sistema borroso genético ha resultado el mejor en la comparación con los otros controladores.

Ha existido un deterioro generalizado de la capacidades de todos los acumuladores utilizados. En este sentido, se están realizado actualmente diversas pruebas, utilizando otro fabricante de baterías, con objeto de verificar los resultados.

Los individuos obtenidos, incorporan un número reducido de variables y reglas, por lo que es factible su implementación en un sistema embebido, basado en un microcontrolador con capacidad de proceso de información y recursos limitados.

Referencias

[Bago-99] Bago, J.C. Magdalena, L. K*nowledge based evaluation of knowledge bases.* EUSFLAT – ESTYLF JOINT CONFERENCE 1999.

[Cord-01] CORDON, O., HERRERA, F., HOFFMANN, F., MAGDALENA, L. *Genetic Fuzzy Systems.* Ed. World Scientific. 2001.

[Gala-99] Galan, S.G., Velasco, J.R. *Rules evaluation of knowledge bases.* Eusflat - Estylf join conference, pp 195-198. Palma de Mallorca, Spain 1999..

[Gala-04] S.G.Galán, y otros. "Controlador jerárquico en sistemas fotovoltaicos autónomos" ESTTYLF04.. por publicar, 2004.

[Gold-89] GOLDBERG, D.E. *Genetic algorithms in search, optimization, and machine learning.* Adisson Wesley, 1989.

[Lore-94] LORENZO, E. *Electricidad Solar. Ingeniería de los sistemas fotovoltaicos.* Editorial PROGENSA. Sevilla, 1994.

CONTROLADOR EVOLUTIVO POR INSERCIÓN DE REGLAS EN SISTEMAS FOTOVOLTAICOS AUTÓNOMOS

S. G. Galán, J. Aguilera
Investigación y desarrollo en Energía Solar y Automática (IDEA).
Dpto. de Electrónica.
Universidad de Jaén.
E.U.P. de Linares Alfonso X El Sabio, 28.
23700 Linares (Jaén) Spain
E-mail: {sgalan,, aguilera}@ujaen.es

Juan Ramón Velasco Pérez
Dpto. de Automática
Universidad de Alcalá.
Escuela Politécnica.
Crta. N-II, Km 31,600
28871 Alcalá de Henares
Madrid Spain
E-mail: juanra@aut.uah.es

Luis Magdalena Layos
Dpto. Matemática Aplicada a las Tecnologías de la Información
Universidad Politécnica de Madrid.
E.T.S.I. de Telecomunicación.
Ciudad Universitaria s/n, 28040
Madrid Spain.
E-mail: llayos@mat.upm.es

Resumen

El presente trabajo describe el algoritmo utilizado en la obtención de forma evolutiva, de un controlador de carga para sistemas fotovoltaicos autónomos. El algoritmo que se presenta, se basa en la continua inserción de reglas, previa verificación de su utilidad. El mecanismo utilizado para evaluar el comportamiento, de cada una de las reglas candidatas a ser insertadas, en la base de conocimiento, se basa en un sistema experto, el cual realiza evaluaciones instantáneas del comportamiento del sistema fotovoltaico autónomo. Dicho sistema de evaluación también será comentado en el presente trabajo. El controlador de carga obtenido mediante el algoritmo propuesto será igualmente presentado así como algunos resultados obtenidos al utilizarlo en una instalación real.

Palabras Clave: Aplicación, Regulador de carga, Sistemas fotovoltaicos autónomos , inserción de reglas, base de conocimiento evolutiva, evaluación experta.

1 INTRODUCCIÓN

La energía solar fotovoltaica es, hoy en día y sin lugar a dudas, una forma limpia y fiable de producción de energía eléctrica a pequeña escala. En ocasiones, como por ejemplo en emplazamientos aislados, cuyo abastecimiento a través de la red eléctrica no resulta fácil, se utilizan los denominados Sistemas Fotovoltaicos Autónomos (SFA)[8].

En dichos sistemas, el controlador de carga es el elemento que se encarga de proteger al sistema de acumulación, sin embargo, el funcionamiento de los mismos en la actualidad, en la práctica presenta un comportamiento manifiestamente mejorable [2].

En este trabajo se presenta un controlador de carga borroso [3] obtenido mediante aprendizaje automático [3][5]. Dicho controlador ha sido fruto de la optimización del comportamiento de un controlador previo, denominado jerárquico [1]. Este controlador jerárquico se ha desarrollado a partir de otros trabajos [4][6][7] previos realizados con el objeto de verificar la viabilidad del uso de un controlador de carga borroso en sistemas fotovoltaicos autónomos.

La metodología de aprendizaje propuesta se basa en inserción de reglas en las bases de conocimiento del controlador, previa verificación de la bondad del comportamiento de las mismas. Dicha verificación se lleva a cabo mediante la valoración comparada del comportamiento del controlador con la regla y sin la regla. Por lo tanto una parte importante en la metodología presentada es la valoración del comportamiento de un controlador de carga. Para realizar esta función se ha utilizado un sistema experto basado en conocimiento.

Este trabajo se estructura de la siguiente forma: En el apartado 2, se realiza la presentación de la metodología utilizada para evaluar el comportamiento de un controlador de carga. En el apartado 3, se presenta el algoritmo de aprendizaje automático propuesto, basado en la inserción de reglas previa verificación de su bondad. En el apartado 4 se presenta el controlador de carga obtenido a partir de controlador jerárquico, mostrando las reglas incorporadas a cada una de las bases de conocimiento. En el apartado 5, se presentan resultados de operación obtenidos al utilizar el regulador presentado en una instalación real, y una comparación con los obtenidos por un regulador comercial de gama alta en una instalación de

similares características. Finalmente el apartado 6, se presentan algunas conclusiones y acciones futuras derivadas del presente trabajo.

2 EVALUACIÓN EXPERTA DEL COMPORTAMIENTO DEL CONTROLADOR BORROSO DE CARGA.

Una parte importante del algoritmo, que se presenta en este trabajo, para la adquisición de conocimiento, consiste en la valoración del comportamiento del controlador de carga borroso. De forma que las posibles mejoras que puedan obtenerse con dicho algoritmo estarán enmarcadas dentro de los objetivos que se valoren del comportamiento.

En este sentido, cabe destacar dos estrategias a la hora de valorar el comportamiento del controlador de carga borroso, una centrándose en el funcionamiento global del sistema y otra centrándose en las relaciones internas existentes en el controlador de carga borroso.

Inicialmente se optó por realizar una evaluación centrada en las relaciones existentes entres las variables internas del controlador de carga. Sin embargo, finalmente se ha optado por la segunda estrategia, es decir, centrándose en el funcionamiento global del sistema. La razón fundamental que ha justificado este cambio consiste en la siguiente circunstancia: Inicialmente se estaban obteniendo unas evaluaciones demasiado estáticas, resultando poco significativas las aportaciones de las nuevas reglas, lo cual dificulta, aunque sin impedir, el funcionamiento del algoritmo que se presenta.

En definitiva, la valoración del comportamiento del controlador de carga borroso, se realiza con un sistema experto, de forma que pueden materializarse reglas que definan funcionamientos deseables del controlador de carga borroso. Éstas reglas se incluyen en una base de conocimiento que será la encargada de evaluar el comportamiento global del controlador de carga borroso.

En la tabla 1 se presentan las reglas que están siendo utilizadas en la actualidad, por el sistema de evaluación experto. Es importante señalar que el sistema de evaluación utilizado, permite la modificación o inclusión de reglas que permitan ir ajustando el comportamiento deseado del controlador borroso de carga.

Tabla 1: Reglas de evaluación

SOC	V.OBJ.	EVAL.
BAJO	BAJA	M. BAJA
BAJO	ALTA	M. ALTA
ALTO	BAJA	M. ALTA
ALTO	ALTA	M. BAJA

La forma en que se calcula la evaluación definitiva del comportamiento del controlador de carga borroso es la siguiente:

Se contemplan cuatro puntos de partida, P_1, P_2, P_3 y P_4, que representan posibles estados iniciales de la batería.

Los cuatro puntos que se han seleccionado, se consideran lo suficientemente representativos de las posibles situaciones por las que se encontrará una batería, a lo largo de su tiempo de vida útil. Concretamente, se han seleccionado dos variables, el estado de carga de la batería (SOC) y la edad, y se les han asignado los siguientes valores normalizados:

P_1:	SOC=0.8	EDAD=0.8
P_2:	SOC=0.8	EDAD=0.2
P_3:	SOC=0.2	EDAD=0.8
P_4:	SOC=0.2	EDAD=0.2

Para cada uno de estos puntos de partida, se lanza una periodo de simulación, obteniendo una valoración parcial del comportamiento de controlador, E_{P1}, E_{P2}, E_{P3} y E_{P4}.

Para calcular cada evaluación parcial E_{Pi}, es necesario tener en consideración los siguientes aspectos:

- Cada evaluación parcial E_{Pi} se calcula como la media aritmética de todas las evaluaciones instantáneas (una por minuto).
- En los casos en los que la tensión de la batería no esté comprendida entre unos valores considerados de seguridad, [11, 15.3], la evaluación instantánea será igual a cero.
- Para valorar negativamente comportamientos que conduzcan variaciones elevadas en la tensión de la batería, cada evaluación parcial E_{Pi} se verá decrementada hasta un 10% en función de dichas variaciones.

Una vez obtenida todas y cada una de las evaluaciones parciales E_{Pi}, la evaluación definitiva se calcula como la media aritmética de dichas evaluaciones, sin embargo dicha evaluación se verá decrementada en un 20%, al no cumplir completitud, si en cualquier momento a lo largo de la obtención de las evaluaciones parciales, la tensión de la batería se ha salido del margen de seguridad. Este mismo porcentaje para valorar la completitud ya fue el mismo que se utilizo en [13].

3. ALGORITMO DE INSERCIÓN DE REGLAS

El algoritmo utilizado para la adquisición de conocimiento, se presenta en la figura 1. Como puede

observarse, el algoritmo se basa en la generación de reglas al azar y su posterior inserción en la base de conocimiento adecuada, siempre que la inclusión de esta regla mejore el comportamiento del controlador de carga. Lógicamente, este comportamiento es evaluado según el procedimiento comentado en el apartado anterior.

Dado que el controlador de carga borroso dispone de tres bases de conocimiento, antes de generar una regla al azar e insertarla en la base de conocimiento, es preciso determinar en cual de las tres bases de conocimiento existentes será insertada.

Para verificar la posible mejora que se obtiene con la inserción de la regla no hay más que comparar las evaluaciones de los comportamientos con dicha regla y sin ella, y seleccionar la mejor situación.

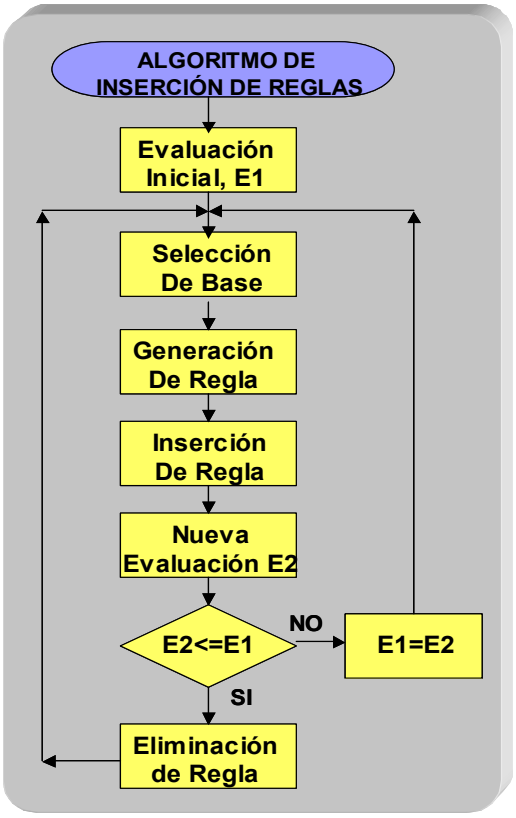


Figura 1: Algoritmo de inserción de reglas.

En relación al funcionamiento de este algoritmo, es importante destacar algunos aspectos:

- Como ya se ha comentado, el controlador borroso dispone de tres sistemas borrosos, y por lo tanto, el aprendizaje podría ser selectivo, es decir, se podrían generar reglas sólo para un sistema borroso. Sin embargo, en este trabajo se ha optado por extender el ámbito del aprendizaje a los tres sistemas borrosos que componen el controlador de carga.
- Se pueden elegir distintos mecanismos de parada para la ejecución del algoritmo presentado: Número de reglas, obtención de cierta mejora, o combinación de ambos. El mecanismo utilizado se corresponde con el número de reglas.

4. REGULADOR OBTENIDO.

El regulador de carga propuesto se ha seleccionado atendiendo a la mejora obtenida con respecto al regulador jerárquico [1]. Otra opción posible consiste en seleccionar aquellas reglas que más ocasiones se han incorporado a lo largo de todos los experimentos realizados. Ésta última opción se mantiene a la espera de futuros experimentos, constituyendo una vía en principio considerada como viable.

A continuación se presentarán las reglas que han mejorado el comportamiento del regulador de carga jerárquico y que por lo tanto han sido incorporadas al regulador de carga borroso propuesto. Es conveniente recordar que las funciones de pertenencia de los conjuntos borrosos no han sufrido modificación alguna con respecto a las del regulador de carga jerárquico.

4.1. Reglas incorporadas en Gaseo.

Además de las propias reglas que el regulador de carga jerárquico contiene en su base de conocimiento de gaseo, el regulador propuesto contiene las reglas mostradas en la tabla 2:

Tabla 2: Reglas incorporadas a la base de conocimiento de gaseo.

BASE DE CONOCIMIENTO DE GASEO		
	ANTECEDENTES	**TIEMPO GASEO**
G13	**IF** “EDAD=JOVEN”	MEDIO
G14	**IF** “PDDANTER=ALTA”	MEDIO
G15	**IF** “PDDANTER=ALTA”	ALTO
G16	**IF** “TEMP=MEDIA” **and** “EDAD=VIEJA”	MUY BAJO
G17	**IF** “TEMP=MEDIA” **and** “PDDANTER=ALTA”	MUY BAJO
G18	**IF** “TEMP=MEDIA” **and** “PDDANTER= ALTA”	ALTO
G19	**IF** “EDAD=JOVEN” **and** “PDDANTER=BAJA”	BAJO
G20	**IF** “TEMP=MEDIA” **and** “EDAD=JOVEN” **and** “PDDANTER= BAJA”	MEDIO
G21	**IF** “TEMP=MEDIA” **and** “EDAD=JOVEN” **and** “PDDANTER=BAJA”	MEDIO
G22	**IF** “TEMP=MEDIA” **and** “EDAD=JOVEN” **and** “PDDANTER= ALTA”	MUY BAJO

Hay que considerar que de todas las reglas incluidas, la G20 y la G21 son idénticas Por otra parte es necesario indicar que algunas de las reglas incluidas presentan inconsistencia, así las reglas G17 y G18, a pesar de tener idénticos antecedentes presentan consecuentes contrapuestos. El motivo de esta circunstancia se puede deber a un intento por parte del regulador propuesto de contrarrestar alguna situación para intentar mantener un comportamiento que incorpore excesivas variaciones en los parámetros.

En relación con la bondad o no de las reglas incluidas, es necesario comentar que existen reglas a priori son totalmente razonables, otras que a priori no son nada razonables y otras que en principio un experto no podría incluirlas en ninguna de las dos categorías. Estas posibles situaciones se muestran en la tabla 3.

Tabla 3: Evaluación subjetiva de las reglas incorporadas en la base de conocimiento de gaseo.

EVALUACIÓN SUBJETIVA DE LAS REGLAS INCORPORADAS EN GASEO.	
Razonables	**G15 – G18 – G19 – G20 – G21**
No razonables	**G16 - G17 - G22**
¿?	**G13 - G14**

La aparición de reglas con un comportamiento no razonable se puede deber, al igual que en el caso de inconsistencia, a un intento del regulador propuesto de contrarrestar alguna situación para intentar mantener un comportamiento que incorpore excesivas variaciones en los parámetros.

4.2. Reglas incorporadas en Incremento.

Al igual que ocurre con la base de conocimiento de gaseo, además de las propias reglas que el regulador de carga jerárquico contiene en su base de conocimiento de incremente, el regulador propuesto contiene las reglas mostradas en la tabla 4:

Tabla 4: Reglas incorporadas a la base de conocimiento de incremento.

BASE DE CONOCIMIENTO DE INCREMENTO		
	ANTECEDENTES	**INCRE.**
R13	**IF** "TEMP=MEDIA"	NEGAT
I14	**IF** "EDAD=JOVEN"	NEGAT
I15	**IF** "EDAD=VIEJA"	MUY NEGAT
I16	**IF** "EDAD=VIEJA"	NEGAT
I17	**IF** "PDDANTER=BAJA"	MUY NEGAT
I18	**IF** "PDDANTER=BAJA"	NEGAT
I19	**IF** "PDDANTER=ALTA"	MUY POSIT
I20	**IF** TEMP=MEDIA" **and** "EDAD=JOVEN"	NEGAT
I21	**IF** "EDAD=VIEJA" **and** "PDDANTER=ALTA"	POSIT
I22	**IF** "TEMP=MEDIA" **and** "EDAD=JOVEN" **and** "PDDANTER= ALTA"	MUY POSIT
I23	**IF** "TEMP=MEDIA" **and** "EDAD=VIEJA" **and** "PDDANTER= ALTA"	MUY POSIT
I24	**IF** "TEMP=MEDIA" **and** "EDAD=VIEJA" **and** "PDDANTER= ALTA"	MUY POSIT

En esta ocasión también aparecen reglas repetidas, concretamente la reglas I23 e I24. Igualmente en esta ocasión existen reglas con idénticos antecedentes y distintos consecuentes, concretamente las reglas I17 e I18 por un lado, y las reglas I15 e I16 por otro. Sin embargo, en esta ocasión no se produce inconsistencia en un sentido estricto, ya que las diferencias existentes en los consecuentes son en realidad distintos grados de un mismo significado.

En la tabla 5 se presenta un cuadro resumen de la valoración previa que puede realizarse a cerca de las reglas incluidas en la base de conocimiento de incremento.

Tabla 5: Evaluación subjetiva de las reglas incorporadas en la base de conocimiento de incremento.

EVALUACIÓN SUBJETIVA DE LAS REGLAS INCORPORADAS EN INCREMENTO.	
Razonables	**I14 – I17 – I-18 - I19 – I20 I21 – I23 – I24**
No razonables	**I13 - I15 – I16**
¿?	**I22**

4.3. Reglas incorporadas en Regulación.

Las reglas que se han incluido en la base de conocimiento de regulación se muestran en la tabla 6.

Tabla 6: Reglas incorporadas a la base de conocimiento de regulación.

BASE DE CONOCIMIENTO DE REGULACIÓN		
	ANTECEDENTES	**VREG.**
R13	**IF** "SOC=ALTO"	FLOTA
R14	**IF** "SOC=ALTO"	FLOTA
R15	**IF** "SOC=ALTO"	FLOTA
R16	**IF** "SOC=ALTO"	FLOTA
R17	**IF** "SOC=ALTO"	FLOTA
R18	**IF** "SOC=ALTO"	FLOTA
R19	**IF** "SOC=BAJO" **and** "TPGASEO=ALTO"	FLOTA
R20	**IF** "SOC=BAJO" **and** "TPGASEO=ALTO"	FLOTA
R21	**IF** "SOC=ALTO" **and** "TPGASEO=BAJO"	FLOTA
R22	**IF** "SOC=ALTO" **and** "TPGASEO=BAJO"	FLOTA
R23	**IF** "TPGASEO=ALTO"	FLOTA
R24	**IF** "TPGASEO=ALTO"	FLOTA
R25	**IF** "TPGASEO=ALTO"	FLOTA

En esta ocasión el numero de reglas repetidas ha subido considerablemente, este se debe a que el espacio de búsqueda es más reducido para la base de conocimiento de regulación (sólo 16 reglas posibles). Así, en realidad sólo han aparecido 4 reglas diferentes.

En este caso no han aparecido reglas con una valoración previa negativa, ni siquiera con incertidumbre en la valoración, es decir, la valoración previa que se hace de todas las nuevas reglas es razonable en todos los casos.

5. RESULTADOS DE OPERACIÓN.

Para poder valorar el comportamiento del controlador de carga obtenido mediante la metodología presentada, se han instalado dos sistemas fotovoltaicos autónomos con las mismas características, uno con el controlador obtenido y otra con un controlador de carga jerárquico.

El periodo de pruebas consiste en someter a los sistemas de acumulación a un continuo ciclado. Así, durante periodos de ausencia de radiación solar, se conectarán las cargas (alógenos) de forma que se producirá una descarga de los sistemas de acumulación. Por otra parte, durante los periodos de existencia de radiación solar y en ausencia de cargas conectadas se producirá el cargado de los sistemas de acumulación. Evidentemente, estos periodos de carga de estarán supervisados por los controladores de carga de cada uno de los sistemas fotovoltaicos autónomos.

El proceso de ciclado comenzó el día 4 de diciembre de 2003, y finalizó el día 7 de enero de 2004. A continuación se presenta una tabla con un resumen de los datos de capacidad medidos al inicial y al final de periodo de pruebas

Tabla 7: Capacidades iniciales y finales de los sistemas de acumulación utilizados.

SISTEMA	Capacidad (AH)		% Capacidad perdida
	Inicial	Final	
JERARQUICO	50,82	26,94	46,98
EVOLUTIVO	44,99	37,57	16,49

6. CONCLUSIONES Y ACCIONES FUTURAS.

En este documento se ha presentado un controlador de carga borroso para sistemas fotovoltaicos autónomos obtenido mediante aprendizaje automático. El algoritmo utilizado para dicho aprendizaje ha sido el de inserción de reglas. El controlador presentado presenta un comportamiento optimizado con respecto al controlador jerárquico inicial.

El controlador de carga borroso obtenido será utilizado en más pruebas con objeto de valorar con más precisión y exactitud su comportamiento .

Referencias

[1] S.G. Galán, J.C. Bago. J. Aguilera, J.R. Velasco, L. Magdalena. "Controlador jerárquico en sistemas fotovoltaicos autónomos". ESTYLF2004, Septiembre, 2004.

[2] Preiser, K. J. Kuhmann, O. Parodi. "*Quality Issues for Solar Home Systems",* 13th European Photovoltaic Solar Energy Conference. Nice, France, pp 510-513. (1995).

[3] Cordón, O., Herrera, F., Hoffmann, F., Magdalena, L.."Genetic Fuzzy Systems. Ed. World Scientific. 2001.

[4] Bago, J.C. Galán, S.G. Aguilera, J. Velasco, J. R. Magdalena, L. "Aplicación de la lógica borrosa a los reguladores de carga en sistemas fotovoltaicos autónomos". ESTYLF2002, pp 65-70, Septiembre, 2002.

[5] Goldberg, D.E. "Genetic algorithms in search, optimization, and machine learning". Adisson Wesley, 1989.

[6] Galán, S.G. Bago, J.C. Aguilera, J. Magdalena, L. "Fuzzy charge controller for stand-alone photovoltaic systems". EUSFLAT 01, pp160-163, 2001.

[7] J.C. Bago, S.G. Galán, J. Aguilera, J.R. Velasco, L. Magdalena. "Fuzzy Charge Applications in Stand-Alone Photovoltaic Systems" Mathware & Soft Computing. Vol .IX . n. 1 (2002). Pp 85-105.

[8] Lorenzo, E. "Electricidad solar". Ingeniería de los sistemas fotovoltaicos. Editorial Progensa. Sevilla 1994.

Extended homogenization for multi-adjoint logic programs

Jesús Medina Moreno
Dept. Matemática Aplicada.
Universidad de Málaga
jmedina@ctima.uma.es

Manuel Ojeda Aciego *
Dept. Matemática Aplicada.
Universidad de Málaga
aciego@ctima.uma.es

Resumen

The theory of multi-adjoint logic programs has been introduced as a unifying framework to deal with uncertainty, imprecise data or incomplete information. From the applicative part, a neural net based implementation of homogeneous propositional multi-adjoint logic programming on the unit interval has been presented elsewhere, but restricted to the case in which the only connectives involved in the program were the usual product, Gödel and Łukasiewicz together with weighted sums.

A generalization of the homogenization process needed for the neural implementation is presented here in order to deal with a more general family of adjoint pairs, but maintaining the advantage of the existing implementation. Its soundness is proved and the complexity of the transformation is analysed.

1 Introduction

The study of reasoning methods under uncertainty, imprecise data or incomplete information has received increasing attention in the recent years. A number of different approaches have been proposed with the aim of better explaining observed facts, specifying statements, reasoning and/or executing programs under some type of uncertainty whatever it might be.

One important and powerful mathematical tool that has been used for this purpose at theoretical level is fuzzy logic. From the applicative side, neural networks have a massively parallel architecture-based dynamics which are inspired by the structure of human brain, adaptation capabilities, and fault tolerance. The recent paradigm of soft computing promotes the use and integration of different approaches for the problem solving.

The main advantages of fuzzy logic systems are the capability to express nonlinear input/output relationships by a set of qualitative if-then rules, and to handle both numerical data and linguistic knowledge, especially the latter, which is extremely difficult to quantify by means of traditional mathematics. The main advantage of neural networks, on the other hand, is the inherent learning capability, which enables the networks to adaptively improve their performance.

Recently, a new approach presented in [2] introduced a hybrid framework to handling uncertainty, expressed in the language of multi-adjoint logic but implemented by using ideas from the world of neural networks. The handling of uncertainty inside their logic model is based on the use of a generalised set of truth-values as a generalization of [6]. On the other hand, multi-adjoint logic programming [4] generalizes residuated logic programming [1] in that several different implications are allowed in the same program, as a means to facilitate the task of the specification.

Considering several implications in the same program is interesting because it provides a more flexible framework for the specification of problems, for instance, in situations in which connectives are built from the users preferences. In these contexts, it is likely that knowledge is described by a many-valued logic program where connectives have many-valued truth functions and, perhaps, aggregation operators (such as arithmetic mean or weighted sum) where different implications could be needed for different purposes, and different aggregators are defined for different users, depending on their preferences.

In [2] a neural-like implementation of multi-adjoint logic programming was presented with the restriction that the only connectives involved in the program were the usual product, Gödel and Łukasiewicz together

*Partially supported by TIC2003-09001-C02-01.

with weighted sums. A key point of the implementation was a preprocessing of the initial program to transform it into a *homogeneous* program, detailed in [3]. As the theoretical development of the multi-adjoint framework does not rely on particular properties of the product, Gödel and Łukasiewicz adjoint pairs, it seems convenient to allow for a generalization of the implementation to admit, at least, a family of continuous t-norms (recall that any continuous t-norm can be interpreted as the ordinal sum of product and Łukasiewicz t-norms).

The purpose of this paper is to present a "finer" homogenization process for multi-adjoint logic programs such that conjunctors which are built as ordinal sums of product, Gödel and Łukasiewicz t-norms are decomposed into its components so that, as a result, the original neural approach is still applicable.

The structure of the paper is as follows: In Section 2, the syntax and semantics of multi-adjoint logic programs are introduced, together with the homogenization procedure; in Section 3, the translation from multi-adjoint programs into homogeneous programs is extended in order to cope with conjunctors defined as ordinal sums, the preservation of the semantics is proved under this extended transformation, and the complexity of the translation is studied. The paper finishes with some conclusions and pointers to future work.

2 Preliminary definitions

To make this paper as self-contained as possible, the necessary definitions about multi-adjoint structures are included in this section. For motivating comments, the interested reader is referred to [4].

Multi-adjoint logic programming is a general theory of logic programming which allows the simultaneous use of different implications in the rules and rather general connectives in the bodies.

The first interesting feature of multi-adjoint logic programs is that a number of different implications are allowed in the bodies of the rules. The basic definition is the generalization of residuated lattice given below:

Definition 1 *Let $\langle L, \preceq\rangle$ be a lattice. A* multi-adjoint lattice *$\mathcal{L}$ is a tuple $(L, \preceq, \leftarrow_1, \&_1, \ldots, \leftarrow_n, \&_n)$ satisfying the following items:*

1. *$\langle L, \preceq\rangle$ is bounded, i.e. it has bottom and top elements;*
2. *$\top \,\&_i\, \vartheta = \vartheta \,\&_i\, \top = \vartheta$ for all $\vartheta \in L$ for $i = 1, \ldots, n$;*
3. *$(\&_i, \leftarrow_i)$ is an* adjoint pair *in $\langle L, \preceq\rangle$ for $i = 1, \ldots, n$; i.e.*
   (a) *Operation $\&_i$ is increasing in both arguments,*
   (b) *Operation $\leftarrow_i$ is increasing in the first argument and decreasing in the second argument,*
   (c) *For any $x, y, z \in P$, we have that $x \preceq (y \leftarrow_i z)$ holds if and only if $(x \,\&_i\, z) \preceq y$ holds.*

In the rest of the paper we restrict to the unit interval, although the general framework of multi-adjoint logic programming is applicable to a general lattice.

2.1 Syntax of multi-adjoint logic programs

A *multi-adjoint program* is a set of weighted rules $\langle F, \vartheta\rangle$ satisfying the following conditions:

1. F is a formula of the form $A \leftarrow_i \mathcal{B}$ where A is a propositional symbol called the *head* of the rule, and $\mathcal{B}$ is a well-formed formula, which is called the *body*, built from propositional symbols $B_1, \ldots, B_n$ $(n \geq 0)$ by the use of monotone operators.
2. The *weight* ϑ is an element (a truth-value) of $[0, 1]$.

Facts are rules with body[1] 1 and a *query* (or *goal*) is a propositional symbol intended as a question $?A$ prompting the system.

2.2 Semantics of multi-adjoint logic programs

Once presented the syntax of multi-adjoint programs, the semantics is given below.

Definition 2 *An* interpretation *is a mapping I from the set of propositional symbols Π to the lattice $\langle [0, 1], \leq\rangle$.*

Note that each of these interpretations can be uniquely extended to the whole set of formulas, and this extension is denoted as $\hat{I}$. The set of all the interpretations is denoted $\mathcal{I}_{\mathfrak{L}}$.

The ordering $\leq$ of the truth-values L can be easily extended to $\mathcal{I}_{\mathfrak{L}}$, which also inherits the structure of complete lattice and is denoted $\sqsubseteq$. The minimum element of the lattice $\mathcal{I}_{\mathfrak{L}}$, which assigns 0 to any propositional symbol, will be denoted $\triangle$.

Definition 3

1. *An interpretation $I \in \mathcal{I}_{\mathfrak{L}}$* satisfies *$\langle A \leftarrow_i \mathcal{B}, \vartheta\rangle$ if and only if $\vartheta \leq \hat{I}\,(A \leftarrow_i \mathcal{B})$.*

[1] It is also customary to use write $\top$ instead of 1, and even not to write any body.

2. *An interpretation $I \in \mathcal{I}_{\mathfrak{L}}$ is a* model *of a multi-adjoint logic program $\mathbb{P}$ iff all weighted rules in $\mathbb{P}$ are satisfied by I.*

3. *An element $\lambda \in L$ is a* correct answer *for a program $\mathbb{P}$ and a query $?A$ if for any interpretation $I \in \mathcal{I}_{\mathfrak{L}}$ which is a model of $\mathbb{P}$ we have $\lambda \leq I(A)$.*

The operational approach to multi-adjoint logic programs used in this paper will be based on the fixpoint semantics provided by the immediate consequences operator, given in the classical case by van Emden and Kowalski [5], which can be generalised to the framework of multi-adjoint logic programs by means of the adjoint property, as shown below:

Definition 4 *Let $\mathbb{P}$ be a multi-adjoint program; the* immediate consequences operator*, $T_{\mathbb{P}} \colon \mathcal{I}_{\mathfrak{L}} \to \mathcal{I}_{\mathfrak{L}}$, maps interpretations to interpretations, and for $I \in \mathcal{I}_{\mathfrak{L}}$ and $A \in \Pi$ is given by*

$$T_{\mathbb{P}}(I)(A) = \sup\left\{\vartheta \,\&_i\, \hat{I}(\mathcal{B}) \mid \langle A \leftarrow_i \mathcal{B}, \vartheta\rangle \in \mathbb{P}\right\}$$

As usual, it is possible to characterise the semantics of a multi-adjoint logic program by the post-fixpoints of $T_{\mathbb{P}}$; that is, an interpretation I is a model of a multi-adjoint logic program $\mathbb{P}$ iff $T_{\mathbb{P}}(I) \sqsubseteq I$. The $T_{\mathbb{P}}$ operator is proved to be monotonic and continuous under very general hypotheses.

Once one knows that $T_{\mathbb{P}}$ can be continuous under very general hypotheses [4], then the least model can be reached in at most countably many iterations beginning with the least interpretation, that is, the least model is $T_{\mathbb{P}} \uparrow \omega(\triangle)$.

2.3 Obtaining a homogeneous program

Regarding the implementation as a neural network of [2], the introduction of the so-called *homogeneous rules*, provided a simpler and standard representation for any multi-adjoint program.

Definition 5 *A weighted formula is said to be* homogeneous *if it has one of the following forms:*

- $\langle A \leftarrow_i \&_i(B_1, \ldots, B_n), \vartheta\rangle$
- $\langle A \leftarrow_i @(B_1, \ldots, B_n), 1\rangle$
- $\langle A \leftarrow_i B_1, \vartheta\rangle$

where $A, B_1, \ldots, B_n$ are propositional symbols.

In the rest of this section we briefly recall the procedure for translating a multi-adjoint logic program into one containing only homogeneous rules. The procedure handles separately rules and facts, the latter are not related to the purpose of this paper, therefore we will only recall the procedure presented for homogenizing rules.

Two types of transformations are considered: The first one handles the main connective of the body of the rule, whereas the second one handles the subcomponents of the body.

$T1$. A weighted rule $\langle A \leftarrow_i \&_j(\mathcal{B}_1, \ldots, \mathcal{B}_n), \vartheta\rangle$ is substituted by the following pair of formulas:

$$\langle A \leftarrow_i A_1, \vartheta\rangle$$
$$\langle A_1 \leftarrow_j \&_j(\mathcal{B}_1, \ldots, \mathcal{B}_n), 1\rangle$$

where A_1 is a fresh propositional symbol, and $\langle \leftarrow_j, \&_j\rangle$ is an adjoint pair.

For the case $\langle A \leftarrow_i @(\mathcal{B}_1, \ldots, \mathcal{B}_n), \vartheta\rangle$ in which the main connective of the body of the rule happens to be an aggregator, the transformation is similar:

$$\langle A \leftarrow_i A_1, \vartheta\rangle$$
$$\langle A_1 \leftarrow @(\mathcal{B}_1, \ldots, \mathcal{B}_n), 1\rangle$$

where A_1 is a fresh propositional symbol, and $\leftarrow$ is a designated implication.

$T2$. A weighted rule $\langle A \leftarrow_i \Theta(\mathcal{B}_1, \ldots, \mathcal{B}_n), \vartheta\rangle$, where Θ is either $\&_i$ or an aggregator, and a component $\mathcal{B}_k$ is assumed to be either of the form $\&_j(\mathcal{C}_1, \ldots, \mathcal{C}_l)$ or $@(\mathcal{C}_1, \ldots, \mathcal{C}_l)$, is substituted by the following pair of formulas in either case:

$$\langle A \leftarrow_i \Theta(\mathcal{B}_1, \ldots, \mathcal{B}_{k-1}, A_1, \mathcal{B}_{k+1}, \ldots, \mathcal{B}_n), \vartheta\rangle$$
$$\langle A_1 \leftarrow_j \&_j(\mathcal{C}_1, \ldots, \mathcal{C}_l), \top\rangle$$

or

$$\langle A \leftarrow_i \Theta(\mathcal{B}_1, \ldots, \mathcal{B}_{k-1}, A_1, \mathcal{B}_{k+1}, \ldots, \mathcal{B}_n), \vartheta\rangle$$
$$\langle A_1 \leftarrow @(\mathcal{C}_1, \ldots, \mathcal{C}_l), \top\rangle$$

where A_1 is a fresh propositional symbol.

The procedure to transform the rules of a program so that all the resulting rules are homogeneous, is presented in Fig. 2. It is based in the two previous transformations, and in its description by abuse of notation the terms T1-rule (resp. T2-rule) are used to mean an adequate input rule for transformation T1 (resp. T2).

3 Considering compound conjunctors

As stated in the introduction, a neural net implementation of the immediate consequences operator of

Program Homogenization
  begin
    repeat
      for each T1-rule **do**
        Apply transformation T1
      end-for
      for each T2-rule **do**
        Apply transformation T2
      end-for
    until neither T1-rules nor T2-rules exist
  end

Figure 1: Pseudo-code for translating into a homogeneous program.

an homogeneous program was introduced in [2] for the case of the multi-adjoint lattice $([0,1], \leq, \&_P, \leftarrow_P, \&_G, \leftarrow_G, \&_L, \leftarrow_L)$. As the theoretical development of the multi-adjoint framework does not rely on particular properties of these three adjoint pairs, it seems convenient to allow for a generalization of the implementation to, at least, a family of continuous t-norms, for any continuous t-norm can be interpreted as the ordinal sum of product and Łukasiewicz t-norms.

In order to maintain the most of the proposed implementation, it makes sense to consider an extra process in the homogenization process in order to further translate a program with new types of t-norms into one on which the original neural approach is still applicable.

Recall the definition of ordinal sum of a family of t-norms:

Definition 6 *Let $(\&_i)_{i\in A}$ be a family of t-norms and a family of non-empty pairwise disjoint subintervals $[a_i, b_i]$ of $[0,1]$. The* ordinal sum *of the summands $(a_i, b_i, \&_i)$, $i \in A$ is the t-norm $\&$ defined as*

$$\&(x,y) = \begin{cases} a_i + (b_i - a_i)\,\&_i\left(\frac{x-a_i}{b_i-a_i}, \frac{y-a_i}{b_i-a_i}\right) & x, y \in [a_i, b_i] \\ \min(x,y) & \text{otherwise} \end{cases}$$

In order to simplify the notation of ordinal sum, let us introduce suitable functions for change of scale, to be able to switch between the intervals $[0,1]$ and $[a_i, b_i]$. Given the unit interval $[0,1]$ and a subset $[a_i, b_i] \subseteq [0,1]$, the function

$$f_i \colon [a_i, b_i] \to [0,1], \text{ with } f_i(x) = \frac{x - a_i}{b_i - a_i}$$

is bijective, and its inverse is

$$f_i^{-1} \colon [0,1] \to [a_i, b_i], \text{ with } f_i^{-1}(x) = a_i + (b_i - a_i)x$$

Now, given a t-norm $\&_i$ consider the following (adapted) t-norm

$$x \,\&_i^* \, y = \begin{cases} f_i^{-1}(f_i(x)\,\&_i\, f_i(y)) & \text{if } x, y \in [a_i, b_i] \\ \min\{x, y\} & \text{otherwise} \end{cases}$$

for all $x, y \in [0,1]$.

With the notations above it is obvious that the ordinal sum of the summands $(a_i, b_i, \&_i)$, $i \in A$ can be written as

$$x \,\&\, y = \min\{x \,\&_i^*\, y \mid i \in A\} \tag{1}$$

With this expression for the ordinal sum, we can introduce a third type of transformation, to be applied to those rules of a homogeneous program with an ordinal sum in the body.

First of all, it is convenient to consider these sum-like conjunctors as general aggregator operators, so that after the (standard) homogenization process we can assume that the weight of the obtained rule is 1.

*T*3. The homogeneous rule[2] $\langle A \leftarrow_i B_1 \,\&\, B_2, 1\rangle$, where $\&$ is expressed as in (1) is substituted by the following $n+1$ formulas:

$$\begin{aligned} \langle A_1 &\leftarrow B_1 \,\&_1^*\, B_2, 1\rangle \\ &\vdots \\ \langle A_n &\leftarrow B_1 \,\&_n^*\, B_2, 1\rangle \\ \langle A &\leftarrow_G \&_G(A_1, \ldots, A_n), 1\rangle \end{aligned}$$

where $A_1, \ldots, A_n$ are fresh propositional symbols.

After applying this transformation to a homogeneous program, we obtain another homogeneous program in whose bodies no t-norm appears as an ordinal sum. This kind of homogenous program is called *sum-free homogeneous program* (in short *sf-homogeneous program*).

Example 1 *Consider the elements $a_1 = 0.1$, $b_1 = 0.5$, $a_2 = 0.7$, $b_2 = 0.9$ and the ordinal sum $\&_s$ defined as*

$$x \,\&_s\, y \begin{cases} f_1^{-1}(f_1(x)\,\&_P\, f_1(y)) & \text{if } x, y \in [a_1, b_1] \\ f_2^{-1}(f_2(x)\,\&_L\, f_2(y)) & \text{if } x, y \in [a_2, b_2] \\ \min\{x, y\} & \text{otherwise} \end{cases}$$

Then if we have the homogeneous rule $\langle A \leftarrow_s B_1 \,\&_s\, B_2, 1\rangle$, applying T3, this is transformed in

$$\begin{aligned} \langle A_1 &\leftarrow B_1 \,\&_P^*\, B_2, 1\rangle && \textit{sf-homogeneous} \\ \langle A_n &\leftarrow B_1 \,\&_L^*\, B_2, 1\rangle && \textit{sf-homogeneous} \\ \langle A &\leftarrow_G \&_G(A_1, \ldots, A_n), 1\rangle && \textit{sf-homogeneous} \end{aligned}$$

[2] To simplify the presentation, let us assume that the body has just two arguments.

The procedure to transform the rules of a program so that all the resulting rules are sf-homogeneous, is presented in Fig. 2. In its description by abuse of notation we use the terms T1-rule (resp. T2-rule, T3-rule) to mean an adequate input rule for transformation T1 (resp. T2, T3).

Program sf-Homogenization
  begin
    repeat
      for each T1-rule **do**
          Apply transformation T1
      end-for
      for each T2-rule **do**
          Apply transformation T2
      end-for
      for each T3-rule **do**
          Apply transformation T3
      end-for
    until neither T1- nor T2- nor T3-rules exist
    end

Figure 2: Pseudo-code for translating into a sf-homogeneous program.

3.1 Preservation of the semantics

It is necessary to check that the semantics of the initial program has not been changed by the transformation. The following results will show that every model of the sf-homogenized program $\mathbb{P}^*$ is also a model of the original program $\mathbb{P}$ and, in addition, the minimal model of $\mathbb{P}^*$ is also the minimal model of $\mathbb{P}$.

Theorem 1 *Let $\mathbb{P}$ be a homogeneous program, then every model of the program $\mathbb{P}^*$, obtained after to apply the sf-homogenization process to $\mathbb{P}$, is also a model of $\mathbb{P}$ when restricted to the variables occurring in $\mathbb{P}$.*

Proof: It will be sufficient to show that the transformation T3 satisfies that every model of its output is also a model of its input, as the proof for T1 and T2 was given in [3].

Assume that I is a model of the rules

$$\langle A_1 \leftarrow B_1 \,\&_1^*\, B_2, 1\rangle, \quad \ldots, \quad \langle A_n \leftarrow B_1 \,\&_n^*\, B_2, 1\rangle \qquad \langle A \leftarrow_G \&_G(A_1, \ldots, A_n), 1\rangle$$

therefore we have

$$\hat{I}(B_1 \,\&_i^*\, B_2) \leq I(A_i) \quad \text{and} \quad \hat{I}(\&_G(A_1, \ldots, A_n)) \leq I(A)$$

for all $i \in \{1, \ldots, n\}$. Now, by monotonicity, we have

$$\&_G(\hat{I}(B_1 \,\&_1^*\, B_2), \ldots, \hat{I}(B_1 \,\&_n^*\, B_2)) \leq I(A)$$

Recall that we want to prove that I satisfies the rule $\langle A \leftarrow_i B_1 \,\&\, B_2, 1\rangle$, that is, $\hat{I}(B_1 \,\&\, B_2) \leq I(A)$, where $\&$ is an ordinal sum of $\&_i^*$, $i \in \{1, \ldots, n\}$.

But this is true by Equation (1) since we have

$$\begin{aligned} I(B_1) \,\&\, I(B_2) &= \min_{i\in\{1,\ldots,n\}}\{I(B_1) \,\&_i^*\, I(B_2)\} \\ &= \&_G(\hat{I}(B_1 \,\&_1^*\, B_2), \ldots, \hat{I}(B_1 \,\&_n^*\, B_2)) \\ &\leq I(A) \end{aligned}$$ □

Theorem 2 *Given a program $\mathbb{P}$, the minimal model of the program $\mathbb{P}^*$ obtained after applying sf-homogenization, is also a model of $\mathbb{P}$ when restricted to variables in $\mathbb{P}$.*

The idea underlying the proof is to consider any model I of $\mathbb{P}$, then extend it to $\mathbb{P}^*$ in such a way that it is also a model of $\mathbb{P}^*$, finally use minimality on $\mathbb{P}^*$. The key point is to notice that, for every "fresh" propositional variable A_i introduced by the process, there is only one rule headed with A_i in the resulting program. This feature allows the extension of any model I to these new symbols in purely recursive terms.

Proof:

Let M^* be the minimal model of $\mathbb{P}^*$, and let M denote its restriction to $\mathbb{P}$. By the Theorem 1 we have that M is also a model of $\mathbb{P}$, so we have only to prove that it is minimal.

Once again, only the behaviour of transformation T3 has to be taken into account.

Given a model I of $\mathbb{P}$, consider a rule $\langle A_i \leftarrow B_1 \,\&_i^*\, B_2, 1\rangle$, where A_i is a propositional variable in $\mathbb{P}^*$ but not in $\mathbb{P}$. We argued above that there can be only one such rule headed with A_i, therefore the following extension of I makes sense:

$$I^*(A_i) = \hat{I}(B_1 \,\&_i^*\, B_2)$$

Obviously, by definition this extension I^* is also a model of $\mathbb{P}^*$, therefore the minimal model M^* of $\mathbb{P}^*$ satisfies $M^* \sqsubseteq I^*$. Now, by restricting the domain back to the variables in $\mathbb{P}$ we obtain $M \sqsubseteq I$. Therefore, M is the minimal model of $\mathbb{P}$. □

3.2 Complexity issues

In [3] it was shown that the complexity of the algorithm for transforming a multi-adjoint program into a homogeneous one is linear on the size of the program. Specifically, the following theorem was stated and proved

Theorem 3 ([3]) *Let $\langle A \leftarrow_i \Theta(\mathcal{B}_1, \ldots, \mathcal{B}_l), \vartheta\rangle$ be a rule with n connectives in the body ($n \geq 1$), then:*

- *The number of homogeneous rules obtained after applying the procedure is n, if either $\Theta = \&_i$ or $\Theta = @$ with $\vartheta = 1$; and $n + 1$ otherwise.*
- *The number of transformations obtained after applying the procedure is $n - 1$ if either $\Theta = \&_i$ or $\Theta = @$ with $\vartheta = 1$; and n otherwise.*

It has to be pointed out that the number of rules obtained (or transformations applied) is *bounded* by the integer number stated in the theorem. The equality is obtained when all the associative operators are effectively grouped into one 'flexible arity' operator.

Back to the sf-homogenization process, it can also be shown to be linear on the size of the program. The following theorem shows a precise calculation of the complexity of the homogenization procedure.

Theorem 4 *Let $\langle A \leftarrow_i \Theta(\mathcal{B}_1, \ldots, \mathcal{B}_l), \vartheta \rangle$ be a rule with n connectives in the body ($n \geq 1$), out of which exactly m are constructed as ordinal sums of k_i basic connectives for each $i \in \{1, \ldots, m\}$ then:*

- *The number of homogeneous rules obtained after applying the procedure is bounded by $n + m + \sum_{i=1}^{m} k_i$, if either $\Theta = \&_i$ or $\Theta = @$ with $\vartheta = 1$; and $n + m + \sum_{i=1}^{m} k_i + 1$ otherwise.*
- *The number of transformations T_i applied by the procedure is $n + 2m - 1$ if either $\Theta = \&_i$ or $\Theta = @$ with $\vartheta = 1$; and $n + 2m$ otherwise.*

Proof: Concatenate the thesis of Theorem 3 with the fact that an application of T3 generates exactly $k + 1$ rules, provided that the ordinal sum was built out of k intervals, because in each homogeneous rule there is only one operator in the body. □

4 Conclusions and future work

It has been shown that the homogenization process for multi-adjoint logic programs can be adapted so that it can "decompose" conjunctors defined as an ordinal sum.

This is a very convenient extension of the previous procedure, in that the existing neural-like implementation of the fixpoint multi-adjoint semantics only considers product, Gödel, and Łukasiewicz connectives and weighted sums. Assuming that one knows the definition of a compound connective as an ordinal sum of product and Łukasiewicz connectives, then it would be possible to use the existing neural implementation to execute the program.

A different approach to this problem would be to modify directly the neural implementation so that the ordinal sum constructor can be conveniently represented by the net. Then, it would be interesting to make a comparison of the efficiency of the treatment of compound conjunctors via this extended sf-homogenization an the modified neural approach.

References

[1] C.V. Damásio and L. Moniz Pereira. Monotonic and residuated logic programs. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU'01*, pages 748–759. Lect. Notes in Artificial Intelligence, 2143, 2001.

[2] J. Medina, E. Mérida-Casermeiro, and M. Ojeda-Aciego. A neural approach to extended logic programs. In *7th Intl Work Conference on Artificial and Natural Neural Networks, IWANN'03*, pages 654–661. Lect. Notes in Computer Science 2686, 2003.

[3] J. Medina and M. Ojeda-Aciego. Homogenizing multi-adjoint logic programs. In *Proc. of Intl Conference on Fuzzy Logic and Technology, EUSFLAT'03*, pages 640–644, 2003.

[4] J. Medina, M. Ojeda-Aciego, and P. Vojtáš. Multi-adjoint logic programming with continuous semantics. In *Logic Programming and Non-Monotonic Reasoning, LPNMR'01*, pages 351–364. Lect. Notes in Artificial Intelligence 2173, 2001.

[5] M. H. van Emden and R. Kowalski. The semantics of predicate logic as a programming language. *Journal of the ACM*, 23(4):733–742, 1976.

[6] P. Vojtáš. Fuzzy logic programming. *Fuzzy Sets and Systems*, 124(3):361–370, 2001.

A termination theorem for sorted multi-adjoint logic programming

C.V. Damásio *
Centro Inteligência Artificial
Universidade Nova de Lisboa
cd@di.fct.unl.pt

J. Medina **
Dept. Matemática Aplicada
Universidad de Málaga
jmedina@ctima.uma.es

M. Ojeda-Aciego **
Dept. Matemática Aplicada
Universidad de Málaga
aciego@ctima.uma.es

Resumen

In this paper we present a general framework of logic programming allowing for the combination of several adjoint lattices of truth-values. The main contribution is a new sufficient condition which guarantees termination of all queries for the fixpoint semantics for an interesting class of programs.

Keywords: Fuzzy Logic Programming, Termination Results.

1 Introduction

A number of approaches have been developed for the so-called inexact or fuzzy or approximate reasoning have been proposed, involving either fuzzy or annotated or probabilistic or similarity-based logic programming, e.g. [7, 11, 6, 5, 12, 8] as a result of the increasing interest in models of reasoning under "imperfect" information.

In this work we propose a sorted version of the multi-adjoint paradigm of logic programming, where each sort identifies an underlying lattice of truth-values (weights) which must satisfy adjoint conditions. We restrict to the ground case but allow infinite programs, and thus do not loose generality.

The semantics of sorted multi-adjoint logic program is characterised, as usual, by the post-fixpoints of the immediate consequence operator $T_{\mathbb{P}}$, which is proved to be monotonic and continuous under very general hypotheses, see [9]. The current proposal is an important enhancement of our previous works [1, 4, 9].

The major contribution of this paper is a termination result for some classes of sorted multi-adjoint logic programs complementing results about Hybrid Probabilistic Logic Programs [6, 5]. In particular, the case of programs obtained by arbitrary composition of operators obeying the boundary condition $\vartheta \otimes 1 = 1 \otimes \vartheta \leq \vartheta$ over the unit interval are shown to be terminating. As a consequence, we illustrate the application of the theorems to obtain known termination results for the class of HPLPs with conjunctive probabilistic strategies.

The structure of the paper is as follows. In Section 2, we introduce the preliminary concepts necessary for the definition of the syntax and semantics of sorted multi-adjoint logic programs, presented in Section 3. In Section 4, we state the basic results regarding the termination properties of our semantics, which are extended later in Section 5. The paper finishes with some conclusions and pointers to future work.

2 Preliminary Definitions

We will make extensive use of the constructions and terminology of universal algebra, in order to define formally the syntax and the semantics of the languages we will deal with. A minimal set of concepts from universal algebra, which will be used in the sequel in the style of [4], is introduced below.

2.1 Some Definitions from Universal Algebra

The notions of signature and Σ-algebra will allow the interpretation of the function and constant symbols in the language, as well as for specifying the syntax.

Definition 1 *A* signature *is a pair* $\Sigma = \langle S, F \rangle$ *where* S *is a set of elements, designated* sorts*, and* F *is a collection* F *of pairs* $\langle f, s_1 \times \cdots \times s_k \to s \rangle$ *denoting func-*

* Partially supported by Acção Integrada Luso-Espanhola E-42/02, by FSE/FEDER project TARDE (POSI/EEI/12097/2001), by European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779 (cf. http://rewerse.net).

** Partially supported by Acción Integrada HP2001-0078, and Spanish MCYT project TIC2003-09001-C02-01.

tions, such that $s, s_1, \ldots, s_k$ are sorts and no symbol f occurs in two different pairs. The number k is the arity of f. If k is 0 then f is a constant symbol.

To simplify notation, we write $f\colon \tau$ to denote a pair $\langle f, \tau\rangle$ belonging to F.

Definition 2 *Given a signature $\Sigma = \langle S, F\rangle$, a* Σ-algebra *$\mathfrak{A}$ is a pair $\left\langle \{A^s\}_{s\in S}, I\right\rangle$ where:*

1. *Each A^s is a nonempty set called the carrier of sort s,*
2. *and I is a function which assigns a map*

$$I(f) : A^{s_1} \times \cdots \times A^{s_k} \to A^s$$

to each $f\colon s_1 \times \cdots \times s_k \to s \in F$, where $k > 0$, and an element $I(c) \in A^s$ to each constant symbol $c\colon s$ in F.

2.2 Multi-Adjoint Lattices and Multi-Adjoint Algebras

The main concept we will need in this section is that of *adjoint pair*.

Definition 3 *Let $\langle P, \preceq\rangle$ be a partially ordered set and let $(\leftarrow, \&)$ be a pair of binary operations in P such that:*

(a1) Operation $\&$ is increasing in both arguments

(a2) Operation $\leftarrow$ is increasing in the first argument and decreasing in the second argument.

(a3) For any $x, y, z \in P$, we have that

$$x \preceq (y \leftarrow z) \quad \textit{iff} \quad (x \,\&\, z) \preceq y$$

Then $(\leftarrow, \&)$ forms an adjoint pair *in $\langle P, \preceq\rangle$.*

Extending the results in [4, 3, 12] to a more general setting, in which different implications (Łukasiewicz, Gödel, product) and thus, several modus ponens-like inference rules are used, naturally leads to considering several *adjoint pairs* in the lattice. More formally,

Definition 4 *Let $\langle L, \preceq\rangle$ be a lattice. A* multi-adjoint lattice *$\mathcal{L}$ is a tuple $(L, \preceq, \leftarrow_1, \&_1, \ldots, \leftarrow_n, \&_n)$ satisfying the following items:*

(l1) $\langle L, \preceq\rangle$ is bounded, i.e. it has bottom ($\bot$) and top ($\top$) elements;

(l2) $(\leftarrow_i, \&_i)$ is an adjoint pair in $\langle L, \preceq\rangle$ for all i;

(l3) $\top \,\&_i\, \vartheta = \vartheta \,\&_i\, \top = \vartheta$ for all $\vartheta \in L$ for all i.

Remark: Note that residuated lattices are a special case of multi-adjoint lattice, in which the underlying poset has a lattice structure, has monoidal structure wrt & and $\top$, and only one adjoint pair is present.

From the point of view of expressiveness, it is interesting to allow extra operators to be involved with the operators in the multi-adjoint lattice. The structure which captures this possibility is that of a multi-adjoint algebra.

Definition 5 *A Σ-Algebra $\mathfrak{L}$ is a* Multi-Adjoint Σ-Algebra *whenever:*

- *The carrier L^s of each sort is a lattice under a partial order $\preceq^s$.*
- *Each sort s contains operators $\leftarrow_i^s\colon s \times s \to s$ and $\&_i^s : s \times s \to s$ for $i = 1, \ldots, n^s$ (and possibly some extra operators) such that the tuple $\mathcal{L}^s$*

$$(L^s, \preceq^s, I(\leftarrow_1^s), I(\&_1^s), \ldots, I(\leftarrow_n^s), I(\&_n^s))$$

is a multi-adjoint lattice.

Multi-Adjoint Σ-Algebras can be found underlying the probabilistic deductive databases framework of [8] where our sorts correspond to disjunctive modes and the adjoint operators to different conjunctive modes for combining probabilistic knowledge. Our framework is richer since we do not restrain ourselves to a single and particular carrier set and allow more operators.

In practice, we will usually have to assume some properties on the extra operators considered. These extra operators will be assumed to be either aggregators, or conjunctors or disjunctors, all of which are monotone functions (the latter, in addition, are required to generalize their Boolean counterparts).

3 Syntax and Semantics of Sorted Multi-Adjoint Logic Programs

Sorted multi-adjoint logic programs will be constructed from the abstract syntax induced by a multi-adjoint Σ-algebra on a set of sorted propositional symbols (or variables). Specifically, we will consider a multi-adjoint Σ-algebra $\mathfrak{L}$ whose extra operators can be arbitrary monotone operators. This algebra will host the manipulation the truth-values of the formulas in our programs.

In addition, let Π be an infinite set of sorted propositional symbols, disjoint from the set of function symbols in $\mathfrak{L}$, and the corresponding term Σ-algebra[1] of

[1] Shortly, this corresponds to the algebra freely generated from Π and the set of function symbols in $\mathfrak{L}$, respecting sort assignments.

formulas $\mathfrak{F} = Terms(\Sigma, \Pi)$. To denote that a symbol $A \in \Pi$ has sort s we will often write $A \in \Pi^s$.

Remark: As we are working with two Σ-algebras, and to discharge the notation, we introduce a special notation to clarify which algebra a function symbols belongs to, instead of continuously using either $\sigma_{\mathfrak{L}}$ or $\sigma_{\mathfrak{F}}$. Let σ be a function symbol in Σ, its interpretation under $\mathfrak{L}$ is denoted $\dot{\sigma}$ (a dot on the operator), whereas σ itself will denote $\sigma_{\mathfrak{F}}$ when there is no risk of confusion.

3.1 Syntax of Sorted Multi-Adjoint Logic Programs

The definition of sorted multi-adjoint logic program is given, as usual, as a set of rules and facts. The particular syntax of these rules and facts is given below:

Definition 6 *A* sorted multi-adjoint logic program *is a set $\mathbb{P}$ of rules of the form $\langle A \leftarrow_i^s \mathcal{B}, \vartheta \rangle$ such that:*

1. *The* rule *($A \leftarrow_i^s \mathcal{B}$) is a formula (an algebraic term) of $\mathfrak{F}$;*
2. *The* weight *ϑ is an element (a truth-value) of $\mathcal{L}^s$;*
3. *The* head *of the rule A is a propositional symbol of Π of sort s.*
4. *The* body *$\mathcal{B}$ is a formula of $\mathfrak{F}$ with sort s, built from sorted propositional symbols $B_1, \ldots, B_n$ ($n \geq 0$) by the use of function symbols in Σ.*
5. Facts *are rules with body $\top^s$, the top element of lattice $\mathcal{L}^s$.*
6. *A* query *(or* goal*) is a propositional symbol intended as a question $?A$ prompting the system.*

Sometimes, we will represent bodies of formulas as $@[B_1, \ldots, B_n]$, where[2] $B_1, \ldots, B_n$ are the propositional variables occurring in the body and $@$ is the aggregator obtained as a composition.

3.2 Semantics of Sorted Multi-Adjoint Logic Programs

Definition 7 *An* interpretation *is a mapping $I \colon \Pi \to \bigcup_s L^s$ such that for every propositional symbol p of sort s then $I(p) \in L^s$. The set of all interpretations of the sorted propositions defined by the Σ-algebra $\mathfrak{F}$ in the Σ-algebra $\mathfrak{L}$ is denoted $\mathcal{I}_{\mathfrak{L}}$.*

Note that by the unique homomorphic extension theorem, each of these interpretations can be uniquely extended to the whole set of formulas $\mathfrak{F}$.

[2]Note the use of square brackets in this context.

The orderings $\preceq^s$ of the truth-values L^s can be easily extended to the set of interpretations as follows:

Definition 8 *Consider $I_1, I_2 \in \mathcal{I}_{\mathfrak{L}}$. Then, $\langle \mathcal{I}_{\mathfrak{L}}, \sqsubseteq \rangle$ is a lattice where $I_1 \sqsubseteq I_2$ iff $I_1(p) \preceq^s I_2(p)$ for all $p \in \Pi^s$. The least interpretation $\triangle$ maps every propositional symbol of sort s to the least element $\bot^s \in \mathcal{L}^s$.*

A rule of a sorted multi-adjoint logic program is satisfied whenever the truth-value of the rule is greater or equal than the weight associated with the rule. Formally:

Definition 9 *Given an interpretation $I \in \mathcal{I}_{\mathfrak{L}}$, a weighted rule $\langle A \leftarrow_i^s \mathcal{B}, \vartheta \rangle$ is* satisfied *by I iff $\vartheta \preceq^s \hat{I}(A \leftarrow_i^s \mathcal{B})$. An interpretation $I \in \mathcal{I}_{\mathfrak{L}}$ is a* model *of a sorted multi-adjoint logic program $\mathbb{P}$ iff all weighted rules in $\mathbb{P}$ are satisfied by I.*

Definition 10 *An element $\lambda \in \mathcal{L}^s$ is a* correct answer *for a program $\mathbb{P}$ and a query $?A$ of sort s if for an arbitrary interpretation I which is a model of $\mathbb{P}$ we have $\lambda \preceq^s I(A)$.*

The immediate consequences operator, given by van Emden and Kowalski, can be easily generalised to the framework of sorted multi-adjoint logic programs.

Definition 11 *Let $\mathbb{P}$ be a sorted multi-adjoint logic program. The* immediate consequences operator *$T_{\mathbb{P}}$ maps interpretations to interpretations, and is defined by*

$$T_{\mathbb{P}}(I)(A) = \bigsqcup_s \{\vartheta \,\&_i^s\, \hat{I}(\mathcal{B}) \mid \langle A \leftarrow_i^s \mathcal{B}, \vartheta \rangle \in \mathbb{P}\}$$

where A is an arbitrary propositional symbol of sort s, and $\bigsqcup_s$ is the least upper bound in the lattice $\mathcal{L}^s$.

The semantics of a sorted multi-adjoint logic program can be characterised, as usual, by the post-fixpoints of $T_{\mathbb{P}}$; that is, an interpretation I is a model of a sorted multi-adjoint logic program $\mathbb{P}$ iff $T_{\mathbb{P}}(I) \sqsubseteq I$. The single-sorted $T_{\mathbb{P}}$ operator is proved to be monotonic and continuous under very general hypotheses, see [9], and it is remarkable that these results are true even for non-commutative and non-associative conjunctors. In particular, by continuity, the least model can be reached in at most countably many iterations of $T_{\mathbb{P}}$ on the least interpretation. These results immediately extend to the sorted case.

4 Termination Results

In this section we recall the termination properties of the $T_{\mathbb{P}}$ operator. In what follows we assume that every operator is computable. If only monotone and

continuous operators are present in the underlying sorted multi-adjoint Σ-algebra $\mathfrak{L}$ then the immediate consequences operator reaches the least fixpoint at most after ω iterations. It is not difficult to show examples in which exactly ω iterations may be necessary to reach the least fixpoint.

The termination property we investigate is stated in the following definition:

Definition 12 *Let $\mathbb{P}$ be a sorted multi-adjoint logic program with respect to a multi-adjoint Σ-algebra $\mathfrak{L}$ and a sorted set of propositional symbols Π. We say that $T_{\mathbb{P}}$* terminates for every query *iff for every propositional symbol A there is a finite n such that ${T_{\mathbb{P}}}^n(\triangle)(A)$ is identical to $lfp(T_{\mathbb{P}})(A)$.*

We presented in [2] several results in order to guarantee that every query can be answered after a finite number of iterations. In particular, this means that for finite programs the least fixpoint of $T_{\mathbb{P}}$ can also be reached after a *finite* number of iterations, ensuring computability of the semantics. Nevertheless, the results are applicable to a special class of infinite sorted multi-adjoint logic programs, designated programs with finite dependencies.

The *dependency graph* of $\mathbb{P}$ has a vertex for each propositional symbol in Π, and there is an arc from a propositional symbol A to a propositional symbol B iff A is the head of a rule with body containing an occurrence of B. The dependency graph for a propositional symbol A is the subgraph of the dependency graph containing all nodes accessible from A and corresponding edges.

Definition 13 *A sorted multi-adjoint logic program $\mathbb{P}$ has* finite dependencies *iff for every propositional symbol A the number of edges in the dependency graph for A is finite.*

We proceed by presenting our new major termination result valid for an important class of sorted multi-adjoint logic programs, where neither acyclicity nor finiteness properties are required. To begin with, the following definition is needed:

Definition 14 *A multi-adjoint Σ-algebra is said to be* local *when the following conditions are satisfied:*

- *For every pair of sorts s_1 and s_2 there is a unary monotone casting function symbol $c_{s_1 s_2} \colon s_2 \to s_1$ in Σ.*
- *All other function symbols have types of the form $f : s \times \cdots \times s \to s$, i.e. are closed operations in each sort, satisfying the following boundary conditions for every $v \in \mathcal{L}^s$:*

$$\begin{aligned} I(f)(v, 1^s, \ldots, 1^s) &\preceq^s v \\ I(f)(1^s, v, 1^s, \ldots, 1^s) &\preceq^s v \\ &\vdots \\ I(f)(1^s, \ldots, 1^s, v) &\preceq^s v \end{aligned}$$

  where 1^s is the top element of $\mathcal{L}^s$. In particular, if f is a unary function symbol then $I(f)(v) \preceq^s v$.
- *The following property is obeyed:*

$$(c_{ss_1} \circ c_{s_1 s_2} \circ \ldots \circ c_{s_n s})\,(v) \preceq^s v$$

  for every $v \in \mathcal{L}^s$ and finite composition of casting functions with overall sort $s \to s$.

In local sorted multi-adjoint Σ-algebras the non-casting function symbols are restricted to operations in a unique sort. In order to combine values from different sorts, one is deemed to use explicitly the casting functions in the appropriate places.

Definition 15 *Let $\mathbb{P}$ be a multi-adjoint program, and $A \in \Pi^s$.*

- *The set $R^I_{\mathbb{P}}(A)$ of* relevant values *for A with respect to interpretation I is the set of maximal values of the set $\{\vartheta \,\&^s_i\, \hat{I}(\mathcal{B}) \mid \langle A \leftarrow^s_i \mathcal{B}, \vartheta\rangle \in \mathbb{P}\}$*
- *The* culprit set *for A with respect to I is the set of rules $\langle A \leftarrow^s_i \mathcal{B}, \vartheta\rangle$ of $\mathbb{P}$ such that $\vartheta \,\&^s_i\, \hat{I}(\mathcal{B})$ belongs to $R^I_{\mathbb{P}}(A)$. Rules in a culprit set are called* culprits.
- *The* culprit collection *for ${T_{\mathbb{P}}}^n(\triangle)(A)$ is defined as the set of culprits used in the tree of recursive calls of $T_{\mathbb{P}}$ in the computation.*

The rationale is to use the set of relevant values for a propositional symbol A to collect the maximal values contributing to the computation of A in an iteration of the $T_{\mathbb{P}}$ operator. The non-maximal values are irrelevant for determining the new value for A by $T_{\mathbb{P}}$.

Theorem 1 ([2]) *Let $\mathbb{P}$ be a sorted multi-adjoint logic program with respect to a local multi-adjoint Σ-algebra $\mathfrak{L}$ and the set of sorted propositional symbols Π, and having finite dependencies.*

If for every iteration n and propositional symbol A of sort s the set of relevant values for A with respect to $T^n_{\mathbb{P}}(\triangle)$ is a singleton, then $T_{\mathbb{P}}$ terminates for every query.

The proof is based on the bounded growth of the culprit collection for ${T_{\mathbb{P}}}^n(\triangle)(A)$. By induction on n, it can be proved that if $T^{n+1}_{\mathbb{P}}(\triangle)(A) \succ^s T^n_{\mathbb{P}}(\triangle)(A)$ for

$A \in \Pi$, then the culprit collection for $T_{\mathbb{P}}^{n+1}(\triangle)(A)$ has cardinality at least $n+1$. Since the number of rules in the dependency graph for A is finite then the $T_{\mathbb{P}}$ operator must terminate after a finite number of steps, by using all the rules relevant for the computation of A.

5 Termination of Hybrid Probabilistic Logic Programs

Hybrid Probabilistic Logic Programs [6] have been proposed for constructing rule systems which allow the user to reason with and combine probabilistic information under different probabilistic strategies. The conjunctive (disjunctive) probabilistic strategies are pairwise combinations of t-norms (t-conorms, respectively) over pairs of real numbers in the unit interval $[0, 1]$, i.e. intervals. The termination results presented in [5] either only allow constant annotation or (finite) ground programs. From the analysis of the fixpoint construction one can see that only a finite number of different intervals can be generated.

Theorem 1 above can be generalized to a case which will allow us to prove, in an abstract way, the termination result for hybrid probabilistic logic programs containing only occurrences of conjunctive basic formulas, without requiring any reasoning about the shape of the transformed program. Specifically, a termination result can also be obtained if the local multi-adjoint Σ-algebra also contains function symbols $g\colon s_1 \times \cdots \times s_l \to s_k$ such that their interpretations are isotonic functions with finite range (that is, the image of g is a finite set). We call this kind of algebra a *local multi-adjoint Σ-algebra with finite operators.*

The major result in this paper is stated below, as an extension of Theorem 1.

Theorem 2 *Let $\mathbb{P}$ be a sorted multi-adjoint logic program with respect to a local multi-adjoint Σ-algebra with finite operators $\mathfrak{L}$ and the set of sorted propositional symbols Π, and having finite dependencies.*

If for every iteration n and propositional symbol A of sort s the set of relevant values for A with respect to $T_{\mathbb{P}}^n(\triangle)$ is a singleton, then $T_{\mathbb{P}}$ terminates for every query.

The intuition underlying the proof of this theorem is simply to apply a cardinality argument. However, the formal presentation of the proof requires introducing some technicalities which offer enough control on the increase of the computation tree for a given query.

On the one hand, one needs to handle the number of applications of rules; this is done by using the concept of *culprit collection*, as in Theorem 1. On the other hand, one needs to consider the applications of the finite operators, which are not adequately considered by the culprit collections. With this aim, given a propositional symbol A, let us consider the subset of rules of the program associated to its dependence graph, and denote it by $\mathbb{P}^A$. This set is finite, for the program has finite dependencies, so we can write:

$$\mathbb{P}^A = \{\langle H_i \leftarrow \mathcal{B}_i, \vartheta_i\rangle \mid i \in \{1, \ldots, s\}\}$$

In addition, let us write each body of the rules above as follows:

$$\mathcal{B}_i = @_i[g_1^i(\mathcal{D}_1^i), \ldots, g_{k_i}^i(\mathcal{D}_{k_i}^i), C_1^i, \ldots, C_{m_i}^i]$$

where $g_j^i(\mathcal{D}_j^i)$ represents the subtrees corresponding to the outermost occurrences of finite operators, the C_j^i are the propositional symbols which are not in the scope of finite operator, and $@_i$ is the operator obtained after composing all the operators in the body not in the scope of any finite operator.

Now, consider $G(\mathbb{P}^A) = \{g_1^1, \ldots, g_{k_1}^1, \ldots, g_1^s, \ldots, g_{k_s}^s\}$, which is a finite multiset, and let us define the following counting sets for the contribution of the finite operators to the overall computation.

Definition 16 *The* counting sets for $\mathbb{P}$ and A *are defined for all $n \in \mathbb{N}$ as follows:*

$$\Xi_n^A = \{k < n \mid \text{ there is } g_j^i \in G(\mathbb{P}^A) \text{ such that } g_j^i(T_{\mathbb{P}}{}^n(\triangle)(\mathcal{D}_j^i)) > g_j^i(T_{\mathbb{P}}{}^{n-1}(\triangle)(\mathcal{D}_j^i))\}$$

With these definitions we can state the main lemma needed in the proof of Theorem 2.

Lemma 1 *Under the hypotheses of Theorem 2, if $T_{\mathbb{P}}{}^{n+1}(\triangle)(A) > T_{\mathbb{P}}{}^n(\triangle)(A)$ then either $|\Xi_{n+1}^A| > |\Xi_n^A|$ or the culprit collection for $T_{\mathbb{P}}{}^{n+1}(\triangle)(A)$ is greater than that for $T_{\mathbb{P}}{}^n(\triangle)(A)$.*

Proof: We will proceed by induction on n.

Base case $n = 0$: for any A it is straightforward that if $T_{\mathbb{P}}(\triangle)(A) > \triangle(A) = \bot$ then a new rule has been used.

Inductive case: Assume that the result is true for any propositional symbol and $n = k$; in order to prove the result for $k + 1$, assume that $T_{\mathbb{P}}{}^{k+1}(\triangle)(A) > T_{\mathbb{P}}{}^k(\triangle)(A)$.

By the singleton hypothesis, there exists a rule labeled by an index $i \in \{1, \ldots, s\}$ such that

$$T_{\mathbb{P}}{}^{k+1}(\triangle)(A) = \vartheta_i \,\&\, T_{\mathbb{P}}{}^k(\triangle)(@_i[g_1^i(\mathcal{D}_1^i), \ldots, g_{r_i}^i(\mathcal{D}_{r_i}^i), C_1^i, \ldots, C_{m_i}^i])$$

now, for that rule indexed by i, by definition of $T_{\mathbb{P}}$ as a least upper bound, we have

$$T_{\mathbb{P}}{}^{k}(\triangle)(A) \geq \vartheta_i \,\&\, T_{\mathbb{P}}{}^{k-1}(\triangle)(@_i[g_1^i(\mathcal{D}_1^i),\ldots,g_{r_i}^i(\mathcal{D}_{r_i}^i),C_1^i,\ldots,C_{m_i}^i])$$

then, by the monotonicity of the connectives in the body, either there exists $j \in \{1,\ldots,r_i\}$ such that

$$g_j^i(T_{\mathbb{P}}{}^{k}(\triangle)(\mathcal{D}_j^i)) > g_j^i(T_{\mathbb{P}}{}^{k-1}(\triangle)(\mathcal{D}_j^i))$$

or there exists $j \in \{1,\ldots,m_i\}$ such that

$$T_{\mathbb{P}}{}^{k}(\triangle)(C_j^i) > T_{\mathbb{P}}{}^{k-1}(\triangle)(C_j^i)$$

In the former case, it is obvious that $|\Xi_{n+1}^A| > |\Xi_n^A|$; in the latter case, then the induction hypothesis on the propositional variable C_j^i applies. □

Proof of the Theorem: The previous lemma is the key of the proof of the main theorem of this section:

- Firstly, since the program has finite dependencies there cannot be infinitely many rules in the culprit collections for A.
- On the other hand, the sequence of cardinals $|\Xi_n^A|$ is upper bounded (since the range of each function g_j^i is finite and $G(\mathbb{P}(A))$ is also finite).

As a result we obtain that $T_{\mathbb{P}}$ terminates for every query. □

We now sketch the application of the previous theorem to prove termination of the particular class of HPLPs with only conjunctive probabilistic strategies. Given the embedding of Hybrid Probabilistic Logic Programs into Residuated ones presented in [4], we know that the transformed HPLP rules can have two different shapes: either the body is the combination of finite image functions, or the combination of t-norms, or the combination of t-conorms.

Under this embedding one could expect to apply the termination theorem to general HPLPs. This is not the case because, in particular, for disjunctive strategies the boundary conditions are not obeyed because of t-conorms. However the result applies straightforwardly for HPLPs with only conjunctive strategies because t-conorms cannot occur.

6 Conclusions

We have presented a sorted multi-adjoint logic programming language, capable of capturing and combining several reasoning paradigms dealing with imprecision and uncertainty. A termination result motivated by the hybrid probabilistic logic programs has been presented, the consequences of this theoretical result and some of its possible modifications will be studied in the particular framework of HPLP. The embedding of other proposals in the literature into our framework will be explored in subsequent work.

References

[1] C.V. Damásio and M. Ojeda-Aciego. On termination of a tabulation procedure for residuated logic programming. 6th Intl Workshop on Termination, pp. 40-43, 2003

[2] C.V. Damásio, J. Medina and M. Ojeda-Aciego. Termination results for sorted multi-adjoint logic programming. In IPMU'04, 1879–1886, 2004.

[3] C. V. Damásio and L. M. Pereira. Monotonic and residuated logic programs. Lect. Notes in Artificial Intelligence 2143, pp. 748–759, 2001.

[4] C. V. Damásio and L. M. Pereira. Hybrid probabilistic logic programs as residuated logic programs. *Studia Logica*, 72(1):113–138, 2002.

[5] M. Dekhtyar, A. Dekhtyar and V.S. Subrahmanian. Hybrid Probabilistic Programs: Algorithms and Complexity. Proc. of Uncertainty in AI conference, 1999

[6] A. Dekhtyar and V.S. Subrahmanian. Hybrid Probabilistic Programs, *Journal of Logic Programming* 43(3):187–250, 2000

[7] M. Kifer and V. S. Subrahmanian, Theory of generalized annotated logic programming and its applications. *J. Logic Programming* 12(4):335–367, 1992

[8] L. Lakhsmanan and F. Sadri, On a theory of probabilistic deductive databases. *Theory and Practice of Logic Programming* 1(1):5–42, 2001

[9] J. Medina, M. Ojeda-Aciego, and P. Vojtáš. Multi-adjoint logic programming with continuous semantics. Lect. Notes in Artificial Intelligence 2173, pp. 351–364, 2001.

[10] J. Medina, M. Ojeda-Aciego, and P. Vojtáš. A procedural semantics for multi-adjoint logic programming. Lect. Notes in Artificial Intelligence 2258, pp. 290–297, 2001.

[11] M. H. van Emden. Quantitative deduction and its fixpoint theory. *Journal of Logic Programming*, 4(1):37–53, 1986.

[12] P. Vojtáš. Fuzzy logic programming. *Fuzzy Sets and Systems*, 124(3):361–370, 2001.

VARIANZA DE UNA VARIABLE ALEATORIA DIFUSA. ESTUDIO DE DISTINTAS DEFINICIONES.

Inés Couso
Dep. Estadística e I.O. y D.M.
Universidad de Oviedo
e-mail: couso@pinon.ccu.uniovi.es

Susana Montes
Dep. Estadística e I.O. y D.M.
Universidad de Oviedo
e-mail: montes@uniovi.es

Luciano Sánchez
Dep. de Informática
Universidad de Oviedo
e-mail: luciano@ccia.uniovi.es

Resumen

En este trabajo introducimos una nueva definición de varianza de una variable aleatoria borrosa en el contexto de la teoría de probabilidades imprecisas y la comparamos con otras definiciones de la literatura. Nuestro objetivo principal es mostrar la utilidad de cada una de ellas en cada situación particular.[1]

Palabras Clave: Variable aleatoria difusa, varianza, probabilidades imprecisas.

1. INTRODUCCIÓN

El concepto de variable aleatoria borrosa, que generaliza la definición clásica de variable aleatoria, fue introducido por Féron [6] en 1976 y modificado posteriormente por otros autores.

En este contexto, las generalizaciones de las definiciones de los parámetros asociados a una distribución de probabilidad pueden dividirse en dos grandes grupos: por una parte, se han definido algunos parámetros como la esperanza [14], la función de distribución en un punto [2, 10], la varianza [9] y la covarianza [12] de una variable aleatoria borrosa como valores difusos. Por otra parte, se ha definido, entre otros, la varianza [8, 11] como valor numérico (nítido). La introducción de estas últimas definiciones está motivada principalmente por el interés por parte de los autores en facilitar la toma de decisiones a partir de parámetros con valores numéricos y no borrosos.

En este trabajo observaremos que el concepto de varianza se puede generalizar al caso de variables aleatorias difusas de diferentes maneras. Consideraremos distintas definiciones de este concepto existentes en la literatura y propondremos una definición adicional, que se podrá enmarcar en un modelo de probabilidades imprecisas. Apoyándonos en ejemplos sencillos, observaremos las ventajas y limitaciones de cada definición en los diferentes contextos.

[1]Este trabajo ha sido financiado por los proyectos BFM2001-3515 y TIC2002-04036-C05-05, parcialmente financiados con fondos FEDER.

2. VARIABLES ALEATORIAS DIFUSAS

Dado que los conjuntos borrosos se han interpretado en la literatura de múltiples maneras, una variable aleatoria borrosa admite tantos otros significados. A continuación, vamos a revisar brevemente dos interpretaciones de las variables aleatorias borrosas ya existentes en la literatura, y vamos a introducir una nueva. De acuerdo con cada una de ellas, se resumirá la información proporcionada por la variable aleatoria borrosa mediante un modelo diferente.

En [14], Puri y Ralescu consideran que las observaciones de algunos experimentos aleatorios no dan lugar a salidas numéricas, sino que vienen representados por términos lingüísticos imprecisos. De acuerdo con esta idea, algunos autores consideran la variable aleatoria borrosa como una función medible, en el sentido clásico, entre cierta σ-álgebra de sucesos del espacio original y una σ-álgebra definida sobre una clase de subconjuntos borrosos de $\mathbb{R}$. En este contexto, se puede utilizar la distribución de probabilidad inducida por la variable aleatoria difusa para resumir la información probabilística que ésta proporciona (por ejemplo, podría generarse un modelo del tipo siguiente: la probabilidad de que el resultado sea "alto" es 0.5, la de que sea "mediano" es 0.25 y la de que sea "bajo" es 0.25).

Kruse y Meyer, por su parte, consideran en [10] una interpretación "posibilística" de los conjuntos borrosos. Según los autores, la variable aleatoria borrosa representa el conocimiento (vago, impreciso) existente acerca de una variable aleatoria (clásica), $X_0 : \Omega \to \mathbb{R}$, a la que denominan "variable aleatoria original". Así,

el grado de pertenencia de un punto x al conjunto borroso $\tilde{X}(\omega)$ representará el grado de posibilidad de que el verdadero valor $X_0(\omega)$ coincida con x. Los autores disponen, así, de los elementos necesarios para definir una medida de posibilidad sobre el conjunto de todas las variables aleatorias. Para cada variable aleatoria, $X : \Omega \to I\!R$, definen su "grado de aceptabilidad" como el valor: $\text{acc}(X) = \inf_{\omega\in\Omega} \tilde{X}(\omega)(X(\omega))$. La función "acc" toma valores en el intervalo unidad. Por tanto, puede considerarse como la distribución de posibilidad asociada a una medida de posibilidad, $\Pi_{\tilde{X}}$, definida sobre el conjunto de todas las variables aleatorias. Así, $\text{acc}(X)$ representa el grado de posibilidad de que X sea la verdadera variable aleatoria que modela el experimento estudiado. En el caso particular de que la variable aleatoria borrosa sea un conjunto aleatorio (es decir, sus imágenes sean subconjuntos nítidos de $I\!R$), la función de aceptabilidad asigna el valor 1 a cierto conjunto de variables aleatorias y el valor 0 a todas las demás. En el caso aún más particular de que la variable aleatoria difusa coincida con una variable aleatoria clásica (cada imagen sea conjunto con un solo elemento) la función de aceptabilidad asignará el valor 1 a una sola variable aleatoria, que es la verdadera variable aleatoria que modela el experimento. En ese caso su observación es totalmente precisa. En este marco hemos construido (véase [3]) una medida de posibilidad sobre el conjunto de todas las distribuciones de probabilidad en $I\!R$. La distribución de posibilidad, $\pi_{P_{\tilde{X}}}$, que caracteriza dicha medida de posibilidad se define de la forma:

$$\pi_{P_{\tilde{X}}}(Q) = \sup\{\text{acc}(X) \mid P_X = Q\} =$$

$$\Pi_{\tilde{X}}(\{X : \Omega \to I\!R \text{ medible} \mid P_X = Q\}).$$

($\pi_{P_{\tilde{X}}}(Q)$ representa el grado de posibilidad de que la variable aleatoria original sea una de aquéllas que inducen la distribución de probabilidad Q en $I\!R$). La medida de posibilidad $\Pi_{P_{\tilde{X}}}$ es una "posibilidad de segundo orden" formalmente equivalente a las consideradas en [5]. Se llama así, porque es una distribución de posibilidad definida sobre un conjunto de medidas de probabilidad. Una medida de posibilidad representa la misma información que un conjunto de medidas de probabilidad (el conjunto de medidas de probabilidad que están por debajo de ella y por encima de la medida de necesidad dual). Por tanto, una medida de posibilidad de segundo orden está asociada a un conjunto de medidas de probabilidad, cada una de ellas definida, a su vez, sobre un conjunto de medidas de probabilidad. Así, una posibilidad de segundo orden nos permitirá establecer afirmaciones como la siguiente: "la probabilidad de que la verdadera probabilidad del valor 7 sea 0.5 está comprendida entre 0.4 y 0.7".

También de acuerdo con la interpretación posibilística de los conjuntos borrosos, se puede proceder de forma algo diferente para describir la información proporcionada por $\tilde{X}$, siguiendo la línea iniciada en [13] para el caso particular de los conjuntos aleatorios. Supongamos que tenemos información parcial acerca de la distribución de probabilidad que modela un experimento compuesto por dos sub-experimentos cuyos espacios muestrales son Ω y $I\!R$, respectivamente. Supongamos, por una parte que la distribución de probabilidad que modela el primero de ambos, $P : \mathcal{A} \to [0, 1]$ está totalmente determinada[2]. Por otra parte, consideraremos una familia de medidas de posibilidad condicionadas $\{\Pi(\cdot \mid \omega)\}_{\omega\in\Omega}$, cada una de ellas determinada por cada conjunto borroso $\tilde{X}(\omega)$. Esta familia de medidas de posibilidad refleja nuestro conocimiento acerca de la relación entre cada resultado del primer sub-experimento y los posibles resultados del segundo. (Si el resultado del primer experimento es ω, entonces el grado de posibilidad de que en el segundo experimento ocurra x es $\tilde{X}(\omega)(x)$. La combinación, mediante técnicas de extensión natural ([15]), de ambas fuentes de información, nos permitiría describir la información disponible acerca de la distribución de probabilidad existente en $\beta_{I\!R}$ (la distribución de probabilidad que modela el segundo sub-experimento) mediante una probabilidad superior (un modelo de probabilidades imprecisas de orden 1, y no un modelo de orden 2 como el descrito anteriormente). Así, se podrán establecer afirmaciones como la siguiente: "la probabilidad del resultado 7 está comprendida entre 0.3 y 0.6".

3. DEFINICIÓN DE VARIANZA

Cada uno de los tres modelos descritos en la sección anterior puede dar lugar a un tipo diferente de definición de varianza, como se verá a continuación.

3.1. Modelo clásico

Consideremos un espacio de probabilidad, $(\Omega, \mathcal{A}, P)$, y una métrica, d, definida sobre la clase de los subconjuntos borrosos de $I\!R$ (o sobre una subclase) y supongamos que $\tilde{X} : \Omega \to \tilde{\mathcal{P}}(I\!R)$ es una función $\mathcal{A} - \beta(d)$ medible[3].

Definición 3.1 *Llamaremos* varianza clásica *de* $\tilde{X}$ *a la cantidad*

$$\text{Var}_{\text{Cl}}(\tilde{X}) = \int_\Omega d(X, E(\tilde{X}))^2 \, dP.$$

Las distintas definiciones de varianza de la literatura que se ajustan a esta formulación se diferencian entre

[2] $\mathcal{A}$ denota una σ-álgebra de sucesos sobre Ω.

[3] $\beta(d)$ representa la σ-álgebra de Borel inducida por d.

sí en la métrica elegida y en la definición de esperanza de variables aleatorias borrosas considerada. En este aspecto, vamos a comentar brevemente algunos detalles de las definiciones de Körner ([8]) y de Lubiano et al. ([11]). Por una parte, en cuanto al concepto de esperanza, Körner considera la definición de Fréchet ([7]) para funciones medibles con valores en un espacio métrico[4]. El autor comprueba que la esperanza de Puri y Ralescu ([14]) es la única esperanza Fréchet para cierta familia de métricas definidas sobre la clase de conjuntos difusos compactos y normales de $I\!R$ a las que denota genéricamente ρ_2. De acuerdo con lo anterior, dada una distancia ρ_2, define la varianza de la v.a.d. $\tilde{X}$ como la cantidad:

$$\mathrm{Var}_{\rho_2}(\tilde{X}) = \int_{\Omega} \rho_2(\tilde{X}, E_{PR}(\tilde{X}))^2 \, dP.$$

En cuanto a la familia de varianzas definidas por Lubiano et al. en [11], la esperanza considerada es también la de Puri-Ralescu y la clase de distancias es la definida por Bertoluzza et al. en [1] que, a su vez, es una subclase de la familia considerada por Körner. En [8] y [11], se pueden consultar propiedades muy interesantes de las familias de varianzas allí definidas. En este trabajo sólo comentaremos algunos aspectos particulares de las mismas, y para ello nos resultará suficiente mostrar su formulación para el caso particular en que $\tilde{X}$ se reduzca a una aplicación multivaluada (una función cuyas imágenes son subconjuntos "nítidos" del espacio final). En ese caso, se puede comprobar fácilmente que las definiciones de Körner y Lubiano et al. se pueden expresar de la siguiente manera:

$$\mathrm{Var}(\tilde{X}) = \pi_1 \mathrm{Var}(X_1) + \pi_2 \mathrm{Cov}(X_1, X_2) + \pi_3 \mathrm{Var}(X_2),$$

donde $\pi_i \geq 0, i = 1, 2, 3$, $\pi_1 + \pi_2 + \pi_3 = 1$ y X_1, X_2 son las variables aleatorias[5] definidas en Ω como $X_1(\omega) = \inf \tilde{X}(\omega)$ y $X_2(\omega) = \sup \tilde{X}(\omega)$, $\forall \omega \in \Omega$, respectivamente. Así, en el caso particular de que π_2 sea nulo (y, por tanto, $\pi_3 = 1 - \pi_1$), la varianza del conjunto aleatorio resultará ser una combinación lineal convexa de las varianzas de sus extremos. Por otra parte, si $\pi_3 = (1 - \sqrt{\pi_1})^2$, la varianza de $\tilde{X}$ coincide con la varianza de la combinación lineal convexa de X_1 y X_2 dada por la expresión $\sqrt{\pi_1} X_1 + (1 - \sqrt{\pi_1}) X_2$. Es decir, en este caso, para cada elemento del espacio muestral, ω, se puede elegir un punto representativo, $\alpha X_1(\omega) + (1 - \alpha) X_2(\omega)$ (con $\alpha \in [0, 1]$), de la imagen de la v.a.d. y, a continuación, calcular la varianza de la v.a. (clásica) resultante. Con estos ejemplos, observamos que las familias de varianzas así definidas permiten cuantificar la dispersión de los "valores" de $\tilde{X}$ si ésta es considerada como una función medible desde un punto de vista clásico, lo cual puede ser útil cuando las imágenes de $\tilde{X}$ son etiquetas lingüísticas.

Ahora bien, si consideramos la variable aleatoria borrosa como la observación imprecisa de una variable aleatoria, tal como sugieren Kruse y Meyer, esta definición "clásica" de varianza no es, quizás, la más adecuada. Observemos el siguiente ejemplo:

Ejemplo 3.1 *Consideremos, por una parte, un espacio muestral unitario (que modelará un experimento determinista),* $\Omega_1 = \{\omega_1\}$, *dotado de estructura de espacio probabilístico con la única* σ*-álgebra que se puede definir sobre él,* $\mathcal{A}_1 = \mathcal{P}(\Omega_1)$ *y la única medida de probabilidad posible,* P_1*. Definamos un conjunto aleatorio,* $\Gamma_1 : \Omega_1 \to \mathcal{P}(I\!R)$*, de la forma* $\Gamma_1(\omega_1) = [-K, K]$ *Consideremos, por otra parte, el espacio probabilístico correspondiente al lanzamiento de una moneda equilibrada,* $(\Omega_2, \mathcal{A}_2, P_2)$*, y el conjunto aleatorio* $\Gamma_2 : \Omega_2 \to \mathcal{P}(I\!R)$ *definido de la forma* $\Gamma_2(c) = \Gamma_2(+) = [-K, K]$*). En este ejemplo,* Γ_1 *sirve para representar la salida (conocida de forma imprecisa) de un experimento determinista. Por ejemplo, permite representar la cantidad de dinero que, con total seguridad, vamos a recibir, si únicamente sabemos que está comprendida entre* $-K$ *y* K*. Por su parte,* Γ_2 *puede utilizarse para representar la ganancia que obtendremos tras lanzar una moneda equilibrada: la cantidad que vamos a recibir depende del resultado de la moneda y está fijada antes de realizar el experimento, pero nosotros sólo la conocemos de forma imprecisa. Ambos conjuntos aleatorios, considerados como funciones medibles clásicas, inducen la misma distribución de probabilidad (degenerada en el "punto" [-K,K]). De esta forma, ambos tienen la misma esperanza de Aumann (que coincide con su única imagen) y tienen varianza "clásica" nula, ya que ambos son funciones constantes. Sin embargo, si, de acuerdo con la interpretación de Kruse y Meyer, suponemos que cada uno de ellos representa la observación imprecisa de una v.a. clásica, en el primer caso sabemos con seguridad que la varianza de la v.a. original es 0. En el segundo caso sólo sabemos que es un valor comprendido entre 0 y* K^2*.*

Éste es sólo un ejemplo sencillo que sirve para ilustrar las limitaciones de la varianza "clásica" en la cuantificación de la información disponible acerca de la varianza de la v.a. original, cuando se utiliza una interpretación "posibilística" de las v.a. difusas. En la práctica, una v.a.d. puede utilizarse para representar la observación imprecisa de cierta característica de los elementos de una población Ω. Para representar la información que ésta proporciona acerca de la varianza de la v.a.

[4] Fréchet define la esperanza de una función medible, Z, con valores en un espacio métrico (M, d) como una solución, $E^{(d)}(Z)$, (no necesariamente única) del problema: $E[d(Z, E^{(d)}(Z))^2] = \mín_{a \in M} E[d(Z, a))^2]$.

[5] Bajo las condiciones de medibilidad exigidas a $\tilde{X}$ por los autores, las funciones X_1 y X_2 son $\mathcal{A} - \beta_{I\!R}$ medibles.

(clásica) que modela dicha característica debemos recurrir a la varianza definida por Kruse y Meyer.

3.2. Modelo impreciso de 2º orden

En [9], Kruse define la varianza de una variable aleatoria borrosa, $\tilde{X} : \Omega \to \tilde{\mathcal{P}}(I\!R)$, como el único conjunto borroso cuyos α-cortes vienen dados de la forma:

$$[\mathrm{Var}_{\mathrm{Kr}}(\tilde{X})]_\alpha := \{\mathrm{Var}(X) \mid X \in S(\tilde{X}_\alpha)\},$$

donde $S(\tilde{X}_\alpha)$ representa el conjunto de todas las selecciones medibles de la aplicación multivaluada α-corte de $\tilde{X}$.

Se puede comprobar fácilmente que la función de pertenencia de este conjunto borroso viene dada por la expresión:

$$\mathrm{Var}_{\mathrm{Kr}}(\tilde{X})(x) = \sup\{\mathrm{acc}(X) \mid \mathrm{Var}(X) = x\},\ \forall\, x \in I\!R.$$

Observamos que esta definición concuerda con el modelo de posibilidad de orden 2 mostrado en la sección 2. Así, el grado de pertenencia de un valor x al conjunto borroso $\mathrm{Var}_{\mathrm{Kr}}(\tilde{X})$ representa el grado de posibilidad de que la variable aleatoria original sea una de aquéllas cuya varianza coincide con x. En el caso particular de que $\tilde{X}$ sea una aplicación multivaluada, $\mathrm{Var}_{\mathrm{Kr}}(\tilde{X})$ será el conjunto clásico formado por las varianzas de todas sus selecciones medibles. En otras palabras, el conjunto de posibles valores de la varianza de la v.a. original. Así, en el ejemplo 3.1, las respectivas varianzas son, según esta definición, $\mathrm{Var}_{\mathrm{Kr}}(\Gamma_1) = \{0\}$ y $\mathrm{Var}_{\mathrm{Kr}}(\Gamma_2) = [0, K^2]$. Observamos así que la varianza de Kruse permite distinguir entre dos variables aleatorias difusas con la misma distribución de probabilidad "clásica" cuando se utilizan en este contexto. Sin embargo, no siempre asocia distintos valores a dos v.a.d. con distinta varianza "clásica", como veremos en el ejemplo siguiente.

Ejemplo 3.2 *Consideremos el espacio de probabilidad $(\Omega_2, \mathcal{A}_2, P_2)$ del ejemplo 3.1 y el conjunto aleatorio constante Γ_2 definido allí. Definamos, por otra parte, el conjunto aleatorio $\Gamma_3 : \Omega_2 \to \mathcal{P}(I\!R)$ de la forma siguiente: $\Gamma_3(c) = [-K, 0]$ y $\Gamma_3(+) = [0, K]$. En ambos casos, la varianza de Kruse da como resultado el intervalo $[0, K^2]$. Sin embargo, la varianza clásica asignaría el valor 0 a Γ_2 y un valor estrictamente positivo a Γ_3. Este ejemplo nos permite observar que la varianza de Kruse no sirve, en general, para cuantificar la dispersión de las imágenes de una v.a. difusa, cuando ésta se considera como una función medible clásica.*

La varianza de Kruse refleja la imprecisión inherente (en el caso de que ésta exista) en las observaciones de la salida de un experimento aleatorio. Por tanto, da como resultado un conjunto (nítido o difuso) y no un valor real, como la varianza "clásica". En la sección siguiente propondremos un modelo en el que también refleja la imprecisión, pero de manera algo diferente. Se trata de un modelo de probabilidades imprecisas de orden 1 y no de orden 2, de manera que asignará a cualquier v.a.d. un conjunto clásico. Con la ayuda de ejemplos sencillos, trataremos de mostrar las similitudes y diferencias con el modelo presente.

3.3. Modelo impreciso de 1er orden

En esta sección introduciremos una nueva definición de varianza. Para ello, debemos considerar el modelo de probabilidades imprecisas de orden 1 mostrado al final de la sección 2. Tal como señalábamos allí, vamos a considerar, por una parte, la medida de probabilidad, P (definida sobre $\mathcal{A}$), que modela el primer sub-experimento y, por otra, una familia de medidas de posibilidad condicionadas, $\{\Pi(\cdot \mid \omega)\}_{\omega\in\Omega}$, definidas de la forma siguiente:

$$\Pi(A \mid \omega) = \Pi_{\tilde{X}(\omega)}(A) = \sup_{x\in A} \tilde{X}(\omega)(x),\ \forall\, A \in \beta_{I\!R},\ \forall\, \omega.$$

En la fórmula anterior, $\Pi_{\tilde{X}(\omega)}$ representa la medida de posibilidad determinada por la distribución de posibilidad $\tilde{X}(\omega) : I\!R \to [0, 1]$. Así, el valor $\Pi(A|\omega)$ establece una cota superior para la probabilidad de que el resultado final pertenezca a A, bajo la hipótesis de que el resultado del experimento inicial es ω. Esta familia de medidas de posibilidad representa el conocimiento (impreciso) recogido por $\tilde{X}$ acerca de la relación existente entre el resultado del primer sub-experimento y el conjunto de posibles resultados en el segundo. En este contexto, todo lo que conocemos acerca de la distribución de probabilidad que modela el segundo experimento es que pertenece al conjunto:

$$\{Q_2 \mid Q_2 \text{ marginal de } P \text{ y } Q(\cdot|\cdot),\ Q(\cdot|\cdot) \in \mathcal{C}\},$$
$$\text{donde}\quad \mathcal{C} = \{Q(\cdot|\cdot) \text{ probabilidad de transición} \mid Q(A|\omega) \le \Pi(A|\omega)\ \forall\, A \in \beta_{I\!R},\ \omega \in \Omega\}.$$

Puede observarse fácilmente que así se generaliza el concepto de probabilidad inducida por una variable aleatoria clásica: si $\Pi(\cdot|\omega)$ es, en particular, la medida de probabilidad degenerada en un punto $X(\omega)$, entonces el conjunto anterior se reduce al conjunto unipuntual $\{P_X\}$. (En este caso, la probabilidad inducida por $X : \Omega \to I\!R$ en $\beta_{I\!R}$ es la única medida de probabilidad compatible con P y $\Pi(\cdot|\cdot)$). Por otra parte, la varianza de una v.a. clásica, $\mathrm{Var}(X) = \int_\Omega [X - E(X)]^2 dP$, se puede expresar de forma alternativa como la siguiente integral de Lebesgue con respecto a P_X:

$$\mathrm{Var}(P_X) = \int_{I\!R} \left(\mathrm{id} - \int_{I\!R} \mathrm{id}\, dP_X\right)^2 dP_X,$$

donde id:$\mathbb{R} \to \mathbb{R}$ es la función identidad[6]. Así, en el modelo de probabilidades imprecisas propuesto, todo lo que conocemos acerca de la varianza de las salidas del segundo sub-experimento es que pertenece al conjunto:

$$\mathrm{Var}_{\mathrm{Im}\ 1}(\tilde{X}) = \{\mathrm{Var}(Q_2) \mid Q_2 \text{ marginal de } P \text{ y } Q(\cdot|\cdot),$$

$$Q(\cdot|\cdot) \in \mathcal{C}\}.$$

Se puede comprobar el siguiente resultado acerca del conjunto $\mathrm{Var}_{\mathrm{Im}\ 1}(\tilde{X})$, que nos permitirá facilitar los cálculos (omitiremos la demostración por falta de espacio):

Teorema 3.1 $\mathrm{Var}_{\mathrm{Im}\,1}(\tilde{X}) = \{\mathrm{Var}(Q) \mid Q(A) \leq \overline{P}_{\tilde{X}}(A),\ \forall A \in \beta_{\mathbb{R}}\}$ *donde* $\overline{P}_{\tilde{X}}$ *es la función de conjuntos subaditiva dada por*

$$\overline{P}_{\tilde{X}}(A) = \int_0^1 P^*_{\tilde{X}_\alpha}(A)\, d\alpha, \forall\, A \in \beta_{\mathbb{R}}$$

y, para cada $\alpha \in (0,1]$, $P^*_{\tilde{X}_\alpha}$ *es la probabilidad superior de Demspter del conjunto aleatorio* $\tilde{X}_\alpha$.

$\mathrm{Var}_{\mathrm{Im}\,1}(\tilde{X})$ es el conjunto de posibles valores de la varianza del segundo sub-experimento de acuerdo con la información disponible. Vamos a comparar en un ejemplo la información proporcionada por $\mathrm{Var}_{\mathrm{Im}\,1}$ y $\mathrm{Var}_{\mathrm{Kr}}$ acerca de la varianza de la distribución de probabilidad "original".

Ejemplo 3.3 *Consideremos el intervalo unidad,* $\Omega = [0,1]$*, dotado de estructura de espacio probabilístico mediante la medida de Lebesgue. Consideremos la variable aleatoria borrosa* $\tilde{X} : \Omega \to \tilde{\mathcal{P}}(\mathbb{R})$ *constante en el conjunto difuso* $\tilde{A}$ *determinado por los* α*-cortes* $\tilde{A}_\alpha = [-(1-\alpha), 1-\alpha]$*. Se puede comprobar fácilmente que* $[\mathrm{Var}_{\mathrm{Kr}}(\tilde{X})]_\alpha = [0, (1-\alpha)^2]$, $\forall\, \alpha > 0$*. Por otra parte, si nos apoyamos en el resultado del teorema 3.1, observamos que* $\mathrm{Var}_{\mathrm{Im}\,1}(\tilde{X})$ *es el intervalo [0,1/3]. Observamos claramente que este intervalo está estrictamente contenido en el soporte de* $\mathrm{Var}_{\mathrm{Kr}}(\tilde{X})$*. Así, bajo el modelo de primer orden aquí descrito, la varianza de los resultados del experimento es, con seguridad, inferior o igual a 1/3, mientras que, según el modelo de segundo orden, se asigna un grado de posibilidad estrictamente positivo a los valores comprendidos entre 1/3 y 1.*

Pasemos ahora a examinar la relación entre ambos modelos en el caso particular de que $\tilde{X}$ sea un conjunto aleatorio. En ese caso, de acuerdo con el teorema 3.1, se observa la igualdad

$$\mathrm{Var}_{\mathrm{Im}\,1}(\tilde{X}) = \{\mathrm{Var}(Q) \mid Q(A) \leq P^*(A),\ \forall\, A \in \beta_{\mathbb{R}}\},$$

donde P^* es la probabilidad superior del conjunto aleatorio. Por otra parte, se satisface claramente la igualdad

$$\mathrm{Var}_{\mathrm{Kr}}(\tilde{X}) = \{\mathrm{Var}(Q) \mid Q = P_X, \text{ con } X \in S(\Gamma)\}.$$

Dado que el conjunto de medidas de probabilidad $P(\Gamma) = \{P_X \mid X \in S(\Gamma)\}$ está contenido en $\mathrm{M}(P^*) = \{Q \mid Q(A) \leq P^*(A),\ A \in \beta_{\mathbb{R}}\}$, la varianza de Kruse es, en este caso, un conjunto contenido en la varianza del modelo de primer orden aquí descrito. En el caso de que $P(\Gamma)$ esté estrictamente contenido en $\mathrm{M}(P^*)$ existirá la posibilidad de que $\mathrm{Var}_{\mathrm{Kr}}(\tilde{X})$ esté también estrictamente contenida en $\mathrm{Var}_{\mathrm{Im}\,1}(\tilde{X})$. Las relaciones existentes entre $P(\Gamma)$ y $\mathrm{M}(P^*)$ han sido estudiadas con detalle en [3] y [4], entre otros. A continuación mostramos un ejemplo en el que ambos conjuntos son distintos y esta diferencia se refleja en el cálculo de la varianza.

Ejemplo 3.4 *Volvamos a considerar los conjuntos aleatorios considerados en el ejemplo 3.1. Según el modelo descrito en esta sección, en el primer caso, el primer sub-experimento es determinista y la relación entre los dos sub-experimentos viene determinada por* Γ_1*. Este conjunto aleatorio representa una distribución de posibilidad condicionada "vacía" sobre* $[-K, K]$*. Así que el conjunto de medidas de probabilidad condicionadas,* $Q(\cdot|\omega_1)$*, compatibles con ella es el conjunto de todas las medidas que asignan probabilidad 1 al conjunto* $[-K, K]$*. Así, se refleja la siguiente información: una vez realizado el primer experimento, se elige un número al azar entre* $-K$ *y* K*, no fijado de antemano. En el caso de* Γ_2*, el primer subexperimento consiste en el lanzamiento de una moneda. Una vez observado el resultado, se elige, cualquiera que sea éste (cara o cruz), un número al azar entre* $-K$ *y* K*. Así, en este ejemplo se ve intuitivamente que, en lo que concierne a la salida del segundo sub-experimento, se podría obviar el lanzamiento de la moneda (no como en el caso del modelo de orden 2) y, por tanto,* Γ_1 *y* Γ_2 *reflejan, desde este punto de vista, la misma información. Así, observamos que* $\mathrm{Var}_{\mathrm{Im}\,1}(\Gamma_1) = \mathrm{Var}_{\mathrm{Im}\,1}(\Gamma_2) = [0, K^2]$.

Pasemos ahora a comentar algunas relaciones existentes entre la varianza correspondiente a este modelo impreciso de orden 1 y la varianza clásica de la sección 3.1. Se puede comprobar que las probabilidades superiores $P^*_{\tilde{X}_\alpha}$, $\alpha \in (0,1]$ se pueden expresar como funciones de la medida de probabilidad inducida por $\tilde{X}$, $P_{\tilde{X}}$. Así, de acuerdo con el resultado mostrado en

[6]Dado que la varianza de una v.a. clásica es función de su distribución de probabilidad inducida, a partir de ahora, cometeremos un pequeño abuso del lenguaje y la expresaremos como la varianza de dicha distribución de probabilidad.

el teorema 3.1, se observa que la varianza del modelo de probabilidades imprecisas considerado en esta sección es función de la medida de probabilidad inducida por $\tilde{X}$. Ahora bien, no es función, en general, de la varianza clásica, tal como mostraremos en el siguiente ejemplo:

Ejemplo 3.5 *Consideremos, por una parte, el conjunto aleatorio* Γ_1 *definido en el ejemplo 3.1 y, por otra, el conjunto aleatorio,* Γ_5*, definido sobre el mismo espacio, de la forma* $\Gamma_5(\omega_1) = \{0\}$*. La varianza "clásica" asigna el valor 0 a ambos conjuntos aleatorios, mientras que la varianza imprecisa asigna, al primero el conjunto de valores* $[0, K^2]$ *y, al segundo, el conjunto unipuntual* $\{0\}$.

Igualmente, se puede comprobar que la varianza clásica tampoco se puede expresar como función de la varianza considerada en esta sección. Para ello basta observar los conjuntos aleatorios del ejemplo 3.2.

4. CONCLUSIONES

En este trabajo hemos estudiado distintas formas de generalizar el concepto de varianza de una variable aleatoria real al caso de variables aleatorias borrosas. En el trabajo de Körner ([8]) se establece una definición más general para el caso en que el espacio final es $\mathbb{R}^n$, con $n \in \mathbb{N}$ arbitrario. En aquel trabajo se define la varianza de $\tilde{X}$ como la esperanza del cuadrado de la distancia de sus imágenes a su esperanza Fréchet. En el caso particular de que $\tilde{X}$ sea un vector aleatorio clásico y la distancia elegida sea la euclídea, el resultado de ese cálculo es un momento de inercia. De esta manera, en el caso n-dimensional se generaliza, mediante el procedimiento de Körner, un concepto que puede tener utilidad en la medición de la dispersión de las imágenes de la variable aleatoria difusa, pero no el concepto de matriz de varianzas-covarianzas. Si, por el contrario, lo que se pretende es generalizar este último, quizás se puede utilizar el procedimiento sugerido por Kruse sin demasiadas modificaciones. Mediante razonamientos similares a los del autor, se puede obtener como resultado un conjunto borroso sobre una la clase de matrices cuadradas que asocie, a cada matriz particular, un grado de posibilidad. Se indicará, con este conjunto difuso, el conocimiento impreciso disponible acerca de la matriz de varianzas-covarianzas del vector aleatorio "original". En [12], Meyer propone una definición de covarianza siguiendo razonamientos similares a los de Kruse. Nuestra intuición nos dice que, probablemente, la combinación de las informaciones proporcionadas por separado acerca de la varianza de cada componente del vector y de la covarianza entre ellas será más imprecisa que la información proporcionada directamente acerca de la matriz de varianzas-covarianzas.

Referencias

[1] C. Bertoluzza, A. Salas, N. Corral (1995) On a new class of distances between fuzzy numbers. *Mathware and Soft Computing* **2** 71-84.

[2] I. Couso, S. Montes, P. Gil (1998) Función de distribución y mediana de variables aleatorias difusas, *Actas del Congreso ESTYLF'98*. Pamplona, España.

[3] I. Couso, S. Montes, P. Gil (2002) Second order possibility measure induced by a fuzzy random variable. En *Statistical modeling, analysis and management of fuzzy data* (eds: C. Bertoluzza, M. A. Gil y D. A. Ralescu), 127-144. Physica-Verlag, Heidelberg, Alemania.

[4] I. Couso, L. Sánchez, P. Gil (2004) Imprecise distribution function associated to a random set, *Information Sciences* **159** 109-123.

[5] G. de Cooman, P. Walley (2002) An imprecise hierarchical model for behaviour under uncertainty, *Theory and Decision* **52** 327-374.

[6] R. Féron (1976) Ensembles aléatoires flous, *C.R. Acad. Sci. Paris Ser. A* **282** 903-906.

[7] M. Fréchet (1948) Les éléments aleatoires de natures quelconque dans un éspace distancié, *Ann. Inst. H. Poincaré* **10** 215-310.

[8] R. Körner (1997), On the variance of fuzzy random variables, *Fuzzy Sets and Systems* **92** 83-93.

[9] R. Kruse (1987) On the variance of random sets, *J. Math. Anal. Appl.* **122** 469-473.

[10] R. Kruse, K.D. Meyer (1987) *Statistics with vague data* D. Reidel Publishing Company.

[11] M.A. Lubiano, M.A. Gil, M. López-Díaz, M. T. López-García (2000), The $\vec{\lambda}$-mean squared dispersion associated with a fuzzy random variable, *Fuzzy Sets and Systems* **111** 307-317.

[12] K.D. Meyer, R. Kruse (1990) On calculating the covariance in the presence of vague data. *En: Progress in Fuzzy Sets and Systems.* W.H. Janko, M.Roubens y H.J. Zimmermann (Eds.) Kluwer Academic Publishers, Dordrecht, Alemania.

[13] E. Miranda, G. de Cooman, I. Couso (2004) Imprecise probabilities induced by multi-valued mappings, *J. Stat. Plann. Inference.* En prensa.

[14] M.L. Puri, D. Ralescu (1986) Fuzzy Random Variables, *J. Math. Anal. Appl.* **114** 409-422.

[15] P. Walley (1991) *Statistical Reasoning with Imprecise Probabilities.* Chapman and Hall, Londres, Reino Unido.

Lógica Difusa en Control de Calidad de Procesos

M.Delgado C.
Dpto. de Ciencias de la Computación e Informática.
Universidad de Granada España.
e-mail: mdelgado@ugr.es

P. Olavarrieta S.
Dpto. de Electricidad.
Universidad Tecnológica Metropolitana. Chile.
e-mail: polavarr@utem.cl

P. Vergara V.
Dpto. de Industria
Universidad Tecnológica Metropolitana Chile
e-mail: pvergara@utem.cl

Resumen

El control de calidad de procesos industriales hasta ahora se han realizado a través de los clásicos gráficos de Shewhart y de gráficos de control con límites probabilísticos, utilizando los estadísticos muestrales $\bar{x}$ y s o sus derivados C_p y C_{pk}, para determinar si el proceso es preciso o poco preciso. La estadística se ha formulado empleando matemática crisp, sin embargo entrega formas impregnadas de vaguedad. Por ejemplo, se habla de procesos precisos o poco precisos. Por este motivo parece natural e interesante introducir herramientas que proceden del ámbito de la lógica difusa para la formulación de modelos para control de calidad.

Se propone realizar el estudio de la capacidad de calidad de un proceso, generar un control simultáneo bilateral para la tendencia central y unilateral para la variabilidad, con lógica difusa. Se definen reglas lingüísticas para realizar este control y se definen funciones de pertenencia para $\bar{x}$ y s.

Palabras Clave: Control, Fuzzy, Estadístico, Calidad

1 Introducción

La característica imprecisa que resulta en los criterios empleados en control de calidad, lleva a preguntarse si el precio que se paga por trabajar con intervalos de valores amplios, tiene otra alternativa. La incorporación de conceptos de calidad a la producción lleva a su vez a encajonar objetos en ellos (producción aceptada/rechazada, o proceso preciso/poco preciso) lo que a su vez muestra que hay objetos que pueden estar en un concepto o en el otro, pero los hay que podrían estar en ambos. Se dirá que existe una zona difusa en que quedan *encajonados* estos últimos objetos. Desde mediados de la década de los sesenta se conoce una sistematización de la lógica difusa [12] que ha permitido una aplicación creciente de ella y permite el trabajo de objetos que tienen una zona difusa.

Lo que se ha hecho hasta la fecha es un formalismo que obtiene resultados precisos con grandes números de producción, a pesar de ello maneja la imprecisión en muestras, pero no trata el traslape de imprecisión entre dos medidas (el concepto de zona difusa). La lógica difusa presenta características de ser una solución a los problemas planteados, ya que opera con objetos naturalmente imprecisos.

Una observación por lo cual se ha visto poco trabajo publicado bajo esta perspectiva, es que el *control* de la calidad de un proceso por muestreo estadístico, tiene pocas características del control realimentado normal que se usa en maquinaria. En control de calidad, el lazo motríz de realimentación lo forma un operario experto, que debe detener una máquina en producción, cuando esta produce fuera de especificación, averiguar la causa y reparar su falla. Este último paso ha evolucionado a sistemas con un banco de datos que describe fallas y de redes neuronales que diagnostican la causa de falla [2].

Se propone estudiar la aplicación de Lógica Difusa en control de calidad de procesos, con el fin de obtener modelos que permitan manejar la incertidumbre no probabilística que es intrínseca en sus medidas. Existen propuestas en la literatura, como los trabajos de Rowlands[6], Wang et al[10] y Raz et al[5]; en los que se puede colegir que quedan espacios no cubiertos en el tema.

Se realizarán estudios en base a la capacidad de calidad del proceso (Cp y Cpk) [4], [7], [8] y [9] ya que es a través de ellos que se define la precisión del proceso. De esto se propondrán métodos para controlar

un proceso respecto del valor nominal asignado a la característica, que se controlará utilizando líneas de atención y de intervención en forma bilateral para la tendencia central, y un control simultáneo unilateral de la variabilidad del proceso, con los usuales riesgos del 5% y 0,1% respectivamente, ahora sobre una base difusa.

2 Marco estadístico.

El producto de un proceso de fabricación debe ser especificado en a lo menos una magnitud fundamental (p.e. resistencia de un cinturón de seguridad), ella se conoce como centro de tolerancia CT. Alrrededor de ella debe haber un valor mínimo, la tolerancia inferior TI y uno máximo, la tolerancia superior TS . Cuando el proceso entrega piezas dentro del intervalo de tolerancia $TI - TS$, se dice que estas son aceptables, en caso contrario son inaceptables o de rechazo. El proceso de fabricación a su vez está inherentemente afectado de causas que producen desviaciones de la medida central especificada, conocidas como causas asignables y de azar, ello lleva a que la produción entregará en algún momento piezas defectuosas. El control estadístico de calidad de un proceso permite identificar si la desviación observada es debida a causa asignable o del azar. Si se está dentro de los límites del azar, el proceso está bajo control, si no se está fuera de control.

2.1 Indicadores de capacidad del proceso.

Antes de producir es necesario hacer una prueba de factibilidad, por lo que se debe hacer un muestreo tal que permita tener una buena estimación del valor central y la variabilidad del proceso. Se recomienda [4] tomar en K intervalos muestras de tamaño n de modo que $Kn > 100$. De esto salen los estadígrafos estimados m y $\hat{\sigma}$. Si se define [4] por k el número de $\hat{\sigma}$ que contiene el intervalo de tolerancia especificado, a su vez define un índice de variavilidad C_p:

$$C_p = \frac{k}{6} \tag{1}$$

Si C_p es mayor que 1 el proceso es preciso y si es menor que 1 es poco preciso.

Si se define descentramiento [4] D entonces se puede definir un índice de variavilidad bajo descentramiento C_{pk}:

$$C_{pk} = \frac{k}{6} - \frac{D}{3} \tag{2}$$

2.2 Proceso bajo especificación.

Un proceso se encuentra bajo especificación, cuando desde m hasta el extremo más próximo (TI o TS), existan más de $3\hat{\sigma}$ de distancia, es decir:

$$1 \leq C_p - \frac{D}{3} = C_{pk} \tag{3}$$

Entonces un proceso bajo especificación es un proceso descentrado bajo control. Con el objetivo de trabajar [4] con valores continuos de k y D, se realizó una modelación del riesgo de rechazo en función de k y D, usándose posteriormente para modelar C_p y C_{pk}, ambos en planilla de cálculo Excel. Esto se muestra en la tabla 1 y la tabla 2 en que aparecen los valores de C_p y C_{pk}, como variables derivadas de las anteriores. Se aprecia que el C_{pk} reproduce el valor original de C_p cuando al primero se le suman $2D$ al valor de k sin descentramiento.

3 Control del proceso.

Estas metodologías se hacen operativas en la práctica como cartas de control provenientes de pruebas de hipótesis con riesgo al 5 % y 0,2 %. Se debe realizar una prueba de hipótesis para cada variable a controlar.

3.1 Zonas de decisión.

En un proceso preciso se ha definido un nivel de confianza de 99,73% o mayor, que contiene a lo menos 6σ dentro del intervalo de tolerancia especificado, el resto define la zona de rechazo. Usualmente se define una probabilidad del 95% para definir una interzona que se llamará zona de atención. Quedan tres zonas, una de rechazo, una de control y en el medio una zona de atención. Veremos primero el caso preciso, dejando el poco preciso para el final a fin de considerar desde donde se llega al caso preciso, relación(3).

3.2 Control de $\bar{x}$. Caso preciso

Para el caso de control de la media muestral $\bar{x}$, se definen las funciones ζ y Ψ en que n, es el tamaño de la muestra, como:

$$\zeta = 3.09(1 - \frac{1}{\sqrt{n}}) \tag{4}$$

$$\Psi = 1 + \frac{0.3657}{\sqrt{n} - 1} \tag{5}$$

Estas son evaluadas en la Tabla 3, para muestras de tamaño 3 a 11.

Tabla 1: Riesgo de rechazo en función de k y D. Indicadores derivados C_p y C_{pk}

D	k	3,0	3,5	4,0	4,5	5,0	5,5	6,0	6,5
0,00	C_p	0,5	0,58333	0,66667	0,75	0,83333	0,91667	1,0	1,08333
	p	0,13721	0,08254	0,04684	0,02508	0,01266	0,00603	0,00271	0,00115
0,25	C_{pk}	0,41667	0,5	0,58333	0,66667	0,75	0,83333	0,91667	1,0
	p	0,14887	0,09203	0,05381	0,02975	0,01555	0,00769	0,00359	0,00159
0,50	C_{pk}	0,33333	0,41667	0,5	0,58333	0,66667	0,75	0,83333	0,91667
	p	0,18263	0,12014	0,07494	0,04429	0,02478	0,01311	0,00656	0,0031
0,75	C_{pk}	0,25	0,33333	0,41667	0,5	0,58333	0,66667	0,75	0,83333
	p	0,23477	0,16554	0,11062	0,06996	0,04184	0,02365	0,01262	0,00636
1,00	C_{pk}	0,16667	0,25	0,33333	0,41667	0,5	0,58333	0,66667	0,75
	p	0,29899	0,22525	0,16057	0,10818	0,06884	0,04136	0,02345	0,01255
1,25	C_{pk}	0,08333	0,16667	0,25	0,33333	0,41667	0,5	0,58333	0,66667
	p	0,3666	0,29402	0,22281	0,15944	0,10769	0,06864	0,04128	0,02342
1,50	C_{pk}	0	0,08333	0,16667	0,25	0,33333	0,41667	0,5	0,58333
	p	0,42749	0,36416	0,29289	0,22232	0,15924	0,10761	0,06861	0,04127

Tabla 2: Continuación Tabla 1 con k entre 6,5 y 10,0

D	k	6,5	7,0	7,5	8,0	8,5	9,0	9,5	10,0
0,00	Cp	1,08333	1,16667	1,25	1,33333	1,41667	1,5	1,58333	1,66667
	p	0,00115	0,00046	0,00017	6,2E-05	2,1E-05	6,6E-06	2E-06	5,6E-07
0,25	Cpk	1,0	1,08333	1,16667	1,25	1,33333	1,41667	1,5	1,58333
	p	0,00159	0,00066	0,00026	9,7E-05	3,4E-05	1,1E-05	3,6E-06	1,1E-06
0,50	Cpk	0,91667	1,0	1,08333	1,16667	1,25	1,33333	1,41667	1,5
	p	0,0031	0,00139	0,00059	0,00023	8,8E-05	3,1E-05	1E-05	3,3E-06
0,75	Cpk	0,83333	0,91667	1,0	1,08333	1,16667	1,25	1,33333	1,41667
	p	0,00636	0,00303	0,00136	0,00058	0,00023	8,7E-05	3,1E-05	1E-05
1,00	Cpk	0,75	0,83333	0,91667	1,0	1,08333	1,16667	1,25	1,33333
	p	0,01255	0,00634	0,00302	0,00136	0,00058	0,00023	8,7E-05	3,1E-05
1,25	Cpk	0,66667	0,75	0,83333	0,91667	1,0	1,08333	1,16667	1,25
	p	0,02342	0,01254	0,00633	0,00302	0,00136	0,00058	0,00023	8,7E-05
1,50	Cpk	0,58333	0,66667	0,75	0,83333	0,91667	1,0	1,08333	1,16667
	p	0,04127	0,02342	0,01254	0,00633	0,00302	0,00136	0,00058	0,00023

Tabla 3: Evaluación de ζ y Ψ para muestras de tamaño 3 a 11.

$\zeta \backslash n$	3	4	5	6	7	8	9	10	11
ζ	1,31	1,55	1,71	1,83	1,92	2,00	2,06	2,11	2,16
$\zeta\Psi$	1,96	2,11	2,21	2,29	2,35	2,40	2,44	2,47	2,50
$(\zeta\Psi - \zeta)/2$	0,33	0,28	0,25	0,23	0,21	0,20	0,19	0,18	0,17

Tabla 4: Υ_{01} y Υ_{25} para riesgos de rechazo de 2,5 y 0,1%.

$\Upsilon \setminus n$	3	4	5	6	7	8	9	10	11
Υ_{01}	2,63	2,33	2,15	2,03	1,93	1,86	1,81	1,76	1,72
Υ_{25}	1,92	1,77	1,67	1,60	1,55	1,51	1,48	1,45	1,43
$(\Upsilon_{01}$-$\Upsilon_{25})/2$	0,35	0,28	0,24	0,21	0,19	0,18	0,16	0,15	0,14

Tabla 5: Valores de $1,58\beta$ *y* β para tamaños muestrales entre 3 y 11.

$\beta \setminus n$	3	4	5	6	7	8	9	10	11
$1,58\beta$	1,78	1,55	1,38	1,26	1,17	1,09	1,03	0,98	0,93
β	1,13	0,98	0,88	0,80	0,74	0,69	0,65	0,62	0,59
$(1,58\beta - \beta)/2$	0,33	0,28	0,25	0,23	0,21	0,20	0,19	0,18	0,17

Las líneas de atención inferior y superior definen entre ellas la zona de control y fuera de ellas la zona de atención:

$$MAI = TI + \zeta\ \Psi\ \hat{\sigma}$$
$$MAS = TS - \zeta\ \Psi\ \hat{\sigma}$$

Las líneas de intervención inferior y superior definen entre ellas y hasta las líneas de atención mas próximas la zona atención y valores mayores que ellas definen la zona de intervención:

$$\begin{aligned} MII &= TI + \zeta\ \hat{\sigma} \\ MIS &= TS - \zeta\ \hat{\sigma} \end{aligned}$$

3.3 Control de la variabilidad s. Caso preciso

Para resolver si s está dentro de la desviación estadística que permite la prueba de hipótesis, será necesario probar si la variabilidad de la muestra está en la zona de control o a lo menos en la zona de atención, de acuerdo a las relaciones(6) y (7). Para ello, sean $\Upsilon_{25,n}$ y $\Upsilon_{01,n}$ definidos por la dócima unilateral de χ^2.

$$\Upsilon_{25,n} = \sqrt{\frac{\chi^2(p \leq 0.025, \nu = n-1)}{n-1}}$$

$$\Upsilon_{01,n} = \sqrt{\frac{\chi^2(p \leq 0.001, \nu = n-1)}{n-1}}$$

Son función del tamaño de la muestra y del riesgo de rechazo y son evaluados en la Tabla 4.

Las líneas de atención son:

$$SAS = \hat{\sigma}\Upsilon_{25,n} \tag{6}$$

Línea de intervención son:

$$SIS = \hat{\sigma}\Upsilon_{01,n} \tag{7}$$

3.4 Reglas de control

Cuando ambos estadígrafos se encuentran en zona de no rechazo de H_o, el proceso está bajo control tanto en tendencia central como en variabilidad. Cuando uno o ambos estadígrafos se encuentran en la zona de atención pero no en la de intervención, el proceso se encuentra en posible fuera de control, para dilucidarlo se necesita tomar una muestra instantánea, si los nuevos estadígrafos del muestreo se sitúan en la misma zona inicial, o se ha(n) extendido(s) a la zona de intervención, se debe intervenir el proceso inmediátamente. Si en caso contrario se sitúan en zona de control exclusívamente, se trató de una falsa alarma. Cuando uno de los estadígrafos se encuentra dentro de la zona de intervención, el proceso se debe intervenir inmediátamente, pues está fuera de control.

3.5 Control de caso poco preciso.

Un caso poco preciso es aquel en que en el intervalo de tolerancia caben inicialmente menos de $6\hat{\sigma}$, o que habiéndolo hay descentramiento. Diversas consideraciones de tipo práctico pueden llevar a trabajar con un proceso en estas condiciones, por lo que se darán unas directivas para su control. Se realiza en este caso un control de la media $\bar{x}$, desde el CT. Sea β función n:

$$\beta = \frac{1.96}{\sqrt{n}} \tag{8}$$

Definición de las líneas de atención

$$MA = CT \pm \beta\hat{\sigma} \tag{9}$$

Definición de las líneas de intervención:

$$MI = CT \pm 1,58\beta\hat{\sigma} \tag{10}$$

Estas se han evaluado en la Tabla 5 y definen un control de $\bar{x}$ alrededor de CT, lo cual en esencia controla el descentramiento, sin embargo no controla la variabilidad. En este caso deberá bajarse el límite de intervención al número de k con que se aceptó la prueba de factibilidad. Juntando el control de la media y de la variabilidad para el caso poco preciso, con los límites de zonas acomodados de acuerdo a la prueba de factibilidad, se debe operar con la metodología de la subsección (3.4)

4 Variables de control difusas.

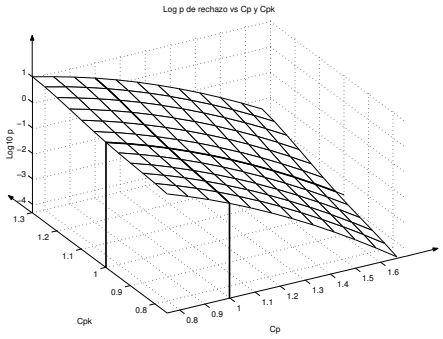


Figura 1: Límite de C_p cuando pasa de poco preciso a preciso. Permite definir una zona difusa.

En la Tabla (1) y la Tabla (2) se muestran algunos valores de una modelación continua del riesgo de rechazo en función de k y D; mediante una adaptación de los datos al manejador de gráficos de Matlab se consigue la figura (1). Ella muestra el comportamiento monótono que tiene la probabilidad de fabricar una pieza defectuosa en función de C_p y C_{pk}, esto permite de acuerdo

a lo anterior, definir directamente funciones difusas de C_p y C_{pk}, de forma triangular o definir la funcion difusa de s. Usaremos la cuarta línea de las Tablas (3) y (4) para fijar el intervalo difuso de C_p. Esto se hace posible si solamente el intervalo definido es igual o mayor que la imprecisión de la medida obtenida [11], [3] , [1]. Para muestreo de cinco o más tomas, se pueden modelar intervalos difusos siguientes:

1. Zona de transición control-atención:

$$1,05\sigma \pm 0,25\sigma \tag{11}$$

2. Zona de transición atención-fuera de control:

$$2,75\sigma \ \leftrightarrow \ 3\sigma \tag{12}$$

4.1 Conversión del indicador C_p

Si tomamos la zona difusa un poco más ancha, digamos del orden de un σ (la diferencia entre 3,09 y 1,96, que da 1,13), podemos obtener un C_p del caso poco preciso:

$$C_p = \frac{6,18 - 1,13}{6,18} = 0,82 \tag{13}$$

Y para el caso preciso:

$$C_p = \frac{6,18 + 1,13}{6,18} = 1,18 \tag{14}$$

Véase la figura (2).

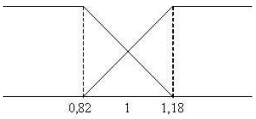


Figura 2: Zona difusa de la variable C_p.

5 Simulación del motor de inferencia

Se hace una modelación simple de $\bar{x}$ y de s con una salida de alarma que se codifica en tres valores V, A y R por los colores verde, amarillo y rojo. La variable se denomina Campana. Para $\bar{x}$ definimos IN e IP, como zonas de intervención negativa y positiva, AN y AP, como zonas de atención negativa y positiva y C como zona bajo control. Para s se definen C, A e I, simplemente.

5.1 Motor difuso

Se ha implementado un motor de inferencia difuso con las entradas usuales del control de calidad clásico, $\bar{x} \in [-3, 3]$ y $\sigma \in [1, 3]$, modelándose la función de pertenencia para el promedio de la muestra en la figura (3).



Figura 3: Función de pertenencia de $\bar{x}$

La modelación de la función de pertenencia de la respuesta lógica del motor de inferencia, en la figura (4). Se muestran también como el sistema de inferencia al desfuzificar su salida entrega una respuesta plana en la zona de control y atención, disminuyendo a cero su respuesta a medida que nos alejamos de dicha zona.

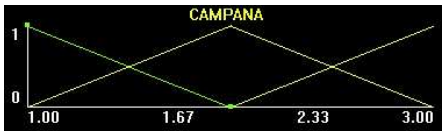


Figura 4: Salida del motor de inferencia

Como son gráficos en dos dimensiones se graficará la salida para una de las dos entradas, manteniendo la otra en un valor constante. En la figura siguiente (5) se muestra a campana en función de sigma de la muestra para un valor de 1,5 de la media.

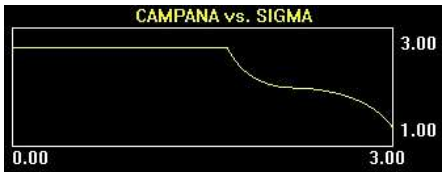


Figura 5: Desfuzificado en función de s para $\bar{x} = 1{,}5$

La respuesta de salida en función de las dos variables difusas de entrada, realisado esto en Matlab, se ve en la figura (6)para valores altos de la función de salida el sistema est bajo control y se sale de l si $\bar{x}$ se va a los extremos o si s toma valores grandes.

6 Discusión

Con respecto a lo hecho se ve que cuando la producción es calificada de poco precisa el método de control ad hoc empieza a ser paulatinamente preponderante sobre

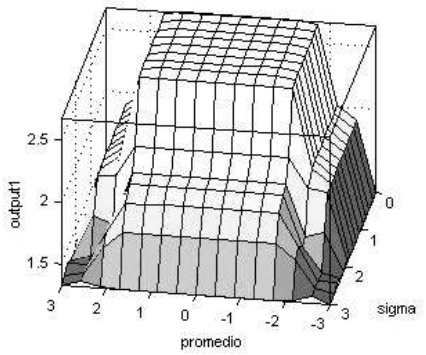


Figura 6: Salida desfuzificada en función de s y $\bar{x}$

el preciso a medida que dicha característica se agranda, hasta ser 100% poco preciso. Lo contrario sucede a medida que nos adentramos en la zona precisa. El sistema tiene mayor tolerancia con respecto a variaciones de la desviación estándar grandes como se ve en la Tabla 3 y 4, al contrario de lo que se observa en la Tabla 5, que es para el caso poco preciso.

Referencias

[1] Bissel A F. How reliable is your capability index?. *App. Stat.* **39** Pág.339-40, 1990

[2] Guh Ruey-Shiang. Integrating Artificial intelligence into on-line stadistical process control. *Qual. Reiab..Engng. Int.*, **19** Pág.1-20, 2003

[3] Kushler R, Hurley P. Confidence bound for capability indices. *Journal of Quality Technology*, **4**, 1992

[4] Pepio, C. Polo, M. Vergara, P. Análisis crítico de los índices de calidad C_p y C_{pk} *Trilogía*, **18** Pág.10-23, 1990-91.

[5] Raz T, Wang JH. Probabilistic and membership approaches in the construction of control chart for linguistic data. *Production planning and control*, **1** Pág.157-167, 1998

[6] Rowlands, H. Wang L. R. An approach of fuzzy logic evaluation and control in SPC. *Qual. Reliab. Engng.* **16** Pág.91-98, 2000.

[7] Vergara, P. Nuevas técnicas acumulativas de control de calidad. Control conjunto por medias móviles, ponderadas exponencialmente para la tendencia central y variabilidad.*Tesis doctoral* ISBN 84-7653-206-7 depósito legal B.2.202.92 Universidad Politécnica de Cataluña,1992.

[8] Vergara P. Comparación de la Eficiencia entre Métodos de Control de Calidad. , Barrera R. *Actas del VI Congreso de Estadística de la Zona Sur.* Universidad de la Frontera,1994.

[9] Vergara P., Gonzalez H., Sepúlveda C., Buccioni R. Efectos de la introducción de Señales en un Método de Control de Calidad Acumulativo. *Actas del VI Congreso de Estadística de la Zona Sur.* Universidad de la Frontera,1994.

[10] Wang JH, Tzvy R. Applying ffuzy set theory in the development of quality control chart. *International Industrial Engeneering Conference Proceedings*, Pág.30-35, 1998.

[11] Yongting C. Fuzzy quality analysis on fuzzy probabilities. *Fuzzy set and system*, **88** Pág.283-90, 1996.

[12] Zadeh L.A. Fuzzy Sets. *Information Control*, **8** Pág. 338-353, 1965.

RELACIONES CAUSALES EN REGLAS DE ASOCIACION

M. Amparo Vila F.
Universidad de Granada
DECSAI-España
vila@decsai.ugr.es

Daniel Sánchez F.
Universidad de Granada
DECSAI-España
daniel@decsai.ug.es

Luis Escobar R.
U. Tecnológica Metropolitana
Depto de Informática Chile
laescoba@utem.cl

Resumen

A pesar de que existen técnicas desarrolladas para descubrir relaciones causales entre datos controlados (o experimentales), no está resuelto como puede hacerse en datos solamente observados (capturados y almacenados, y sobre los cuales no es posible experimentar). Para encontrar relaciones causales en datos observados, se han usado métodos basados en grafos, los que resultan excesivamente complejos cuando son aplicados en un típico gran conjunto de datos.

La teoría de la inferencia causal en Minería de Datos consta de dos partes : la inferencia causal en datos estadísticos (la que contempla la matemática, y la filosofía causal) y la parte de Minería de Datos que tiene que ver con técnicas que permitan seleccionar un conjunto de datos observados posibles de relacionarse causalmente. Todo esto en grandes volúmenes de datos almacenados en Base de Datos.

Las técnicas existentes de minería causal se basan en técnicas más complejas de descubrimiento causal en datos estadísticos y en datos experimentales. De esta manera a partir de la filosofía causal y la matemática es posible también inferir relacione causales en grandes Bases de Datos. Es necesario destacar que la búsqueda de causalidad en Bases de Datos es compleja, en el sentido que no es posible partir de una hipótesis, sin embargo, restringiéndose a un subconjunto de datos a través de la utilización de la técnica de minería de datos denominada "reglas de asociación" se podría obtener información de presunta causalidad entre ellos.

Palabras Clave: Minería de Datos, Causalidad, Reglas de Asociación, KDD

1. INTRODUCCIÓN

1.1. El problema

El interés en las reglas de asociación es que ellas ofrecen la promesa (o ilusión) de causalidad, o al menos entregan relaciones predictivas. Sin embargo, las reglas de asociación sólo calculan la frecuencia de ocurrencias de uniones entre los atributos; Ellas no expresan una relación causal. Sería de mucha utilidad poder descubrir las relaciones causales a partir de reglas de asociación en el contexto de la minería de datos.

1.2. Objetivos y motivación de la investigación

El objetivo de este trabajo de investigación es realizar una revisión de los conceptos relativos a la causalidad desde diversos puntos de vista, filosófico, matemático, etc., para ello se han recopilado diferentes modelos matemáticos y técnicas utilizados para representar relaciones causales y para juzgar cuando una relación causal es válida o no lo es. Para lo cual se desea proponer dentro del ámbito de la Minería de Datos, una "Técnica de Minería de Relaciones Causales" a través del post-procesamiento de los resultados entregados por la búsqueda de reglas de asociación.

En cuanto a la motivación, se puede precisar como sigue; "En muchas ocasiones cuando un usuario de un sistema de minería de datos se encuentra buscando y obteniendo reglas de asociación, lo que este usuario busca y/o como interpreta dichas reglas de asociación son las relaciones causales, pero en la práctica una regla de asociación no representa en general una relación causal.

2. INTRODUCCION A LA MINERIA DE DATOS Y EXTRACCION DE CONOCIMIENTO EN BASES DE DATOS

Se conoce como Descubrimiento de Conocimiento en Bases de Datos (KDD: Knowledge Discovery in Data Bases) al proceso de búsqueda de nuevo conocimiento a partir de datos almacenados en Bases de Datos. El proceso incluye el análisis inteligente de los datos y los pasos que permiten entregar conocimiento.

La minería de datos se refiere a las técnicas que permiten extraer conocimientos de un gran volumen de datos almacenados en Bases de Datos. En general el concepto Minería de Datos lo utilizan analistas de sistemas, estadísticos y especialistas en gestión informática, mientras KDD es usado por especialistas de aprendizaje en computadores e inteligencia artificial.

Fundamentalmente Data Mining utiliza técnicas para realizar predicción o descripción de eventos, los cuales

puedan obtenerse en forma automática a partir del gran volumen de datos almacenados en las Bases de Datos. La predicción involucra la utilización de atributos de una Base de Datos para predecir el comportamiento futuro de otros valores (conocer a priori como se presentarán datos desconocidos) y que son de interés para los usuarios, y la descripción se relaciona con encontrar patrones que permitan a los usuarios interpretar los valores de los atributos almacenados (descripción de datos y patrones).

Los objetivos inherentes a la predicción y descripción es posible lograrlos a través del Data Mining, para ello se aplican diversas técnicas, tales como reglas de asociación, árboles de decisión, clustering (agrupamiento), rted neuronal, inteligencia artificial, etc.

3. ASPECTOS FILOSOFICOS DE LA CAUSALIDAD

Las generalizaciones habituales nos dicen cómo son las cosas: Las montañas son irregulares. Forman juicios descriptivos de la realidad. A diferencia de ellas, las generalizaciones causales nos explican el porqué de las cosas: ¿Por qué llueve? ¿Por qué hay sol? ¿Por qué hay guerras?

En los años de la antigua Grecia se comenzó a hablar y pensar filosóficamente respecto al concepto de causa. Aristóteles distinguió cuatro tipos de causas:

1) la causa material (la materia de la pintura: bastidores, tela, tinta, etc.)
2) la causa formal (es la idea representada por el artista en el cuadro)
3) la causa eficiente (es la acción del artista sobre la tela)
4) la causa final (es el fin para el cual se ha hecho la pintura).

Aristóteles [1] creo la conocida y popular doctrina aristotélica de las causas, que persistió en la cultura oficial de occidente hasta el Renacimiento.

En la ciencia moderna, las causas formales y finales fueron dejadas de lado por considerárselas fuera del alcance del proceso de experimentación científica. Lo cual significó llegar a la conclusión que en la moderna concepción del mundo, la materia es esencialmente el sujeto del cambio, no aquello de que una cosa está hecha y que persiste. "Por lo tanto, de las 4 causas de Aristóteles, sólo la causa eficiente es merecedora de investigación científica" [2].

Galileo Galilei encuentra una doble idea de causa, como sucesión temporal y como necesidad racional, para Hobbes (3), la causa será considerada como la suma de los antecedentes del efecto. Para Galileo la causa eficiente es la condición necesaria y suficiente para la aparición de algo, aquella y no otra debe llamarse causa, a cuya presencia siempre sigue el efecto y a cuya eliminación el efecto desaparece. El problema de la definición de Galileo es que inutiliza el concepto de causa, pues entonces los análisis causales serían imposibles, debido a la infinidad de factores presuntamente integrantes de la causa y la prueba empírica de la hipótesis causal sería igualmente imposible, pues la supresión de cualquiera de los infinitos factores introducirá una diferencia, y por lo tanto, sería necesario llevar cuenta de una infinidad de parámetros.

El siglo XVII, recogiendo un proceso de crítica intelectual a las causas finales y materiales aristotélicas se pone fin a esta concepción. Para Descartes, la causa final deberá reservarse para el conocimiento de Dios. Respecto a la material, sostendrá que no podrá entenderse como causa de algo su propia materia [4] .

Hume señaló en su "Tratado de la Naturaleza Humana" y en la "Investigación sobre el entendimiento humano", que un examen empírico de la causalidad muestra que no hay una conexión necesaria entre causa y efecto, sino una mera secuencia en el tiempo.

4. LA MATEMÁTICA, LA FILOSOFIA CAUSAL Y MINERIA CAUSAL

4.1. Causalidad y la Matemática

Entrando en materia podemos decir que a teoría de la inferencia causal une tres partes, dos partes de matemática y una parte de filosofía. Las dos partes matemáticas son grafos acíclicos dirigidos (DAGs) y la teoría de la probabilidad (enfocada en la independencia condicional de variables), y la parte filosofía involucra la causalidad entre variables.

Uno de los grandes problemas que presenta la búsqueda de relaciones causales en un universo de datos, guarda relación con la selección del subconjunto de datos que prometa corresponder a relaciones causales.

Los algoritmos actuales que son usados para el descubrimiento en datos observados frecuentemente usan correlación e independencia probabilísticas para encontrar posibles relaciones causales. Por ejemplo, si dos variables son estadísticamente independientes, se puede afirmar que entre ellas no hay relación causal.

Existen dos enfoques para el descubrimiento de causalidad en datos observados, estas son:

Basadas en reglas (constrain-Based) y
Enfoque basado en redes Bayesianas.

El descubrimiento causal en redes Bayesianas, consiste en encontrar la estructura causal más probable y los

parámetros correspondientes a la estructura. El primer punto de complejidad encontrado es el número de modelos posibles. Para solo tres variables existen veinticinco modelos posibles. Claramente, no seria viable enumerar todos los modelos posibles incluso para un conjunto pequeño de datos. Este problema, prácticamente hace inviable aplicar el algoritmo Bayesiano a datos observados.

En general,. Los algoritmos que se aplican en la búsqueda de causalidad en datos observados son los algoritmos Basados en reglas (constrain-Based).

En este contexto es posible separar la teoría en tres partes: grafos dirigidos, la probabilidad, y la causalidad. La teoría conecta la estructura causal a la probabilidad. Para inferir desde un conjunto de datos estadísticos una estructura causal es necesario hacer una serie de supuestos.

Cabe hacer notar que existen muchas otras teorías que permiten extraer conclusiones causales para datos estadísticos, se ha escogido la Markoviana ya que el objetivo es explicar en algún sentido la relación entre la matemática, la filosofía y la minería causal.

Markov plantea un método para determinar causalidad en datos estadísticos, el cual se fundamente en la determinación de dependencia probabilística. Explicando este método, se mostrara la relación entre la matemática la filosofía causal.

Un Dag es un objeto matemático totalmente abstracto. En la teoría de inferencia causal, los DAGs tienen dos funciones distintas. En la primera representan conjuntos de las distribuciones de probabilidad y en la segunda representan las estructuras causales. La manera en que representan las distribuciones de probabilidad es dada por la condición de Markov, en la cuál (el DAGs) se convierte en una relación gráfica más útil: d-separation (Pearl 1988) es una relación entre tres conjuntos disjuntos de vértices en un grafo dirigido. La idea básica de la d-separation supone verificar si un conjunto Z de vértices bloquea todas las conexiones (de cierto tipo) entre X e Y en un grafo G. Si es así, entonces X e Y son d-separados por Z en G, es decir, X e Y tienen cierto grado de independencia.

Por ejemplo, se tiene representado en un DAG tres vértices: {X1, X2, X3}, y un conjunto de dos arcos entre estos vértices: {X1-> X2, X2-> X3}. Casi siempre se represen los DAGs con una figura, o diagrama de ruta, por ejemplo, este Dag se representa: X1->X2-> X3. En este ejemplo, al aplicar d-separación al DAG se tiene que: X1 y X3 son d-separados por X2. Luego, se puede suponer que en todas distribuciones que este DAG puede representar, X1 es independiente de X3 condicionado en X2. Usando la notación para la independencia presentada por Phil Dawid (1979); X1 _ | | _ X3 | X2 quiere decir: X1 y X3 son independientes condicionadas a X2.
Es importante destacar que mientras no se le dé una interpretación a los DAGs, estos son sólo objetos matemáticos que se pueden conectar a las distribuciones de probabilidad de cualquier manera que se desee. Lo importante es que de cualquier forma los DAGs permiten identificar conjuntos de independencias (conjunto de variables independientes). Cuando se da una interpretación causal (filosofía causal) a un DAG, entonces tenemos que la d-separación es la conexión entre un DAG causal y las distribuciones de probabilidad.

Normalmente hay muchos DAGs distintos que representan exactamente el mismo conjunto de relaciones de independencia, y por lo tanto, el mismo conjunto de distribuciones.

En este contexto se han desarrollados distintos algoritmos que permiten calcular d-separación para cualquier grafo, y que además son capaces de generar todos los DAGs posibles dado un conjunto especifico (y conocido) de relaciones de independencia (Figura 4.1).

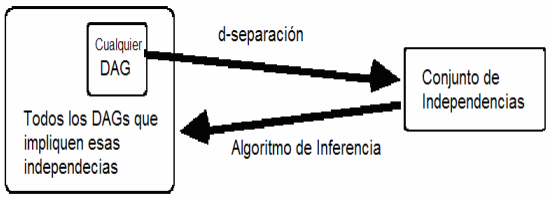


Figura 4.1: Dags generado a partir de un conjunto de relaciones de variables independientes.

Uno de los algoritmos es el denominado "algoritmo PC". Su entrada es un conjunto de relaciones de independencia sobre un conjunto de variables y su resultado es un conjunto de DAGs sobre estas variables que son su equivalente d-separado, o Markoviano. Al aplicar el algoritmo PC sobre la relación de independencia "X1 y X3 son independientes condicionadas a X2", se tiene que es posible generar otros dos DAGs que son d-separación equivalentes, esto se puede ver en la figura 4.2. El algoritmo PC genera todos los DAGs que son equivalente en d-separación.

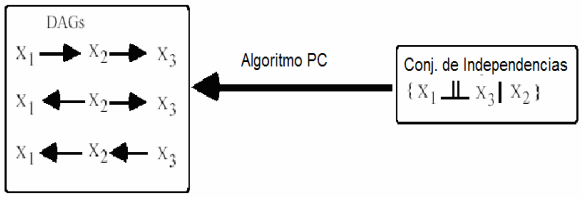


Figura 4.2: DAGs generados por algoritmo PC a partir de la relación de independencias "X1 y X3 son independientes condicionadas a X2".

Si se considera solamente la d-separación y la condición de Markov como sólo la matemática que conecta a DAGs y las distribuciones de probabilidad y no se involucra la causalidad en nada. Entonces, se cuenta con una teoría matemática para generar representaciones de estructuras de independencia. Sin embargo, si se interpreta un Dag en forma causal (con la filosofía causal) la condición de Markov y la d-separación son de hecho la conexión correcta entre la estructura causal y la independencia probabilística. En este contexto a la interpretación causal de la condición de Markov se le denomina "condición causal de Markov".

Los DAGs que son interpretados causalmente se denominan grafos causales. Por ejemplo, en un conjunto V de variables se tiene que S es una variable que representa el habito de fumar, la variable Y representa los dedos manchados por nicotina, y C la variable que representa la enfermedad de cáncer al pulmón, entonces el siguiente grafo causal (Figura 4.3.) representa lo qué puede ser la estructura causal entre estas variables.

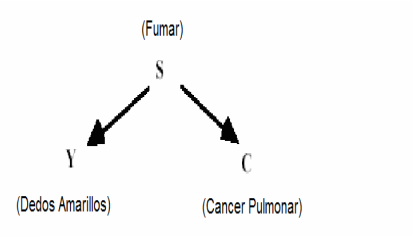


Figura 4.3: Dag causal "habito de fumar y cáncer al pulmón"

Los grafos causales se suponen completos en un sentido y no en otro. Son incompletos en el sentido de que no incluyen todas las causas de cada una de las variables en el sistema. Por ejemplo en el DAG anterior se han omitido: 1) muchas causas del cáncer al pulmón, por ejemplo, polución ambiental, inhalación de asbesto, factores genéticos, etc. 2) muchas variables que pueden estar entre una causa especificada y su efecto, por ejemplo, un problema bronquial podría estar entre el "Verdadero" camino causal de fumar a tener cáncer al pulmón. Sin embargo, estas omisiones no tienen relación con la característica de que el grafo anterior sea completo o incompleto.

Se dice que un grafo causal es completo en el sentido que cumple las siguientes dos características: Primero, deben estar presentes todas las causas comunes de las variables especificadas y segundo, deben estar representadas todas las relaciones causales entre las variables especificadas.

La teoría supone un grafo completo en el sentido que deben estar presentes (incluidas en el grafo) todas las causas comunes de las variables especificadas. Por ejemplo, si se observa el grafo del ejemplo ilustrado en la figura 4.3. es posible detectar que falta la variable "factor genético" que es causa (común) de la conducta de fumar y de la enfermedad del cáncer. Como en el grafo no se ha representado esta causa común a las dos variables se puede afirmar que el grafo causal (figura 4.3) no es una representación exacta de la estructura causal entre estas tres variables, es decir, es un grafo incompleto.

El grafo causal también se supone estar completo en el sentido de que todas las relaciones causales entre las variables especificadas están representadas en el grafo. Por ejemplo, el grafo de la Figura 4.3. no tiene ningún arco dirigido de Y a S, si las manchas de nicotina en los dedos fueran causa de la conducta de fumar esta seria un fundamento para afirmar que el grafo es incompleto o no "exacto".

4.2. La causalidad, la matemática y la minería de causalidad

En la búsqueda de relaciones causales es muy importante tener la posibilidad de manipular los datos, es decir, poder manejar las variables de modo que sea posible observar el comportamiento de una o más cuando se producen cambios en las otras variables.

La minería causal trata de generar métodos eficientes para descubrir las relaciones causales Bases de Datos observacionales. A la búsqueda de causalidad entre datos de una Base de Datos, se le denomina "Minería Causal o Descubrimiento Causal".

Por otra parte, es importante la búsqueda de causalidad en datos observados ya que es posible encontrar en ellos causalidades que tal vez sería prácticamente imposible encontrarlas en datos experimentales. Por lo tanto hay un gran potencial para el uso de técnicas de búsqueda de causalidad en grandes Bases de Datos.

La minería causal corresponde a la etapa posterior a la Minería de Datos, ya que a través de esta se interpretan los datos generados por la Minería de Datos.

La teoría de la inferencia causal en Minería de Datos consta de dos partes : la inferencia causal en datos estadísticos (la que contempla la matemática, y la filosofía causal) y la parte de Minería de Datos que tiene que ver con técnicas que permitan seleccionar un conjunto de datos observados posibles de relacionarse causalmente. Todo esto en grandes volúmenes de datos almacenados en Base de Datos.

Las técnicas existentes de minería causal se basan en técnicas más complejas de descubrimiento causal en datos estadísticos y en datos experimentales. De esta manera a partir de la filosofía causal y la matemática es posible también inferir relacione causales en grandes

Bases de Datos. Es necesario destacar que la búsqueda de causalidad en Bases de Datos es compleja, en el sentido que no es posible partir de una hipótesis, sin embargo, restringiéndose a un subconjunto de datos a través de la utilización de la técnica de minería de datos denominada "reglas de asociación" se puede obtener información de presunta causalidad entre ellos. En realidad, las reglas de asociación entregan información respeto a la frecuencia con que se relacionan dos o más atributos o Datos en la Base. En este contexto, es posible afirmar que los datos identificados por las reglas de asociación podrían corresponder a simples relaciones entre datos, por ejemplo, se puede encontrar a través de aplicación de reglas de asociación que la masa, la aceleración y la fuerza se asocian con alta frecuencia en el conjunto de datos, sin embrago, al analizar este hallazgo, se tiene que la masa y la aceleración no son causa de la fuerza, solo corresponden a una relación en que la fuerza es el resultado del producto masa-aceleración. También, podemos tener casos en que la asociación encontrada corresponde a una correlación y no a causalidad, por ejemplo, se puede tener que la cesantía aumenta y aumenta la delincuencia, recordemos que estadísticamente, se dice, que la correlación no necesariamente implica causalidad.

4.3.- Reglas de asociación

"El Descubrimiento de reglas de asociación, se define como el problema de encontrar todas las reglas de asociaciones existentes con un grado de confianza y soporte mayor que el especificado por el usuario, valores denominados minconf y minsup respectivamente" [5].

5. ALGORITMOS BASADOS EN ILCD

El planteamiento de Silverstein [6] para minar estructuras causales construidas sobre el algoritmo LCD[7] mantienen una complejidad de tiempo polinomial. Estos algoritmos no intentan descubrir la estructura causal completa como los algoritmos IC y PC. El algoritmo LCD encuentra estructuras causales en la forma de cadenas y ramificaciones. Sin añadir complejidad, se encuentra una estructura causal adicional; el autor nombra a esta estructura causalidad CCU. La causalidad CCU esta representada por las estructuras a -> b <- c, y también es conocida como las v-estructuras.

Los autores utilizan un método llamado soporte (soporte significa que un valor particular debe estar sobre una cierta frecuencia en los datos). Esto restringe la prueba de causalidad a los ítem de mayor interés. Un valor de umbral de confianza es utilizado como un nivel de confianza estadístico para determinar si dos variables son dependientes. La prueba estadística de chi-cuadrado ($\chi 2$) es utilizada para determinar la dependencia de dos variables. El coeficiente $\beta 2$ es calculado como: (O-E)/E donde O es el valor observado para la variable y E es el valor esperado para la variable. Si $\beta 2$ es mayor que $\chi 2\alpha$ donde α es el nivel de confianza de la prueba, entonces las dos variables son dependientes. Utilizando estadística en combinación con soporte y confianza, el error en determinar dependencia es reducido.

Silverstein[6] no asume que en el caso a -> b, a es la causa directa. Esto permite que existan posibles variables ocultas. Otras suposiciones que serian deseables de eliminar son tomadas. Se asumen solo datos Booleanos sin datos perdidos. Por ejemplo este método puede ser utilizado en Datos Boléanos extraídos de la canasta de compras.

Al eliminar la representación grafica, este planteamiento es más viable en términos computacionales. Pero, no supera la dificultad de identificar estructuras con cierto grado de confianza. Últimamente estos métodos solo asumen cierta correlación estadística en una orientación particular entre las variables para decidir sí una variable causa a otra. Usualmente, el orden temporal de las variables se supone conocido de antemano. Cuando el orden temporal es desconocido, Pearl [8] introduce la noción de tiempo estadístico, a cualquier orden de variables que coincidan con la estructura causal.

6. CONCLUSIONES

5.1. Ideas principales de la causalidad en Descubrimiento de Conocimiento sobre Bases de Datos.

La Minería de Datos cuenta con técnicas que permiten en alguna medida extraer objetos o información en los grandes volúmenes de datos, sin embargo, estos objetos no necesariamente representan conocimiento.

El procesamiento posterior de los resultados entregados por la Minería de Datos es una de las etapas del proceso de Descubrimiento de Conocimiento sobre Bases de Datos, y consiste en el procesamiento de los objetos de conocimiento producidos en etapas previas con el fin de simplificar, depurar, validar y visualizar el conocimiento extraído.

La transformación de estos objetos en conocimiento depende de la interpretación que realice el usuario. En muchos casos la obtención de conocimiento se desprende sin mayor análisis de los objetos, sin embargo, en otros casos requiere de un análisis exhaustivo y complicado, en estos casos es necesario contar con técnicas o herramientas de ayuda en el análisis posterior a la Minería de Datos.

Una de las técnicas utilizadas en Minería de Datos, las reglas de asociación, permite obtener patrones y relaciones entre los datos. El análisis posterior del resultado obtenido al aplicar la técnica de reglas de

asociación, consiste en un análisis semántico realizado en forma manual e íntegramente por el usuario, ya que la Minería de Datos no lo realiza. Esta forma de realizar este proceso de análisis es limitada y puede conducir a errores, ya que aunque puedan utilizarse herramientas estadísticas para algunos de los procesos que involucra dicho análisis, el usuario es siempre el que decide en el proceso de decisión final.

Las reglas de asociación indican la fuerza de la asociación de dos o más atributos de datos. El interés en las reglas de asociación es que ellas entregan la promesa (o ilusión) de causalidad, o al menos de relaciones predictivas. Sin embargo las reglas de asociación solo calculan la frecuencia de ocurrencias de uniones de dos o más atributos de datos; Ellas no expresan una relación causal. Si fuera posible descubrir las relaciones causales, esto seria muy útil para descubrir conocimiento. El objetivo de este trabajo ha sido explorar la causalidad en el contexto de la Minería de Datos.

El descubrimiento de causalidad en datos observados se apoya en técnicas que reúnen elementos matemáticos y filosóficos. Sin embargo, el problema no esta resuelto. Los algoritmos para el descubrimiento en datos observacionales por lo general utilizan correlación e independencia probabilística. La aplicación de estos algoritmos solo permite afirmar que dos variables son estadísticamente independientes, y de ello se puede asegurar de que no existe causalidad asociada. Sin embargo, lo inverso no es necesariamente verdadero. Dada esta situación, el usuario de realizar un trabajo de análisis manual muy exhaustivo para poder determinar si existe causalidad entre las reglas de asociación generadas por la Minería de Datos. De otra forma, cuando se encuentran dos variables que están correlacionadas, ¿como se puede decidir que una causa a la otra?

5.2. Trabajo Futuro

En el desarrollo de este trabajo se han comentado ciertas materias que, si bien en algunos casos han quedado resueltas, pueden mejorarse, dejando abiertas varias líneas de investigación de cara al desarrollo de trabajos futuros. A continuación se explica brevemente una propuesta de investigación.

Adaptar y aplicar (probar) en una Base de Datos especifica el algoritmo planteado por Silverstein (1998) "Scaleable Techniques For Mining Causal Structures" para encontrar estructuras causales en datos observados.

El algoritmo planteado por Silverstein (1998) para encontrar estructuras causales, utiliza la prueba estadística de chi-cuadrado ($\chi 2$) para determinar la independencia de dos variables en el universo de datos observados. De esta forma obtiene el subconjunto de datos observados sobre los cuales aplica el algoritmo de búsqueda de causalidad entre los datos.

La implementación de esta adaptación se podría realizar aplicando la técnica de reglas de asociación sobre una Base de Datos determinada, seleccionando las reglas que supere un cierto umbral y con ellas montar un dígrafo para discutir cuales de las reglas obtenidas podrían corresponder a relaciones causales y cuales no a partir del algoritmo.

En este trabajo se debiera de hacer uso factores de certeza en la búsqueda de reglas de asociación . Para ello aplicar (probar) el factor de certeza propuesto por el Dr. Daniel Sánchez Fernández [5]. El factor propuesto es capaz de detectar el grado de dependencia o independencia entre antecedente y consecuente de la regla. Es importante probar el factor de certeza propuesto por el Dr. Sánchez ya que normalmente los otros métodos no son capaces detectar el grado de dependencia o independencia entre antecedente y consecuente de la regla.

Este trabajo permitiría evaluar la utilidad que podría obtener una organización en la búsqueda de causalidad en sus Bases de Datos, aplicando la técnica de reglas de asociación en la selección de datos a analizar con el algoritmo propuesto por Silverstein (1998) [6] .

Referencias

[1] Aristóteles, "La Metafísica".

[2] Mario Bunge. Causalidad. Editorial Universitaria, Buenos Aires, 1978.

[3] Hobbes, Thomas. 1996 (1651). Leviatán (México, Fondo de Cultura Económica).

[4] Jean Wahl. Introducción a la filosofía .Fondo Cultura Económica, México, 1954.

[5] Fernando Berzal, Ignacio Blanco, Daniel Sánchez. and María Amparo Vila, Measuring the accuracy and interest of association rules: A new framework, Department of Computer Science and Artificial Intelligence, University of Granada, E.T.S.I.I, March 2002

[6] C. Silverstein, S. Brin, et al. [1998] "Scaleable Techniques For Mining Causal Structures," Proceedings. 1998 International Conference Very Large Data Bases, NY, 594-605

[7] G. Cooper [1997] "A Simple Constraint-Based Algorithm for Efficiently Mining Observational For Causal Relationships" in Data Mining and Knowledge Discouvery, v 1, n 2, 203-224

[8] J. Pearl, J. [2000] Causality: Models, Reasoning, And Inference, Cambridge University Press, NY.

[9] W. Frawley and G. Piatetsky-Shapiro and C. Matheus, Knowledge Discovery in DataBases: An Overview. AI Magazine, Fall 1992, pgs 213-228.

Sistema de Soporte de Decisiones: Aplicación a la Gestión de la Contaminación en la ciudad de Santiago de Chile

Miguel Delgado C.
Universidad de Granada
DECSAI-España
mdelgado@ugr.esm

Daniel Sánchez F.
Universidad de Granada
DECSAI-España
daniel@decsai.ug.es

Santiago Zapata C.
U. Tecnológica Metropolitana
Depto de Informática Chile
szapata@utem.cl

Resumen

La contaminación del aire es uno de los mayores problemas que afecta a los países en casi cualquier parte del mundo. El incremento en las cantidades de gases y partículas potencialmente dañinas para la salud y el medio ambiente se ha constatado a escala mundial, y cada día es mas obvio que la respuesta a estos problemas se concentra en la búsqueda de soluciones inteligentes.

En Chile, por ley, se establece la obligación de desarrollar planes de descontaminación en áreas donde los niveles de contaminantes excedan sistemáticamente las normas ambientales, y planes de prevención donde dichas normas se encuentren en peligro de ser superadas.

Durante el período otoño-invierno la población de la ciudad de Santiago, se ve afectada por un aumento repentino en los niveles de contaminación del aire. Esta situación es conocida como episodio crítico de contaminación atmosférica, y se produce cuando se registran elevados niveles de concentración de contaminantes durante períodos de corta duración. Estos episodios se originan a partir de la convergencia de una serie de factores meteorológicos que impiden la buena ventilación de la cuenca de Santiago y/o debido a un incremento en las emisiones previas al evento.

De acuerdo a la normativa vigente, los criterios para decretar un episodio crítico de contaminación esta referido al índice ICAP, que es generado por los datos de la Red de Monitoreo de gases y partículas que son ingresados a un Modelo de Pronóstico desarrollado por el físico J. Cassmassi. [4], [5], [11].

De esta forma, la autoridad toma las decisiones con respecto a los episodios críticos que pueden afectar a la ciudad de Santiago, apoyándose en las predicciones que entrega el modelo.

Nuestro trabajo de investigación se encuadra en la línea de buscar metodologías inteligentes para predicción y control de problemas medioambientales en la ciudad de Santiago y su Área Metropolitana.

Palabras Clave: Toma de Decisiones, DSS, KDD, Contaminación. Polución.

1 INTRODUCCIÓN

1.1. El Problema

La contaminación del aire en la ciudad de Santiago se origina principalmente durante el período otoño-invierno, en estos meses la población de la Región Metropolitana se ve afectada por un aumento repentino en los niveles de contaminación del aire.

Estas situaciones, conocidas genéricamente como episodios críticos de contaminación atmosférica, se producen cuando se registran elevados niveles de concentración de ciertos agentes contaminantes durante períodos de corta duración. Estos episodios se originan a partir de la convergencia de una serie de factores meteorológicos que impiden la buena ventilación de la cuenca de Santiago y también debido a un incremento en las emisiones de los agentes contaminantes.

El Índice de Calidad del Aire por Material Particulado, ICAP [1], [2], [3] es el indicador que a través de los datos emanados del modelo de pronóstico Cassmassi sirve como antecedente para que la autoridad de gobierno pueda determinar sí la ciudad de Santiago esta en presencia de un episodio crítico de contaminación. El modelo Cassmassi pronostica el valor máximo de concentración promedio de 24 horas de material particulado respirable PM10, para el período de 00 a 24 horas del día siguiente.

Para este pronóstico de episodios un equipo de meteorólogos expertos prepara diariamente el pronóstico de condiciones meteorológicas asociadas a episodios de contaminación atmosférica para la Región Metropolitana,

en base a información meteorológica actualizada, nacional e internacional, y a los datos de calidad de aire de la red de monitoreo automático de calidad del aire y meteorología Red MACAM [2], [6].

Es importante destacar que debido a que el modelo pronostica la calidad del aire del día siguiente, la declaración de un episodio por parte de la autoridad no implica que el aire haya empeorado, sino que podría llegar a empeorar. Es decir, los episodios se decretan en forma preventiva para evitar alcanzar los índices pronosticados, y así proteger la salud de la población.

1.2. Manejo del problema

En la actualidad y a nivel mundial, el estudio y control de problemas de contaminación medioambiental, se aborda con el apoyo de los llamados DSS (Decision Support Systems, Sistemas de Soporte de Decisiones en castellano) que son una clase específica de sistemas de información que apoyan los procesos de toma de decisiones en las organizaciones. Estos DSS son sistemas interactivos que ayudan a la toma de decisiones facilitando el manejo de los datos, documentos, conocimiento y/o modelos que se usan para resolver problemas dentro de las organizaciones.

Una de los ámbitos en que los DSS son de gran utilidad es en el apoyo a la toma de decisiones para análisis y control de problemas medioambientales, en particular el apoyo que puede brindar la Minería de Datos (Data Mining DM) al permitir extraer patrones, modelos, relaciones, tendencias, etc., que finalmente permiten encontrar "reglas" o "patrones" ("conocimiento") a partir de los datos y comunicarlos al usuario a través de los DSS.

1.3. Objetivos de la Investigación

El objetivo general de nuestra investigación es proponer una metodología basada en DSS como una alternativa al método actualmente usado para medir y tomar decisiones con respecto a las emergencias ambientales en la ciudad de Santiago.

2. GESTION ACTUAL DE LA CONTAMINACIÓN EN LA CIUDAD DE SANTIAGO

2.1. Introducción

Uno de los grandes problemas que posee la Región Metropolitana, especialmente la comuna de Santiago de Chile es la concentración en el ambiente de partículas contaminantes que dañan la salud de las personas que viven y transitan por esta ciudad. Por esto es importante contar con un buen sistema de apoyo a las decisiones que permita analizar los datos para obtener resultados que ayuden a tomar mejores decisiones de prevención para la ciudadanía.

Una entidad responsable que se encarga de velar por el cuidado del medio ambiente es el SESMA [4], quien a través de una red de medición llamada Red MACAM [6] van registrando diariamente los niveles de contaminación en el medio ambiente.

2.2. Red MACAM

La red oficial de Monitoreo Automático de Calidad del Aire y Meteorología de la ciudad de Santiago de Chile (Red MACAM) fue puesta en marcha en 1988 con 5 estaciones de monitoreo, mayoritariamente ubicadas en el sector céntrico de Santiago. En el invierno de 1997, dicha red fue renovada y ampliada a 8 estaciones de monitoreo automático, en lo que hoy se denomina como Red MACAM-2, dependiente del Servicio de Salud Metropolitano del Ambiente (SESMA). El sitio de Internet que permite obtener los datos es el siguiente:

http://www.minsal.cl/sesma/pvcasesma/default.htm

2.3. Normas Chilenas de Calidad del Aire

2.3.1. Normas

La Resolución Nº369, del Ministerio de Salud del 12 de abril de 1988, publicada en el diario oficial el 26 de Abril de 1988, establece los índices de calidad del aire para determinar el nivel de contaminación atmosférica de la Región Metropolitana.

2.3.2 Índices ICA E ICAP

Tanto el ICA como el ICAP, dan origen a la siguiente calificación de la situación de acuerdo al valor obtenido:

INDICE	CALIFICACION
0 – 100	BUENO
101 – 200	REGULAR
201 – 300	MALO
301 – 400	CRITICO
401 – 500	PELIGROSO

2.4. Modelo Predictivo de Joseph Cassmassi

2.4.1. Introducción

Este modelo es usado para pronosticar los episodios de Alerta, Preemergencia o Emergencia, cuyas medidas que

toma la autoridad han sido descritas en la introducción de la presente investigación, la información entregada por el modelo implica adelantarse al aumento excesivo de los Índices de Calidad del Aire considerando las variables meteorológicas y los niveles de contaminación por Material Particulado en suspensión.

El Modelo de Predicción usa ocho estaciones de monitoreo distribuidas geográficamente en la ciudad de Santiago de Chile y para su estimación considera: el comportamiento histórico de la contaminación atmosférica y su asociación con parámetros meteorológicos, niveles de contaminación atmosférica y su variación diaria. Todos estos factores son ingresados a un Polinomio matemático que fue desarrollado por el físico americano Joseph Cassmassi en el año 1999, basado en técnicas estadísticas clásicas de regresión lineal múltiple [5].

2.4.2. Modelo de pronóstico para episodios

El Plan de Prevención y Descontaminación Atmosférica para la Región Metropolitana (PPDA) estableció la necesidad de crear un modelo de pronóstico de episodios de contaminación, debido a la importancia de anticipar las medidas de control y de resguardo a la población frente a la ocurrencia de situaciones críticas.

Por su parte, el Decreto Supremo N° 59 de 1998 del Ministerio Secretaría General de la Presidencia, que establece la norma de calidad primaria para material particulado respirable (MP10), incluyó los requerimientos que debe satisfacer una metodología de pronóstico de calidad de aire.

De acuerdo con dichos requerimientos, mediante resolución Nº12.612 de 1998 del Servicio de Salud Metropolitano del Ambiente de la Región Metropolitana (SESMA), se oficializó la primera aplicación de un modelo de pronóstico de calidad de aire en la ciudad de Santiago, en uso a partir de julio de 1998.

Como una forma de optimizar esta herramienta, en 1999 CONAMA encargó un estudio para mejorar la metodología de pronóstico de calidad de aire en la Región Metropolitana. El estudio dio como resultado un nuevo modelo de pronóstico, denominado modelo Cassmassi.

2.4.3. Variables utilizadas

La metodología de pronóstico de concentraciones de PM10 está basada en algoritmos de cálculo desarrollados mediante aplicación de técnicas estadísticas de regresión múltiple, enfocadas a encontrar relaciones entre posibles variables predictivas y la variable a predecir. Los predictores incluyen variables meteorológicas observadas, índices de condiciones meteorológicas observadas y pronosticadas, concentraciones de contaminantes observadas, índices de variaciones esperadas de emisiones y otros.

2.4.4. Potencial meteorológico de contaminación atmosférica (Algoritmos usados)

La aplicación operacional de esta metodología considera dos algoritmos de predicción para cada estación monitora:

Un primer algoritmo incluye el "Índice de Potencial Meteorológico de Contaminación Atmosférica" (PMCA) pronosticado para el día siguiente. La figura 6.1 muestra que el PMCA tiene una buena correspondencia con los promedios de 24 horas de las concentraciones observadas de PM10.

El segundo algoritmo está basado solamente en observaciones (del mismo día y del día anterior). De esta forma, si el primer algoritmo no se puede aplicar por insuficiencia de la información, se utiliza el segundo algoritmo.

2.4.5. Validación del modelo

En 1999/2000, se realizó una evaluación del modelo Cassmassi por parte del Departamento de Geofísica de la Universidad de Chile. Para ello, se utilizó la información del período que se extiende desde el 1 de abril al 17 de septiembre de 1999. De esta manera, se garantizó la independencia de los datos usados para validar el modelo de los usados en su construcción. La validación muestra que el modelo Cassmassi cumple con los requisitos del Decreto Supremo. Nº 59 de 1998.

La descripción detallada del modelo Cassmassi se encuentra en el estudio "Improvement of the Forecast of Air Quality and of the Knowledge of the Local Meteorological Conditions in the Metropolitan Region" (Cassmassi 1999), [7], [8].

2.5. Fallas del Modelo Predictivo de Cassmassi

El modelo predictivo es ineficiente para predecir niveles críticos de contaminación, por ejemplo, en el año 2003 solo acertó 8 veces la predicción de situaciones criticas, con 30 falsas alarmas y 12 episodios subestimados. Según estudios (Revista Que Pasa, 16 de Mayo del 2003), [9] indica que una de las causas principales de las fallas es que no se ha actualizado en 5 años, y que además de usar el modelo predictivo de Cassmassi oficialmente, se apoya en otros cinco modelos no oficiales, para estimar el grado de certeza del primero. Por ejemplo, un día cualquiera el pronóstico indicaba que había un 80% de probabilidades de alerta ambiental, pero con una incerteza del 35% de que se transformara en preemergencia ambiental.

Este modelo considera 8 estaciones de monitoreo en forma aislada y en muchas ocasiones se ha observado que

durante la ocurrencia de episodios críticos, las estaciones interactúan entre sí. El adoptar el enfoque reduccionista en lugar del enfoque sistémico ha generado la ocurrencia de innumerables errores.

2.6. Decisiones a partir de los resultados del modelo

Diferentes son los efectos nocivos que puede traer a la salud de las personas la contaminación del aire que actualmente enfrenta la ciudad de Santiago. Sin embargo, éstos se tornan especialmente peligrosos en el caso de exposiciones a niveles muy elevados durante cortos períodos de duración, y que es lo que ocurre cuando estamos frente a un episodio crítico. Es entonces fundamental poder anticiparse a las situaciones de mayor riesgo para detener las negativas consecuencias que acarrea una emergencia ambiental.

Los resultados del modelo pronóstico son utilizados como herramienta de información preventiva por parte de las autoridades regionales, de esta manera, si el valor arrojado por el modelo pronóstico supera en alguna de las ocho estaciones:

El **nivel ICAP 200** según este modelo debiera decretarse
Alerta Ambiental
El **nivel ICAP 300** según este modelo debiera decretarse
Preemergencia Ambiental
El **nivel ICAP 500** según este modelo debiera decretarse
Emergencia Ambiental

2.7. Costo de los errores del Modelo Predictivo

Según estudios económicos, los costos asociados a los errores del modelo predictivo superaron los $ 2.000 millones de pesos en el año 2001 unos 2,7 millones de Euros, sobre los $ 5.000 millones de pesos en el año 2002 unos 6,9 millones de Euros, y se estimaba que para el año 2003 esta cifra podría duplicarse alcanzando valores cercanos a los 14 millones de Euros, [9], [10].

Como medidas correctivas, estos estudios recomendaban revisar el análisis de efectividad de las medidas adoptadas durante los episodios críticos; evaluar los costos asociados a la paralización de fuentes industriales y en tercer lugar revisar y mejorar la calidad predictiva de los modelos usados.

3. PROPUESTA DE TRABAJO

3.1. Introducción

De acuerdo a lo mostrado anteriormente para la gestión de los problemas ocasionados por la contaminación ambiental en la ciudad de Santiago, concluimos que al problema puede aportarse una solución como detallamos a continuación.

El **tratamiento actual** del problema presentado se puede esquematizar como lo muestra la figura 3.1., en la cual se puede destacar que el modelo de Cassmassi (utiliza herramientas estadísticas clásicas), no se ha actualizado desde su concepción inicial hace aproximadamente seis años, no justifica claramente los episodios críticos y basa su predicción en un promedio de comportamiento de las variables involucradas del día anterior, no es sistémico dado que no relaciona el efecto sinergético de los datos generados por las diferentes estaciones de registro de datos que están distribuidas geográficamente en la ciudad de Santiago.

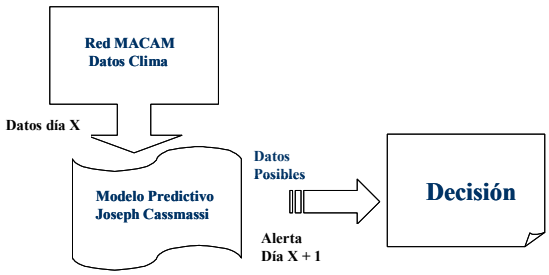


Fig. 3.1. Tratamiento actual de gestión de decisiones ambientales

La **propuesta alternativa**, propone disponer los datos básicos entregados por las estaciones de monitoreo de datos de la Red MACAM y la información meteorológica de la predicción del clima entregada por los servicios meteorológicos de la ciudad, en un **Modulo llamado Almacén de Datos Histórico (Datawarehouse)**, para estructurar estos datos en el almacén de datos puede utilizarse la fecha de creación como una dimensión principal en la futura base de datos multidimensional, las ventajas de estructurar las dimensiones de esta manera permitirá: discriminar las alertas ambientales según la época del año (estacionalidad) , evidenciando que habrá niveles críticos probablemente en épocas del año frías (invierno) potenciadas además por el fenómeno climático de la inversión térmica a que se ve afectada la ciudad de Santiago en épocas invernales y frías, además, podrá presentarse información del comportamiento de los datos por días, semanas, meses, etc.

El **módulo KDD** permite incorporar elementos que ayuden en forma inteligente al decisor, o sea el sistema debe ser capaz de realizar procesos de extracción de conocimiento, por ejemplo minería de datos, a la información histórica almacenada en el Datawarehouse y en un proceso de tipo Batch efectúe por ejemplo análisis de comportamiento de las variables involucradas en los últimos periodos, por ejemplo, los últimos dos años generando conocimiento a partir de los procesos de minería de datos realizada sobre los datos básicos históricos contenidos en el almacén de datos. Los resultados obtenidos por el tratamiento de los datos con

las herramientas de Data Warehousing y de Minería de Datos serán presentados al usuario para apoyar su toma de decisiones a través de un **Módulo DSS Sistema de Soporte de Decisiones**, mostrado figura 3.2.

Este **módulo DSS** facilita al usuario decisor aplicar medidas con respecto a los episodios críticos de contaminación (alerta, preemergencia, emergencia), con interfases usuarias que permitan a través de consultas al almacén de datos ver que pasa si (What is/ What if) si se toman decisiones, además de proporcionaren forma automática ayudas o sugerencias que asistan la tomador de decisiones.

La predicción de la calidad del aire es un tema apropiado para aplicar extracción de conocimiento y por ello usar minería de datos. Al tener almacenado información acerca de la calidad del aire que contienen una gran cantidad de datos, estos pueden ser analizados para encontrar entre ellos relaciones, regularidades y patrones de comportamiento.

Al tener un almacenamiento multidimensional se facilitará tratamientos a los datos, y técnicas como la Minería de Datos permitirá extraer conocimiento no explorado a partir de los datos básicos contenidos en el almacén de datos históricos, intentando además que el propio sistema propuesto pueda justificar la predicción ante el decisor (explicación).

Esta característica proporciona una consistencia mayor al sistema propuesto con respecto a lo entregado actualmente bajo el enfoque de usar el modelo de Cassmassi, ya que el decisor tiene antecedentes en cuanto al porque se ha producido una alerta y además podrá discriminar si la alerta se ha producido por un alza en los limites permitidos en las variables que se relacionan con los gases o en las variables que se relacionan con el material particulado o puede explicar que la conjunción de algunas variables esta gatillando un evento de alerta ambiental. .

Por ejemplo, al usar minería de datos puede que sea útil construir árboles de decisión, ya que la predicción que se desea realizar se hará en base a categorías (bueno, regular, malo, critico).

Otro aporte de la propuesta es que incorpora el enfoque sistémico para entregar una predicción, usando toda la sinergia que se puede lograr al tratar los datos de la ciudad como un todo en forma de sistema.

En resumen, la propuesta que presentamos en esta investigación es más robusta que el sistema actualmente en uso por las autoridades en lo referente al tratamiento de las emergencias ambientales usando el modelo de Cassmassi y la información entregada por la Red MACAM y los organismos en cargados de las predicciones climáticas.

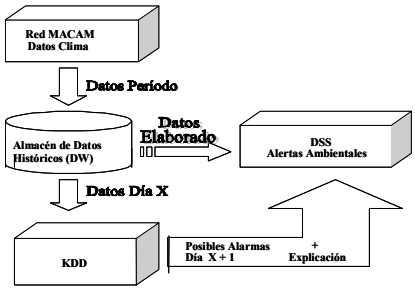


Fig. 3.2 Propuesta alternativa para apoyar la gestión ambiental

4. Conclusiones y Trabajo Futuro

En términos generales, la contaminación del aire puede definirse como cualquier condición atmosférica en que ciertas sustancias alcanzan concentraciones lo suficientemente elevadas sobre su nivel ambiental normal, lo que puede producir un efecto negativo en las personas, en los animales y en la vegetación. En las grandes ciudades, la mayor parte de la contaminación es generada por emisiones del parque automotor y de las fuentes industriales. En la ciudad de Santiago esta situación se agrava debido a su ubicación geográfica y las condiciones meteorológicas que la caracterizan

El modelo predictivo de Cassmassi que se usa actualmente en la ciudad de Santiago pronostica el valor promedio de concentración máxima de material particulado respirable (PM10) para las 24 horas del día siguiente. Lo anterior lo realiza a partir de la información entregada por las estaciones de monitoreo de la Red MACAM que maneja el SESMA, además de la condición meteorológica esperada a escala local. El índice ICAP sirve como antecedente para que el modelo de Cassmassi determine la probable presencia de un episodio critico de contaminación, debido a lo cual la autoridad correspondiente toma decisiones en forma preventiva.

Se puede concluir que el modelo predictivo de Cassmassi apoya a las autoridades ambientales a tomar decisiones con una precisión cercana al 65%, que en múltiples ocasiones las predicciones no han correspondido a las reales condiciones ambientales que se han presentado al día siguiente y se han tomado decisiones extremas cuando no ha correspondido y no se han tomado decisiones extremas cuando si la situación ambiental lo amerita, todo lo anterior genera altos costos en la salud de las personas y también costos económicos por la paralización de fuentes industriales cuando no ha sido necesario.

Tal como hemos indicado en el desarrollo de la presente investigación, el modelo predictivo de Cassmassi se basa en un polinomio matemático que incorpora técnicas estadísticas clásicas, tales como el análisis de regresión

lineal múltiple. Se ha criticado fuertemente al modelo por no estar concebido en forma sistémica, es decir no considera la sinergia que produce el efecto de interacción en cada una de las estaciones de las variables involucradas, por ejemplo, material particulado (PM10), ozono, dióxido de nitrógeno, monóxido de carbono, etc.

Se han realizado intentos por encontrar soluciones alternativas, en efecto, se han desarrollado en el mundo modelos de predicción que incorporan en su concepción, técnicas modernas de Redes Neuronales, de Lógica Difusa, de cambios dinámicos generados por los cambios climáticos globales de la tierra, parámetros de flujo de vehículos y ciclos de conducción en la ciudad, etc.

Por ejemplo un grupo de investigadores de la Universidad de Santiago de Chile, en el Departamento de Física, crearon un modelo basado en redes neuronales, que ha sido probado en paralelo con el modelo predictivo de Cassmassi, obteniendo resultados superiores al modelo de Cassmassi. Trabajos realizados por estos investigadores han sido presentados a congresos nacionales e internacionales [11], pero hasta la fecha la autoridad de gobierno no los ha considerado como una alternativa viable para implementarla en la ciudad de Santiago.

Dentro de la revisión realizada al uso de los DSS como herramienta de apoyo a la gestión de problemas ambientales, se ha encontrado que existen esfuerzos en el mundo por desarrollar herramientas que combinen técnicas de minería de datos y sistemas de soporte de decisiones, pero para aplicaciones específicas a lugares geográficamente definidos, tal como se ha mostrado en el Capitulo 5 de la presente memoria.

Cuando se inició esta investigación el objetivo principal era proponer una metodología basada en DSS y Minería de Datos como alternativa al modelo predictivo usado actualmente en la ciudad de Santiago de Chile, dado que existían muchas molestias de los habitantes de la ciudad cuando el modelo efectuaba predicciones erradas. Este objetivo esta presentado como parte de esta investigación en el punto 2.

Tal como se indica en la propuesta de trabajo, esta permitiría depositar los datos básicos generados por la Red MACAM y los datos meteorológicos referentes al clima en un almacén de datos, para luego efectuar operaciones con ellos, ya sea de ver si cumple con los niveles máximos permitidos o encontrar relaciones ocultas entre las variables por algún proceso de minería de datos, como se muestra gráficamente en la punto 3.

Como una forma de mostrar la posible funcionalidad de la propuesta, hemos desarrollado un prototipo que muestra en forma ilustrativa como podría funcionar la propuesta alternativa. Este prototipo por tener el carácter de ilustrativo, fue desarrollado con herramientas de cuarta generación, tales como Excel, Visual Basic y para mostrar el efecto de la Minería de Datos se uso el software de distribución gratuita llamado Weka. Los datos empleados para este prototipo corresponden a los meses de Enero y Julio del año 2001, que se sitúan en el periodo de verano e invierno en la ciudad de Santiago.

El modelo desarrollado puede complementarse con técnicas avanzadas de predicción de series temporales, dado que de este modo se podrán analizar con mas precisión las proyecciones de datos que son los elementos sobre los que realmente se toman las decisiones para alertas y establecimiento de episodios críticos.

El trabajo futuro es promisorio, pues encontrar soluciones alternativas al problema de la contaminación en la ciudad de Santiago de Chile se ha transformado en una cruzada de carácter humanitario, ya que los problemas a la salud de la población y a la economía del país están alcanzando niveles que sobrepasan lo aceptable por la comunidad.

Referencias

[1] www.sesma.cl/sitio/pag/aire/Indexjs3aire005.asp

[2] www.sesma.cl/sitio/pag/aire/Indexjs3airesist_monit.asp

[3] http://www.conama.cl/portal/1255/channel.html

[4] SESMA (Servicio de Salud Metropolitano del Ambiente), www.sesma.cl

[5] http://www.cleanairnet.org/lac/1471/article-40847.html.

[6] RED MACAM (Red de Monitoreo Automático de Calidad del Aire y Meteorología), www.conama.cl/rm/568/article-1114.html

[7] http://www.conama.cl/rm/568/article-1183.html.

[8] http://www.conama.cl/rm/568/article-2581.html.

[9] http://www.quepasa.cl/revista/2003/05/16/t-16.05.QP.NAC.CONTAMINACION.html

[10] http://www.sustentable.cl/portada/Descontaminacion/2453.asp

[11] http://www.geofisica.cl/English/pics3/FUM6.htm, http://www.geofisica.cl/Meteorol.htm

[12] Power, D. J. A Brief History of Decision Support Systems. DSSResources.COM, World Wide Web, http://DSSResources.COM/history/dsshistory.html, version 2.6, November 12, 2002 .

[13] J. P. Shim, Merril Warkentin, James F. Courtney, Daniel J. Power, Ramesh Sharda, Christer Carlsson Past, Present, and Future of Decision Support Technology, Decision Support System 32 (2002) 111-126, www.elsevier.com/locate/dsw

[14] Biss, A. "Dynasty Triage Advisor Enables Medical Decision-Support", 2002, at URL DSSResources.COM. www.dynasty.com, http://dssresources.com/cases/dynasty.html

RATIONALITY AND CONGRUENCE AXIOMS FOR FUZZY CHOICE FUNCTIONS

Irina Georgescu
Turku Centre for Computer Science,
Institute for Advanced Management Systems Research,
Lemminkäisenkatu 14, FIN-20520 Turku, Finland
irina.georgescu@abo.fi

Resumen

In this paper we study the congruence and the revealed preference axioms for arbitrary fuzzy choice functions. Secondly we define some concepts of rationality and normality of fuzzy choice functions and we investigate some of their properties.

Keywords: Fuzzy choice function, Rationality, Normality.

1 INTRODUCTION

The theory of fuzzy revealed preference relations and fuzzy choice functions is a topic a vast literature has been devoted to (see for example the references of [10]). Particularly, Banerjee studies in fuzzy context different forms of the axioms of revealed preference and congruence (see [2]). Banerjee works with choice functions defined on the family of non-empty finite subsets of the universe of alternatives X. The range of such a choice function is made of non-zero fuzzy subsets of X with finite support.

In [5, 6, 7] we have considered fuzzy choice functions whose domain is any non-empty family of non-zero fuzzy subsets of X. Banerjee fuzzifies only the range of a choice function; we use a fuzzification of both the domain and the range of a choice function.

This paper continues the line of [5, 6] and aims at generalizing to fuzzy choice functions some results of Hansson [9] and Suzumura [13].

Section 2 contains some notions and facts on fuzzy relations.

Section 3 is concerned with our framework of fuzzy choice functions. We define the revealed fuzzy preference relations R^* and R_* and we formulate the revealed preference axioms $WAFRP$, $WAFRP^\circ$, $SAFRP$, $SAFRP^\circ$, $HAFRP$ and the congruence axioms $WFCA$, $SFCA$. We obtain two main theorems

- the axioms $WFCA$ and $WAFRP^\circ$ are equivalent;
- the axioms $SFCA$ and $HAFRP$ are equivalent.

Section 4 deals with the notions of M-rationality and G-rationality. For any fuzzy preference relation Q we define the functions $M(.,Q)$, $G(.,Q)$, then we define the concepts of M-rationality, G-rationality, M-normality and G-normality related to fuzzy choice functions. We explore the connections between these four notions.

2 PRELIMINARIES

The results of this paper are formulated in the framework of the fuzzy set theory based on Gödel t-norm [3, 8]. They will be used without mention in the next sections.

Let $\wedge$ be the Gödel t-norm on the interval $[0,1]$ and $\rightarrow$ its residuum:

$a \wedge b = \min(a,b)$;

$a \rightarrow b = \bigvee\{c \in [0,1] | a \wedge c \leq b\}$.

The residuum $\rightarrow$ introduces the negation $\neg a = a \rightarrow 0$ for each $a \in [0,1]$.

The biresiduum $\leftrightarrow$ is defined by $a \leftrightarrow b = (a \rightarrow b) \wedge (b \rightarrow a)$.

Let X be a non-empty set (*the universe*). A *fuzzy subset* of X is a function $A : X \rightarrow [0,1]$; if $x \in X$ then $A(x)$ is called the *degree of membership* of x. We denote by $\mathcal{P}(\mathcal{X})$ the powerset of X and by $\mathcal{F}(\mathcal{X})$ the family of fuzzy subsets of X. For $U \subseteq X$ we denote by χ_U the characteristic function of U: $\chi_U(x) = \begin{cases} 1 & \text{if } x \in U \\ 0 & \text{if } x \notin U. \end{cases}$

If we identify U with its characteristic function χ_U then $\mathcal{P}(\mathcal{X})$ can be considered a subset of $\mathcal{F}(\mathcal{X})$. For any $A, B \in \mathcal{F}(\mathcal{X})$ we write $A \subseteq B$ if $A(x) \leq B(x)$ for all $x \in X$. A fuzzy subset A of X is *non-zero* if $A(x) \neq 0$ for some $x \in X$; a subset A of X is *normal* if $A(x) = 1$ for some $x \in X$.

The *support* of $A \in \mathcal{F}(\mathcal{X})$ is defined by $supp\ A = \{x \in X | A(x) > 0\}$.

A *fuzzy relation* R on X is a function $R : X^2 \to [0, 1]$. Let R be a fuzzy relation on X. R will be called

$\circ$ reflexive if $R(x, x) = 1$ for any $x \in X$;

$\circ$ transitive if $R(x, y) \wedge R(y, z) \leq R(x, z)$ for all $x, y, z \in X$;

$\circ$ total if $R(x, y) > 0$ or $R(y, x) > 0$ for all distinct $x, y \in X$;

$\circ$ strongly total if $R(x, y) = 1$ or $R(y, x) = 1$ for all distinct $x, y \in X$.

The *transitive closure* of a fuzzy relation R on X is the intersection $T(R)$ of all transitive fuzzy relations on X including R. Of course R is transitive if and only if $T(R) = R$.

For any fuzzy relation R on X let P_R denote the *asymmetric part* of R, defined by $P_R(x, y) = R(x, y) \wedge \neg R(y, x)$ for any $x, y \in X$.

If $x_1, \ldots, x_n \in X$ then $[x_1, \ldots, x_n]$ will be the fuzzy subset of X defined by

$$[x_1, \ldots, x_n](y) = \begin{cases} 1 & \text{if} \quad y \in \{x_1, \ldots, x_n\} \\ 0 & \text{otherwise.} \end{cases}$$

$[x_1, \ldots, x_n]$ is the characteristic function of $\{x_1, \ldots, x_n\}$.

If $A, B \in \mathcal{F}(\mathcal{X})$ then we denote

$I(A, B) = \bigwedge_{x \in X} (A(x) \to B(x)) = \bigwedge_{x \in X} \{B(x) | x \in X, A(x) > B(x)\}$;

$E(A, B) = \bigwedge_{x \in X} (A(x) \leftrightarrow B(x)) = \bigwedge_{x \in X} \{A(x) \wedge B(x) | x \in X, A(x) \neq B(x)\}$.

$I(A, B)$ is called the *subsethood degree* of A in B and $E(A, B)$ the *degree of equality* of A and B.

3 FUZZY REVEALED PREFERENCE AND CONGRUENCE AXIOMS

Let X be a non-empty set of alternatives and $\mathcal{B}$ a non-empty family of non-zero fuzzy subsets of X. If $S \in \mathcal{B}$ and $x \in X$ then the real number $S(x)$ can be taken as the *availability degree* of the alternative x. A *fuzzy choice function* on $\langle X, \mathcal{B} \rangle$ is a function $C : \mathcal{B} \to \mathcal{F}(\mathcal{X})$ such that for any $S \in \mathcal{B}$, $C(S)$ is a non-zero fuzzy subset of X and $C(S) \subseteq S$. In terms of consumers theory, the pair $\langle X, \mathcal{B} \rangle$ is a *fuzzy choice space* and a fuzzy choice function $C : \mathcal{B} \to \mathcal{F}(\mathcal{X})$ is a *fuzzy consumer* on $\langle X, \mathcal{B} \rangle$. The elements of X are "bundles" and the elements of $\mathcal{B}$ are "fuzzy budgets"'. If $x \in X$ is a bundle and $S \in \mathcal{B}$ is a fuzzy budget then the real number $C(S)(x)$ can be interpreted as the *degree* to which the bundle x is chosen subject to the fuzzy budget S.

This notion of fuzzy choice functions generalizes the notion introduced by Banerjee in [2], where the domain of a fuzzy choice function is the family of non-empty finite subsets of X. Banerjee fuzzifies only the range of a choice function; in our approach both the domain and the range of a choice function are fuzzified.

Let C be a fuzzy choice function on $\langle X, \mathcal{B} \rangle$. We introduce the *revealed preference* fuzzy relations R^* and R_* on X by putting

$$R^*(x, y) = \bigvee_{S \in \mathcal{B}} (C(S)(x) \wedge S(y)) \tag{1}$$

$$R_*(x, y) = \bigwedge_{S \in \mathcal{B}} [(S(x) \wedge C(S)(y)) \to C(S)(x)] \tag{2}$$

for any $x, y \in X$. $R_*(x, y)$ can be written

$$R_*(x, y) = \bigwedge_{S \in \mathcal{B}} [S(x) \to (C(S)(y) \to C(S)(x))] \tag{3}$$

The definition (3) extends the crisp notion introduced in [13]. A regard at it would hint at the fuzzy relation R_1 on X defined by

$$R_1(x, y) = \bigwedge_{S \in \mathcal{B}} (\neg S(x) \vee C(S)(x) \vee \neg C(S)(y)) \tag{4}$$

for any $x, y \in X$.

Lemma 3.1 $R_1 \subseteq R_*$.

In [7] we have introduced the fuzzy relations $\tilde{P}$ and $\tilde{R}$ by

$$\tilde{P}(x, y) = \bigvee_{S \in \mathcal{B}} [C(S)(x) \wedge S(y) \wedge \neg C(S)(y)] \tag{5}$$

$$\tilde{R}(x, y) = \neg \tilde{P}(y, x) \tag{6}$$

for any $x, y \in X$.

In the classical choice functions theory an important role is played by the axioms WCA (Weak Congruence Axiom), SCA (Strong Congruence Axiom), $WARP$ (Weak Axiom of Revealed Preference) and $SARP$ (Strong Axiom of Revealed Preference) [1, 11]. We recall in brief some crisp notions and these axioms.

A *choice function* on the choice space $\langle X, \mathcal{B}\rangle$ is a function $C : \mathcal{B} \to \mathcal{P}(\mathcal{X})$ such that for any $S \in \mathcal{B} \subseteq \mathcal{P}(\mathcal{X}) \setminus \{\emptyset\}$, $\emptyset \neq C(S) \subseteq S$.

For a fixed choice function C on the choice space $\langle X, \mathcal{B}\rangle$ the crisp preference relations R^*, $\tilde{P}$ and $\tilde{R}$ on X are given by

$R^* = \{(x, y) \in X^2 | x \in C(S)$ and $y \in S$ for some $S \in \mathcal{B}\}$;

$\tilde{P} = \{(x, y) \in X^2 | x \in C(S)$ and $y \in S \setminus C(S)$ for some $S \in \mathcal{B}\}$;

$\tilde{R} = \{(x, y) \in X^2 | (y, x) \notin \tilde{P}\}$.

In [1, 13] the revealed preference axioms $WARP$ and $SARP$ are introduced:

$WARP$ If $(x, y) \in \tilde{P}$ then $(y, x) \notin R^*$.

$SARP$ If $(x, y) \in T(\tilde{P})$ then $(y, x) \notin R^*$.

where $T(\tilde{P})$ is the transitive closure of $\tilde{P}$.

In [11, 12] the congruence axioms WCA and SCA are introduced:

WCA For any $S \in \mathcal{B}$ and $x, y \in X$,

$(x, y) \in R^*, x \in S$ and $y \in C(S) \Rightarrow x \in C(S)$.

SCA For any $S \in \mathcal{B}$ and $x, y \in X$,

$(x, y) \in T(R^*), x \in S$ and $y \in C(S) \Rightarrow x \in C(S)$.

where $T(R^*)$ is the transitive closure of R^*.

The following congruence and revealed preference axioms were formulated in [7] as a natural generalization of WCA, SCA, $WARP$ and $SARP$.

$WFCA$ For any $S \in \mathcal{B}$ and $x, y \in X$, $R^*(x, y) \wedge C(S)(y) \wedge S(x) \leq C(S)(x)$.

$SFCA$ For any $S \in \mathcal{B}$ and $x, y \in X$, $T(R^*)(x, y) \wedge C(S)(y) \wedge S(x) \leq C(S)(x)$.

$WAFRP$ For all $x, y \in X$, $\tilde{P}(x, y) \leq \neg R^*(y, x)$.

$SAFRP$ For all $x, y \in X$, $T(\tilde{P})(x, y) \leq \neg R^*(y, x)$.

The next two axioms extend to fuzzy choice functions the form of $WARP$ and $SARP$ formulated in terms of C-connected sequences ([13]).

$WAFRP^\circ$ For any $x, y \in X$ and $S_1, S_2 \in \mathcal{B}$

$$[S_1(x) \wedge C(S_2)(x)] \wedge [S_2(y) \wedge C(S_1)(y)] \leq E(S_1 \cap C(S_2), S_2 \cap C(S_1))$$

$SAFRP^\circ$ For any $x_1, \ldots, x_n \in X$ and $S_1, \ldots, S_n \in \mathcal{B}$

$$\bigwedge_{k=1}^{n} [S_k(x_k) \wedge C(S_{k+1})(x_k)] \leq \bigvee_{j=1}^{n-1} E(S_j \cap C(S_{j+1}), S_{j+1} \cap C(S_j))$$

where $S_{n+1} = S_1$.

Remark 3.2 *In the crisp case $WAFRP^\circ$ says: if there exist an element $x \in S_1 \cap C(S_2)$ and an element $y \in S_2 \cap C(S_1)$ then $S_1 \cap C(S_2) = S_2 \cap C(S_1)$.*

This is exactly the axiom $WARP$ formulated in terms of C-connected sequences, hence $WAFRP^\circ$ is a generalization of $WARP$. Similarly, $SAFRP^\circ$ generalizes $SARP$.

We add the axiom $HAFRP$ as a fuzzy extension of $HARP$:

$HAFRP$ For any $x_1, \ldots, x_n \in X$ and $S_1, \ldots, S_n \in \mathcal{B}$:

$$\bigwedge_{k=1}^{n} [S_k(x_k) \wedge C(S_{k+1})(x_k)] \leq \bigwedge_{j=1}^{n-1} E(S_j \cap C(S_{j+1}), S_{j+1} \cap C(S_j))$$

where $S_{n+1} = S_1$.

It is clear that $HAFRP$ implies $SAFRP^\circ$.

Lemma 3.3 $\tilde{P} \subseteq R^*$.

Theorem 3.4 *For a fuzzy choice function $C : \mathcal{B} \to \mathcal{F}(\mathcal{X})$ the following are equivalent:*

(i) C satisfies $WFCA$; (ii) $R^* \subseteq R_*$; (iii) C satisfies $WAFRP^\circ$.

Theorem 3.5 *For any fuzzy choice function $C : \mathcal{B} \to \mathcal{F}(\mathcal{X})$ the following are equivalent:*

(i) C satisfies $HAFRP$;

(ii) C satisfies $SFCA$.

4 M-RATIONAL AND G-RATIONAL FUZZY CHOICE FUNCTIONS

In this section we study the concepts of M-rational and G-rational fuzzy choice functions extending some results proved by Suzumura for crisp choice functions [13].

Let Q be a fuzzy preference relation on the universal set of alternatives X and $\mathcal{B}$ a non-empty family of non-zero fuzzy subsets of X. For any $S \in \mathcal{B}$ let us define the fuzzy subsets $M(S,Q)$ and $G(S,Q)$ of X:

(a) $M(S,Q)(x) = S(x) \wedge \bigwedge_{y \in X} [(S(y) \wedge Q(y,x)) \rightarrow Q(x,y)]$

$= S(x) \wedge \bigwedge_{y \in X} [S(y) \rightarrow (Q(y,x) \rightarrow Q(x,y))]$;

(b) $G(S,Q)(x) = S(x) \wedge \bigwedge_{y \in X} [S(y) \rightarrow Q(x,y)]$.

(a) and (b) generalize to fuzzy choice functions the Q-maximal point sets and the Q-greatest point sets (see [13]).

Lemma 4.1

(i) $G(S,Q) \subseteq M(S,Q)$ for each $S \in \mathcal{B}$;

(ii) If Q is strongly total then $G(S,Q) = M(S,Q)$.

Definition 4.2 *A fuzzy choice function C on $\langle X, \mathcal{B} \rangle$ will be called G-rational (resp. M-rational) if there exists a fuzzy preference relation Q on X such that*

$C(S) = G(S,Q)$ (resp. $C(S) = M(S,Q)$) for any $S \in \mathcal{B}$.

Remark 4.3 *Let $C : \mathcal{B} \rightarrow \mathcal{F}(\mathcal{X})$ be a G-rational fuzzy choice function and Q the fuzzy preference relation such that $C(S) = G(S,Q)$ for each $S \in \mathcal{B}$. By Lemma 4.1 (ii), if Q is strongly total $M(S,Q) = G(S,Q)$ for each $S \in \mathcal{B}$; in this case we say that C is strongly total rational.*

Lemma 4.4 *Any M-rational fuzzy choice function is G-rational.*

For any fuzzy choice function C on $\langle X, \mathcal{B} \rangle$ let us consider the functions $G^* : \mathcal{B} \rightarrow \mathcal{F}(\mathcal{X})$ and $M^* : \mathcal{B} \rightarrow \mathcal{F}(\mathcal{X})$ defined by

$G^*(S)(x) = G(S,R^*)(x) = S(x) \wedge \bigwedge_{y \in X} [S(y) \rightarrow R^*(x,y)]$;

$M^*(S)(x) = M(S,R^*)(x) = S(x) \wedge \bigwedge_{y \in X} [(S(y) \wedge R^*(y,x)) \rightarrow R^*(x,y)]$

for any $S \in \mathcal{B}$ and $x \in X$.

Lemma 4.5 *If C is a fuzzy choice function then $C(S) \subseteq G^*(S) \subseteq M^*(S)$ for any $S \in \mathcal{B}$.*

Definition 4.6 *A fuzzy choice function C is G-normal (resp. M-normal) if $C(S) = G^*(S)$ (resp. $C(S) = M^*(S)$) for any $S \in \mathcal{B}$.*

If C is G-normal (resp. M-normal) then it is G-rational (resp. M-rational).

Proposition 4.7 *Any M-normal fuzzy choice function C is G-normal.*

Proposition 4.8 *For a fuzzy choice function C the following are equivalent:*

(1) C is G-rational;

(2) C is M-normal.

Proposition 4.9 *Let C be a fuzzy choice function on $\langle X, \mathcal{B} \rangle$ such that $C(S)$ is a normal fuzzy subset of X for any $S \in \mathcal{B}$. If C satisfies $WAFRP^\circ$ then C is M-normal.*

Corollary 4.10 *Any fuzzy choice function satisfying $WAFRP^\circ$ is G-rational.*

Proposition 4.11 *Any M-normal fuzzy choice function C is strongly total rational.*

5 A PARTICULAR CLASS OF FUZZY CHOICE FUNCTIONS

Let $X = \{x, y\}$ and $\mathcal{B} = \{\mathcal{A}, \mathcal{B}\}$ where $A, B \in \mathcal{F}(\mathcal{X})$ are given by

$A = \frac{\alpha}{x} + \frac{1}{y}$; $B = \frac{1}{x} + \frac{\beta}{y}$ $(0 \leq \alpha, \beta \leq 1)$.

Consider the function $C : \mathcal{B} \rightarrow \mathcal{F}(\mathcal{X})$ defined by:

(c) $C(A) = \frac{\gamma}{x} + \frac{1}{y}$; $C(B) = \frac{1}{x} + \frac{\delta}{y}$ $(0 \leq \gamma \leq \alpha, 0 \leq \delta \leq \beta)$.

For any $\alpha, \beta, \gamma, \delta$ as above, C is a fuzzy choice function on $\langle X, \mathcal{B} \rangle$. In this section we will investigate some properties of the fuzzy choice functions introduced by (c). We will obtain some new information on the connections between various conditions studied in the previous sections.

First we compute the fuzzy preference relation R^*:

$$R^* = \begin{pmatrix} 1 & \beta \vee \gamma \\ \alpha \vee \delta & 1 \end{pmatrix}. \tag{7}$$

We also compute the fuzzy relation R_*:

$$R_* = \begin{pmatrix} 1 & \alpha \rightarrow \gamma \\ \beta \rightarrow \delta & 1 \end{pmatrix}. \tag{8}$$

Proposition 5.1 *The fuzzy choice function C defined by (c) satisfies $WAFRP^\circ$ if and only if $\beta \wedge (\alpha \vee \delta) = \delta$ and $\alpha \wedge (\beta \vee \gamma) = \gamma$.*

Let us compute the fuzzy relation $\tilde{P}$:

$$\tilde{P} = \begin{pmatrix} 0 & \beta \wedge \neg\delta \\ \alpha \wedge \neg\gamma & 0 \end{pmatrix}. \tag{9}$$

Proposition 5.2 *The fuzzy choice function C verifies $WAFRP$ if and only if $\alpha \wedge \beta \wedge \neg\delta = \alpha \wedge \beta \wedge \neg\gamma = 0$.*

Remark 5.3 *Let us take $\alpha = 1, \gamma = \frac{1}{2}, \beta = \delta = \frac{2}{3}$. Since $\neg\delta = \neg\gamma = 0$, by Proposition 5.2 one can infer that C satisfies $WAFRP$. We remark that $\alpha \wedge (\beta \vee \gamma) = 1 \wedge (\frac{2}{3} \vee \frac{1}{2}) = \frac{2}{3} \neq \gamma$, hence, by Proposition 5.1, the axiom $WAFRP^\circ$ does not hold. In conclusion the implication $WAFRP \Rightarrow WAFRP^\circ$ fails and $WAFRP$, $WAFRP^\circ$ are not equivalent conditions.*

Remark 5.4 *We observe that*

$$T(\tilde{P}) = \begin{pmatrix} \alpha \wedge \beta \wedge \neg\gamma \wedge \neg\delta & \beta \wedge \neg\delta \\ \alpha \wedge \neg\gamma & \alpha \wedge \beta \wedge \neg\gamma \wedge \neg\delta \end{pmatrix} \tag{10}$$

Since $\alpha \wedge \beta \wedge \neg\delta = 0$ and $\alpha \wedge \beta \wedge \neg\gamma = 0$ implies $\alpha \wedge \beta \wedge \neg\gamma \wedge \neg\delta = 0$, by Proposition 5.2 we also obtain the equivalence

C satisfies $SAFRP$ if and only if $\alpha \wedge \beta \wedge \neg\delta = \alpha \wedge \beta \wedge \neg\gamma = 0$.

In our case $\mathcal{B}$ has two members A, B then the axioms $WAFRP^\circ$, $SAFRP^\circ$ and $HAFRP$ are equivalent. Then using again the argument in Remark 5.3 the implication $SAFRP \Rightarrow HAFRP$ does not hold.

Proposition 5.5 *For the fuzzy choice function C defined by (c) the following are equivalent:*

(i) C is G-normal;

(ii) C is M-normal;

(iii) $\alpha \wedge \beta = \gamma \wedge \delta$.

Theorem 5.6 *Let C be the fuzzy choice function defined by (c). Then the following assertions are equivalent*

(i) C satisfies $WAFRP^\circ$;

(ii) C is G-normal;

(iii) C is M-normal;

(iv) $\alpha \wedge \beta = \gamma \wedge \delta$;

(v) $\alpha \wedge (\beta \vee \gamma) = \gamma$ and $\beta \wedge (\alpha \vee \delta) = \delta$.

References

[1] K.J.Arrow (1959). Rational choice functions and orderings. *Economica*, volume 26, pp. 121-127.

[2] A. Banerjee (1995). Fuzzy choice functions, revealed preference and rationality. *Fuzzy Sets and Systems*, volume 70, pp. 31-43.

[3] R. Bělohlávek (2002). Fuzzy relational systems. Foundations and principles, Kluwer.

[4] C. Carlsson, R. Fullér (2002). Fuzzy reasoning in decision making and optimization. Physica-Verlag.

[5] I. Georgescu (2003). Rational and congruous fuzzy consumers. In *Proceedings of FIP 2003*, volume 1, pp. 133-137, Beijing, China, March 2003.

[6] I. Georgescu (2003). A fuzzy analysis of a Richter theorem in fuzzy consumers. In *Proceedings of EUSFLAT 2003*, pp. 423-428, Zittau, Germany, September 2003.

[7] I. Georgescu (2004). On the axioms of revealed preference in fuzzy consumer theory. *Journal of Systems Sciences and Systems Engineering*, forthcoming.

[8] P. Hájek (1998). Metamathematics of fuzzy logic. Kluwer.

[9] B. Hansson (1968). Choice structures and preference relations. *Synthese*, volume 18, pp. 443-458.

[10] P. Kulshreshtha, B. Shekar (2000). Interrelationship among fuzzy preference - based choice function and significance of rationality conditions: a taxonomic and intuitive perspective. *Fuzzy Sets and Systems*, volume 109, pp. 429-445.

[11] M. Richter (1966). Revealed preference theory. *Econometrica*, volume 34, pp. 635-645.

[12] A.K. Sen (1971). Choice functions and revealed preference. *The Review of Economic Studies*, volume 38, pp 307-312.

[13] K. Suzumura (1976). Rational choice and revealed preference. *The Review of Economic Studies*, volume 43, pp. 149-159.

ALGUNAS PROPIEDADES ALGEBRAICAS DE LOS CONJUNTOS BORROSOS INTUICIONISTAS*

Elena Castiñeira
Dept. Matemática Aplicada-F. Informática
Universidad Politécnica de Madrid
28660 Boadilla del Monte
ecastineira@fi.upm.es

Susana Cubillo
Dept. Matemática Aplicada-F. Informática
Universidad Politécnica de Madrid
28660 Boadilla del Monte
scubillo@fi.upm.es

Resumen

En este artículo se profundiza sobre ciertas propiedades algebraicas de los conjuntos borrosos intuicionistas. En concreto se estudia el principio de no-contradicción. Además, se introduce el concepto de incompatibilidad en dichos conjuntos, estudiando la relación con la contradicción en los mismos.

Palabras Clave: Conjuntos borrosos intuicionistas, principio de no-contradicción, incompatibilidad, contradicción.

1 INTRODUCCIÓN

1.1 Los conjuntos borrosos intuicionistas fueron introducidos por Atanassov en 1983 y recogen como caso particular los conjuntos borrosos. Desde entonces se han publicado muchos trabajos, tanto en el campo teórico de estos conjuntos como en el de sus aplicaciones. En [1] K.T. Atanassov propone como conjunción y disyunción dos operadores que son extensiones naturales utilizando la t-norma del mínimo y la t-conorma del máximo; también introduce un primer operador de negación. Con posterioridad, en [5] se definen de forma axiomática los conceptos de t-norma y t-conorma intuicionistas, que engloban como caso particular los propuestos por Atanassov. Asimismo, en [3] se establece una definición más general de operador de negación para los conjuntos borrosos intuicionistas, a la vez que se estudian formas de generar este tipo de operadores.

Por tanto, en la familia de los conjuntos borrosos intuicionistas sobre un conjunto no vacío X, que notaremos por $\mathcal{IF}(X)$, se tienen definidas dos operaciones binarias: la intersección en base a una t-norma intuicionista $\mathcal{T}$, que notaremos por $\cap_{\mathcal{T}}$, y la unión basada en una t-conorma intuicionista $\mathcal{S}$, que denotaremos por $\cup_{\mathcal{S}}$. Por último, en $\mathcal{IF}(X)$ también se dispone de una operación unaria que modeliza el complemento $^{C_{\mathcal{N}}}$ determinada por un operador de negación intuicionista $\mathcal{N}$. Por consiguiente, es interesante el estudio de las propiedades algebraicas que se puedan verificar en estos conjuntos y, en suma, determinar la estructura algebraica de $(\mathcal{IF}(X), \cap_{\mathcal{T}}, \cup_{\mathcal{S}}, ^{C_{\mathcal{N}}})$. En este trabajo se continúa en el estudio de la representación de $\cap_{\mathcal{T}}$ y $\cup_{\mathcal{S}}$ (sección 2), y se investiga en concreto la ley del principio de no-contradicción (sección 3).

1.2 Por otra parte, la importancia de obtener *outputs* no contradictorios en los procesos de inferencia y, por tanto, la necesidad de abordar el problema de la contradicción en el campo de la lógica borrosa ya fue apuntado y tratado en [10] y [11]. En esta línea, en [4] se introdujo y estudió el concepto de contradicción en el campo de los conjuntos borrosos intuicionistas. Como es bien sabido, los conceptos de contradicción e incompatibilidad son equivalentes en un álgebra de Boole, como sucede en el caso de los conjuntos clásicos, es decir, en $(\mathcal{P}(X), \cap, \cup, ')$, donde $\mathcal{P}(X)$ indica partes de X. Sin embargo, esto no sucede cuando se consideran las estructuras de lógicas borrosas $(\mathcal{F}(X), \cap_T, \cup_S, ^{C_N})$, donde $\mathcal{F}(X)$ denota el conjunto de los subconjuntos borrosos de X, su intersección viene determinada por la t-norma T, su unión por la t-conorma S y el complementario por la negación fuerte N (ver [11]). En la sección 4 de este trabajo trataremos la relación de los mencionados conceptos en la estructura de lógica borrosa intuicionista $(\mathcal{IF}(X), \cap_{\mathcal{T}}, \cup_{\mathcal{S}}, ^{C_{\mathcal{N}}})$.

1.3 Para finalizar esta introducción recogemos la definición de conjunto borroso intuicionista propuesta por Atanassov.

Definición 1 ([1]) *Un conjunto borroso intuicionista*

* Este trabajo está parcialmente subvencionado por la CICYT, dentro del Proyecto TIC00-1420

A en el universo $X \neq \emptyset$, es un conjunto de la forma

$$A = \{(x, \mu_A(x), \nu_A(x)) \,/\, x \in X\}$$

tal que para todo $x \in X$ es $\mu_A(x)+\nu_A(x) \leq 1$, y donde las funciones $\mu_A : X \to [0,1]$, $\nu_A : X \to [0,1]$ reciben el nombre de grados de pertenencia y no-pertenencia, respectivamente.

Este conjunto puede ser considerado como un L-conjunto borroso en el sentido de Goguen ([8]). En este caso,

$$L = \{\mathbf{x} = (x_1, x_2) \in [0,1]^2 : x_1 + x_2 \leq 1\}$$

es un conjunto parcialmente ordenado respecto a la relación $\leq_L$ definida como sigue: dados $\mathbf{x} = (x_1, x_2)$, $\mathbf{y} = (y_1, y_2) \in L$,

$$\mathbf{x} \leq_L \mathbf{y} \iff x_1 \leq y_1 \;\&\; x_2 \geq y_2$$

De esta manera, $(L, \leq_L)$ es un retículo completo donde el menor elemento es $\mathbf{0_L} = (0,1)$, y el mayor $\mathbf{1_L} = (1,0)$. Sin embargo, existen elementos no comparables y, por lo tanto, no se trata de un retículo lineal.

Así, un conjunto borroso intuicionista A queda determinado por su función de pertenencia/no-pertenencia:

$$\chi^A \in L^X = \{\chi^A : X \to L\}$$

definida para cada $x \in X$ como $\chi^A(x) = (\mu_A(x), \nu_A(x))$.

Notaremos de forma abreviada un IFS, considerando su terminología inglesa, para referirnos a un conjunto borroso intuicionista, es decir, un elemento de $\mathcal{IF}(X)$.

2 SOBRE LOS CONECTIVOS EN $\mathcal{IF}(X)$

En [1] se definen la intersección y la unión de IFSs como sigue: si $A, B \in \mathcal{IF}(X)$ siendo sus funciones de pertenencia/no pertenencia (μ_A, ν_A) y (μ_B, ν_B), respectivamente, es

$$A \cap B = \left\{ \begin{array}{l} (x, \mathrm{Min}(\mu_A(x), \mu_B(x)), \mathrm{Max}(\nu_A(x), \nu_B(x))), \\ x \in X \end{array} \right\}$$

$$A \cup B = \left\{ \begin{array}{l} (x, \mathrm{Max}(\mu_A(x), \mu_B(x)), \mathrm{Min}(\nu_A(x), \nu_B(x))), \\ x \in X \end{array} \right\}$$

Es decir, se obtienen las funciones $\chi^{A\cap B}$ y $\chi^{A\cup B}$ aplicando los operadores Min y Max convenientemente a las coordenadas de χ^A y χ^B. El paso siguiente sería encontrar una definición adecuada de operadores $\mathcal{T}$ y $\mathcal{S}$ que permitiesen obtener

$$\chi^{A \cap_{\mathcal{T}} B} = \mathcal{T}(\chi^A, \chi^B)$$

$$\chi^{A \cup_{\mathcal{S}} B} = \mathcal{S}(\chi^A, \chi^B)$$

Una definición axiomática de estos operadores, que generalizan las t-normas y t-conormas, se introduce en [5] y también es tratada en [2], [6] y [7].

Definición 2 ([5]) *Una función $\mathcal{T} : L^2 \to L$ se dice que es una t-norma intuicionista (IF t-norma) si es conmutativa, asociativa y creciente (en cada coordenada respecto al orden $\leq_L$), con elemento neutro $\mathbf{1_L} = (1,0)$. De la misma forma, una función $\mathcal{S} : L^2 \to L$ se dice que una t-conorma intuicionista (IF t-conorma) si es conmutativa, asociativa , creciente y con elemento neutro $\mathbf{0_L} = (0,1)$.*

En [5] se demuestra el siguiente resultado.

Teorema 3 ([5]) *Si T es una t-norma y S una t-conorma tales que $T(x,y) \leq 1 - S(1-x, 1-y)$ para cualesquiera $x, y \in [0,1]$, entonces las funciones $\mathcal{T}, \mathcal{S} : L^2 \to L$ definidas para cada $\mathbf{x} = (x_1, x_2)$, $\mathbf{y} = (y_1, y_2) \in L$ por*

$$\mathcal{T}(\mathbf{x}, \mathbf{y}) = (T(x_1, y_1), S(x_2, y_2))$$

$$\mathcal{S}(\mathbf{x}, \mathbf{y}) = (S(x_1, y_1), T(x_2, y_2))$$

son respectivamente una IF t-norma y una IF t-conorma.

Nótese que la condición $T(x,y) \leq 1 - S(1-x, 1-y)$ es necesaria, pues si existiesen $x, y \in [0,1]$ tales que $T(x,y) > 1 - S(1-x, 1-y)$, los operadores $\mathcal{T}$ y $\mathcal{S}$ no estarían bien definidos. En efecto, $\mathcal{T}((x, 1-x), (y, 1-y)) = (T(x,y), S(1-x, 1-y))$, pero $T(x,y) + S(1-x, 1-y) > 1$, y $\mathcal{T}((x, 1-x), (y, 1-y)) \notin L$.

Respecto al recíproco del teorema anterior, Deschrijver *et al.* comprueban en [7] que dada una IF t-norma $\mathcal{T}$ (y lo mismo para IF t-conorma $\mathcal{S}$), no siempre existen una t-norma T y una t-conorma S que permitan representarla como indica el anterior resultado. Denominan t-representables a las que sí lo verifican, y a las que notaremos por $\mathcal{T} = \mathcal{T}_{T-S}$ y $\mathcal{S} = \mathcal{S}_{T-S}$.

A continuación establecemos una condición suficiente para asegurar la t-representabilidad de las IF t-normas y t-conormas.

Teorema 4 *Si $\mathcal{T} : L^2 \to L$ es una IF t-norma tal que dados $\mathbf{x} = (x_1, x_2)$, $\mathbf{y} = (y_1, y_2) \in L$ se verifica*

$$(Pr_1 \circ \mathcal{T})(\mathbf{x}, \mathbf{y}) = t_1(x_1, y_1)$$

$$(Pr_2 \circ \mathcal{T})(\mathbf{x}, \mathbf{y}) = t_2(x_2, y_2)$$

donde $Pr_1 : L \to [0,1]$ y $Pr_2 : L \to [0,1]$ representan la proyección en la primera y segunda coordenadas, respectivamente, entonces t_1 es una t-norma y t_2 es una t-conorma, y por tanto, $\mathcal{T}$ es t-representable.

Demostración: i) Conmutatividad: se sigue fácilmente de dicha propiedad para $\mathcal{T}$.

ii) Asociatividad: Dados $x, y, z \in [0,1]$, como $\mathcal{T}$ es asociativa, en particular, se verifica:

$$\mathcal{T}((x,0),\mathcal{T}((y,0),(z,0)) = \mathcal{T}(\mathcal{T}((x,0),(y,0)),(z,0))$$

y aplicando Pr_1 a ambos miembros:

$$\begin{aligned}(Pr_1 \circ \mathcal{T})((x,0),\mathcal{T}((y,0),(z,0))) \\ = \quad t_1(x, Pr_1(\mathcal{T}((y,0),(z,0)))) = t_1(x,t_1(y,z))\end{aligned}$$

ha de ser igual a

$$\begin{aligned}(Pr_1 \circ \mathcal{T})(\mathcal{T}((x,0),(y,0)),(z,0)) \\ = \quad t_1(Pr_1(\mathcal{T}((x,0),(y,0))),z) = t_1(t_1(x,y),z)\end{aligned}$$

iii) Monotonía: Sean $x, y, \alpha, \beta \in [0,1]$ tales que $x \leq \alpha$ e $y \leq \beta$, entonces es $(x,0) \leq_L (\alpha,0)$, $(y,0) \leq_L (\beta,0)$. Así, se verifica:

$$\begin{aligned}t_1(x,y) &= (Pr_1 \circ \mathcal{T})((x,0),(y,0)) \\ &\leq (Pr_1 \circ \mathcal{T})((\alpha,0),(\beta,0)) = t_1(\alpha,\beta)\end{aligned}$$

pues Pr_1 es creciente respecto a los órdenes $\leq_L$ en L y el usual en [0,1], y $\mathcal{T}$ es creciente. De la misma manera se prueba para t_2 teniendo en cuenta que Pr_2 es decreciente y que $(0,\alpha) \leq_L (0,x)$ y $(0,\beta) \leq_L (0,y)$.

iv) Elemento neutro: $t_1(x,1) = (Pr_1 \circ \mathcal{T})((x,0),\mathbf{1_L}) = Pr_1(x,0) = x$ y $t_2(x,0) = (Pr_2 \circ \mathcal{T})((0,x),\mathbf{0_L}) = Pr_2(0,x) = x$ □

De forma análoga se probaría el siguiente resultado para las IF t-conormas.

Teorema 5 *Si $\mathcal{S} : L^2 \to L$ es una IF t-conorma tal que dados $\mathbf{x} = (x_1,x_2)$, $\mathbf{y} = (y_1,y_2) \in L$ se verifica*

$$(Pr_1 \circ \mathcal{S})(\mathbf{x},\mathbf{y}) = s_1(x_1,y_1)$$

$$(Pr_2 \circ \mathcal{S})(\mathbf{x},\mathbf{y}) = s_2(x_2,y_2)$$

entonces s_1 es una t-conorma y s_2 es una t-norma, y $\mathcal{S}$ es t-representable.

Corolario 6 *Sean $\mathcal{T},\mathcal{S} : L^2 \to L$ IF t-norma y t-conorma, respectivamente; se verifica:*

i) $\mathcal{T}$ es t-representable si y sólo si $Pr_1 \circ \mathcal{T}$ es función sólo de las primeras coordenadas de cada variable y $Pr_2 \circ \mathcal{T}$ es función sólo de las segundas coordenadas de cada variable.

ii) $\mathcal{S}$ es t-representable si y sólo si $Pr_1 \circ \mathcal{S}$ es función sólo de las primeras coordenadas de cada variable y $Pr_2 \circ \mathcal{S}$ es función sólo de las segundas coordenadas de cada variable.

Por otra parte, recordemos que una función decreciente $\mathcal{N} : L \to L$ es una negación borrosa intuicionista fuerte (IFN) si $\mathcal{N}(\mathbf{O_L}) = \mathbf{1_L}$, $\mathcal{N}(\mathbf{1_L}) = \mathbf{0_L}$, y para todo $\mathbf{x} \in L$, se verifica $\mathcal{N}(\mathcal{N}(\mathbf{x})) = \mathbf{x}$. Estas funciones de negación, $\mathcal{N}$, permiten definir complementos en la familia de los IFSs: ${}^{C_\mathcal{N}} : \mathcal{IF}(X) \to \mathcal{IF}(X)$ tal que dado $A \in \mathcal{IF}(X)$, $A^{C_\mathcal{N}} \in \mathcal{IF}(X)$ viene determinado por $\chi^{A^{C_\mathcal{N}}} = \mathcal{N} \circ \chi^A$.

Bustince *et al.* en [3] introducen generadores borrosos intuicionistas que permiten la construcción de negaciones borrosas intuicionistas, y Deschrijver *et al.* en [7] estudian el problema demostrando que toda IFN fuerte $\mathcal{N}$ viene determinada por una negación fuerte $N : [0,1] \to [0,1]$ mediante la fórmula

$$\mathcal{N}(x_1,x_2) = (N(1-x_2), 1-N(x_1))$$

para todo $(x_1,x_2) \in L$. Por otra parte, también se verifica que si N es una negación fuerte, entonces $\mathcal{N}$ definida según la anterior fórmula es una IFN fuerte. En cuanto a las negaciones fuertes, fueron caracterizadas por Trillas en [9], de tal manera que N es una negación fuerte si y sólo si existe un automorfismo de orden del intervalo unidad, $\varphi : [0,1] \to [0,1]$, tal que $N(x) = \varphi^{-1}(1-\varphi(x))$, para todo $x \in [0,1]$. Por lo tanto, una IFN fuerte $\mathcal{N}$ también queda determinada por un automorfismo de orden φ; por lo que escribiremos $\mathcal{N} = \mathcal{N}_\varphi$ cuando queramos notar este hecho.

Una vez revisados los distintos conectivos, resultará interesante el estudio de las propiedades que dotan de cierta estructura algebraica a $(\mathcal{IF}(X), \cap_\mathcal{T}, \cup_\mathcal{S}, {}^{C_\mathcal{N}})$, o equivalentemente a $(L^X, \mathcal{T}, \mathcal{S}, \mathcal{N})$.

3 PRINCIPIO DE NO-CONTRADICCIÓN EN $\mathcal{IF}(X)$

Las estructuras reticulares de las lógicas multivaluadas borrosas $(\mathcal{F}(X), \cap_T, \cup_S, {}^{c_N})$ (o equivalentemente $([0,1]^X, T, S, N)$) no gozan de todas las propiedades de un álgebra de Boole, sino que dependiendo de T, S y N se verificarán unas u otras. Como en el caso clásico, se verifica el principio de no-contradicción en $\mathcal{F}(X)$ si para todo $A \in \mathcal{F}(X)$ es $A \cap_T A^{c_N} = \emptyset$, o de forma equivalente, considerando la estructura funcional, si para todo $x \in [0,1]$ es $T(x,N(x)) = 0$. Análogamente, el principio del tercero excluido se verifica en $\mathcal{F}(X)$ si $A \in \mathcal{F}(X)$ es $A \cup_S A^{c_N} = X$, o de forma equivalente, para todo $x \in [0,1]$ es $S(x,N(x)) = 1$. Es bien sabido que en el caso de t-normas y t-conormas estrictas, estos principios no se verifican. Sin embargo, para cualquier t-norma asociada a la de Lukasiewick por un automorfismo de orden, es decir $T_\varphi = \varphi^{-1} \circ W \circ (\varphi \times \varphi)$ donde $W(x,y) = \mathrm{Max}(0, x+y-1)$ es

la t-norma de Lukasiewick, se verifica el principio de no-contradicción considerando la negación N_φ generada por dicho automorfismo. De la misma manera, se verifica el principio del tercero excluido para la t-conorma $S_\varphi = \varphi^{-1} \circ W^* \circ (\varphi \times \varphi)$, siendo $W^*(x,y) = \text{Min}(1, x+y)$ la t-conorma de Lukasiewick, y considerando la mencionada N_φ.

A continuación, se estudia qué sucede respecto a estos principios en el campo de los IFSs, extendiendo para ello las definiciones que sean necesarias. En $(\mathcal{IF}(X), \cap_{\mathcal{T}}, \cup_{\mathcal{S}}, ^{C_{\mathcal{N}}})$, de forma análoga al caso clásico y al borroso, el principio de no-contradicción dice que para cualquier $A \in \mathcal{IF}(X)$ debe ser $A \cap_{\mathcal{T}} A^{C_{\mathcal{N}}} = \emptyset$, o equivalentemente, para todo $\mathbf{x} \in L$ ha de ser $\mathcal{T}(\mathbf{x}, \mathcal{N}(\mathbf{x})) = \mathbf{0_L}$. Del mismo modo, el principio del tercero excluido dice que para todo $A \in \mathcal{IF}(X)$ debe ser $A \cup_{\mathcal{S}} A^{C_{\mathcal{N}}} = X$, o equivalentemente, para todo $\mathbf{x} \in L$ ha de ser $\mathcal{S}(\mathbf{x}, \mathcal{N}(\mathbf{x})) = \mathbf{1_L}$.

Aunque los resultados de esta sección se refieren a la no-contradicción, otros similares pueden deducirse para el tercero excluido, pero no los abordaremos debido al límite que se establece para este trabajo.

Definición 7 *Una IF t-norma $\mathcal{T}$ es estricta si $\mathcal{T}(\mathbf{x}, \mathbf{y}) = \mathbf{0_L}$ implica que $\mathbf{x} = \mathbf{0_L}$ ó $\mathbf{y} = \mathbf{0_L}$.*

Una IF t-conorma $\mathcal{S}$ es estricta si $\mathcal{S}(\mathbf{x}, \mathbf{y}) = \mathbf{1_L}$ implica que $\mathbf{x} = \mathbf{1_L}$ ó $\mathbf{y} = \mathbf{1_L}$.

Teorema 8 *Si $\mathcal{T}$ es una IF t-norma tal que $\mathcal{T} = \mathcal{T}_{T-S}$, se verifica:*

i) Si S es estricta, entonces $\mathcal{T}$ es estricta.

ii) Si T y S son duales y $\mathcal{T}$ es estricta, entonces T y S son estrictas.

Demostración: i) Si $\mathcal{T}(\mathbf{x}, \mathbf{y}) = \mathbf{0_L}$, con $\mathbf{x} = (x_1, x_2)$ e $\mathbf{y} = (y_1, y_2)$, entonces $S(x_2, y_2) = 1$; de donde se sigue que $x_2 = 1$ ó $y_2 = 1$, ya que S es estricta. Si $x_2 = 1$, de la condición $x_1 + x_2 \leq 1$, se deduce que $x_1 = 0$ y, por tanto, es $\mathbf{x} = \mathbf{0_L}$. De la misma manera, si $y_2 = 1$ se obtiene que $\mathbf{y} = \mathbf{0_L}$.

ii) Veamos que T es estricta, la demostración de que S también es estricta es totalmente análoga. Si $T(x,y) = 0$, teniendo en cuenta que T y S son duales (i.e. $S(\alpha, \beta) = 1 - T(1-\alpha, 1-\beta)$), se obtiene que

$$\mathcal{T}((x, 1-x), (y, 1-y)) = (T(x,y), 1 - T(x,y)) = \mathbf{0_L}$$

y, dado que $\mathcal{T}$ es estricta, entonces $(x, 1-x) = \mathbf{0_L}$, siendo así $x = 0$, o $(y, 1-y) = \mathbf{0_L}$, en cuyo caso sería $y = 0$. □

De igual modo, se demuestra el siguiente resultado que es análogo para IF t-conormas.

Teorema 9 *Si $\mathcal{S}$ es una IF t-conorma tal que $\mathcal{S} = \mathcal{S}_{T-S}$, se verifica:*

i) Si T y S son estrictas, entonces $\mathcal{S}$ es estricta.

ii) Si T y S son duales y $\mathcal{S}$ es estricta, entonces T y S son estrictas.

Respecto a la verificación del principio de no-contradicción, tenemos los siguientes resultados.

Teorema 10 *Si en $\mathcal{IF}(X)$ se considera la intersección dada por una IF t-norma t-representable, $\mathcal{T}_{T-S}$, no se verifica el principio de no-contradicción, para cualquiera que sea el complemento determinado por una IFN fuerte, $\mathcal{N}$.*

Demostración: Atendiendo a la representación de una IFN fuerte mediante una negación fuerte N, se verifica que $\mathcal{T}_{T-S}((x_1,x_2), \mathcal{N}(x_1,x_2)) = \mathbf{0_L}$ si y sólo si $\mathcal{T}_{T-S}((x_1,x_2), (N(1-x_2), 1-N(x_1))) = \mathbf{0_L}$, lo que equivale a que $T(x_1, N(1-x_2)) = 0$ y $S(x_2, 1-N(x_1)) = 1$. Sin embargo, eligiendo $x_2 = 0$, $x_1 < 1$, se tiene que $S(x_2, 1-N_g(x_1)) = 1 - N_g(x_1) < 1$. □

Podemos preguntarnos qué ocurre en el caso de las t-normas no representables.

En [7] los autores presentan una IF t-norma no representable definida para cada $\mathbf{x} = (x_1, x_2), \mathbf{y} = (y_1, y_2) \in L$,

$$\mathcal{T}(\mathbf{x}, \mathbf{y}) = \begin{cases} \mathbf{x}, & \text{si } \mathbf{y} = \mathbf{1_L} \\ \mathbf{y}, & \text{si } \mathbf{x} = \mathbf{1_L} \\ (\text{Min}(x_1, y_1), \\ \text{Max}(1-x_1, 1-y_1)) & \text{en otro caso} \end{cases}$$

Su consideración nos lleva a la siguiente familia:

$$\mathcal{T}_T(\mathbf{x}, \mathbf{y}) = \begin{cases} \mathbf{x}, & \text{si } \mathbf{y} = \mathbf{1_L} \\ \mathbf{y}, & \text{si } \mathbf{x} = \mathbf{1_L} \\ (T(x_1, y_1), \\ T^*(1-x_1, 1-y_1)) & \text{en otro caso} \end{cases}$$

donde T es cualquier t-norma y T^* su dual. $\mathcal{T}_T$ es una IF t-conorma. En efecto, observemos en primer lugar que está bien definida pues $T(x_1, y_1) + T^*(1-x_1, 1-y_1) = 1$. Las propiedades de contorno y la monotonía se deducen directamente de la definición. Comprobemos la asociatividad: dados $\mathbf{x}, \mathbf{y}, \mathbf{z} \in L$, teniendo en cuenta la asociatividad de T y S, y que son duales,

$$\begin{aligned} \mathcal{T}_T(\mathcal{T}_T(\mathbf{x}, \mathbf{y}), \mathbf{z}) &= \mathcal{T}_T((T(x_1, y_1), T^*(1-x_1, 1-y_1)), \mathbf{z}) \\ &= (T(T(x_1, y_1), z_1)), T^*(1 - T(x_1, y_1), 1 - z_1)) \\ &= (T(T(x_1, y_1), z_1)), T^*(T^*(1-x_1, 1-y_1), 1-z_1))) \\ &= (T(x_1, T(y_1, z_1)), T^*(1-x_1, T^*(1-y_1, 1-z_1))) \\ &= (T(x_1, T(y_1, z_1)), T^*(1-x_1, 1-T(y_1, z_1))) \\ &= \mathcal{T}_T(\mathbf{x}, (T(y_1, z_1), T^*(1-y_1, 1-z_1))) = \mathcal{T}_T(\mathbf{x}, \mathcal{T}_T(\mathbf{y}, \mathbf{z})) \end{aligned}$$

Respecto a esta familia de IF t-normas y su verificación del principio de no contradicción, tenemos los siguientes resultados:

Proposición 11 *El principio de no-contradicción no se verifica en $\mathcal{IF}(X)$ cuando se considera la intersección determinada por $\mathcal{T}_T$, siendo T una t-norma estricta, y $\mathcal{N}$ cualquier IFN fuerte.*

Demostración: En efecto, si $\mathcal{N}$ está determinada por N, dado $\mathbf{x} = (x_1, x_2) \in L$, es $\mathcal{T}_T(\mathbf{x}, \mathcal{N}(\mathbf{x})) = (T(x_1, N(1-x_2)), T^*(1-x_1, 1-N(1-x_2)))$, que es distinto de $(0,1)$, siempre que sean $0 < x_1, x_2 < 1$, con $x_1 + x_2 \leq 1$. □

Proposición 12 *Se verifica el principio de no contradicción en $(L^X, \mathcal{T}_W, \mathcal{S}_{W^*}, \mathcal{N}_s)$, donde W y W^* son la t-norma y la t-conorma de Lukasiewick, respectivamente, y $\mathcal{N}_s(x_1, x_2) = (x_2, x_1)$ es la IFN estándar.*

Demostración: Dado $\mathbf{x} = (x_1, x_2) \in L$,

$$\begin{aligned} \mathcal{T}_W(\mathbf{x}, \mathcal{N}_s(\mathbf{x})) &= \mathcal{T}_W((x_1, x_2), (x_2, x_1)) \\ &= (W(x_1, x_2), W^*(1-x_1, 1-x_2)) \end{aligned}$$

Por lo que $\mathcal{T}_W(\mathbf{x}, \mathcal{N}_s(\mathbf{x})) = \mathbf{0_L}$ si y sólo si $\mathrm{Max}(0, x_1 + x_2 - 1) = 0$ y $\mathrm{Min}(2 - x_1 - x_2, 1) = 1$, lo cual se verifica por ser $x_1 + x_2 \leq 1$. □

También en [7] se propone otra t-norma intuicionista no representable: Dados $\mathbf{x} = (x_1, x_2), \mathbf{y} = (y_1, y_2) \in L$

$\mathcal{T}(\mathbf{x}, \mathbf{y}) = (\mathrm{Max}(0, x_1 + y_1 - 1), \mathrm{Min}(1, x_2 + 1 - y_1, y_2 + 1 - x_1))$.

De nuevo este ejemplo nos sugiere otra familia de t-normas intuicionistas no representables: Para cada t-norma T, siendo T^* su t-conorma dual:

$$\begin{aligned} \mathcal{T}^T(\mathbf{x}, \mathbf{y}) &= (T(x_1, y_1), \\ &\quad \mathrm{Min}(T^*(1-x_1, y_2), T^*(1-y_1, x_2))) \end{aligned}$$

En efecto, está bien definida:

$$\begin{aligned} T(x_1, y_1) + \mathrm{Min}(T^*(1-x_1, y_2), T^*(1-y_1, x_2)) &\leq T(x_1, y_1) + T^*(1-x_1, y_2) \\ &= T(x_1, y_1) + 1 - T(x_1, 1-y_2) \\ &\leq T(x_1, y_1) + 1 - T(x_1, y_1) = 1. \end{aligned}$$

Las condiciones de contorno, monotonía y conmutatividad se deducen de las mismas propiedades de las t-normas y t-conormas, además, teniendo en cuenta la dualidad de T y T^*, se obtiene la asociatividad.

De nuevo, respecto al principio de no contradicción obtenemos algunos resultados.

Proposición 13 *El principio de no-contradicción no se verifica en $\mathcal{IF}(X)$ cuando se considera la intersección dada por $\mathcal{T}^T$, siendo T una t-norma estricta, y $\mathcal{N}$ cualquier IFN fuerte.*

Demostración: $\mathcal{T}^T((x_1, x_2), (N(1-x_2), 1-N(x_1))) = (T(x_1, N(1-x_2)), \mathrm{Min}(T^*(1-x_1, 1-N(x_1)), T^*(1-N(1-x_2), x_2))) \neq (0,1)$, siempre que $0 < x_1, x_2 < 1$, con $x_1 + x_2 \leq 1$.

Proposición 14 *Se verifica el principio de no-contradicción en $(L^X, \mathcal{T}^W, \mathcal{S}^{W^*}, \mathcal{N}_s)$, donde W y W^* son la t-norma y la t-conorma de Lukasiewick, respectivamente, y $\mathcal{N}_s(x_1, x_2) = (x_2, x_1)$ es la IFN estándar.*

Demostración: Dados $\mathbf{x} = (x_1, x_2), \mathbf{y} = (y_1, y_2) \in L$ es

$\mathcal{T}^W(\mathbf{x}, \mathbf{y}) =$
$= (W(x_1, y_1), \mathrm{Min}(W^*(1-x_1, y_2), W^*(1-y_1, x_2)$
$= (\mathrm{Max}(0, x_1 + y_1 - 1), \mathrm{Min}(1 - x_1 + y_2, 1 - y_1 + x_2, 1))$

Por tanto, $\mathcal{T}^W(\mathbf{x}, \mathcal{N}_s(\mathbf{x})) = \mathcal{T}^W((x_1, x_2), (x_2, x_1)) = (\mathrm{Max}(0, x_1 + x_2 - 1), 1) = \mathbf{0_L}$ dado que $x_1 + x_2 \leq 1$. □

4 T-INCOMPATIBILIDAD

Cuando se trabaja en un álgebra de Boole, como sucede en $(\mathcal{P}(X), \cap, \cup, ^c)$, la equivalencia entre $A \cap B = \emptyset$ (A y B incompatibles) y $A \subset B^c$ (A y B contradictorios) resulta de la aplicación de las leyes asociativa, de idempotencia (para ver que incompatible implica contradictorio) y distributiva (para el recíproco). Por lo que es de esperar que esta equivalencia no se verifique en un retículo cualquiera. Así, por ejemplo, en la terna de De Morgan $(\mathcal{F}(X), \cap_{\mathrm{Min}}, \cup_{\mathrm{Max}}, ^c)$ sobre los conjuntos borrosos de X, donde ^{c} es el complemento determinado por la negación estándar $N_s(x) = 1 - x$, no se verifica. En [11] se comprueba este hecho con $X = [0,1]$, $\mu_A(x) = \mathrm{Min}(x, 1/2)$ y $\mu_B(x) = 1 - x$. A continuación vamos a estudiar estos conceptos en el campo de los conjuntos borrosos intuicionistas.

Definición 15 *Sean $A, B \in \mathcal{IF}(X)$, determinados por $\chi^A, \chi^B \in L^X$, y $\mathcal{T}$ una IF t-norma. Diremos que A y B (ó χ^A y χ^B) son $\mathcal{T}$-incompatibles si $A \cap_{\mathcal{T}} B = \emptyset$, o equivalentemente si $\mathcal{T}(\chi^A, \chi^B) = \chi^{\mathbf{0_L}}$, es decir, para todo $x \in X$ es $\mathcal{T}(\chi^A(x), \chi^B(x)) = (0,1)$.*

Observación Dada $\mathcal{T}_{T-S}$, entonces $\chi^A = (\mu_A, \nu_A)$ y $\chi^B = (\mu_B, \nu_B)$ son $\mathcal{T}_{T-S}$-incompatibles si y sólo si para todo $x \in X$

$$\begin{aligned} &\mathcal{T}_{TS}((\mu_A(x), \nu_A(x)), (\mu_B(x), \nu_B(x))) = \\ &= (T(\mu_A(x), \mu_B(x)), S(\nu_A(x), \nu_B(x))) = (0,1), \end{aligned}$$

lo que equivale a

$$T(\mu_A(x), \mu_B(x)) = 0 \ \text{ y } \ S(\nu_A(x), \nu_B(x)) = 1$$

para todo $x \in X$.

En [4] se define la contradicción entre conjuntos borrosos intuicionistas, de la siguiente forma.

Definición 16 *Dados $A, B \in \mathcal{IF}(X)$, determinados respectivamente por χ^A, $\chi^B \in L^X$, y una IFN fuerte $\mathcal{N}$, se dice que A y B son $\mathcal{N}$-contradictorios si $\chi^A \leq_L \mathcal{N} \circ \chi^B$.*

Además, en el mismo trabajo se demuestra que $\chi^A = (\mu_A, \nu_A)$ y $\chi^B = (\mu_B, \nu_B)$ son $\mathcal{N}$-contradictorios si y sólo si para todo $x \in X$

$$\begin{cases} \varphi(\mu_A(x)) + \varphi(1 - \nu_B(x)) \leq 1 \\ \varphi(\mu_B(x)) + \varphi(1 - \nu_A(x)) \leq 1 \end{cases}$$

siendo φ el automorfismo tal que $\mathcal{N} = \mathcal{N}_\varphi$.

Proposición 17 *Sean χ^A, $\chi^B \in L^X$ y $\mathcal{T}_{T-S}$ IF t-norma t-representable mediante cualquier t-norma T y una t-conorma S estricta. Si χ^A y χ^B son $\mathcal{T}_{T-S}$-incompatibles, entonces son $\mathcal{N}$-contradictorios para toda IFN fuerte $\mathcal{N}$.*

Demostración: Por ser χ^A y χ^B $\mathcal{T}_{T-S}$-incompatibles, $T(\mu_A(x), \mu_B(x)) = 0$ y $S(\nu_A(x), \nu_B(x)) = 1$ para todo $x \in X$. Por ser S estricta, será $\nu_A(x) = 1$ ó $\nu_B(x) = 1$, para todo $x \in X$. Sea φ tal que $\mathcal{N} = \mathcal{N}_\varphi$; si $\nu_A(x) = 1$ entonces $\varphi(\mu_B(x)) + \varphi(1 - \nu_A(x)) = \varphi(\mu_B(x)) \leq 1$ y, dado que $\mu_A(x) + \nu_A(x) \leq 1$, es $\mu_A(x) = 0$, con lo que $\varphi(\mu_A(x)) + \varphi(1 - \nu_B(x)) = \varphi(1 - \nu_B(x)) \leq 1$. Análogamente si fuese $\nu_B(x) = 1$. □

Proposición 18 *Si χ^A, χ^B son $\mathcal{T}_{T-S}$-incompatibles, donde $T = \varphi^{-1} \circ W \circ (\varphi \times \varphi)$, $S = \varphi^{-1} \circ W^* \circ (\varphi \times \varphi)$, y $\varphi \leq id$, entonces son $\mathcal{N}_s$-contradictorios (donde $\mathcal{N}_s$ es la negación estándar).*

Demostración: Por la nota anterior, χ_A, χ_B son $\mathcal{T}_{T-S}$-incompatibles si y sólo si $T(\mu_A(x), \mu_B(x)) = 0$ y $S(\nu_A(x), \nu_B(x)) = 1$ para todo x, lo que es equivalente a que para todo $x \in X$

$$\varphi(\mu_A(x)) + \varphi(\mu_B(x)) \leq 1 \text{ y } \varphi(\nu_A(x)) + \varphi(\nu_B(x)) \geq 1$$

Además, al ser *id* el automorfismo que determina $\mathcal{N}_s$, dichos conjuntos son $\mathcal{N}_s$-contradictorios si para todo x es

$$\begin{cases} \mu_A(x) + 1 - \nu_B(x) \leq 1 \\ \mu_B(x) + 1 - \nu_A(x) \leq 1 \end{cases}$$

o, equivalentemente,

$$\begin{cases} \mu_A(x) \leq \nu_B(x) \\ \mu_B(x) \leq \nu_A(x) \end{cases}$$

Así, teniendo en cuenta la condiciones que deben cumplir las funciones de pertenencia/no pertenencia de un conjunto borroso intuicionista, se tiene:

$\mu_A(x) + \nu_A(x) \leq 1 \leq \varphi(\nu_A(x)) + \varphi(\nu_B(x)) \leq \nu_A(x) + \nu_B(x)$
y por tanto, $\mu_A(x) \leq \nu_B(x)$, a la vez que

$\mu_B(x) + \nu_B(x) \leq 1 \leq \varphi(\nu_A(x)) + \varphi(\nu_B(x)) \leq \nu_A(x) + \nu_B(x)$
y por tanto, $\mu_B(x) \leq \nu_A(x)$. □

Referencias

[1] K.T. Atanassov. Intuitionistic fuzzy sets. *(Physica-Verlag, Heidelberg, New York)*, 1999.

[2] M. Baczynski. On Some Properties of Intuitionistic Fuzzy Implication. *Proc. Third Eusflat*, Pág. 168-171, 2003.

[3] H. Bustince, J. Kacpryk, V. Mohedano. Intuitionistic fuzzy generators: Applications to intuitionistic fuzzy complementation. *Fuzzy Sets and Systems, 114*, Pág. 485-504, 2000.

[4] S. Cubillo, E. Castiñeira. Contradiction in Intuitionistic Fuzzy Sets. Aparecerá en *Proceedings of IPMU'04.*

[5] C. Cornelis, G. Deschrijver. The Compositional Rule of Inference in Intuitionistic Fuzzy Logic Framework. *Proceedings of ESSLLI 2001 Student Session (Kristina Striegnitz, ed.)* Kluwer Academic Publishers, Pág. 83-94, 2001.

[6] C. Cornelis, G. Deschrijver, E. Kerre. Classification of Intuitionistic Fuzzy Implicatiors: an algebraic Approach. *Proceedings of the FT & T'02, Durham, North Carolina*, Pág. 83-94, 2002.

[7] G. Deschrijver, C. Cornelis, E. Kerre. Intuitionistic fuzzy connectives revisited *Proceedings of IPMU'02*, Pág. 1839-1844, 2002.

[8] J.A. Goguen. *L*-Fuzzy Sets, *Journal of Mathematical Analysis and Applications 18 (1)*, Pág. 623-668, 1967.

[9] E. Trillas. Sobre funciones de negación en la teoría de conjuntos difusos. *Stochastica III/1*, Pág. 47-60 (in Spanish), 1979. Reprinted (English version) (1998) in *Advances of Fuzzy Logic* (eds. S. Barro et altri-Universidad de Santiago de Compostela), Pág. 31-43, 1998.

[10] E. Trillas, C. Alsina, J. Jacas. On Contradiction in Fuzzy Logic, *Soft Computing* **3 (4)**, Pág. 197-199, 1999.

[11] E. Trillas, S. Cubillo. On non-contradicictory Input/Ouput couples in Zadeh's CRI. *Proceedings of NAFIPS 1999*, Pág. 28-32, New York, 1999.

UTILIZACION DE REGLAS DE ASOCIACION EN LA TEORIA DE CONCEPTOS L-FUZZY[1]

Ana Burusco
Dpto. de Automática y Computación
Universidad Pública de Navarra
31006 Pamplona, Spain
burusco@unavarra.es

Ramón Fuentes-González
Dpto. de Automática y Computación
Universidad Pública de Navarra
31006 Pamplona, Spain
rfuentes@unavarra.es

Cristina Alcalde
Dpto. de Matemática Aplicada
Escuela Universitaria Politécnica
Universidad del País Vasco
20018 San Sebastián, Spain
mapalvac@sp.ehu.es

Resumen

En este trabajo mostramos cómo la teoría de conceptos L-Fuzzy puede beneficiarse de algunas ideas de minería de datos.
En la mayor parte de las situaciones, el tamaño del retículo de conceptos es exponencial con respecto al del contexto del que se deriva, por eso se deben buscar estrategias que permitan organizar la información que se desprende de los conceptos.
Partiendo de la idea de conjuntos frecuentes, definiremos nuestros conceptos L-Fuzzy frecuentes y estableceremos reglas de asociación que nos servirán para definir implicaciones entre atributos y extraer de esa forma información del contexto L-Fuzzy original.
Finalmente, ilustraremos estos resultados mediante un ejemplo.

Palabras Clave: Conceptos y contextos L-Fuzzy, conjuntos frecuentes, reglas de asociación, implicaciones entre atributos, adquisición y representación del conocimiento.

1 INTRODUCCION

La teoría formal de conceptos desarrollada por R. Wille [9] intenta extraer conocimiento de una tabla binaria de doble entrada que representa un contexto formal (G,M,I) con G y M conjuntos de objetos y atributos respectivamente e $I \subseteq GxM$. Esta información se recoge a través de conceptos formales que son pares (A,B) con $A \subseteq G$, $B \subseteq M$ verificando $A^*=B$ y $B^*=A$ (siendo * el operador derivación que asocia a cada conjunto A de objetos, el conjunto B de atributos relacionados con él y viceversa). Al primer elemento del par se le llama extensión y al segundo intensión.

El conjunto de conceptos que se derivan de un contexto (G,M,I) tiene estructura de retículo completo y es habitual su representación mediante diagramas de líneas.
Dado un contexto formal (G,M,I) y dados $C,D \subseteq M$, se dice que $C \rightarrow D$ sí y solo sí $D \subseteq C^{**}$. Esta es la manera formal de definir la implicación entre los conjuntos de atributos C y D (todo objeto que comparte los atributos de C también comparte los atributos de D). Estas implicaciones se visualizan a partir del diagrama de líneas que representa el retículo de conceptos.

Nosotros definimos en trabajos anteriores ([2],[3]) los contextos L-Fuzzy (L,X,Y,R), con L retículo completo, X e Y conjuntos de objetos y atributos respectivamente y $R \in L^{XxY}$ una relación L-Fuzzy entre los objetos y atributos, como una extensión al caso Fuzzy de los contextos formales de R. Wille cuando se quieren estudiar relaciones entre los objetos y los atributos con valores en un retículo completo L en lugar de valores binarios.
A la hora de trabajar con estos contextos L-Fuzzy utilizamos los operadores derivación 1 y 2 definidos mediante las siguientes expresiones:
$\forall A \in L^X$, $\forall B \in L^Y$

$$A_1(y)=\inf_{x \in X} (A'(x) \vee R(x,y)) \quad (1)$$

$$B_2(x)=\inf_{y \in Y} (B'(y) \vee R(x,y)) \quad (2)$$

con A' y B' complementarios de A y B respectivamente, donde A_1 representa de forma difusa a los atributos relacionados con los objetos de A y B_2, a los objetos relacionados con los atributos de B .

La información almacenada en el contexto se visualiza mediante los conceptos L-Fuzzy que son pares $(M,M_1) \in (L^X,L^Y)$ con $M \in fix(\varphi)$, conjunto de puntos fijos del operador φ, siendo éste definido a partir de los operadores derivación 1 y 2 antes mencionados como $\varphi(M)=(M_1)_2=M_{12}$. Estos pares, cuyas primera y segunda componentes son la extensión e intensión respectivamente, representan de forma vaga a un grupo

[1] Este trabajo ha sido financiado con el proyecto TIC 2003-08693 del Ministerio de Ciencia y Tecnología.

de objetos que comparten unos atributos.

El conjunto $\mathcal{L} = \{(M,M_1)/\ M \in fix(\varphi)\}$ con la relación de orden $\leq$ definida como:
$\forall (M,M_1),(N,N_1) \in \mathcal{L},\ (M,M_1) \leq (N,N_1)$ si $M \leq N$ (o equiv. $N_1 \leq M_1$) es un retículo completo al que llamamos ([2],[3]) retículo de conceptos L-Fuzzy.

Podemos establecer relaciones entre la teoría de conceptos L-Fuzzy y la minería de datos a través de las reglas de asociación. El primer paso será encontrar todos los conjuntos frecuentes ([1],[5],[7]). Dado un umbral al que llamaremos soporte mínimo, un conjunto se llama frecuente si su soporte es mayor o igual que dicho soporte mínimo, entendiendo por soporte el cociente entre el número de veces que aparecen los elementos del conjunto dividido por el número máximo posible.

Trasladando esta idea a un contexto formal, dado $B \subseteq M$, $sop(B)=\frac{|B^*|}{|G|}$ siendo B^* (subconjunto derivado de B según la teoría de R. Wille) el conjunto de objetos relacionados con todos los atributos de B.
Diremos que B es frecuente cuando $sop(B) \geq sopmin$, con un $sopmin \in [0,1]$ fijado. Esta idea sirve a G. Stumme y otros ([8]) para definir los conceptos frecuentes sin más que mirar si su intensión es frecuente y de ahí intentar quedarse con los conceptos de mayor interés de un determinado contexto dando lugar a lo que llama retículo de conceptos iceberg.

Como ya hemos dicho, ligada a la idea de conjuntos frecuentes aparecen las reglas de asociación ([1],[5],[7]) y su aplicación a la extracción de conocimiento en grandes bases de datos. Agrawal ([1]) en 1993 diseñó un algoritmo para determinar, para unos determinados valores de soporte y confianza $s,c \in [0,1]$, todas las reglas de asociación cuyos soportes y grados de confianza quedan por encima de s y c.

G. Stumme ([8]) utiliza estas reglas de asociación en términos de la teoría formal de conceptos como sigue:
Dado un contexto formal (G,M,I), una implicación $B \rightarrow C$, con $B,C \subseteq M$ se cumple con un soporte

$$sop(B \rightarrow C) = sop(B \cup C) = \frac{|(B \cup C)^*|}{|G|}$$

y una confianza

$$conf(B \rightarrow C) = \frac{sop(B \cup C)}{sop(B)}$$

donde se entiende por $B \cup C$ el conjunto formado por los elementos de B y C.

Reglas de asociación que ocurren sólo con un cierto grado de confianza han sido investigadas por Luxenburger ([6]) en el marco de la teoría formal de conceptos. A partir de este punto, existen en la literatura varios algoritmos para realizar minería de reglas de asociación. Todos ellos pasan por determinar los conjuntos frecuentes y a partir de ellos las reglas de asociación $B \rightarrow C$ cuyo valor de confianza queda por encima de un valor mínimo determinado.

A continuación, pasamos a mostrar cómo podemos utilizar estas herramientas de minería de datos en el marco de la Teoría de conceptos L-Fuzzy. Ésta será una primera aproximación al tema que esperamos profundizar en trabajos futuros.

2 CONCEPTOS L-FUZZY FRECUENTES

2.1. OBJETOS Y ATRIBUTOS L-FUZZY FRECUENTES

Dado un contexto L-Fuzzy, en [4] hicimos un estudio en profundidad sobre el proceso de reducción del mismo. En ese trabajo eliminábamos aquellos objetos y atributos que no jugaban un papel relevante en el contexto tras el cálculo de sus conceptos L-Fuzzy asociados.
En el caso en que el tamaño del contexto sea grande, puede ser interesante disponer de una herramienta que nos permita prescindir de algunos de esos objetos o atributos sin más que realizar sencillos cálculos.

Con este fin, adaptaremos la definición de conjuntos frecuentes al caso de contextos L-Fuzzy:

Sea (L,X,Y,R) un contexto L-Fuzzy.
Dado $A \in L^X$, definimos la función soporte como el cociente de cardinales

$$sop(A)= \frac{\sum A_1(y)}{|Y|} \qquad (3)$$

Análogamente, dado $B \in L^Y$, definimos

$$sop(B)= \frac{\sum B_2(x)}{|X|} \qquad (4)$$

Podemos reducir el tamaño del contexto L-Fuzzy original si eliminamos todos los objetos y atributos L-Fuzzy poco frecuentes, es decir, aquellos en los que el soporte sea menor que un soporte mínimo determinado. Esto nos permitirá disponer de un contexto más manejable, sin embargo, tenemos que ser conscientes de que el hecho de eliminar algunos objetos y atributos puede ser interesante para trabajar con contextos más pequeños, pero puede ser desaconsejable en el caso de que queramos realizar el estudio de ciertos objetos o atributos a pesar de sus bajos valores. (Pensemos en el caso de un producto nuevo que se introduce en el mercado y que todavía no tiene muchas ventas)

2.1.1. Ejemplo

Supongamos que queremos hacer un estudio del consumo de una serie de productos (atributos) en distintos establecimientos de una cadena de alimentación (objetos). La siguiente tabla recoge la cantidad de cada uno de los productos vendidos en cada establecimiento:

Tabla 1: Venta de los productos Pi en los establecimientos Ej

T	P1	P2	P3	P4	P5	P6	P7
E1	541	1264	1350	1361	570	535	534
E2	1700	1462	2976	2217	1676	3256	2827
E3	1750	5280	4917	4715	1841	1411	1421
E4	460	1107	1110	1042	540	484	331
E5	1261	1714	3208	3332	2120	3153	4505
E6	345	638	563	1039	213	453	238

Abordaremos este tema a través de un contexto L-Fuzzy (L,X,Y,R) donde L={0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9, 1}, X será el conjunto de establecimientos, Y el conjunto de productos y R la relación que resulte de transformar la tabla T dividiendo todos los elementos por el mayor de la tabla y realizando los correspondientes redondeos para que los valores pertenezcan a la cadena L.

Tabla 2: Relación L-Fuzzy R

R	P1	P2	P3	P4	P5	P6	P7
E1	0.1	0.2	0.3	0.3	0.1	0.1	0.1
E2	0.3	0.3	0.6	0.4	0.3	0.6	0.5
E3	0.3	1	0.9	0.9	0.3	0.3	0.3
E4	0.1	0.2	0.2	0.2	0.1	0.1	0.1
E5	0.2	0.3	0.6	0.6	0.4	0.6	0.9
E6	0.1	0.1	0.1	0.2	0	0.1	0

En este caso, tomaremos, por ejemplo, como soporte mínimo 0.1 (10%). Es decir, eliminaremos aquellos objetos y atributos cuyo soporte quede por debajo de 0.1.
Para hacerlo asociaremos a cada objeto o atributo su correspondiente conjunto L-Fuzzy asociado.
Así, por ejemplo, el conjunto L-Fuzzy asociado a E6 será *E6*={E1/0, E2/0, E3/0, E4 /0, E5/0, E6/1} y definiremos su soporte mediante las expresiones dadas en (1) y (3).
Como sop(*E6*)= 0.6/7<0.1 y el resto de los soportes son mayores que 0.1, eliminaremos sólamente el objeto E6.

Si una vez eliminado E6 construimos el retículo de conceptos L-Fuzzy asociado al contexto reducido, obtenemos un total de 39 conceptos L-Fuzzy.
Para interpretar un concepto L-Fuzzy se pondrá atención en aquellos objetos y atributos cuyos grados de pertenencia destaquen frente al resto. Así, por ejemplo, del concepto L-Fuzzy (A37, B37) con

A37={E1/0.6, E2/0.6, E3/0.7, E4 /0.6, E5/0.6}
B37={P1/0.3,P2/0.4,P3/0.4,P4/0.4,P5/0.3,P6/0.3,P7/0.3})

se puede concluir que "*los productos P2, P3 y P4 son los que mayor consumo tienen en el establecimiento E3, y sólamente en este establecimiento"*.

A continuación mostramos el diagrama de líneas de este retículo de conceptos donde los círculos representan los conceptos L-Fuzzy. Además, se puede ver la relación de orden existente entre los distintos conceptos de forma que (Ai,Bi)≤(Aj,Bj) si existe una línea ascendente del concepto que tiene la etiqueta i al concepto que tiene la etiqueta j:

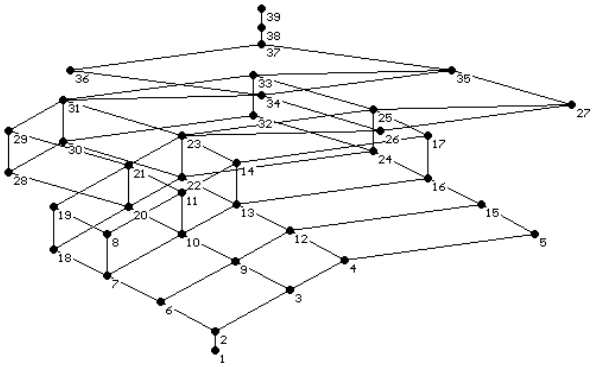


Figura 1: Retículo de conceptos L-Fuzzy

2.2. CONCEPTOS L-FUZZY FRECUENTES

Debido a la existencia de un excesivo número de conceptos L-Fuzzy en la mayoría de los contextos que se manejan (como se puede ver en el ejemplo anterior), se hace necesario articular métodos que permitan organizar la información que se desprende de los mismos.

Si lo que se pretende es tener una visión global del retículo de conceptos, se pueden establecer clasificaciones de los conceptos L-Fuzzy como ya hicimos en [4].

Sin embargo, en ocasiones resulta interesante quedarnos con aquellos conceptos L-Fuzzy que tengan mayor relevancia, entendiendo por relevancia que en ellos intervengan un número importante de objetos.
Esta es la misma filosofía que hemos seguido al definir los objetos y atributos frecuentes y que queremos trasladar al ámbito de los conceptos L-Fuzzy.

2.2.1. Definición

Dado sopmin∈[0,1], diremos que un concepto L-Fuzzy (A,B) ∈ $\mathcal{L}$ es frecuente si su intensión es frecuente, es decir, sop(B) ≥sopmin.

2.2.2. Definición

Sea $\mathcal{L}$ el retículo de conceptos L-Fuzzy del contexto (L,X,Y,R). Dado un valor sopmin$\in$[0,1], el retículo de conceptos frecuentes $\mathcal{L}_f$ será el conjunto :
$\mathcal{L}_f$={(A,B) $\in$ $\mathcal{L}$ / sop(B) $\geq$sopmin}.

Debido a que la función soporte es monótona decreciente, el retículo de conceptos frecuentes es un sup-semirretículo del retículo de conceptos L-Fuzzy al que tendremos que añadir el elemento mínimo para convertirlo en retículo.

Por otra parte, si observamos la representación gráfica del retículo de conceptos L-Fuzzy vemos que para un soporte mínimo determinado, el retículo de conceptos L-Fuzzy frecuentes estará formado por los conceptos L-Fuzzy que quedan en la parte superior del gráfico y que representan la información más importante que se obtiene del contexto, de forma que cuando disminuimos ese soporte mínimo aparecen nuevos conceptos que nos permiten ver más detalles del contexto como se puede observar en el ejemplo. (Cuanto más pequeño sea el soporte mínimo dispondremos de una información más rica, aunque también más extensa).

2.2.3. Ejemplo

Si volvemos al retículo de conceptos del ejemplo anterior, y fijamos como soporte mínimo 0.62 (62%) nos quedaremos con los siguientes conceptos L-Fuzzy frecuentes (en general los soportes son bajos porque los valores de la tabla son bajos):

A37={E1/0.6, E2/0.6, E3/0.7, E4 /0.6, E5/0.6}
B37={P1/0.3,P2/0.4,P3/0.4,P4/0.4,P5/0.3,P6/0.3,P7/0.3}

A38={E1/0.7, E2/0.7, E3/0.7, E4 /0.7, E5/0.7}
B38={P1/0.3,P2/0.3,P3/0.3,P4/0.3,P5/0.3,P6/0.3,P7/0.3}

A39={E1/0.8, E2/0.8, E3/0.8, E4 /0.8, E5/0.8}
B39={P1/0.2,P2/0.2,P3/0.2,P4/0.2,P5/0.2,P6/0.2,P7/0.2}

Los conceptos L-Fuzzy en los que coinciden todos los grados de pertenencia de los objetos y todos los grados de pertenencia de los atributos no son interesantes para nosotros como ya analizamos en [3]. Por tanto, en este caso sólo nos fijaremos en el concepto L-Fuzzy (A37, B37).

Como ya hemos comentado antes, en él "*se estudia el alto consumo de los productos P2, P3 y P4 en el establecimiento E3*".

Veamos dónde quedan estos conceptos en el gráfico (los hemos rodeado de un cuadrado):

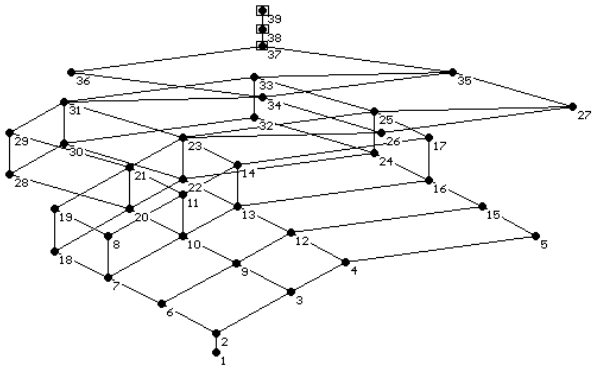


Figura 2: Conceptos L-Fuzzy frecuentes con soporte mínimo 62%.

Si bajamos el nivel de exigencia mediante el soporte mínimo a 54% añadimos a la lista anterior los conceptos L-Fuzzy frecuentes que aparecen a continuación y que nos aportan nueva información donde intervienen nuevos establecimientos (E2 y E5). Así, por ejemplo, podemos ver en el concepto L-Fuzzy (A27,B27) que "*lo que se consume fundamentalmente en los establecimientos E2, E3 y E5 (sobretodo en E3) son los productos P2, P3 y P4 (sobretodo P3)*".

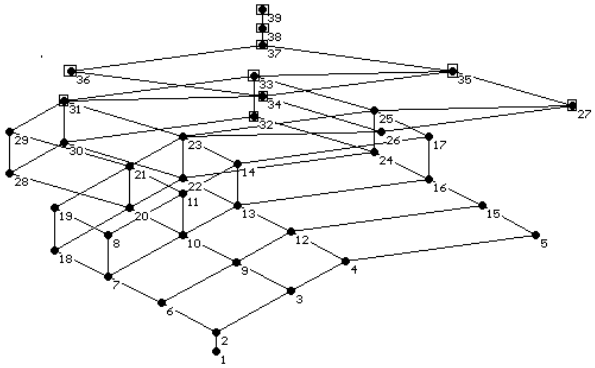


Figura 3: Conceptos L-Fuzzy con soporte mínimo 54%.

A27={E1/0.4, E2/0.6, E3/0.7, E4 /0.4, E5/0.6}
B27={P1/0.3,P2/0.4,P3/0.6,P4/0.4,P5/0.3,P6/0.3,P7/0.3}

A31={E1/0.5, E2/0.5, E3/0.6, E4 /0.5, E5/0.6}
B31={P1/0.4,P2/0.4,P3/0.5,P4/0.5,P5/0.4,P6/0.4,P7/0.4}

A32={E1/0.5, E2/0.5, E3/0.7, E4 /0.5, E5/0.5}
B32={P1/0.3,P2/0.5,P3/0.5,P4/0.5,P5/0.3,P6/0.3,P7/0.3}

A33={E1/0.5, E2/0.5, E3/0.7, E4 /0.5, E5/0.6}
B33={P1/0.3,P2/0.4,P3/0.5,P4/0.5,P5/0.3,P6/0.3,P7/0.3}

A34={E1/0.5, E2/0.6, E3/0.6, E4 /0.5, E5/0.6}
B34={P1/0.4,P2/0.4,P3/0.5,P4/0.4,P5/0.4,P6/0.4,P7/0.4}

A35={E1/0.5, E2/0.6, E3/0.7, E4 /0.5, E5/0.6}
B35={P1/0.3,P2/0.4,P3/0.5,P4/0.4,P5/0.3,P6/0.3,P7/0.3}

A36={E1/0.6, E2/0.6, E3/0.6, E4 /0.6, E5/0.6}
B36={P1/0.4,P2/0.4,P3/0.4,P4/0.4,P5/0.4,P6/0.4,P7/0.4}

Finalmente, con un soporte mínimo del 42% tendremos que analizar también, además de los anteriores, otros 23 conceptos L-Fuzzy nuevos (donde aparecen nuevos productos), entre los que destacaría:

A8={E1/0.4, E2/0.4, E3/0.4, E4 /0.4, E5/0.6}
B8={P1/0.4,P2/0.4,P3/0.6,P4/0.6,P5/0.4,P6/0.6,P7/0.6}

En este caso, se observa que "*E5 es el único establecimiento que tiene un alto consumo de P3, P4, P6 y P7*".

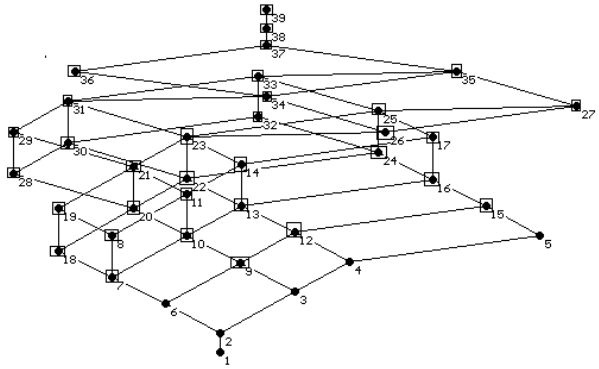


Figura 4: Conceptos L-Fuzzy con soporte mínimo 42%.

3 IMPLICACIONES ENTRE ATRIBUTOS L-FUZZY DEFINIDAS MEDIANTE REGLAS DE ASOCIACIÓN

Uno de los temas que quedan pendientes por desarrollar en la teoría de conceptos L-Fuzzy es el de definir implicaciones entre los distintos atributos L-Fuzzy que nos permitan establecer las relaciones entre ellos. Debido al carácter difuso del contexto, resulta difícil poder definir estas implicaciones con un grado de certeza del 100% tal y como sucede en la teoría formal de conceptos. Sin embargo, es posible establecer implicaciones entre dichos atributos a distintos niveles de confianza utilizando reglas de asociación.

Con este fin, tomaremos las definiciones dadas de soporte y grado de confianza de una regla, así como nuestra propia definición de soporte de un subconjunto de atributos para establecer la implicación entre atributos B⇒D.

3.1. Definición

Sea (L,X,Y,R) un contexto L-Fuzzy y sean $B,D \subseteq L^Y$.
El soporte y la confianza de esta implicación B⇒D serán

$$sop(B \Rightarrow D) = sop(B \cup D) = \frac{\sum (B \cup D)_2(x)}{|X|} \quad (5)$$

$$conf(B \Rightarrow D) = \frac{sop(B \cup D)}{sop(B)} = \frac{\sum (B \cup D)_2(x)}{\sum B_2(x)} \quad (6)$$

y representarán, en el primer caso, el porcentaje de objetos que comparten los atributos de la premisa y la conclusión, y en el segundo, dentro de esos objetos, cuál es el porcentaje de los que verifican la implicación.

Para nosotros, está será la forma de representar la idea de que todo objeto que tiene los atributos de B en un determinado grado, tiene también los de D en el mismo grado.

3.2. Ejemplo

Siguiendo el ejemplo 2.1.1., vamos a ver cuánto valen el soporte y grado de confianza de algunas reglas.
Obsérvese que aunque en este caso tal vez las implicaciones podrían leerse desde la tabla, no siempre es así, debido al tamaño de la tabla o a que la premisa y la conclusión sean compuestas.

Sean *P1*={P1/1, P2/0,...P7/0} y *P2*={P1/0, P2/1,P3/0,... P7/0}.

- Implicación *P1*⇒*P2*

sop(*P1*⇒*P2*)=1/5 (20%)
conf(*P1*⇒*P2*)=1 (100%)

- Implicación *P2*⇒*P1*

sop(*P2*⇒*P1*)=1/5 (20%)
conf(*P2*⇒*P1*)=1/2 (50%)

De estas dos reglas podemos concluir que "*el porcentaje de establecimientos donde se venden los productos P1 y P2 al mismo tiempo es bajo (soporte 20%) y siempre que se vende P1 también se vende P2, aunque no al contrario (grados de confianza de 100% y 50% respectivamente)*".

- Implicación *P3*⇒*P4*

sop(*P3*⇒*P4*)=2.4/5 (48%)
conf(*P3*⇒*P4*)=2.4/2.6 (92,30%)

- Implicación *P4*⇒*P3*

sop(*P4*⇒*P3*)=2.4/5 (48%)
conf(*P4*⇒*P3*)=2.4/2.4 (100%)

En este caso, podemos decir que "*hay un mayor porcentaje de establecimientos en los que se venden P3 y P4 (soporte 48%) y ambos grados de confianza son muy*

altos (siempre que se vende P4 se vende P3 y un alto número de las veces en que se vende P3, se vende P4)".

Veamos a continuación dos implicaciones donde la premisa o la conclusión son compuestas:

Sean para nosotros *P3*={P1/0, P2/0, P3/1, P4/0...P7/0} y *P6 y P7*={P1/0, P2/0,..., P6/1, P7/1}.

- Implicación *P3*⇒*P6 y P7*

sop(*P3*⇒*P6 y P7*)=1.6/5 (32%)
conf(*P3*⇒*P6 y P7*)=0.32/0.52 (61.53%)

- Implicación *P6 y P7*⇒*P3*

sop(*P6 y P7*⇒*P3*)=2.4/5 (32%)
conf(*P6 y P7*⇒*P3*)=0.32/0.32 (100%)

"El porcentaje de establecimientos donde se venden al mismo tiempo P3, P6 y P7 es del 32%, de forma que siempre que se venden P6 y P7 se vende P3 (100% de confianza), aunque no ocurre lo mismo en sentido contrario (61.53% de confianza)".

Queremos destacar la importancia de conocer estas implicaciones entre atributos de cara a mejorar la política de ventas en los establecimientos de la empresa.

LINEAS FUTURAS

También en este caso tendremos el problema de la existencia de un excesivo número de implicaciones entre atributos.
Será por tanto necesario realizar en trabajos futuros minería de reglas de asociación L-Fuzzy, que consistirá en determinar todas las implicaciones cuyo grado de confianza y soporte quedan por encima de unos determinados umbrales.

Referencias

[1] R. Agrawal, T. Imielinski, A. Swami. Mining association rules between sets of items in large databases. *Proc. ACM SIGMOD*, *Washington,* Pág. 207-216, 1993.

[2] A. Burusco y R. Fuentes-González. The study of the L-Fuzzy Concept Lattice. *Mathware and Soft Computing*, Vol. 1, Nº 3, Pág. 209-218, 1994.

[3] A. Burusco y R. Fuentes-González. Construction of the L-Fuzzy Concept Lattice. *Fuzzy Sets and Systems*, Vol. 97, Nº 1, Pág. 109-114, 1998.

[4] A. Burusco y R. Fuentes-González. Extracción de información relevante en contextos L-Fuzzy. *Información tecnológica*, Vol. 14, Nº 4, Pág. 65-70, 2003.

[5] J. Han, J. Pei, Y. Yin. Mining frequent patterns without candidate generation. *Proc. ACM SIGMOD*, *Dallas,* Pág. 1-12, 2000.

[6] M. Luxenburger. Implications partielles dans un contexte. *Mathemátiques, informatique et sciences humaines*, Vol. 113, Pág. 35-55, 1991.

[7] N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal. Discovering frequent closed itemsets for association rules. *ICDT, Jerusalem,* Pág. 398-416, 1999.

[8] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, L. Lakhal. Computing iceberg concept lattices with Titanic. *Journal on Knowledge and Data Engineering*, Vol. 42, Nº 2, Pág. 189-222, 2002.

[9] R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. *I. Rival(ed.) Ordered Sets, Reidel, Dordrecht-Boston* , Pág. 445-470, 1982.

A New Linguistic Modelling of the Symmetrical Threshold Semantics*

E. Herrera-Viedma
Dept. of Computer Science and Artificial Intelligence, Library Science Studies School, University of Granada, 18017 – Granada, Spain
viedma@decsai.ugr.es

A. G. López-Herrera
Dept. of Computer Science and Artificial Intelligence, Library Science Studies School, University of Granada, 18017 – Granada, Spain
agabriel@ugr.es

L. Hidalgo
Office of Transference of Investigation Results, University of Granada, E-18071 – Granada, Spain
lhidalgo@ugr.es

Abstract

In this contribution a new matching function for a symmetrical threshold semantics is presented. It can be interpreted as a tuning of that proposed in [10] which was defined using an ordinal fuzzy linguistic approach. This new matching function is defined using a 2-tuple fuzzy linguistic approach and, consequently, avoids the loss of precision and information in final results. On the other hand, it softens the behaviour of that proposed in [10] by processing in a more consistent way the query threshold weights. In such a way, this new linguistic matching function improves the results of linguistic information retrieval systems and therefore, can help to improve the users' satisfaction.

Keywords: Fuzzy Information Retrieval, Linguistic Modelling, Weighted Queries.

1. Introduction

The main activity of an Information Retrieval System (IRS) is the gathering of pertinent archived documents that better satisfy the user queries. IRSs present three components to carry out this activity [10, 11]: i) a *database*: which stores the documents and the representation of their information contents (index terms), ii) a *query subsystem*: which allows users to formulate their queries by means of a query language, iii) a*n evaluation subsystem:* which evaluates the documents for a user query obtaining a Retrieval Status Value (RSV) form each document. The query subsystem supports the user-IRS interaction, and therefore, it should be able to account for the imprecision and vagueness typical of human communication. This aspect may be modelled by means of the introduction of weights in the query language. Many authors have proposed weighted IRS models using Fuzzy Set Theory [2, 3, 4, 7, 8, 9, 16, 19, 20, 21]. Usually, they assume numeric weights associated with the queries (values in [0, 1]). However, the use of query languages based on numeric weights forces the user to quantify qualitative concepts (such as "importance"), ignoring that many users are not able to provide their information needs precisely in a quantitative form but in a qualitative one. Some fuzzy linguistic IRS models [5, 6, 10, 11, 12, 17] have been proposed using a *fuzzy linguistic approach* [24] to model the query weights and document scores. A useful fuzzy linguistic approach which allows us to reduce the complexity of the design for the IRSs [10,11] is called the *ordinal fuzzy linguistic approach* [14, 15, 22]. In this approach, the query weights and document scores are ordered linguistic terms.

On the other hand, we have to establish the semantics associated with the query weights to formalize fuzzy linguistic weighted querying. There are four semantic possibilities [3, 10, 17]: i) weights as a measure of the importance of a specific element in representing the query, ii) as a threshold to aid in matching a specific document to the query, iii) as a description of an ideal or perfect document, and iv) as a limit on the amount of documents to be retrieved for a specific element.

In [10] a variant for a threshold semantics, called symmetrical threshold semantics, was proposed. This semantics has a symmetric behaviour in both sides of the mid threshold value. It assumes that a user may use presence weights or absence weights in the formulation of weighted queries. Then, it is symmetrical with respect to the mid threshold value, i.e., it presents the usual behaviour for the threshold values which are on the right of the mid linguistic value (presence weights), and the opposite behaviour for the values which are on the left (absence weights or presence weights with low value). To evaluate this semantics, in [10] was defined a parameterized symmetrical linguistic matching function. This function has like main limitation the loss of

* Este trabajo ha sido financiado por el proyecto: *GLIRS-II: Un Sistema de Recuperación de Información Documental Basado en Información Lingüística Difusa y Algoritmos Genéticos*. Ministerio de Ciencia y Tecnología. Ref. TIC-2003-07977.

information and precision in final results, i.e. in the computation of the linguistic RSVs of documents.

In this contribution we present a new modelling of the symmetrical threshold semantics defined in [10] which overcomes its difficulties. We present a new and alternative definition of the symmetrical matching function that synthesizes the symmetrical threshold semantics, softens the behaviour of that defined in [10], and allows to achieve more precise RSVs, improving the results of the retrieval, and consequently, allowing to increase the users' satisfaction. This new symmetrical matching function is defined in a 2-tuple fuzzy linguistic approach [13]. By using the 2-tuple fuzzy linguistic representation model we improve the precision in the representation of linguistic information and by using the 2-tuple computational model we avoid the loss of information in the combination operations of linguistic information.

The paper is structured as follows. Section 2 presents the 2-tuple fuzzy linguistic approach. Section 3 defines the new linguistic symmetrical matching function and accomplishes a study of its performance. And finally, in Section 4, some concluding remarks are pointed out.

2. A 2-tuple fuzzy linguistic approach

The *ordinal fuzzy linguistic approach* is an approximate technique appropriate to deal with qualitative aspects of problems [15]. An ordinal fuzzy linguistic approach is defined by considering a finite and totally ordered label set $S = \{s_0, \ldots, s_T\}$, $T+1$ is the cardinality of S in the usual sense, and with odd cardinality (7 or 9 labels). The mid term representing an assessment of "approximately 0.5" and the rest of the terms being placed symmetrically around it [1]. The semantics of the linguistic terms set is established from the ordered structure of the terms set by considering that each linguistic term for the pair (s_i, s_{T-i}) is equally informative. For each label s_i is given a fuzzy number defined on the [0,1] interval, which is described by a membership function. The computational model to combine ordinal linguistic information is based on the following operators:

1. Negation operator: $Neg(s_i) = s_j, j = T - i$.
2. Maximization operator: $MAX(s_i, s_j) = s_i$ if $s_i >= s_j$.
3. Minimization operator: $MIN(s_i, s_j) = s_i$ if $s_i <= s_j$
4. Aggregation operators: Usually to combine ordinal linguistic information we use aggregation operators based on symbolic computation, e.g. the LOWA operator [15] or the LWA operator [14].

Let S be a linguistic term set, if a symbolic method aggregating linguistic information obtains a value $\beta \in [0, T]$, and $\beta \notin \{0, \ldots, T\}$ then an approximation function (app(.)) is used to express the index of the result in S [13].

For example, in the LOWA, app(.) is the simple function *round.*

Definition 1. [13] Let $\beta \in [0, T]$ be the result of an aggregation of the indexes of a set of labels assessed in a linguistic term set S, i.e., the result of a symbolic aggregation operation. Let $i = round(\beta)$ and $\alpha_i = \beta - i$ be two values, such that, $i \in \{0, 1, .., T\}$ and $\alpha_i \in [-.5, .5)$ then α_i is called a *Symbolic Translation*.

From this concept, F. Herrera and L. Martínez developed a linguistic representation model which represents the linguistic information by means of 2-tuples (s_i, α_i), $s_i \in S$ and $\alpha_i \in [-.5, .5)$ [13]:

- s_i represents the linguistic label of the information, and
- α_i is a numerical value expressing the value of the translation from the original result β to the closest index label i in S.

This model defines a set of transformation functions between numeric values and linguistic 2-tuples.

Definition 2. [13] Let S be a linguistic term set and $\beta \in [0, T]$, then the 2-tuple that expresses the equivalent information to β is obtained with the following function:

$\Delta: [0, T] \rightarrow S \times [-.5, .5)$;

$$\Delta(\beta) = (s_i, \alpha_i), \text{ with } \begin{cases} s_i & i = round(\beta) \\ \alpha_i = \beta - i & \alpha_i \in [-.5, .5) \end{cases} \quad (1)$$

where s_i has the closest index label to "β" and "α_i" is the value of the symbolic translation.

Proposition 1. [13] Let (s_i, α_i), $s_i \in S$ be a linguistic 2-tuple. There is always a Δ^{-1} function, such that, from a 2-tuple it returns its equivalent numerical value $\beta \in [0, T] \subset \Re$.

Remark 1: [13] From Definition 2 and Proposition 1, it is obvious that the conversion of a linguistic term into a linguistic 2-tuple consists of adding a value 0 as symbolic translation: $s_i \in S \rightarrow (s_i, 0)$.

The 2-tuple linguistic computational model operates with the 2-tuples without loss of information and is based on the following operations [13]:

1. Negation operator of a 2-tuple:

$$Neg(s_i, \alpha_i) = \Delta(T - \Delta^{-1}(s_i, \alpha_i)).$$

2. Comparison of 2-tuples: The comparison of linguistic information represented by 2-tuples is carried out according to an ordinary lexicographic order.

3. Aggregation of 2-tuples: Using the functions Δ and Δ^{-1} any numerical aggregation operator can be easily extended for dealing with linguistic 2-tuples.

Definition 3. [23] Let $A = \{a_1, ..., a_m\}$, $a_k \in [0,1]$ be a set of assessments to aggregated, then the OWA operator, ϕ, is defined as $\phi(a_1, ...,a_m) = W \cdot B^T$, where $W = [w_1, ..., w_m]$, is a weighting vector, such that $w_i \in [0,1]$ and $\Sigma_i w_i = 1$, and $B = \{b_1, ..., b_m\}$ is a vector associated to A, such that, $B = \sigma(A) = \{a_{\sigma(1)}, ..., a_{\sigma(m)}\}$, with σ being a permutation over the set of assessments A, such that $a_{\sigma(j)} \leq a_{\sigma(i)}$ $\forall i \leq j$.

A 2-tuple linguistic extended definition of ϕ would be as follows:

Definition 4. Let $A = \{(a_1, \alpha_1),..., (a_m, \alpha_m)\}$ be a set of assessments in the linguistic 2-tuple domain, then the 2-tuple linguistic OWA operator, ϕ_{2t} is defined as

$$\phi_{2t}((a_1,\alpha_1),...,(a_m,\alpha_m)) = \Delta(W \cdot B^T),$$

$$B = \sigma(A) = \left\{(\Delta^{-1}(a_1,\alpha_1))_{\sigma(1)},...,(\Delta^{-1}(a_m,\alpha_m))_{\sigma(m)}\right\}.$$

3. A new linguistic matching function for a symmetrical threshold semantics

In this section we present a new proposal to model the symmetrical threshold semantics defined in [10] in order to improve its performance. Before presenting it, we show the linguistic IRS assumed.

3.1. An ordinal linguistic weighted IRS based on a symmetrical threshold semantics

The ordinal linguistic weighted IRS that we use in our research presents the following elements to carry out its activity:

1. Database: we assume a database of a traditional fuzzy IRS as in [8, 19, 18]. The database stores the finite set of documents $D = \{d_1,\ldots, d_m\}$ represented by a finite set of index terms $T = \{t_{1,...,}t_l\}$, which describe the subject content of the documents. The representation of a document is a fuzzy set of terms characterized by a numeric indexing function $F: D\times T \rightarrow [0, 1]$, which is called index term weight [19]: $d_j = F(d_j, t_1)/t_1 + F(d_j, t_2) /t_2 + \ldots + F(d_j, t_l) /t_l$. F weighs index terms according to their significance in describing the content of a document. Thus $F(d_j, t_i)$ is a numerical weight that represents the degree of significance of t_i in d_j.

2. *Query subsystem: w*e use a query subsystem with a fuzzy linguistic weighted Boolean query language to express user information needs. With this language each query is expressed as a combination of the weighted index terms that are connected by logical operators AND ($\wedge$), OR ($\vee$), and NOT($\neg$). The weights are ordinal linguistic values taken from a label set S, and they are associated with a symmetrical threshold semantics [10, 11]. As in [5], our atomic components are pairs, $\langle t_i, c_i\rangle$, where t_i is an index term, but defining the linguistic variable *Importance* with the ordinal linguistic approach and associating c_i with a symmetrical threshold semantics. Accordingly, the set $\boldsymbol{Q}$ of the legitimate queries is defined by the following syntactic rules:

1. $\forall q = \langle t_i, c_i\rangle \in \boldsymbol{T} \times S \rightarrow q \in \boldsymbol{Q}$.
2. $\forall q, p \in \boldsymbol{Q} \rightarrow q \wedge p \in \boldsymbol{Q}$.
3. $\forall q, p \in \boldsymbol{Q} \rightarrow q \vee p \in \boldsymbol{Q}$.
4. $\forall q \in \boldsymbol{Q} \rightarrow \neg q \in \boldsymbol{Q}$.
5. *All legitimate queries* $q \in \boldsymbol{Q}$ *are only those obtained by applying rules 1-4, inclusive.*

3. Evaluation subsystem: The evaluation subsystem for weighted Boolean queries acts by means of a constructive bottom-up process based on the *criterion of separability* [9, 19]. The RSVs of the documents are ordinal linguistic values whose linguistic components are taken from the linguistic variable *Importance* but representing the concept of *relevance*. Therefore, the set of linguistic terms S is also assumed to represent the relevance values. The evaluation subsystem acts in two steps:

1. Firstly, the documents are evaluated according to their relevance only to atoms of the query. In this step, the symmetrical threshold semantics is applied in the evaluation of atoms by means of a parameterized linguistic matching function $g : \boldsymbol{D}\times\boldsymbol{T}\times S \rightarrow S$, which is defined as [10]:

$$g(d_j,t_i,c_i) = \begin{cases} s_{Min\{a+\beta,T\}} & s_{T/2} \leq s_b \leq s_a \\ s_{Max\{0,a-\beta\}} & s_{T/2} \leq s_b \wedge s_a < s_b \\ Neg(s_{Max\{0,a-\beta\}}) & s_a \leq s_b < s_{T/2} \\ Neg(s_{Min\{a+\beta,T\}}) & s_b < s_{T/2} \wedge s_b < s_a \end{cases} \quad (2)$$

such that, (i) $s_{b=} c_i$; (ii) s_a is the linguistic index weight obtained as $s_a = Label(F(d_j,t_i))$, being $Label$:$[0,1]\rightarrow S$ a function that assigns a label in S to a numeric value $r \in [0,1]$; and iii) β is a bonus value that rewards/penalizes the relevance degrees of documents for the satisfaction /dissatisfaction of request $<t_i,c_i>$, which can be defined depending on the closeness between $Label(F(d_j,t_i))$ and c_i, for example as $\beta=round(2|b-a|/T)$. We should point out that whereas the traditional threshold matching function are always non-decreasing [17], g is non-decreasing on the right of the mid term and decreasing on the left of the mid term in order to be consistent with the meaning of the symmetrical threshold semantics.

2. Secondly, the documents are evaluated according to their relevance to Boolean combinations of atomic components, and so on, working in a bottom-up fashion until the whole query is processed. In this step, the logical connectives AND and OR are modelled by means of LOWA [15] operators with $orness(W)<0.5$ and $orness(W) \geq 0.5$ respectively, being $orness(W)$ a orness measure introduced by Yager in [23] to classify the aggregation of the OWA operators: $orness(W)=(1/m-1)(\sum^{m}_{i=1} (m-i)\, w_i)$.

Remark 2: We should point out that if we have a negated

query, or a negated subexpression, or a negated atom, their evaluation is obtained from the negation of the relevance results computed for the query, or the subexpression, or atom in a no-negated situation.

3.2. Limitations of the symmetrical threshold semantics modelled by *g*

According to the symmetrical threshold semantics the evaluation subsystem assumes that a user may search for documents with a minimally acceptable presence of one term in their representations (as in the classical interpretation happens [17]) or documents with a maximally acceptable presence of one term in their representations. Then, when a user asks for documents in which the concept(s) represented by a term t_i is (are) with the value *High Importance*, the user would not reject a document with a *F* value greater than *High*; on the contrary, when a user asks for documents in which the concept(s) represented by a term t_i is (are) with the value *Low Importance*, the user would not reject a document with a *F* value less than *Low*. Given a request $<t_i, c_i> \in \boldsymbol{T}\times S;$ this means that the query weights that imply the presence of a term in a document $c_i \geq s_{T/2}$ (e.g. *High, Very High*) they must be treated differently to the query weights that imply the absence of one term in a document $c_i < s_{T/2}$ (e.g. *Low, Very Low*). Then, if $c_i > s_{T/2}$ the request $<t_i, c_i>,$ is synonymous with the request $<t_i,$ *at least* $c_i>,$ which expresses the fact that the desired documents are those having *F* values as high as possible; and if $c_i < s_{T/2}$ is synonymous with the request $<t_i,$ *at most* $c_i>,$ which expresses the fact that the desired documents are those having *F* values as low as possible. The linguistic matching function *g* defined in [10] represents a possible modelling of the meaning of the symmetrical threshold semantics. However, such modelling or interpretation presents the following limitations:

1. *The loss of precision*: This problem is a consequence of ordinal linguistic framework which works with discrete linguistic expression domains and this implies to assume limitations in the representation domain of RSVs. Therefore, as linguistic term sets (*S*) assumed have a limited cardinality (5,7 or 9 labels) to assess the linguistic RSVs, in consequence, it is difficult to distinguish or specify what documents really satisfy better the atomic weighted request $<t_i, c_i>$. Although the system retrieves many documents the possible relevance assessments are limited by the cardinality of the label set considered.

2. *The loss of information*: This problem also is a consequence of the ordinal linguistic approach because it forces us to apply approximation operations in the definition of *g*, in particular, the *rounding* operation used to calculate the parameter $\beta,$ and as it is known [13], in such a case almost always there exists a loss of information

Example 1: Let $S= \{s_0 =$*Null (N),* $s_1=$*Extremely_Low (EL),* $s_2=$*Very_Low (VL),* $s_3=$*Low (L),* $s_4=$*Medium (M),* $s_5=$*High (H),* $s_6=$*Very_High (VH),* $s_7=$*Extremely_High (EH),* $s_8=$*Total (TO)}* be a label set used to assess the linguistic information in a IRS and consider two documents d_1 and d_2, such that, $Label(F(d_1,t_i))=EH$ and $Label(F(d_2,t_i))=TO,$ respectively, then if the atomic request is $<t_i, M>$ we obtain the same relevance degree for both documents as a consequence of the loss of information, $g(d_1, t_i, M)= TO$ and $g(d_2, t_i, M)= TO$.

3. *g tends to overvalue the satisfaction/dissatisfaction of the requests*: This problem is a consequence of the own definition of *g*. For example, if we analyze its definition we can observe that relevance degrees generated when the threshold value is satisfied, i.e. $s_{Min\{a+\beta,T\}}$, always are limited by the index term weight, s_a. This shows a too optimistic evaluation of the satisfaction of threshold value and reduces the possibilities of discrimination among the documents that satisfy the threshold value. Similarly, it happens in the dissatisfaction case.

In the following subsection, we try to overcome these problems by defining a new threshold matching function.

3.3. A 2-tuple linguistic matching function to model the symmetrical threshold semantics

In this section, we present a new symmetrical matching function to model the symmetrical threshold semantics that overcomes the problems of the matching function *g* [10] aforementioned. We design it by using as base the 2-tuple fuzzy linguistic representation model [13] and we call it like 2-tuple linguistic matching function g_{2t}.
Firstly, we should point out that the simple fact to define the new matching function g_{2t} in a 2-tuple linguistic approach allows us to solve the first problem of *g,* given that using the 2-tuple linguistic representation model in its definition g_{2t} inherits its properties, and one of the main properties of the 2-tuple linguistic representation model is to eliminate the loss of precision of the ordinal linguistic model [13]. On the other hand, to overcome the second problem we have to avoid to include approximation operations in the definition of g_{2t}, and to overcome the third problem we have to soften the relevance degrees generared by g_{2t} when threshold value is minimally satisfied by the index term weight.

As aforementioned, symmetrical threshold semantics has a symmetric behaviour in both sides of the mid threshold value because it is defined to distinguish two situations in the threshold interpretation: i) when the threshold value is on the left of the mid term and ii) when it is on the right. It assumes that a user may use presence weights or absence weights in the formulation of weighted queries. Then, it is symmetrical with respect to the mid threshold value. Therefore, analyzing the case of presence weights, i.e. threshold values which are on the right of the mid

threshold value, we rapidly derive the case of absence weights.

When the linguistic threshold weight s_b given by a user is higher, in the usual sense, than middle label of the term linguistic set, $s_{T/2}$, the matching function g is non-decreasing. As aforesaid, in this case the problem of g is that it rewards excessively to those documents whose F values overcomes to the threshold weight s_b and penalizes excessively to those documents whose F values do not overcome s_b. We look for a non-decreasing matching function g_{2t} that softens the behaviour of g. Concretely, to achieve this goal g_{2t} should work as follows: the more the F values exceed the threshold values and the closer they are to the maximum RSV s_T, the greater the RSVs of the documents. However, when the F values are below the threshold values and closer to s_0, the lower the RSVs of the documents and the closer to s_0 they are. These two circumstances are called in the literature oversatisfaction and undersatisfaction [17]. Assuming a continuous numeric domain [0, T], in Figure 1 we represent graphically the desired behaviour of g_{2t} for three possible threshold values *T/2,* u and u', being values *0*, *T/2,* and T the indexes of the following terms of *S:* bottom term, middle term and top term, respectively.

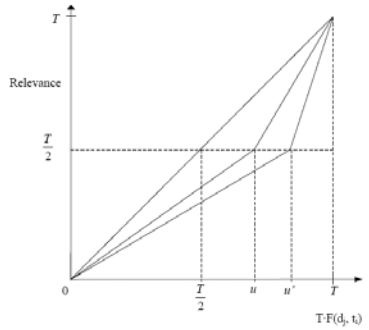


Figure 1. Desired behaviour of the matching function g_{2t}.

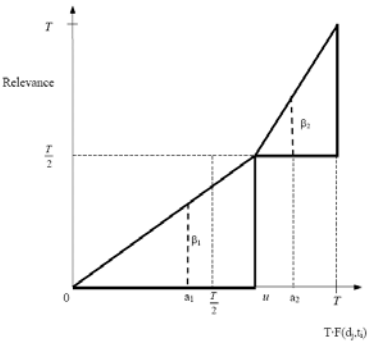


Figure 2. Desired behaviour of g_{2t} for a threshold value on the right of the mid term.

If we focus on the case of threshold value u (see Figure 2), then given two possible values of index term weight $a_1<u$ and $a_2>u$, the relevance degrees obtained by a desired matching function should be β_1 and *(T/2)* + β_2. Assuming this hypothesis the definition of the 2-tuple linguistic matching function g_{2t} on the right of the mid term would be as follows:

$$g_{2t} : \boldsymbol{D}\times\boldsymbol{T}\times(S\times[-.5, .5)) \rightarrow S\times[-.5, .5))$$

$$g_{2t}(d_j,t_i,(s_b,0)) = \begin{cases} \Delta(\beta_2 + \frac{T}{2}) & (s_a,\alpha_a) \geq (s_b,0) \wedge (s_b,0) \geq (s_{T/2},0) \\ \Delta(\beta_1) & (s_a,\alpha_a) < (s_b,0) \wedge (s_b,0) \geq (s_{T/2},0) \end{cases} \quad (3)$$

where $(s_a,\alpha_a) = \Delta(T\cdot F(d_j,t_i))$, $(s_b,0)$ is the representation in the linguistic 2-tuple model of the linguistic threshold weight given by a user, and β_1 and β_2 are numerical values obtained as follows. In Figure 2, two triangles are showing the behaviour of the desired matching function. The triangle on the right of the mid value *T/2* shows the way in which documents that have an index term weight a_2 higher than a threshold value u are rewarded, and the triangle on the left of the mid value shows the way in which documents that have an index term weight a_1 lower than u are penalized. Analysing both triangles we can calculate the following expressions for β_2 and β_1:

$$\frac{T-(\frac{T}{2})}{T-u} = \frac{\beta_2}{a_2-u} \Rightarrow \beta_2 = \frac{T\cdot(a_2-u)}{2\cdot(T-u)}$$

$$\frac{\frac{T}{2}}{u} = \frac{\beta_1}{a_1} \Rightarrow \beta_1 = \frac{a_1\cdot\frac{T}{2}}{u} = \frac{a_1\cdot T}{2\cdot u}.$$

To apply these expressions in the 2-tuple linguistic matching function g_{2t} we must know that:

- $u = \Delta^{-1}(s_b,0)$, being s_b the linguistic threshold value provided by a user,
- a_2 would be the numeric weight of some index term t_i representing the content of a document d_j, i.e., $a_2 = T\cdot F(d_j, t_i)$, and similarly
- a_1 would be the numeric weight of some index term t_i representing the content of a document d_k, i.e., $a_1 = T\cdot F(d_k, t_i)$.

Summarizing, given that g_{2t}, like g, must present a symmetric behaviour in both sides of the mid threshold value, then the complete definition of g_{2t} is easily obtained as follows:

$$g_{2t}(d_j,t_i,(s_b,0)) = \begin{cases} \Delta(\beta_2 + \frac{T}{2}) & if\ (s_a,\alpha_a) \geq (s_b,0) \wedge (s_b,0) \geq (s_{T/2},0) \\ \Delta(\beta_1) & if\ (s_a,\alpha_a) < (s_b,0) \wedge (s_b,0) \geq (s_{T/2},0) \\ \Delta(\beta_2^* + \frac{T}{2}) & if\ (s_a,\alpha_a) \leq (s_b,0) \wedge (s_b,0) < (s_{T/2},0) \\ \Delta(\beta_1^*) & if\ (s_a,\alpha_a) > (s_b,0) \wedge (s_b,0) < (s_{T/2},0) \end{cases}$$

(4)

with $\beta_2 = \frac{T\cdot(a_2-u)}{2\cdot(T-u)}$, $\beta_1 = \frac{a_1\cdot T}{2\cdot u}$, $\beta_2^* = \frac{T\cdot(u-a_1)}{2\cdot u}$,

$\beta_1^* = \frac{T\cdot(T-a_2)}{2\cdot(T-u)}$, $u = \Delta^{-1}(s_b,0)$, $a_1 = T\cdot F(d_k, t_i)$ and

$a_2 = T \cdot F(d_j, t_i)$.

4. CONCLUDING REMARKS

In this paper we have described a new linguistic modelling of the symmetrical threshold semantics [10] in a linguistic framework. We have defined a new symmetrical linguistic matching function to model the meaning of the symmetrical threshold semantics that overcomes the problems found in the linguistic matching function defined in [10]. We have defined this new linguistic matching function in a 2-tuple fuzzy linguistic context [13] to take advantage of the usefulness of the 2-tuple fuzzy linguistic representation model with respect to avoid the problems of loss of precision and information in the results.

In the future, we shall research the impact of the different threshold matching functions existing in the literature in order to define a general application framework that facilitates us their design and use in the IRSs.

References

[1] P.P. Bonissone and K.S. Decker, Selecting uncertainty calculi and granularity: An experiment in trading-off precision and complexity, in: L.H. Kanal and J.F. Lemmer, Eds., *Uncertainty in Artificial Intelligence* (North-Holland, 1986) 217-247.

[2] A. Bookstein, Fuzzy request: An approach to weighted Boolean searches, *Journal of the American Society for Information Science* 31 (1980) 240-247.

[3] G. Bordogna, C. Carrara and G. Pasi, Query term weights as constraints in fuzzy information retrieval, *Information Processing & Management* 27 (1991) 15-26.

[4] G. Bordogna and G. Pasi, Linguistic aggregation operators of selection criteria in fuzzy information retrieval, *International Journal of Intelligent Systems* 10 (1995) 233-248.

[5] G. Bordogna and G. Pasi, A fuzzy linguistic approach generalizing Boolean Information retrieval: A model and its evaluation, *Journal of the American Society for Information Science* 44 (1993) 70-82.

[6] G. Bordogna and G. Pasi. An ordinal information retrieval model. International Journal of Uncertain, Fuzziness and Knowledge System, 9 (2001) 63-76.

[7] D. Buell and D.H. Kraft, Threshold values and boolean retrieval systems, *Information Processing & Management* 17 (1981) 127-136.

[8] D. Buell and D.H. Kraft, A model for a weighted retrieval system, *Journal of the American Society for Information Science* 32 (1981) 211-216.

[9] C.S. Cater and D.H. Kraft, A generalization and clarification of the Waller-Kraft wish list, *Information Processing & Management* 25 (1989) 15-25.

[10] E. Herrera-Viedma, Modelling the retrieval process for an information retrieval system using an ordinal fuzzy linguistic approach, *Journal of the American Society for Information Science and Technology* 52:6 (2001) 460-475.

[11] E. Herrera-Viedma, An information retrieval system with ordinal linguistic weighted queries based on two weighting elements. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems* 9 (2001) 77-88.

[12] E. Herrera-Viedma, O. Cordón, M. Luque, A.G. Lopez, A.M. Muñoz, A model of fuzzy linguistic IRS based on multi-granular linguistic information, *International Journal of Approximate Reasoning* 34 (2003) 221-239.

[13] F. Herrera and L. Martínez, A 2-tuple fuzzy linguistic representation model for computing with words, *IEEE Transactions on Fuzzy Systems* 8:6 (2000) 746-752.

[14] F. Herrera and E. Herrera-Viedma, Aggregation operators for linguistic weighted information, *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans,* 27 (1997) 646-656.

[15] Herrera, F., Herrera-Viedma, E., and Verdegay, J. L. (1996). Direct approach processes in group decision making using linguistic OWA operators. *Fuzzy Sets and Systems*, 79: 175-190.

[16] D.H. Kraft and D.A. Buell, Fuzzy sets and generalized Boolean retrieval systems, *International Journal of Man-Machine Studies* 19 (1983) 45-56.

[17] D.H. Kraft, G. Bordogna and G. Pasi, An extended fuzzy linguistic approach to generalize Boolean information retrieval, *Information Sciences* 2 (1994) 119-134.

[18] S. Miyamoto, *Fuzzy Sets in Information Retrieval and Cluster Analysis* (Kluwer Academic Publishers, 1990).

[19] W.G. Waller and D.H. Kraft, A mathematical model of a weighted Boolean retrieval system, *Information Processing & Management* 15 (1979) 235-245.

[20] R.R. Yager, A Hierarchical Document Retrieval Language, *Information Retrieval* 3 (2000) 357-377.

[21] R.R. Yager, A note on weighted queries in information retrieval system, *Journal of American Society of Information Sciences* 38 (1987) 23-24.

[22] R.R. Yager, An approach to ordinal decision making, *International Journal of Approximate Reasoning* 12 (1995) 237-261.

[23] R.R. Yager, On ordered weighted averaging aggregation operators in multicriteria decision making, *IEEE Transactions on Systems, Man, and Cybernetics* 18 (1988) 183-190.

[24] L.A. Zadeh, The concept of a linguistic variable and its applications to approximate reasoning. Part I, *Information Sciences* 8 (1975) 199-249, Part II, *Information Sciences* 8 (1975) 301-357, Part III, *Information Sciences* 9 (1975) 43-80.

DESAMBIGUACIÓN DE SIGNIFICADOS EN GUMSE BASADA EN EL REFINAMIENTO DEL SENTIDO

Javier de la Mata, José A. Olivas y Jesús Serrano-Guerrero
Universidad de Castilla-La Mancha.
Paseo de la Universidad, 4.
13071 Ciudad Real, España.
Javierl.mata@alu.uclm.es, joseangel.olivas@uclm.es, jesus.serrano3@alu.uclm.es

Resumen

La ambigüedad en el significado de las palabras es una de las causas de la degradación de las prestaciones en los sistemas de recuperación de información. Sobre todo en sistemas de uso general en los que los usuarios utilizan pocos términos en sus consultas. En este artículo se presenta un método de desambiguación del sentido, aplicado en el meta-buscador experimental GUMSe, basado en la realimentación proveniente de los usuarios del sistema.

Palabras Clave: Meta-buscador Internet, Desambiguación del sentido, Refinamiento del sentido.

1 INTRODUCCIÓN

Con el auge de Internet y el crecimiento exponencial de la cantidad de información alojada en ella el proceso de búsqueda se ha vuelto complicado y poco eficiente, desbordando con información poco relevante a los usuarios. El problema se debe a que la mayoría de los buscadores actuales basan su búsqueda en la coocurrencia de palabras clave dentro de los textos, limitándose a aspectos léxicos y en algunos casos sintácticos.

Las líneas de investigación actuales para mejorar la relevancia de los documentos devueltos por el proceso de búsqueda se basan en la incorporación de capacidades semánticas y deductivas mediante el uso de diccionarios, conceptos tales como la sinonimia y la antonimia o el uso de ontologías. Desde el punto de vista procedimental, se basan principalmente en la utilización de tesauros como WordNet [8] que dota al sistema de cualidades semánticas basadas en la hiperonimia, hiponimia o sinonimia.

Mientras que los diccionarios habitualmente están organizados por orden alfabético, WordNet se organiza en base a las relaciones semánticas entre las distintas palabras. Las principales relaciones que se contemplan en WordNet son la sinonimia y la hiponimia entre otras. Así se pueden encontrar conjuntos de palabras sinónimas agrupadas en conjuntos denominados *synsets*, pudiendo darse el caso de que una palabra polisémica pertenezca a distintos *synsets*. La relación de hiponimia (junto con la hipernimia) se establece entre distintos nombres, estableciéndose así una jerarquía entre términos basada en la hiponimia. Estas relaciones proporcionan una información valiosa que permite expandir los términos de la consulta para incorporar información semántica al proceso de búsqueda.

En este artículo se propone un modelo de meta-buscador que requiere de la ayuda del usuario para decidir las acepciones del término más cercanas a su intención de búsqueda. Con este paso previo se pretende minimizar el ruido producido por la polisemia en la búsqueda y mejorar los resultados de la misma. Para lograr este objetivo nos basamos en WordNet 2.0. Además, el sistema incorpora la capacidad de eliminar la ambigüedad producida por los posibles significados de una palabra de forma automática si existe algún tipo de relación semántica entre los términos introducidos por el usuario que permita deducir la acepción más correcta, para lo que también se utiliza WordNet. Pero la utilización de WordNet para la desambiguación de términos polisémicos presenta inconvenientes, por lo que hemos decidido implementar un sistema de desambiguación adicional, basado en la experiencia adquirida a través del usuario en la fase de "refinamiento del sentido", al seleccionar los sentidos más cercanos a su intención de búsqueda de los que ofrece WordNet.

2 ARQUITECTURA Y PROCESO DE BÚSQUEDA DE GUMSE

2.1. ARQUITECTURA DE GUMSE

GUMSe [9] es un meta-buscador formado por un conjunto de agentes con distinta funcionalidad que se comunican entre sí a través de una red. La arquitectura de GUMSe se muestra en la figura 1. Este planteamiento permite la existencia de distintos agentes ejecutándose en máquinas diferentes. De esta forma se puede aumentar la eficiencia del sistema, ya que es posible la ejecución de las distintas tareas en paralelo. Además se permite la ejecución simultanea de varios agentes con funciones similares. Esta posibilidad conlleva un aumento de la eficiencia de la búsqueda y una reducción del tiempo de ejecución de los procesos que requieran mayor carga computacional.

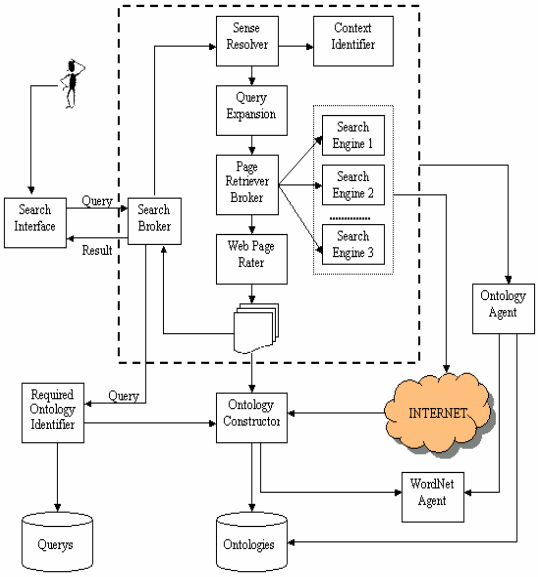


Figura 1: Arquitectura del buscador

El modelo de buscador propuesto se apoya en la utilización de los actuales motores de búsqueda para tratar de obtener un conjunto de documentos más relevantes para el usuario. Es decir, es un meta-buscador que trata de mejorar la relevancia de los resultados mediante la combinación de diferentes motores de búsqueda. Este planteamiento requiere de un posterior proceso de los distintos resultados obtenidos que implica la integración y reajuste de la relevancia a la búsqueda del usuario.

2.2. PROCESO DE BÚSQUEDA

El proceso de búsqueda seguido en este modelo se puede descomponer en una serie de fases coordinadas por al agente *Search Broker*. En cada una de estas fases se va refinando sucesivamente la carga semántica de la consulta para permitir expandir los resultados de la misma con documentos relacionados conceptualmente. Las 4 fases propuestas son las siguientes:

2.2.1. Refinamiento del sentido de la búsqueda

Inicialmente se trata de refinar la búsqueda para evitar el mayor inconveniente de las búsquedas basadas en palabras clave, que es la técnica principalmente utilizada en los buscadores actuales. Este inconveniente es la polisemía de las palabras, es decir, la influencia de los múltiples significados que pueden tomar los distintos términos que intervienen en una consulta, provocando una desviación en la relevancia de los resultados obtenidos. Para tratar de solucionar este problema se utilizará el concepto de acepción o sentido para permitir enfocar la búsqueda y mejorar los resultados. Se trata por tanto, de que el usuario seleccione la acepción que más se adecue a sus intenciones de búsqueda en caso de tratarse de un término con varios sentidos.

2.2.2. Expansión de la consulta

Posteriormente se utiliza la información adquirida de la fase anterior para aportar una mayor carga semántica al conjunto de consultas. Para ello, se construyen una serie de consultas adicionales a la original utilizando los términos que poseen alguna relación semántica (sinonimia, hiperonimia, etc.) con los términos que aparecen en la consulta del usuario.

2.2.3. Recolección de la web

El siguiente paso consiste en lanzar las distintas consultas a una serie de motores de búsqueda (actualmente Google y Altavista) para obtener una colección de páginas entre las que se encontrarán los documentos más relevantes. El proceso consiste en eliminar los duplicados y obtener una evaluación en función únicamente del contenido del documento.

2.2.4. Evaluación de los resultados

Por último se procede a recuperar las páginas obtenidas y mostrarlas al usuario ordenadas bajo un único criterio de relevancia. En el cálculo de la relevancia se utilizan los términos o sinónimos de las acepciones descartadas de forma que influyan de forma negativa en el cálculo de la relevancia, primando aquellos términos relacionados con la acepción correcta. De esta forma, se favorece a los documentos con la acepción seleccionada aumentando el grado de relevancia de la colección de documentos obtenidos.

3 LA DESAMBIGUACIÓN AUTOMÁTICA

Cuando se introducen varios términos en una consulta y alguno de ellos tiene más de una acepción, es necesario conocer la acepción a la que se refiere el usuario. De esta

forma se puede centrar la búsqueda y obtener resultados más acordes con lo que se busca. Para cumplir este propósito se utiliza la etapa de "refinamiento del sentido" en GUMSe, pero como contrapartida implica un gasto en tiempo. Para poder optimizar esta labor se ha desarrollado un sistema de desambiguación automático basado en WordNet con el objetivo de ahorrar el tiempo gastado en el refinamiento si es posible. Existen muchos trabajos que se basan en la utilización de WordNet para intentar obtener el sentido apropiado de una palabra, como por ejemplo los de Agirre [2] y Cañas [3] entre otros.

Pero existen problemas con la utilización de tesauros como fuente para la desambiguación de las palabras. El principal problema que se suele comentar es la granularidad de los sentidos de una palabra. Varios autores, como por ejemplo Slator and Wilks [10], han remarcado que las divisiones del sentido que se encuentran en los diccionarios son frecuentemente demasiado "finas" para los propósitos del procesamiento del lenguaje natural [7]. Este fenómeno introduce distintos efectos combinatorios entre los sentidos que hacen complicado una desambiguación correcta. Se puede distinguir esta problemática presente entre granularidad "fina" y "gruesa", expuesto por Agirre y Lopez de Lacalle [1], en los sentidos del término "channel" de WordNet.

Además, el problema de la granularidad se traslada también al usuario. Los repositorios de sentido "fino" como WordNet hacen difícil incluso para los humanos distinguir de forma adecuada y consistente entre los sentidos de una palabra [4]. Para tratar de paliar este problema se han creado otros repositorios que permiten la organización de la granularidad de los sentidos de forma "fina" o más "gruesa", como realiza SenseVal-2 [6]. De hecho, SenseVal-2 permite puntuar a los distintos sistemas que lo utilizan con los dos tipos de granularidad.

Este no es el único problema que se plantea en los sistemas de recuperación de información en Internet que utilizan la desambiguación del sentido de las palabras. Los tesauros clásicos existentes no suelen contemplar términos de reciente acuñación, como por ejemplo XML, ni tampoco términos muy utilizados en las búsquedas de Internet, como "mp3", "Christina Aguilera" o "kazaa lite" por ejemplo. Este fenómeno hace que la desambiguación se aplique con poca frecuencia, y en su lugar sea necesario una actualización constante de las estructuras de conocimiento utilizadas. Por ejemplo, en la consulta "snes roms" (posición 27 en las consultas más utilizadas en Noviembre 2003 según HitBrain[1]), un tesauro clásico como WordNet solo reconoce el término "rom", y en este caso lo hace con un sentido incorrecto: "read only memory", cuando debería referirse a "cartucho o juego para los emuladores".

[1] www.hitbrain.com

En el meta-buscador GUMSe, existe una etapa de refinamiento del sentido que permite detectar este tipo de situaciones. Esta etapa, además de mejorar la búsqueda incorporando noción del sentido en los términos polisemicos, permite mejorar las estructuras de conocimiento utilizadas de forma progresiva. De este modo, en GUMSe se ha implementado un sistema de desambiguación automática, basada en la experiencia acumulada en la selección de los sentidos correctos de WordNet entre pares de términos.

4 VENTAJAS DE LA FASE DE REFINAMIENTO DEL SENTIDO

4.1. DETECTAR NUEVOS SENTIDOS O ACEPCIONES

Permite, con la colaboración del usuario, detectar nuevos sentidos o acepciones, así como reconocer acrónimos. Se puede indicar al sistema que el término introducido, a pesar de ser reconocido por WordNet, no se refiere a ninguna de las acepciones del mismo. Por ejemplo cuando se utiliza la consulta "snes rom", el término rom no se refiere a la memoria, sino a un cartucho. Otra posibilidad es permitir introducir una frase que describa al término. Por ejemplo, para el término "salt", según WordNet, uno de sus sentidos es el acrónimo de "Strategic Arms Limitation Talks", pero no es el único existente, sino que SALT también puede ser "Society for Applied Learning Technology", acrónimo no reconocido por WordNet.

4.2. RECONOCER LOS SENTIDOS MÁS UTILIZADOS

WordNet, por defecto, muestra en las primeras posiciones aquellas acepciones más frecuentemente utilizadas. Esta posición está determinada fruto del procesamiento del corpus utilizado para la construcción de WordNet. Pero no tiene porque adaptase a la frecuencia de utilización en la realidad, como sucede con el término "java". La utilización de este término en la búsquedas de Internet se refieren en su mayoría a un lenguaje de programación, sin embargo, esta acepción aparece la última (la tercera) en WordNet.

4.3. REDUCCIÓN DE LA GRANULARIDAD

Reconocer sentidos que se pueden agrupar en uno solo. Debido a la "fina" granularidad de los sentidos de WordNet, es posible que distintos usuarios seleccionen distintos significados cuando aparecen en una consulta junto al mismo término. Está tarea se realiza en función de los acuerdos y desacuerdos de los anotadores humanos. Esta idea no es nueva, sino que se ha aplicado anteriormente para tratar de obtener un conjunto más

"grueso" de significados de un término. Por ejemplo, Chklovski [4] utiliza este desacuerdo entre significados para construir una matriz de confusión que se utiliza para calcular la similaridad de significados, a partir de la cual se agrupan los distintos sentidos en clusters. Este método permite crear particiones de los sentidos y obtener, por tanto, conjuntos de significados más "gruesos".

Por ejemplo, en la consulta "java example" se puede producir esta situación. Cuando esta consulta la realizan dos usuarios distintos, y ambos se refieren al término "java" como lenguaje de programación (seleccionando la acepción correcta en la etapa de refinamiento del sentido) puede producirse un desacuerdo en el término "example". Puede suceder que para uno el término "example" se refiera a "algo a ser imitado", o que se refiera a "una tarea realizada o problema resuelto para desarrollar habilidad o entendimiento".

También en nuestro caso es posible utilizar el desacuerdo de los usuarios del buscador para reducir la granularidad. Una reducción de la granularidad implica un conjunto más reducido de sentidos para cada término de la consulta, lo que facilita su manejo, disminuye el volumen de datos que manejar y aumenta la eficiencia. Además, aquellos sentidos agrupados en un mismo cluster pueden ser manejados como un único sentido. Esta situación permite mostrar al usuario sólo uno de los sentidos pertenecientes a un mismo cluster, normalmente el más seleccionado por los usuarios. Las ventajas de este planteamiento son claras, ya que se reduce el número de entradas de cada término que tiene que revisar el usuario en la etapa de refinamiento del sentido, lo que reduce el tiempo de elección y por tanto el tiempo de ejecución de la consulta. Además, se palia en cierto grado el efecto no deseado que la granularidad "fina" tiene sobre el usuario, que provoca dudas en la selección del sentido correcto.

4.4. DESAMBIGUACIÓN AUTOMÁTICA

Por último, la etapa de *refinamiento del sentido* permite la desambiguación automática basada en la acumulación de la experiencia de los usuarios. Cuando se utiliza más de un término en una consulta, se puede establecer una relación entre los términos, pero además, al seleccionar los sentidos correctos en caso de que se utilicen términos polisémicos, se puede establecer una relación entre acepciones.

La etapa de *refinamiento del sentido* permite al sistema aprender las distintas relaciones existentes entre los términos de una consulta. Con la ayuda del usuario, y asumiendo que no existe malintención en su elección, se puede conocer la relación entre un sentido de uno de los términos con otro sentido de otro término. Esta información se puede utilizar para tratar de obtener la acepción correcta en las situaciones en que no se puede obtener por otros medios.

5 ALGORITMO DE DESAMBIGUACIÓN AUTOMÁTICA

Se define el conjunto Q formado por los T términos presentes en una consulta, siendo T mayor que 1 y menor o igual que 10.

$$Q = \{t_1, t_2, \ldots, t_T\} \qquad 1 < T \leq 10$$

El conjunto S_i está formado por las posibles acepciones de un término del conjunto Q (de la consulta).

$$S_i = \{S_{i1}, S_{i2}, \ldots, S_{iA}\},\ A \geq 1,\ 1 \leq i \leq T$$

Cuando el usuario selecciona el sentido correcto de cada uno de los términos que participan en la consulta, se establece una asociación entre pares, cada uno de ellos formado por un término y uno de los sentidos de ese término. Llamaremos *par término-acepción* al par (t_i, S_m), donde t_i es el término i-ésimo de la consulta Q y S_m es el sentido m-ésimo seleccionado por el usuario para el término t_i. Hay que tener en cuenta, que no se utilizarán en este proceso aquellas consultas que no han pasado por la etapa de refinamiento del sentido, es decir, que han sido desambiguadas de forma automática sin la intervención del usuario, ni tampoco las que tengan un único término.

Por tanto, se puede obtener una relación como una tripleta formada por dos pares término-acepción y un valor que determina en qué grado están relacionados. Esta relación la denominaremos *relación términos-acepción*, y se define de la forma:

$$\{(t_i, S_m), (t_j, S_n), V\} \quad , i \neq j,\ 1 \leq i,j \leq T$$

V es el grado borroso asignado a la relación de términos, y toma un valor comprendido entre 0 y 1. El grado de la relación entre dos pares término-acepción, que no entre términos únicamente, será mayor en función de la frecuencia con la que aparezcan en las consultas. Cuanto más frecuentemente aparezcan juntos en una consulta, es decir, más usuarios la utilizan, más fiable será la existencia de la relación. Por tanto, se define el grado de la relación términos-acepción formada por los términos A y B de una consulta y las acepciones correspondientes S_a y S_b mediante la unión borrosa de Dombi [5]:

$$V = \frac{1}{1 + \left[\left(\frac{1}{V_{AB}} - 1 \right)^{-\lambda} + \left(\frac{1}{V_{BA}} - 1 \right)^{-\lambda} \right]^{-1/\lambda}}$$

Donde V_{AB} mide la relación de los términos A y B en las consultas con la acepción S_a del término A, y V_{BA} para la acepción S_b del término B. El valor λ es parametrizable y actualmente se toma $\lambda = 2$. Los valores de V_{AB} y V_{BA} se calculan mediante la fórmula:

$$V_{AB} = \frac{\sum_{i=1}^{Q_A} (1 - \sigma \log_{10}(T_i - 1)) * f_{Bi}}{\max(V_{AX})}$$

Donde Q_A es el número de consultas en las que aparece el término A con la acepción S_a. f_{Bi} es 1 si aparece en la consulta i el término B con la acepción S_b y 0 si no aparece. T_i es el número de términos de la consulta i. $\max(V_{ax})$ es el mayor valor de una relación del término A con acepción S_a con cualquier otro término y acepción, y se utiliza para normalizar la medida.

El parámetro σ se utiliza para ponderar en que grado se tienen en cuenta las consultas de más de 2 términos. Este factor toma un valor experimental (actualmente 0,5) y se utiliza para reducir el impacto en el cálculo de la medida de las consultas en las que aparezcan más de 2 términos. Esto se debe a que puede suceder que no exista una relación directa entre cada uno de los términos, sino que dos de ellos estén mucho más relacionados con un tercer término de la consulta. Por ejemplo, si se busca una casa con jardín y garaje se puede utilizar la consulta "casa jardín garaje", los términos "jardín" y "garaje" no tienen relación por sí mismos, a no ser que se consideren como posibles partes de una "casa".

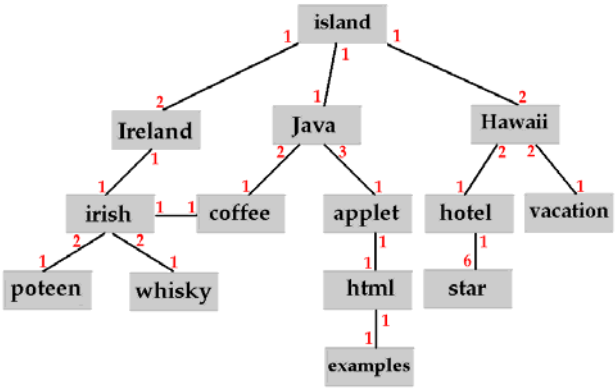


Figura 2: Posible grafo términos-acepción

A partir de la relación términos-acepción es posible construir un grafo no dirigido denominado grafo términos-acepción. Este grafo es una de las estructuras semánticas construidas por el Agente Constructor de Ontologías, y se actualiza periódicamente con las nuevas consultas realizadas por los usuarios. Los nodos del grafo son cada uno de los términos que se han utilizado en las consultas y que han pasado por la etapa de refinamiento del sentido. Los arcos unen dos términos y están etiquetados por el valor V que indica el coste de esa rama. Cada rama que entra o sale de un vértice tiene asociado una acepción perteneciente al término de ese nodo. De esta forma se consigue que este grafo relacione los términos a través de las acepciones. Por ejemplo, como puede verse en la figura 2, el término "Java" se relaciona con el término "island" a través de la acepción 1 de "java" (isla de Indonesia) y de la acepción 1 de "island", mientras que se relaciona con el término "applet" por medio de la acepción 3 de "java" (lenguaje de programación) y la única acepción de "applet".

Cuando se introduce un término polisémico es posible, basándose en este grafo, tratar de encontrar la acepción a la que el usuario se refiere. El mecanismo consiste en buscar un camino simple que relacione los dos nodos correspondientes a dos términos de la consulta. La longitud del camino está limitada para reducir el tiempo de búsqueda, actualmente se restringe a caminos menores o iguales a 5.

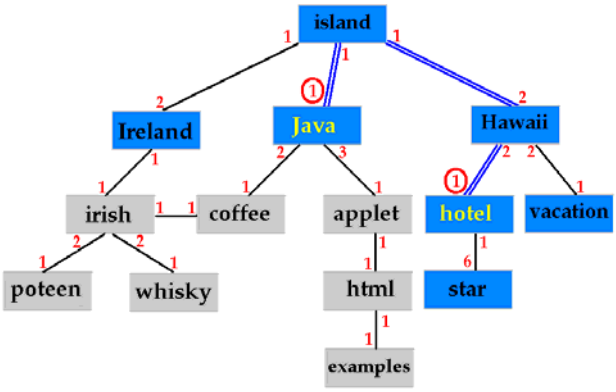


Figura 3: Nodos del grafo términos-acepción implicados en la consulta "java hotel", tomando como origen "hotel"

El camino se halla buscando por cada rama del nodo origen, que será uno de los términos de la consulta, el de menor grado o con menor número de acepciones. Para los nodos visitados, solamente se visitarán los nodos adyacentes del actual a través de una rama con acepción igual a la que se llegó al nodo actual. La búsqueda prosigue mientras no se supere la restricción de longitud y no se ha visitado el nodo destino. Si se encuentra el camino, es decir, se llega al nodo destino, entonces se pueden determinar las acepciones de los dos términos en caso de tener más de una. Para el nodo origen será la acepción de la rama por la que comienza el camino. Para el nodo destino la acepción es la de la rama por la que se llegó al construir el camino.

Por ejemplo, si un usuario busca un hotel situado en la isla de Java puede utilizar la consulta "Java hotel". El término "java" es polisémico, ya que puede tener 3 posibles acepciones (una isla, un café o un lenguaje de programación). Por tanto, se puede utilizar este algoritmo para obtener el sentido apropiado de "Java". Se parte del término "hotel" como nodo origen, ya que tiene una única acepción, y el nodo destino será el término "java". En la figura 3 se puede ver los posibles nodos visitados para la consulta "java hotel" partiendo del nodo "hotel". Una vez

encontrado un camino simple se pueden determinar la acepciones correspondientes, que son la 1 para "hotel", ya que es la única que tiene, y la 1 para "Java" (veáse figura 3).

Una vez encontrado el camino hay que calcular el grado de relación entre los términos, para ello se utilizan el grado de relación V. Se calcula el grado de relación y se determina si es aceptable si supera un cierto umbral. El calculo del grado de la relación del camino C se calcula aplicando una función de agregación como es la media geométrica:

$$V(C) = \sqrt[L(C)]{\prod_{i=1}^{L(C)} V_i}$$

Donde V_i es el valor de la relación de cada rama que forma el camino y L(C) es la longitud del camino. Actualmente se considera válida la relación si V(C) es mayor de 0,5.

6 CONCLUSIONES Y TRABAJOS FUTUROS

En este artículo se ha expuesto un nuevo mecanismo para la desambiguación automática del sentido de palabras polisémicas, basado en la información adquirida en la fase de refinamiento del sentido. Esta fase aporta nuevas posibilidades para mejorar las estructuras de conocimiento utilizadas para aumentar la precisión en las búsquedas que estamos comenzando a explotar.

Todavía queda mucho trabajo por hacer y problemas que resolver en la aplicación efectiva de este método. Una posible mejora es la poda de las ramas con poca relevancia para obtener un grafo más reducido que mejore la eficiencia. También existen problemas debido a la existencia de dos caminos distintos que conducen a distintas acepciones, como por ejemplo cuando se busca una implementación de un "hotel" en el lenguaje java, se podría utilizar también la consulta "java hotel" expuesta como ejemplo anteriormente. Actualmente, se utiliza el valor V(C) para determinar el mejor camino.

También existe el problema de que este método no es aplicable cuando solo aparece un término en la consulta. En este caso, se necesitará de la ayuda del usuario, en la fase de refinamiento del sentido, para determinar el sentido correcto, o bien asumir uno por defecto (el primero de la lista).

Agradecimientos

Financiado parcialmente por el proyecto PBC-03-004 PREDACOM, JCCM, el proyecto TIC2003-08807-C02-02 DIMOCLUST, MCYT, y la *Cons. de Ciencia y Tecnología* – Fondo Social Europeo de la JCCM.

Referencias

[1] Agirre, E.; Lopez de Lacalle, O. (2003): Clustering WordNet Word Senses. *In Proceedings of the Conference on Recent Advances on Natural Language Processing (RANLP '03).* Borovets, Bulgary.

[2] Agirre, E.; Rigau, G. (1996): Word Sense Disambiguation using Conceptual Density. *In Proc. of 15th International Conference on Computational Linguistics, COLING 96.* Copenhagen, Denmark.

[3] Cañas, A. J.; Valerio, A.; Lalinde-Pulido, J.; Carvalho, M.; Arguedas, M. (2003): Using WordNet for Word Sense Disambiguation to Support Concept Map Construction. *In Proc. of SPIRE 2003 – 10th International Symposium on String Processing and Information Retrieval.* Manaus, Brazil.

[4] Chklovski, T.; Mihalcea, R. (2003): Exploiting Agreement and disagreement of Human Annotators for Word Sense Disambiguation. *In Proc. of the Conference on Recent Advances in Natural Language Processing, RANLP 2003.* Borovetz, Bulgaria.

[5] Dombi, J. (1982): A general class of fuzzy operators, the De Morgan class of fuzzy operators and fuzziness measures induced by fuzzy operators. *Fuzzy Sets and Systems*, No. 8, pp. 149-163.

[6] Edmonds, P. (2002): SENSEVAL: The evaluation of word sense disambiguation Systems. *Published in the ELRA Newsletter*, Vol. 7, No. 3.

[7] Ide, N.; Véronis, J. (1998): Word Sense Disambiguation: The State of the Art. *Computational Linguistics*, Vol. 24, No. 1.

[8] Miller, G. (eds) (1990). WORDNET: An Online Lexical Database. *International Journal of Lexicography*, Vol. 3, No. 4, pp. 235 – 244.

[9] Olivas, J. A.; de la Mata, J.; Serrano-Guerrero, J. (2004): Ontology Constructor Agent for improving Web Search with GUMSe. *In Proc. of Information Processing and Management of Uncertainty in Knowledge –Based Systems, IPMU'04*. Perugia, Italia.

[10] Slator, B. M.; Wilks, Y. A. (1987): Towards semantic structures from dictionary entries. *In Proc. of the 2nd Annual Rocky Mountain Conference on Artificial Intelligence.* Boulder, Colorado, pp. 85-96.

MODELOS BORROSOS PARA EL FILTRADO DE CONTENIDOS WEB BASADO EN USABILIDAD

Elena García, Miguel A. Sicilia, Tomasa Calvo, Jesús Lázaro
Dept. CC. Computación
Universidad de Alcalá
Ctra. Barcelona km. 33,6
28871, Alcalá de Henares (Madrid)
{elena.garciab, msicilia, tomasa.calvo,jesus.lazaro}@uah.es

David Palomar
Dept. Informática
Universidad Carlos III
Avd. Universidad 30
28911,Leganés (Madrid)
dpalomar@inf.uc3m.es

Resumen

El filtrado de contenidos permite seleccionar aquellos recursos que son más apropiados para el usuario, de acuerdo a descripciones de sus preferencias, del contexto en que se encuentra, y de los mismos contenidos. Todas esas descripciones están sujetas a diversos tipos de imprecisión o incertidumbre, y algunos trabajos existentes las han tratado desde diferentes perspectivas. Este trabajo se centra en los criterios de filtrado que tienen en cuenta la *forma* de los recursos — es decir, su diseño, en el caso de la Web — y por tanto, sus características de usabilidad. Concretamente, se describe un modelo borroso para el filtrado basado en características de usabilidad que integra aspectos tratados en trabajos anteriores.

Palabras Clave: Filtrado de información, usabilidad, conjuntos borrosos.

1 INTRODUCCIÓN

El filtrado de información en la Web u otros sistemas hipermediales tiene como objetivo seleccionar los elementos de una base de contenidos o *ítems* que mejor satisfagan la *necesidad de información* dada de un usuario, ya sea éste una persona o un agente software.

Las técnicas más utilizadas actualmente para llevar a cabo este tipo de filtrado son las de *recuperación de información* —bien tengan teoría de conjuntos borrosos incorporada [5, 8] o no—, que básicamente realizan la búsqueda de recursos relevantes mediante el uso de índices y la correspondencia entre los términos expresados por el usuario y los descriptores del contenido de los recursos Web [2]. No obstante, el paradigma de la "*Web Semántica*"[6] habilita otras formas de recuperación (o estrategias híbridas), en este caso basadas en representaciones del conocimiento conocidas como ontologías [7]. Estas interfaces basadas en *ontologías* (como la descrita en [14]) requieren de bases de contenidos con *anotaciones* conectadas a las ontologías del dominio relevantes.

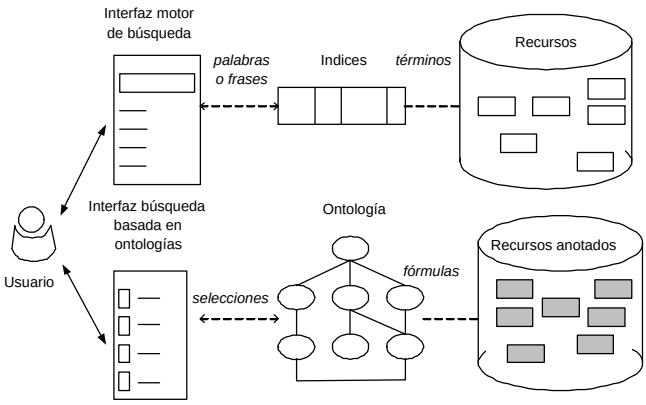


Figura 1: Tipos de filtrado basado en contenidos (FBC).

La Figura 1 sintetiza las diferencias fundamentales entre ambas técnicas de filtrado o recuperación de ítems. En la recuperación de información, esencialmente se usan palabras o frases escritas por el usuario con un índice de documentos, utilizando términos de lenguajes naturales como resumen de los recursos. Sin embargo, cuando se usan ontologías, se tiene algún tipo de interacción entre el usuario y las estructuras conceptuales de la ontología formal seleccionada, y estas se usan para derivar fórmulas en un lenguaje lógico [1] que se usan para recuperar los recursos de acuerdo a sus anotaciones en el mismo lenguaje basado en lógica.

Para llevar a cabo tal selección de contenidos se pueden

tener en cuenta las características tanto de los contenidos, como del dispositivo en el que se visualizarán y del usuario que los solicita. Esto da lugar a diferentes tipos de descripciones, que se sintetizan en la Figura 2.

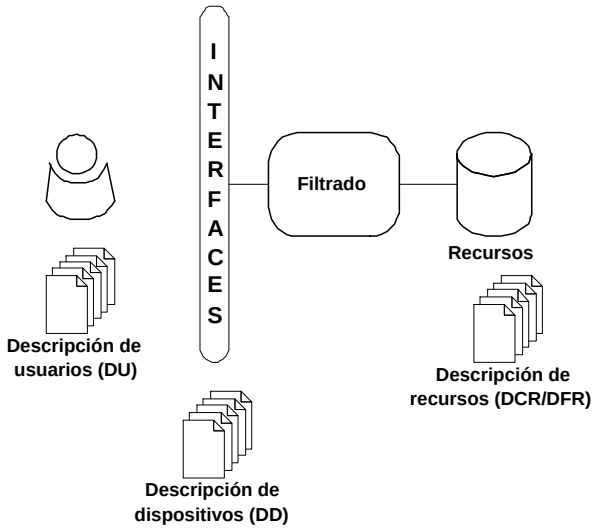


Figura 2: Descripciones utilizadas en el filtrado basado en contenidos.

Las descripciones de los usuarios (DU) o modelos de usuario representan las preferencias, objetivos o características de usuarios o grupos. Las descripciones de los dispositivos (DD) representan las capacidades de interacción del hardware y el software básico de las interfaces. Por último, las descripciones de los recursos nos proporcionan información bien sobre los contenidos de los mismos (DCR), o bien sobre su forma o diseño (DFR).

Las descripciones relacionadas con la *forma* (es decir, con el diseño) de los contenidos, y por tanto, con la usabilidad de los mismos, abren la posibilidad de comportamientos de adaptación como los siguientes:

- Establecer criterios de usabilidad especificados por el usuario en cuanto a sus preferencias o necesidades personales, como por ejemplo el grado de accesibilidad de los contenidos (en el sentido de [17]).

- Utilizar los indicadores proporcionados por analizadores de usabilidad [3] para dar mayor prioridad a los recursos que se consideren más "usables".

- Combinar criterios de usabilidad con descripciones borrosas de dispositivos, que permitan una mayor flexibilidad que las actuales descripciones de capacidades (como por ejemplo, la de FIPA[1]).

La evaluación de la usabilidad está inherentemente sujeta a incertidumbre y vaguedad. Siguiendo las definiciones de Smets [18], la incertidumbre proviene en este caso de la subjetividad de los juicios del experto o del usuario, y la vaguedad está implícita en la naturaleza aproximada de las guías para el diseño y evaluación de sistemas usables.

En este trabajo se describe qué información se debe recoger de cada uno de los elementos que intervienen en un filtrado de contenidos basado en usabilidad de manera que se habilite un tratamiento de la información imperfecta que en ellos se detecta, así como las técnicas utilizadas sobre cada uno de estos elementos para realizar el filtrado bajo estas premisas. Además se propone un marco integrador de todas estas técnicas con objeto de realizar el filtrado de contenidos basado en usabilidad de la manera más ajustada a las necesidades del usuario posible.

El resto de este artículo tiene la siguiente estructura. En la Sección 2, se describe trabajo previo sobre algunas técnicas borrosas que se han utilizado dentro del marco de filtrado de contenidos descrito. Después, en la Sección 3 se presenta un modelo que integra varias de estas técnicas en un marco común para el filtrado basado en la usabilidad. La cuarta sección concluye el artículo, indicando algunas líneas de investigación abiertas al respecto del estudio de la usabilidad como elemento de filtrado de contenidos.

2 MODELOS BORROSOS PARA EL FILTRADO DE INFORMACIÓN

La Tabla 1 resume algunos trabajos previos de los autores sobre técnicas borrosas que pueden utilizarse en el filtrado de recursos, indicando las descripciones que se ven afectadas por la borrosidad.

Como se describe de la información referenciada en la Tabla 1, las técnicas borrosas que pueden utilizarse son muy diversas, y afectan a diferentes aspectos de las descripciones que pueden utilizarse en el filtrado. Algunas de ellas trabajan específicamente con la forma de los contenidos:

- Los aspectos perceptuales de la forma de las páginas Web pueden elicitarse con técnicas experimentales [16], y después integrarse en reglas bor-

[1] http://www.fipa.org/specs/fipa00091/XC00091C.pdf

Técnica	Descripción afectada	Referencias
Directrices (guidelines)	DFR, DD	[16]
Preferencias estructurales	DFR, DU	[9]
Agregación de aspectos de usabilidad	DFR	[11]
Reglas de adaptación cuantificadas	DFR, DCR	[10][9]
Modelos borrosos de grupos	DU	[13]
Segmentación borrosa	DU	[12]

Tabla 1: Algunas técnicas borrosas útiles en el filtrado.

rosas como implementación de directrices (guidelines) de usabilidad, que permiten obtener indicadores analíticos de las páginas. Un ejemplo sería el de una directriz como "si la página tiene *demasiados* enlaces, la usabilidad es baja".

- Pueden utilizarse técnicas de recogida de información de los usuarios para elaborar sus preferencias sobre elementos denominados estructurales de las páginas [9]. Ejemplos de estos elementos son la longitud de áreas de enlaces o listas, la granularidad de los contenidos, o su densidad semántica.

- Dado que la usabilidad de una Web suele analizarse en función de ciertos sub–aspectos cuyas relaciones no están claramente determinada, el uso de esquemas de agregación flexible permite aproximar la obtención de estimaciones de la usabilidad global, teniendo en cuenta sub–aspectos como la eficiencia, facilidad de aprendizaje o satisfacción de los usuarios [11].

Además, es posible también utilizar indicadores estadísticos como los obtenidos por Ivory [4] como aproximadores borrosos para implementar análisis de usabilidad de manera menos costosa que la empleada en el procedimiento de elicitación de características perceptuales. Una aproximación para este tipo de análisis se describe en [15].

3 USO DE CONJUNTOS BORROSOS EN LOS DIFERENTES ASPECTOS DEL FILTRADO

En esta sección esbozamos un modelo de filtrado basado en la usabilidad que integra trabajos anteriores como los descritos en la sección previa. La Figura 3 esquematiza ese tipo de filtrado, combinado con un filtrado basado en los contenidos.

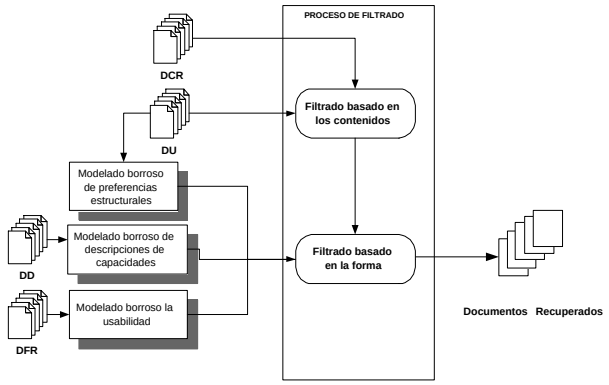


Figura 3: Esquema general de un filtrado combinado de recursos.

Las preferencias estructurales de los usuarios pueden modelarse como relaciones anotadas con preferencias lingüísticas entre los usuarios en el conjunto $\mathcal{U}$ (de igual modo se podrían aplicar a grupos, aunque no lo tratamos aquí) y los diferentes atributos descriptores de las páginas Web $a_i \in \mathcal{A}$.

$$R_{estr} : \mathcal{U} \times \mathcal{A} \rightarrow \mathcal{LS} \quad (1)$$

Donde LS — o mejor LS_{a_i} — denota un conjunto de etiquetas lingüísticas asociadas al atributo considerado en cada caso. Por ejemplo, $(u_i, granularidad) \rightarrow media$ sería un ejemplo de preferencia concreta. Este tipo de preferencias son siempre específicas del contexto, ya que pueden variar para un mismo usuario dependiendo de la aplicación, por lo cual, realmente estaríamos hablando de una familia de relaciones R^{ctx}_{estr} donde el contexto ctx determinaría el subconjunto de $\mathcal{A}$ aplicable. Por ejemplo, el atributo $longitud \in \mathcal{A}$ estaría solamente asociada a elementos tipo lista, y sería diferente dependiendo de la ubicación de cada lista en la página.

Por otro lado, el dispositivo por sí mismo restringe la selección de los elementos en el filtrado. Así, una descripción de las características del dispositivo $\mathcal{C}$ se

comprobaría con las características del ítem en consideración. En ocasiones, algunas de esas características también se expresarían de manera borrosa, o al menos, su compatibilidad estaría sujeta a algún tipo de flexibilidad.

Dadas las características anteriores, se pueden construir filtrados basados en la usabilidad de diferentes tipos, dependiendo del uso y composición de cada uno de los modelos mostrados en la Figura 3. A continuación describimos una posibilidad y el razonamiento de su aplicación.

La siguiente expresión resume un posible modelo de filtrado de contenidos basado en usabilidad, que comienza con un subconjunto de recursos o ítems O, posiblemente proviniente de un filtrado previo basado en contenidos.

$$O_1 = dm(O, \mathcal{C}) \quad (2)$$

$$O_2 = up(O_1, R^{ctx}_{estr}) \quad (3)$$

$$O_3 = su(O_2, \mathcal{K}) \quad (4)$$

El proceso comienza con un filtrado *dm* basado en las características del dispositivo, debido a que estos determinan las posibilidades físicas de visualización, que son precondición a las preferencias.

A continuación se tendrían en cuenta las preferencias estructurales determinadas por el contexto. En este punto se pueden utilizar, por ejemplo, reglas borrosas para determinar la relevancia de la forma de cada una de las páginas para el usuario concreto

Por último, de entre los elementos resultantes se seleccionarán aquellos que, con una valoración similar en los pasos precedentes tienen una mayor usabilidad. Para tomar esta decisión, se utilizará un artefacto de análisis de la usabilidad $\mathcal{K}$ que puede utilizar aproximaciones como las descritas en [11][16] [15]. Nótese que la consideración de la usabilidad sólo se haría para aquellos ítems que ya hayan pasado una cierta conformidad (vaga) con el dispositivo, y que han sido también reordenados de acuerdo a las preferencias del usuario. Este criterio da preferencia al contenido y a la personalización antes que a los criterios generales de usabilidad, evitando así que la imperfección de estos últimos pueda influir en el filtrado. En todos los pasos, la selección implica una ordenación por relevancia (y posiblemente un descarte de elementos de acuerdo a cierto umbral), de modo que cada O_i es una colección de pares (o, r), donde $r \in [0, 1]$ representa la valoración del item concreto para el paso determinado.

En la aproximación descrita, el filtro *dm* correspondería a una extensión flexible de la adaptación automática al dispositivo de las tecnologías existentes como `Mobile.ASP.NET` de Microsoft. Por ejemplo, los tamaños del área de visualización, habitualmente expresados en pixels pueden convertirse a categorías borrosas teniendo en cuenta el estado actual de la tecnología. El resultado de este paso sería que ciertos recursos podrían pasar a la siguiente fase sin tener una compatibilidad completa con el dispositivo para aquellas características c_i que lo permitiesen (como el tamaño de la pantalla). Esos recursos con una compatibilidad "no perfecta" podrían en pasos posteriores ser considerados como mejores debido a las preferencias del usuario, o a la usabilidad global de los contenidos.

4 CONCLUSIONES

El filtrado de recursos en la Web es un problema que puede abordarse mediante el uso de diferentes tipos de descripciones, incluyendo descripciones de los usuarios, de los contenidos, y de los dispositivos de interacción. La imprecisión en estas descripciones ha sido tratada en trabajos precedentes desde diferentes perspectivas. Entre estas descripciones, un subconjunto de ellas afectan a la forma de las interfaces de usuario, y por tanto, a la usabilidad de las mismas. Se ha descrito un modelo integrador de filtrado basado en usabilidad que permite complementar técnicas basadas en el contenido con la consideración de la calidad de las interfaces.

Referencias

[1] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): *The Description Logic Handbook*. Theory, Implementation and Applications. Cambridge (2003)

[2] Baeza-Yates, R., Riviero-Neto, B. *Modern Information Retrieval*. Addison-Wesley, 1999.

[3] Ivory, M. and Hearst, M. The State of the Art in Automated Usability Evaluation of User Interfaces. *ACM Computing Surveys*, 33(4), 1–47 (2001).

[4] Ivory, M. An Empirical Approach to Automated Web Site Evaluation. *Journal of Digital Information Management*, 1(2), 75–102 (2003).

[5] Bordogna G., and Pasi G. The application of fuzzy set theory to model information retrieval. En F. Crestani and G. Pasi (eds.)*Soft computing in Information Retrieval: Techniques and Applications*, Phvsica-Verlag, 207–241 (2000).

[6] Berners-Lee, T., Hendler, J. Scientific publishing on the "Semantic Web". En: *Nature*, Abril (2001).

[7] Gruber, T.R. Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5), 907–928 (1995).

[8] Olivas, J. A.,Garcés, P., Romero, F. P. FISS: Application of Fuzzy Technologies to an Internet Meta-Searcher. En *Proceedings of the 2002 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, 140–145 (2002).

[9] Sicilia, M. A., García, E., Martínez, J. J. Personalización estructural basada en criterios de usabilidad. *Revista Iberoamericana de Inteligencia Artificial* 16(4), 92–99 (2002)

[10] Sicilia, M. A., Díaz, P., Aedo, I., García, E.: Fuzzy Linguistic Summaries in Adaptive Hipermedia Systems. En *Adaptive Hipermedia and Adaptive Web Systems*. Lecture Notes in Computer Science, Vol. 2347. Springer-Verlag, 317–327 (2002).

[11] Sicilia, M.A., García, E, Calvo, T. An Inquiry-based method for Choquet Integral-based Aggregation of Usability Attributes. *Kybernetika*, 39(5) 601–614 (2003).

[12] Sicilia, M.A., García, E. On Fuzziness in Relationship Value Segmentation: Applications to Personalized e-Commerce. *ACM SIGECOM Newsletter* 4(2), 1–10 (2003).

[13] M.A. Sicilia, E. García. Fuzzy Group Models for Adaptation in Cooperative Information Retrieval Contexts. En *Proceedings of SOFSEM 2004*, 324–334 (2004).

[14] Sicilia, M.A., García, E., Aedo, I., Díaz, P. A literature-based approach to annotation and browsing of Web resources.*Information Research* 8(2) (2003).

[15] García, E., Sicilia, M.A., Calvo, T. Evaluating Web Document Quality with Linguistic Variables: Combining Informative and Page Design Quality. En *Proccedings of IPMU 2004* (to appear).

[16] García, E., Sicilia, M.A., Gutiérrez, J.A.: On the Vague Modelling of Web Page Characteristics Regarding Usability. En: *First International Atlantic Web Intelligence Conference*. Lecture Notes in Computer Science, Vol. 2663. Springer-Verlag, 199–207 (2003).

[17] Chisholm, W., Vanderheiden, G. *Web Content Accessibility Guidelines 1.0*. W3C Recommendation 5-May-1999.

[18] Smets, P. Imperfect information: Imprecision-uncertainty. En *Uncertainty Management in Information Systems: From Needs to Solutions*, Kluwer Academic Publishers, 225–254 (1997).

[19] Sicilia, M.A., García, E. Information imperfection as an inherent characteristic of adaptive hypermedia: Imprecise models of users and interactions. En *Adaptable and Adaptive Hypermedia Systems*, Idea Group Publishing Inc. (2004) (to appear).

RELEVANCIA DE LA RECUPERACIÓN DE INFORMACIÓN EN INTERNET PARA EL *SOFT COMPUTING*

Alejandro Sobrino
Dpto. de Lógica y Filosofía Moral. USC
lflgalex@usc.es

Resumen

Los metabuscadores son sistemas informáticos que usan a buscadores especializados para ofrecer resultados sobre aquellas búsquedas que le piden sus usuarios. El punto clave en este proceso es la integración de los órdenes locales de los buscadores en un único orden global, que ofrece el metabuscador. En el caso de búsquedas precisas, hay varias propuestas para integrar órdenes parciales. Pero no se ha estudiado el problema cuando se trata de búsquedas aproximadas. Este trabajo quiere ofrecer un acercamiento a este interesante tema y proponer a los metabuscadores fuzzy como instrumentos para evaluar la bondad de la combinación de diferentes modelos de búsqueda fuzzy usados en el *matching* impreciso propio de las búsquedas conceptuales.
Palabras clave: metabuscadores, búsquedas fuzzy, integración de lógicas

1. INTRODUCCIÓN

El título de esta comunicación es una inversión del título de la sesión en la que se inscribe. El título de la sesión alude a la utilidad del *soft computing* en la recuperación de información. El título que propongo quiere resaltar la utilidad de la recuperación de información para el *soft computing* -más concretamente, para la lógica fuzzy-, en un sentido que, al no ser habitual, debe ser especificado.

Una de las características más señaladas de la lógica fuzzy es su carácter plural. Se suele decir, y con razón, que más que de una lógica, se trata de una familia de lógicas. Ello se debe a que aquello que intentan modelar; a saber, el significado impreciso o vago de ciertos predicados del lenguaje natural, no presenta una faz única y uniforme, sino un rostro variado y desigual. En efecto, cada lógica fuzzy pretende recoger, en su propuesta, ciertas particularidades semánticas de las oraciones que pretende modelar. Así, p. ej., para oraciones compuestas coordinadas, se suele verificar si su contenido es interdependiente o no para justificar el uso de alguna de las tres t-normas más habituales; a saber, *min*, *prod* y *W* (Trillas et al., 1995)

En la vida ordinaria hay problemas complejos que pueden ser abordados de manera modular, proponiendo formas específicas de representación y gestión de conocimiento para aspectos particulares del mismo. A veces, la solución a esos problemas se reclamará global, no una simple sucesión de respuestas parciales. En ese caso, hay que combinar las racionalidades propias de cada módulo a fin de ofrecer una solución integrada y única. Si la gestión de problemas complejos incluye el manejo de conocimiento impreciso, esa solución integrada supondrá combinar distintas lógicas fuzzy.

La combinación de lógicas fuzzy se puede abordar, al menos, desde dos perspectivas diferentes:
• formal, proponiendo una especie de lógica integradora de segundo nivel que agrupe a las lógicas usadas (Gabbay, 1999).
• pragmática, combinando lógicas en la solución a problemas concretos, y donde el éxito o fracaso de la solución proveída será un criterio para evaluar la bondad de la combinación.
En este trabajo nos centraremos, fundamental-mente, en esta última aproximación.

Un ámbito pragmático donde se platea la combinación de racionalidades es en el de los metabuscadores. La secuencia puede ser la siguiente: un usuario introduce una pregunta para obtener información. El metabuscador la trasladada, a su vez, a aquellos buscadores que estima en mejores condiciones de dar respuestas adecuadas. Cada buscador devuelve un orden local y el metabuscador ha de integrarlos en un orden final único que será la respuesta definitiva a la pregunta. Al hacer eso, se puede decir que el metabuscador combina racionalidades y la evaluación de los algoritmos de combinación puede cuantificarse en términos del grado de satisfacción de los usuarios con el resultado que el metabuscador le ofrece.

En este trabajo se pretende estudiar la combinación racional de lógicas fuzzy utilizando para ello metabuscadores que eliciten a buscadores que implementen búsqueda conceptual aproximada (Olivas et al., 2003) y definiciones booleanas fuzzy. El artículo se estructura de la siguiente manera: en el punto siguiente se presentan brevemente las características principales de la

arquitectura de un metabuscador; a continuación, se exponen algunos métodos usuales de combinación de similitudes locales (órdenes locales) para obtener una similitud global (orden global final); en el siguiente apartado se describe un pequeño experimento de combinación de lógicas para el caso de los buscadores fuzzy. El trabajo finalizará relatando las conclusiones y algunas prospecciones de futuro sobre este apasionante tema.

2. METABUSCADORES

De manera simple, puede decirse que un metabuscador es un ingenio informático que hace búsquedas usando como intermediarios a buscadores específicos. Su utilidad se justifica en el crecimiento enorme de la Web y en la necesidad de elicitar y coordinar buscadores especializados, que den cuenta de manera eficiente de una parte del inmenso volumen de contenidos almacenados en esta gran biblioteca digital que es la Web. La dedicación de los buscadores a contenidos concretos atenuará, al mismo tiempo, la impopularidad de las actualizaciones que hacen los arácnidos o *bots*, útiles para dar información renovada, pero contraindicados por provocar lentitud o colapso de los servidores más solicitados. Los metabuscadores contribuirán, de manera decisiva, a paliar el problema de escalabilidad que tienen, y van a tener de manera más acusada en el futuro, los buscadores genéricos.

Los metabuscadores se distinguen de los buscadores en que:
• no mantienen una base de índices propia,
• a menudo tienen información sobre la especialización de los diferentes buscadores, a fin de elicitar el más adecuado en atención a la pregunta.
Activados los buscadores oportunos, cada uno de ellos ofrecerá un orden de resultados. Al metabuscador le queda, entonces, la tarea crucial de comparar los distintos órdenes, provenientes seguramente de bases de datos diferentes y asentados en criterios de combinación conceptual distintos; y confeccionar una única lista de direcciones, que será la respuesta al tema de la pregunta que introduce un usuario.

El siguiente gráfico explica las características principales de la arquitectura de un metabuscador:

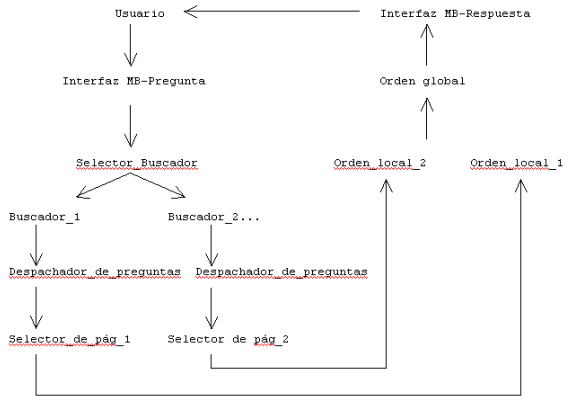


Fig 1. Estructura de un metabuscador

Su secuencia de acción es la siguiente: un usuario escribe en la caja de búsqueda una pregunta. Puede tratarse de una pregunta sencilla o de una frase booleana. El buscador detecta la cuestión y activa el selector de buscadores, eligiendo aquellos especializados en el contenido de esa pregunta. Se traslada la pregunta a cada buscador elegido a través del despachador de preguntas. Cada buscador identifica el número de documentos potencialmente útiles como respuesta a la petición y minimiza la recuperación de documentos inútiles. Usando sus algoritmos de búsqueda, ordena los documentos seleccionados según su relevancia por similitud con el contenido de la pregunta. El metabuscador, atendiendo a sus algoritmos de combinación, seleccionará, de cada orden local, aquellos documentos que mejor contribuyan a un orden global adecuado, que será ofrecido al usuario a través de su interfaz de comunicación como respuesta a la información demandada.

Tareas críticas de un metabuscador son, entre otras, las siguientes:
• Traducir la pregunta a formatos admisibles. La aceptación e interpretación de las preguntas por los buscadores elicitados no siempre es directa. En efecto, puede que:
- la sintaxis de los términos admisibles por el metabuscador y por el buscador sea diferente.
- la semántica de los términos sea distinta –en el caso de búsquedas con frases complejas- en diferentes buscadores.
• Combinar los órdenes locales, que proveen los buscadores activados, en un orden único y global. Se trata de proponer medidas de similitud globales que combinen, de modo adecuado, medidas de similitud parciales entre la pregunta hecha por el usuario y los contenidos de las páginas devueltas como respuesta.
El siguiente apartado de este trabajo incide, para búsquedas léxicas, en estas tareas.

3. DE LA SIMILITUD LOCAL A LA SIMILITUD GLOBAL.

Cada buscador elicitado por un metabuscador puede tener características propias que lo hagan diferente. En efecto, los buscadores pueden diferenciarse en el método de indexación aplicado, en el criterio para atribuir pesos a los términos del índice, en la norma para atribuir pesos a los términos de las preguntas, en las funciones de similitud que evalúan el parecido o proximidad de los términos de las preguntas e índice y en la presentación de los resultados. En este apartado nos centraremos preferentemente en las medidas de similitud entre los términos representativos de las preguntas y los términos del índice.

Las medidas de similitud locales son aquellas que emplean los buscadores especializados y cumplen un papel primordial en la recuperación de información. Una vez que se introduce una pregunta en la caja de búsqueda, el igualamiento entre los términos de la pregunta e índice provocará como respuesta un conjunto de hipertextos que apuntan a direcciones Url seguidas de *snipets*, primeras palabras o líneas de las páginas recuperadas. Usualmente, tanto los términos de las preguntas como los del índice están ponderados y se representan por un vector de pesos. La función de similitud calcula la cercanía entre esos dos vectores y, en consecuencia, provee resultados. En la bibliografía sobre recuperación de información, la función punto y la función coseno son dos de las más utilizadas (Kraft et al, 1999).

El cálculo de la similitud local es un paso previo para hallar la similitud global. Un problema al tratar de averiguar la similitud local es que la atribución de pesos a los términos del índice y de la pregunta atiendan a criterios diferentes. P. ej., que los términos del índice estén ponderados en virtud de su frecuencia y de su frecuencia inversa (*idf*), mientras que los términos de la pregunta sólo estén ponderados por su frecuencia. En ese caso, el metabuscador debe averiguar la frecuencia *idf* de los términos de la pregunta y adjuntarla al vector correspondiente. Hay varias técnicas para hacer esta averiguación, pero no serán desarrollas aquí. Una vez conseguida la homogeneización de criterios en la atribución de pesos, una función de similitud hallará el parecido de los vectores correspondientes, dando como resultando un ordenamiento local de páginas como respuesta a la pregunta. Pero el metabuscador debe hallar la similitud global, que proporcione un orden o lista única de resultados.

Cuando las búsquedas son léxicas y precisas, hay dos estrategias principales en la persecución de este objetivo:

1. Obtener una similitud global por ajuste de similitudes locales. Se trata de usar información adicional para realizar ese ajuste, como la calidad de los buscadores que ofrecen resultados parciales.
2. Obtener la similitud global a partir de los documentos devueltos por los diferentes buscadores.

Para aplicar 1. es necesario identificar si los conjuntos de páginas devueltas son idénticos o diferentes y, en este último caso, si son disjuntos o se solapan. Si las páginas devueltas por los órdenes locales son idénticas, y se especifica la similitud, el problema de ofrecer una lista única se denomina fusión de datos. En este caso, para obtener un orden global se utilizan funciones de combinación lineal como *min*, *max*, *promedio o CombMNZ*. Si las páginas devueltas son disjuntas y se especifican las similitudes locales, estas deben ser normalizadas a un rango común y, dado que las páginas serán únicas, debe puntuarse el sitio al que pertenecen. CORINet, p.ej., halla la similitud global procediendo así: sea s la puntuación de un sitio y s' la puntuación promedio de todos los sitios accedidos. La relevancia de una página para el orden global resultará del producto de s y w, siendo $w=1+N\times(s-s'/s)$ y N es el número de sitios accedidos. Si $s>s'$, w será mayor que 1, y cuanto mayor sea esa desigualdad, mayor será el valor de w; si $s<s'$, w será menor que 1 y vale el razonamiento converso. El resultado global ofrecerá una lista de documentos en orden descendente de similitud entre pregunta y respuesta, ajustada por la calidad del sitio al que pertenece la respuesta. Si no se especifican las similitudes locales, puede seguirse la estrategia de convertir los ordenamientos locales en similitudes. D-WISE sigue el siguiente procedimiento, en el que el escenario de partida es el siguiente: supongamos que r_i es la puntuación atribuida a un sitio S_i y $r_{min}=\min\{r_i\}$ es el mínimo de estas puntuaciones para todos los sitios accedidos por el buscador. p es el orden de la página repuesta a la pregunta en el sitio S_i. La función de conversión g traduce la posición de la página a una similitud: $g=1-(p-1)\times F_i$, donde $F_i=(r_{min})/(m\times r_i)$ y m es el número de páginas que se desea recuperar de entre todos los sitios accedidos.

Para aplicar 2. caben dos aproximaciones principales: 'bajar' los documentos de los órdenes locales para computar aquellos parámetros necesarios para hallar la similitud global o usar el conocimiento descubierto.

Al primer procedimiento se le conoce como *fetching* de documentos y procede así: si un buscador halla respuesta a una consulta lo hace mostrando las Url y los *snipets* que describen brevemente el contenido de las páginas, pero no devolviendo las propias páginas. El método del *fetching* consiste en 'bajarlas' de los servidores locales y en computar las similitudes globales en el metabuscador. Supongamos que la función de similitud global es la función coseno. Este método garantiza que, una vez bajada la página, se pueda determinar la frecuencia local de los términos y, en consecuencia, también su frecuencia

global (si las páginas no se solapan, la frecuencia global es la suma de las frecuencias locales).

El segundo procedimiento es el uso de conocimiento descubierto. Una dificultad en proponer un orden global es que no se puedan comparar las similitudes locales porque diferentes buscadores usan criterios de indexación distintos o computan las similitudes por métodos diferentes. Si esas diferencias se constatan, se evalúa si las similitudes locales son razonablemente comparables, en caso afirmativo, se ajustan y, finalmente, se obtienen similitudes globales a partir de las similitudes locales (Meng et al, 2002).

4. BUSCADORES FUZZY

Algunos sistemas de búsqueda en Internet, disponibles en la Red, se dicen fuzzy, pero esta es, sin duda, una atribución exagerada en algunos casos y dudosa en otros. Algún trabajo reciente (Pal & Talwar, 2002) anuncia buscadores comerciales basados en lógica fuzzy, como Nzsearch, DNS search, Finder, o Aptronix. Nzsearch (www.searchnz.co.nz/) se atribuye el calificativo fuzzy por la posibilidad de corregir deletreos en aquellas direcciones mal tecleadas. Aptronix usa un entorno de inferencia fuzzy para diseñar applets Java inteligentes; no es por tanto, un buscador fuzzy. Si se utiliza un buscador convencional (p. ej., Google) y se le pide información sobre motores de búsqueda fuzzy, devuelve una serie de páginas que, una vez leída su documentación, resultan de dudosa adecuación:

- Infoseek (www.infoseek.com). Se dice fuzzy porque permite a los usuarios especificar la presencia necesaria de un término en una búsqueda colocando un signo + delante del mismo o la ausencia colocando un signo -. Por ejemplo, “+pez" “-peligroso" para buscar peces que no son peligrosos. Como se observa, usa + en vez de "AND" y – en vez de "AND NOT."

- Northern Light (www.northernlight.com). Dice usar un “AND” fuzzy por usar singulares y plurales en la búsqueda, dando, según especifica su documentación, un resultado más amplio que el que proporciona un buscador preciso.

- Discount Engine (http://www.discountdomainsuk.com/glossary/3/499/0), define la búsqueda fuzzy como aquella que tiene éxito en el igualamiento aún cuando las palabras estén parcialmente deletreadas o mal deletreadas. La misma acepción se puede encontrar en http://www.search-marketing.info/search-glossary/engine-terms/fuzzy-search.htm

- Netscape (http://www.netsacape.com) dice usar el operador fuzzy ~, que se escribe después de introducir algunos caracteres. Su misión es obligar al sistema a adivinar lo que viene después de él. P. ej., si recordamos parte de una palabra y escribimos amoxi~, el buscador debería devolver todas las páginas compatibles con ese prefijo, incluidas las que tienen que ver con amoxicilina, nombre completo del término quizás parcialmente olvidado.

No se puede decir que el uso de emparejamientos aproximados para hallar similitud de cadenas les confiera, a estos buscadores, naturaleza fuzzy. Un buscador genuinamente fuzzy debería hacer búsquedas por palabras aproximadas y devolver órdenes graduales en virtud de la vecindad semántica de las palabras de la pregunta e índice; esto es, en virtud de su proximidad de significado, no de la coincidencia de sus caracteres. Ya que no hay buscadores fuzzy, los existentes no pueden ser utilizados para examinar la combinación de lógicas fuzzy. Por eso se hace necesaria una estrategia diferente, que detallamos en el siguiente apartado.

5. SIMILITUD FUZZY EN LOS BUSCADORES Y ÓRDENES GRADUADOS EN LOS METABUSCADORES.

Como se indicó en el párrafo anterior, parece que hay buscadores que implementan algunas funciones sintácticas de naturaleza aproximada, pero no se sabe de buscadores comerciales que implementen algún tipo de semántica fuzzy; es decir, definiciones aproximadas para búsquedas booleanas. Por tanto, para conseguir el objetivo de este trabajo, que es la combinación de lógicas fuzzy y su evaluación global, los buscadores tradicionales no sirven debiendo ser reconvertidos

¿Cómo hacer que un buscador preciso admita búsquedas imprecisas? Si la función de igualamiento es precisa, el buscador sólo devolverá páginas cuyo índice iguale con el término de la pregunta. Si la pregunta emplea operadores booleanos, el emparejamiento será el que determine su semántica bivaluada. Una forma de introducir capacidades de búsqueda imprecisa es asociarle al buscador un diccionario de sinónimos, de manera que para cada términos o términos de la pregunta detecte la acepción y le adjunte el *synset* o conjunto de palabras sinónimas correspondiente acompañados de sus respectivos grados de sinonimia (Fernández Lanza et al, 2003). Las páginas recuperadas ahora serán aquellas que igualan con el término fuente, pero también las que emparejan con los sinónimos. Estas últimas estarían cualificadas con un grado de similitud respecto al orden resultante del emparejamiento exacto, que sería el de la sinonimia entre el sinónimo que posibilitó la recuperación y el término fuente.

Una vez que se dispone de entradas de búsqueda aproximadas, se pueden definir operadores booleanos fuzzy. En efecto, los sinónimos de los términos fuente de

las preguntas estarán ponderados con un grado, que señala la similitud con las palabras de búsqueda originales. Al ser graduados, las combinaciones conceptuales que con ellos se establezcan (AND, OR, NEAR, etc) pueden ser definidas de manera fuzzy. P. ej., para una pregunta que contuviera al operador OR se usaría alguna de las t-conormas conocidas. Estas combinarían los grados de sinonimia de ambos términos y le asociarían el resultado al orden de páginas cuya recuperación hubiesen propiciado; páginas que contendrán, en algún grado, alguna o las dos palabras vinculadas por el operador OR.

Una vez que se dispone de órdenes locales afectados por un grado de similitud queda el reto de combinarlos en un orden global que será también aproximado o ponderado por un grado. Antes de iniciar este proceso deben verificarse algunas cuestiones previas:

- Puede que los grados de similitud entre términos de la pregunta y del índice que da origen a los órdenes locales se hallen, en distintos buscadores, usando funciones de sinonimia diferentes.
- Si las búsquedas son con frases booleanas puede que se usen, en distintos buscadores, diferentes operadores fuzzy.

Antes de comparar órdenes locales hay que uniformizar estos criterios. La información que usen los buscadores respecto a las funciones de sinonimia y operadores booleanos implementados debe ser pública y ha de poder ser rastreada por un metabuscador, para que emprenda transformaciones adecuadas o compare sólo aquellos buscadores que sean homogéneos. Una vez construida o constatada esa homogeneización, queda efectuar la combinación de órdenes locales para obtener el orden global.

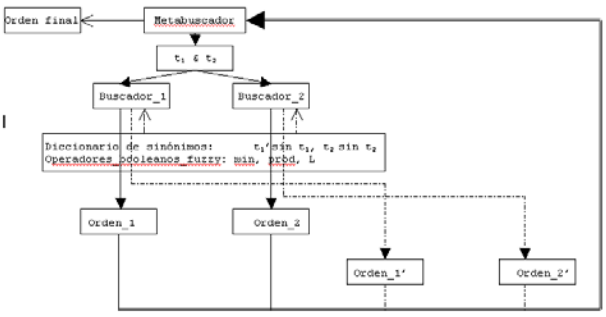


Fig 2. Ciclo de combinación de órdenes parciales en un metabuscador

La tarea de integración o combinación conlleva, de modo estándar, dos pasos: eliminación de duplicados y fusión de similitudes (Schalen et al, 2004). Veamos, para nuestro caso, una posible e inicial forma de operacionalizar este proceso:

- Eliminación de duplicados. La búsqueda por 'coche & rápido' puede devolver las mismas páginas que la búsqueda por 'automóvil & rápido', ya que 'coche' y 'automóvil', por ser sinónimos en un grado alto, pueden estar copresentes en muchos documentos. Para eliminar duplicados habría que detectar las URL, comprobar su fecha de actualización y, en caso de igualdad, proceder a eliminar uno de los documentos.
- Fusión de similitudes. Imaginemos que una búsqueda por 'coche potente' queda asociada, en virtud de la sinonimia de uno de los términos, a 'coche veloz' (1) y a 'coche rápido' (2). Supongamos que, fruto de esa asociación, el buscador devuelve dos listas de 10 páginas, que aquí rotulamos así:

(1)	(2)
a	a
b	b
c	c
d	k
e	e
f	l
g	g
h	h
i	m
j	j

La fusión de similitudes, que deberá efectuar el metabuscador para proporcionar un orden único y global, puede ser el producto de:

-Primar aquellas páginas que tengan términos más sinónimos al término fuente.
-Ponderar el orden según la posición de la página en el ranking.

El cálculo de la sinonimia entre el término fuente de la búsqueda y los términos de su synset lo hace el diccionario electrónico de sinónimos usando alguna de las medidas de similitud que tiene implementadas. A continuación, se asocian a las páginas los grados de sinonimia de las palabras o términos que contienen. La ponderación según la posición de la página en el ranking puede hallarse, en una solución sencilla, usando la siguiente fórmula, p=(11-n)/10, siendo n la posición en el ranking de la página considerada y 10 el número de páginas recuperadas.

Así, para el caso anterior, suponiendo que el grado de sinonimia de 'veloz' y 'potente' es 0.6 y que el grado de sinonimia de 'rápido' y 'potente' es 0.4, la fusión de los dos órdenes locales, ponderados según la posición de las páginas, daría como resultado el siguiente orden global:

a
b
c
d
e
f
k
g

l
h
i

6. CONCLUSIONES

Este trabajo ha querido mostrar como los metabuscadores pueden ser un medio interesante en el que probar asociaciones de lógicas fuzzy. Si el metabuscador combina lógicas fuzzy de modo adecuado, arrojará buenos ordenamientos de páginas y, en consecuencia, será muy solicitado. Por tanto, su éxito puede ser utilizado como un criterio pragmático para diferenciar buenas y malas asociaciones.

Para búsquedas aproximadas se necesitan buscadores fuzzy. Pero no hay todavía buscadores comerciales a los que pueda atribuírsele ese calificativo. Por tanto, hay que adaptar alguno de los existentes para que permita búsquedas imprecisas. La asociación de un diccionario de sinónimos y la implementación de operadores booleanos fuzzy posibilita esta tarea.

Algunos aspectos críticos de la combinación de los órdenes locales ofrecidos por cada buscador fuzzy en el orden global graduado que debe proporcionar el metabuscador son:

• Uniformización de posibles diferencias en la definición de funciones de sinonimia y operadores booleanos fuzzy empleados por cada buscador.

• Indagación en funciones de integración adecuadas para combinar órdenes parciales.

El estudio de estos aspectos debe ser ampliado en un futuro; aquí sólo se ha tratado de dar una aproximación inicial al mismo.

Agradecimientos

Financiación a cargo del proyecto PGIDT02PXIB30501PR de la Xunta de Galicia

Referencias

Bordogna, G. & Pasi, G. & Yager, R. R., 'Soft approaches to distributed information retrieval', *Int. J. of Approximate Reasoning*, 34 (2003), 105-120.

Bordogna, G. & Pasi, G., 'A model for a Soft Fusion of Information Accesses on the web'. To be published in *Fuzzy Sets and Systems*.

Fernández Lanza, S. & Graña, J. & Sobrino, A. 'Introducing FDSA (Fuzzy Dictionary od Synonims and Antonyms) Applications on Information Retrieval and Stand-Alone Use', *Mathware and Soft Computing*, vol. X, nº 2-3 (2003), 57-70.

Gabbay, D., *Fibring Logics*. Clarendom Press, 1999.

Herrera-Viedma, E. et al. 'Incorporating filtering techniques in a fuzzy linguistic multi-agent model for information gathering on the web', To appear in *Fuzzy Sets and Systems*.

Kraft, D. H. & Bordogna, G. & Passi, G., 'Fuzzy Set Techniques in Information Retrieval', in Bezdek, J. C., & Dubois, D. & Prade, H. (eds.), *Fuzzy Sets in Approximate Reasoning and Information Systems*. Kluwer, 1999.

Meng, W. & Yu, C. & Liu, K-L., 'Building Efficient and Effective Metasearch Engines', *ACM Computing Surveys*, vol. 34, nº1, March 2002, pp. 48-89.

Pal, S. P.& Talwar, V., 'Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions', *IEEE Transactions on Neural Networks*, vol. 13, nº 5, sep. 2002

Olivas, J. A. & Garcés, P. J. & Romero, F. P., 'An application of the FISS-CRM model to the FISS metasearcher: Using fuzzy synonymy and fuzzy generality for representing concepts in documents', *International Journal of Approximate Reasoning*. To appear

Schalen, E. & Sattler, K-U. & Saake, G., Efficient similarity-based operations for data integration', *Data & Knowledge Engineering*, 48, 2004, 361-387.

Trillas, E. & Alsina, C. & Terricabras, J. M., *Introducción a la lógica borrosa*. Ariel, 1995.

APRENDIZAJE Y AJUSTE GENÉTICO DE SISTEMAS DIFUSOS BASADO EN LA REPRESENTACIÓN DE 2-TUPLAS LINGÜÍSTICAS

Jesús Alcalá-Fdez, Rafael Alcalá, Francisco Herrera
Departamento de Ciencias de la Computación e Inteligencia Artificial
Universidad de Granada, 18071 Granada, España
{jalcala, alcala, herrera}@decsai.ugr.es

Resumen

El Modelado Difuso Lingüístico permite la obtención de modelos claramente interpretables por el ser humano. Sin embargo, la *precisión* y la *interpretabilidad* de los modelos obtenidos son propiedades claramente contradictorias, provocando una falta de precisión en sistemas complejos. Por ello, en los últimos años se ha tratado de dotar al modelado lingüístico de una mayor precisión sin perder la interpretabilidad en un alto grado.

Siguiendo esta premisa, en este trabajo se presenta un método para la obtención de modelos lingüísticos basado en el aprendizaje evolutivo de la base de datos (granularidad y ajuste de las particiones difusas) y en el uso de un método básico para la generación de reglas. Para reducir el espacio de búsqueda y facilitar el ajuste de las funciones de pertenencia, se considera un nuevo esquema de representación de reglas basado en el modelo de 2-tuplas lingüísticas, que permite la variación lateral de las etiquetas mediante el ajuste de un único parámetro por etiqueta.

Palabras Clave: Sistemas basados en reglas difusas, modelado lingüístico, ajuste lateral, aprendizaje de la granularidad.

1 INTRODUCCIÓN

Uno de los problemas del Modelado Difuso Lingüístico (MDL) es su falta de precisión al modelar sistemas complejos. Esto se debe a que el concepto de variable lingüística impone fuertes restricciones a la estructura de regla difusa [2]. Este inconveniente aleja al MDL del deseado equilibrio entre precisión e interpretabilidad.

Para solucionar este problema, en la literatura [3], se han considerado muchas y diferentes mejoras al MDL. Uno de los enfoques más utilizados para mejorar el comportamiento de los Sistemas Basados en Reglas Difusas (SBRDs), llamado *ajuste*, consiste en el refinamiento de la Base de Datos (BD) una vez que la Base de Reglas (BR) ha sido obtenida. Generalmente, el ajuste es una variación en la forma de las funciones de pertenencia con el principal objetivo de mejorar la precisión del modelo lingüístico. En el enfoque de ajuste clásico, si las etiquetas tienen forma triangular hay que ajustar tres parámetros por etiqueta.

En [1], se presentó un nuevo esquema de representación de reglas lingüísticas para el ajuste evolutivo de la BD mediante un deplazamiento del soporte de las funciones de pertenencia. Este enfoque se basa en el modelo de representación de 2-tuplas lingüísticas [10], que permite la variación lateral de las etiquetas mediante el ajuste de un único parámetro por etiqueta. De esta manera, se consiguieron dos objetivos:

- obtener etiquetas que contengan un conjunto de ejemplos con mejor cubrimiento (búsqueda de precisión) conservando su forma original, y
- reducir el espacio de búsqueda del proceso de ajuste respecto al ajuste clásico para obtener fácilmente un modelo óptimo.

Sin embargo, dicho esquema de aprendizaje parte de una BD inicial y una BR que permanece fija durante todo el proceso de ajuste de las funciones de pertenencia. Por lo tanto, el proceso de aprendizaje estará sesgado por la BD y BR iniciales, quedando fijos aspectos tales como la granularidad de las particiones, o el número y la estructura de las reglas obtenidas.

En este trabajo, nos proponemos extender el método de aprendizaje propuesto en [1] de manera que se puedan aprender Bases de Conocimiento (BCs) completas. Para ello, proponemos un método evolutivo para la obtención de BCs basado en el aprendizaje de la granularidad y el ajuste lateral de la BD, que aprenda al mismo tiempo el número de etiquetas por variable, el desplazamiento lateral de dichas etiquetas

(ajuste) y que, a partir de esto, mediante un método simple de generación de reglas, obtenga la BR correspondiente. A modo de ejemplo, analizaremos este enfoque en un problema real, tanto desde el punto de vista de la precisión como de la interpretabilidad.

En la siguiente Sección, se explica el modelo de representación de reglas basado en 2-tuplas. El esquema de aprendizaje seguido para el aprendizaje de BCs se introduce en la Sección 3. En la Sección 4, se presenta el método de aprendizaje propuesto en este trabajo. La Sección 5 muestra el comportamiento de dicho método en un problema de estimación real. Por último, en la Sección 6, se muestran algunas conclusiones.

2 EL AJUSTE LATERAL

En [1], se propuso un nuevo modelo de ajuste de SBRDs basado en el modelo de representación de 2-tuplas lingüísticas propuesto en [10], que permite el desplazamiento lateral del soporte de una etiqueta y mantiene la interpretabilidad asociada al modelo obtenido. Esta propuesta introduce un nuevo modelo de representación de reglas basado en el concepto de traslación simbólica.

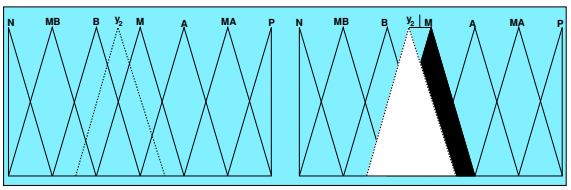


Figura 1: Desplazamiento lateral de la etiqueta M.

La Figura 1 representa la variación lateral de la etiqueta M. La nueva etiqueta "y_2" se situa entre B y M, siendo menor que M pero más cercana a M que a B.

La traslación simbólica de un término lingüístico es un número dentro del intervalo [-0.5, 0.5) que expresa la variación del dominio de una etiqueta, cuando ésta se mueve entre las dos etiquetas laterales a la misma. Formalmente, tenemos el par

$$(s_i, \alpha_i), \;\; s_i \in S, \;\; \alpha_i \in [0.5, -0.5).$$

En [10], se propone tanto el modelo de representación de 2-tuplas lingüísticas, como los elementos para la comparación y agregación de información lingüística necesarios para su uso en modelos de Toma de Decisiones. A continuación, se puede ver su uso para la representación y ajuste de reglas lingüísticas considerando un modelo simple de control.

Consideremos un problema de control con dos variables de entrada, una de salida y una BD definida a partir de expertos:

$$Error \rightarrow \{N, C, P\}, \quad \bigtriangledown Error \rightarrow \{N, C, P\},$$
$$Potencia \rightarrow \{B, M, A\} \;\; .$$

Regla Clásica:
R1: Si el **Error** es Cero y la **Variación del Error** es Positiva entonces la **Potencia** es Alta

Regla con Representación de 2-tuplas:
R1: Si el **Error** es (Cero, 0.3) y la **Variación del Error** es (Positiva, -0.2) entonces la **Potencia** es (Alta, -0.1)

Figura 2: Regla clásica y regla con representación de 2-tuplas.

La figura 2 muestra el concepto de regla clásica y regla con representación de 2-tuplas. Desde el punto de vista de la interpretabilidad, la regla ajustada se podría interpretar como sigue:

Si el **Error** es "*mayor* que Cero" y
la **Variación del Error** es "*menor* que Positiva"
entonces la **Potencia** es "un *poco menor* que Alta".

Esta propuesta reduce la complejidad del problema de ajuste, al reducir los 3 parámetros por etiqueta a 1 parámetro de traslación simbólica.

En [1], se proponen dos enfoques para el ajuste, un enfoque global y uno local. En nuestro caso, el ajuste se produce al nivel de las particiones lingüísticas (enfoque global). En todas las reglas en las que aparece el par (X_i, etiqueta) se mantiene el mismo valor de ajuste. Por ejemplo, X_i es (Alto, 0.3) presentará igual valor para toda regla en la que se encontrase inicialmente el par "X_i es Alto".

3 APRENDIZAJE DE LA BC

Clásicamente, el aprendizaje de la BC de un SBRD se basa en la existencia de una BD previamente definida [12]. Generalmente, la BD es obtenida escogiendo un número de etiquetas para cada cada variable y fijando los parámetros mediante una distribución uniforme de dichas etiquetas. Sin embargo, mediante esta forma de trabajo, la BD tiene una gran influencia en el comportamiento del SBRD finalmente obtenido, ya que la BR obtenida depende de la bondad de la BD.

Otros enfoques realizan un *ajuste* a *posteriori* de la BD [1, 4, 6]. Para ello, aprenden los parámetros o la forma de las funciones de pertenencia, mejorando la precisión del SBRD. No obstante, el proceso de aprendizaje está sesgado por la BD y BR iniciales, quedando fijos aspectos tales como la granularidad de las particiones, o el número y la estructura de las reglas.

En otros trabajos se aprende conjuntamente la BD y la BR [5]. Ésto permite mejores definiciones de la BC,

pero el espacio de búsqueda que se maneja es demasiado grande haciendo muy difícil la búsqueda de una solución óptima. Otra forma de generar la BC consiste en obtener la BD y BR por separado, basándose en el aprendizaje de la BD a *priori* [9] (véase la Figura 3).

El esquema de aprendizaje considerado en este trabajo sigue este último enfoque y, se compone de dos partes con objetivos diferentes:

- Un proceso genético de aprendizaje de la BD, que permite definir:
  - El número de etiquetas de cada variable.
  - El desplazamiento lateral de dichas etiquetas.
- Un método *Ad-hoc* de generación de reglas basado en ejemplos [12], que obtiene la BR a partir de la BD previamente obtenida. Dicho método se ejecuta para cada BD generada por un Algoritmo Genético (AG), construyendo BCs completas que son evaluadas dentro de dicho algoritmo.

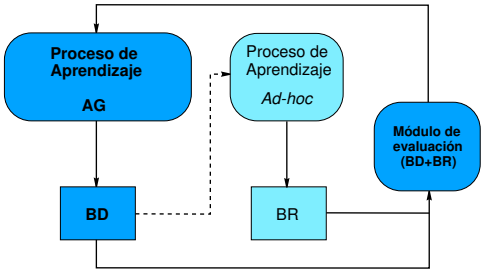


Figura 3: Esquema de aprendizaje de la BC.

4 ALGORITMO EVOLUTIVO DE APRENDIZAJE DE LA BC

Un buen esquema de evolución es el modelo genético de CHC [8]. CHC hace uso de un enfoque de "Selección basado en Poblaciones". N padres y sus hijos son combinados para seleccionar los N mejores individuos que formarán parte de la siguiente población. En este trabajo, se aplica este método con algunas modificaciones.

El proceso de aprendizaje de la BD, se compone de dos fases (propiciadas por la población inicial y el esquema de reinicialización). En la primera, se aprende la granularidad de las particiones difusas fijando los desplazamientos a cero, mientras que en la segunda (tras la primera reinicialización), se fija la granularidad de la mejor solución y se obtienen los desplazamientos. Para ello, se utiliza un esquema de codificación doble (granularidad + desplazamientos).

En adelante, se explicarán los componentes principales del algoritmo propuesto (codificación de la BD, población inicial y reinicialización, función de evaluación y operador de cruce).

4.1 CODIFICACIÓN DE LA BD

Se utiliza un esquema de codificación doble ($C = C_1 + C_2$) para ambas partes, *granularidad* y *parámetros de traslación*:

- Aprendizaje del número de etiquetas (C_1): Es un vector de números enteros de tamaño N (siendo N el número de variables). Los posibles valores considerados son el conjunto $\{3, \ldots, 9\}$:

$$C_1 = (E^1, \ldots, E^N) \ .$$

- Ajuste de la semántica (C_2): Es un vector de tamaño $N * 9$ (N variables con un máximo de 9 etiquetas) en el que se codifican los desplazamientos de las distintas etiquetas, para cada variable. Por supuesto, si un individuo no tiene el máximo número de etiquetas posible en alguna de las variables, las zonas reservadas para los valores de esas etiquetas no se usan. De esta manera, la parte C_2 tiene la siguiente estructura:

$$C_2 = (\alpha_1^1, \ldots, \alpha_{E^1}^1, \ldots, \alpha_1^N, \ldots, \alpha_{E^N}^N)$$

4.2 POBLACIÓN INICIAL Y REINICIALIZACIÓN

En nuestro caso, estos dos componentes están relacionados para conseguir el aprendizaje en dos fases anteriormente mencionado:

1. Primera fase (granularidad). Para ello la población inicial estará compuesta de dos bloques:
   - En el primer bloque, cada cromosoma tiene el mismo número de etiquetas para todas las variables del problema y se consideran particiones unifomes con los parámetros de traslación fijados a cero.
   - En el segundo bloque (de igual tamaño), la única diferencia es que cada variable puede tener un número de etiquetas distinto.

   Puesto que CHC no tiene operador de mutación, los parámetros de traslación permanecen fijos y se aprende el número de etiquetas más prometedor para cada variable. Ésto es así hasta que se produzca la primera reinicialización.
2. Segunda fase (ajuste lateral). Para salir de óptimos locales, se utiliza un mecanismo de reinicialización [8]. En este caso, todos los cromosomas

fijan su parte C_1 a la de la mejor solución global y los parámetros de la parte C_2 se generan aleatoriamente en el intervalo [-0.5, 0.5). Además, si la mejor solución global ha cambiado desde la última inicialización, ésta se incluye en la población (la explotación sigue mientras haya convergencia).

4.3 FUNCIÓN DE EVALUACIÓN

Para evaluar un cromosoma se aplica el algoritmo de generación de reglas de Wang y Mendel [12] sobre la BD codificada en dicho cromosoma. Una vez obtenida la BC completa, se calcula su Error Cuadrático Medio (ECM) y se trata de minimizar la siguiente función:

$$F_C = w_1 \cdot ECM + w_2 \cdot NR,$$

donde, NR es el número de reglas de la BC actual (para penalizar un número excesivo de reglas), $w_1 = 1$ y w_2 se calcula a partir del ECM y el número de reglas de la BC obtenida a partir de una BD con el máximo número de etiquetas permitido (9 etiquetas),

$$w_2 = \alpha \cdot \frac{ECM_{max-et}}{NR_{max-et}} \quad .$$

Por otro lado, para considerar la existencia de ejemplos no cubiertos en el cálculo del ECM, se aplica el siguiente mecanismo:

1. Se busca la regla más cercana al ejemplo no cubierto (distancia euclidea normalizada). Las coordenadas no cubiertas del ejemplo toman como valor el centro de la correspondiente etiqueta.
2. Se busca la segunda regla más cercana. Si los consecuentes de ambas reglas se solapan sólo se dispara la regla más cercana. En otro caso, la salida debería ser por interpolación de ambas reglas. Para ello, las coordenadas del ejemplo que en un inicio estaban cubiertas, se desplazan hacia la segunda regla sin salirse del cubrimiento de la más cercana (se acercan al extremo un 10% del tamaño del soporte). Finalmente, se infiere con el nuevo ejemplo.

4.4 OPERADOR DE CRUCE

Se aplican dos operadores de cruce distintos según la información de los padres: Cuando los padres tienen distinto número de etiquetas en alguna variable, se fija un punto de corte en C_1 y se aplica el operador clásico de cruce en las dos partes C_1 y C_2 (exploración); Cuando los padres tienen igual número de etiquetas para cada variable, se aplica un operador de cruce del tipo de entorno alrededor de los padres basado en el BLX, el operador PBLX [11], sólo sobre la parte C_2 (explotación).

5 EXPERIMENTOS Y ANÁLISIS DE RESULTADOS

Con objeto de analizar el comportamiento del método propuesto, aprendizaje de la granularidad y ajuste lateral global, hemos escogido un problema real, la estimación de la longitud de las líneas de baja tensión de una compañía eléctrica [7]. En la siguiente subsección se puede encontrar una breve descripción de este problema.

Tabla 1: Métodos considerados en los experimentos.

Ref.	Método	Tipo Aprendizaje/Ajuste
[12]	**WM**	*Ad-hoc* Basado en Ejemplos
[6]	**T**	Parámetros (Ajuste Clásico)
[4]	**PAL**	Parámetros + Dominios + Modificadores Lingüísticos
[1]	**GL**	Ajuste Lateral Global
—	**GGL**	Granularidad + Ajuste Lateral Global + BR

La Tabla 1 presenta una breve descripción de los métodos estudiados. El método de WM es utilizado como método simple para la generación de una BC inicial. Los distintos métodos de ajuste (T, PAL y GL) son aplicados una vez que dicha BC es obtenida. El método propuesto (GGL) incluye el algoritmo de WM dentro del propio proceso de aprendizaje de la BD. T representa un algoritmo clásico de ajuste de los parámetros de las funciones de pertenencia. El método PAL ha sido comparado con métodos de ajuste de parámetros, de dominio (factores de escala) y de modificadores lingüísticos y con cualquier combinación de ellos dos a dos, y obtiene mejores resultados [4]. Por ello, sólo consideramos la combinación de los tres componentes (parámetros, dominio y modificadores) en este estudio.

Las particiones lingüísticas consideradas tendrán *cinco términos lingüísticos* con forma triangular para WM y los métodos T, PAL y GL (número con el que presentan el mejor comportamiento). Se utiliza el *centro de gravedad ponderado por el grado de emparejamiento* como operador de defuzzificación y la *t-norma del mínimo* como operador de implicación y conjunción.

Los valores de los parámetros utilizados en todos los experimentos presentados son: 50 individuos en la población, 50000 evaluaciones y 0,6 como probabilidad de cruce, para un PBLX en el caso de GL y GGL (0,2 como probabilidad de mutación por cromosoma y 0,35 para el factor a del operador de cruce max-min-aritmético para T y PAL). El factor α para la función de evaluación en GGL se ha fijado a 1,15.

Tabla 2: Resultados obtenidos por los métodos estudiados

Método	**#R**	**ECM**$_{ent}$	σ_{ent}	**t-test**	**ECM**$_{test}$	σ_{test}	**t-test**
WM	12.4	234712	32073	–	242147	24473	–
WM + T	12.4	158662	6495	–	221613	29986	–
WM + PAL	12.4	**141638**	4722	⋆	189279	22022	–
WM + GL	12.4	166674	11480	–	189216	14743	–
GGL	**8.8**	162295	4059	–	**177507**	16234	⋆

5.1 DESCRIPCIÓN DEL PROBLEMA

Algunas veces existe la necesidad de medir la longitud de las líneas de electricidad que tiene una compañía eléctrica. Esta medida puede ser útil para estimar los costes de mantenimiento de la red, lo cual constituyó el principal objetivo planteado en la provincia de Asturias [7]. La línea de baja tensión resulta costosa de medir en lugares montañosos e irregulares y, una compañía puede servir a más de 10.000 pequeños núcleos. En consecuencia, es necesario determinar un modelo indirecto para calcular la longitud de dichas líneas. De esta manera, el problema consiste en encontrar un modelo que relacione la *longitud total de línea de baja tensión* instalada en un núcleo rural con el *número de habitantes* y su *radio* [7].

Para ello, se cuenta con una muestra de 495 núcleos rurales. En los distintos experimentos, se trabaja con un modelo de *validación cruzada con 5 particiones* de datos, esto es, 5 particiones aleatorias al 20%, y la combinación de cuatro particiones (80%) como entrenamiento y una partición como test. Así se tienen 5 particiones al 80% y 20% en entrenamiento y test.

5.2 RESULTADOS Y ANÁLISIS

Para cada partición, todos los modelos evolutivos se ejecutan 6 veces, y se muestran los resultados medios del error para las 30 ejecuciones de cada algoritmo sobre cada problema. Los resultados obtenidos se muestran en la Tabla 2, donde $\#R$ representa el número de reglas, ECM$_{ent}$ y ECM$_{test}$ el error medio en entrenamiento y test, σ la desviación típica y *t-test* los resultados de aplicar un *test t-student (con 95% de confianza)* al mejor resultado medio de la correspondiente columna comparado 1 a 1 con el resto de resultados medios. La interpretación de esta última columna es la siguiente:

⋆	Indica el resultado con la mejor media
–	Indica peor comportamiento que el mejor

Analizando los resultados presentados en la Tabla 2 podemos indicar las siguientes conclusiones:

- A pesar de que el método puede obtener particiones con una granularidad alta, la base de reglas está aprendida mediante un método de generación de pocas reglas (8,8 sobre 81 posibles si la partición de entrada es de 9x9) que junto con el objetivo de minimizar el número de reglas facilitan la obtención de modelos precisos pero a la vez compactos y, por lo tanto, más interpretables. En este caso se han eliminado casi 4 reglas respecto a los modelos obtenidos por el resto de los métodos (reducción de un 29% en el número de reglas).

- El método propuesto muestra una fuerte reducción del error cuadrático en el conjunto de test en un problema fuertemente no lineal y con puntos de ruido, presentándose robusto frente al sobreaprendizaje y manteniendo un error en entrenamiento similar al del resto de métodos (excepto en el caso de WM+PAL). Además, la desviación típica es de las más bajas. En este caso, hay que destacar que cada partición de datos se comporta como un problema distinto, siendo la desviación en test similar a la de entrenamiento si se calculase respecto a cada partición.

- El ajuste lateral de funciones de pertenencia (un único parametro por función) reduce el espacio de búsqueda respecto a un ajuste clásico y, por tanto, es ideal para ser combinado con el aprendizaje de la granularidad de las particiones difusas, problema en que se maneja un espacio de búsqueda de gran tamaño. Además, los modelos así obtenidos son altamente interpretables puesto que la forma de las etiquetas originales no es alterada.

Las Figuras 4 y 5 muestran la BD evolucionada y la BR obtenida con el método GGL de una de las 30 ejecuciones realizadas. Para facilitar la representación gráfica, en estas figuras las etiquetas se nombran desde 'l1' hasta 'lE^i'. No obstante, dichas etiquetas tienen asociado un significado lingüístico impuesto por un experto. Por ejemplo, si la etiqueta 'l1' en la variable 'X1' significa 'BAJO', 'l1+0,11' se puede interpretar como 'un poco mayor que BAJO' (según la apreciación

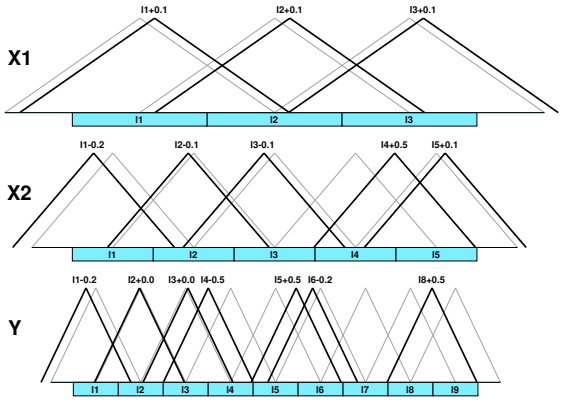


Figura 4: BD sin ajuste (en gris) y BD ajustada (en negro – sólo se muestran las etiquetas utilizadas en la BR) de un modelo obtenido con GGL.

#R: 10
ECM-ent: 167084,468
ECM-pru: 152948,173

X1	X2	Y
I1+0,11	**I1** -0,24	**I1** -0,22
I1+0,11	**I2** -0,07	**I2** -0,03
I1+0,11	**I3** -0,13	**I4** -0,50
I1+0,11	**I4**+0.48	**I3**+0,05
I1+0,11	**I5**+0,11	**I5**+0,46
I2+0,11	**I2** -0,07	**I3**+0,05
I2+0,11	**I3** -0,13	**I6** -0,18
I2+0,11	**I4**+0,48	**I6** -0,18
I3+0,10	**I2** -0,07	**I4** -0,50
I3+0,10	**I3** -0,13	**I8**+0,48

Figura 5: BR y desplazamientos de un modelo obtenido con GGL.

del experto) ó, como en el caso del ajuste clásico, mantener el significado original de dicha etiqueta.

6 CONCLUSIONES

En este trabajo, se presenta un nuevo método para el aprendizaje de BCs basado en el aprendizaje evolutivo de la BD (granularidad y pámetros de traslación). A continuación se presentan algunas conclusiones:

- El esquema de aprendizaje seguido y el uso del ajuste lateral permiten una reducción del espacio de búsqueda que facilita la obtención de modelos precisos y compactos.
- Al seguir un enfoque global y mantener la forma de las etiquetas originales, la interpretabilidad de los modelos se mantiene en un alto grado.

En este momento, se está realizando un estudio sobre distintos problemas para determinar la dependencia entre la complejidad del problema y el funcionamiento de los distintos métodos de aprendizaje.

Agradecimientos

Este trabajo ha sido financiado por el CICYT bajo el proyecto KEEL (TIC-2002-04036-C05-01).

Referencias

[1] R. Alcalá, F. Herrera. Genetic tuning on fuzzy systems based on the linguistic 2-tuples representation. *Procs. of FUZZ-IEEE Int. Conf.*, 2004, por aparecer.

[2] A. Bastian. How to handle the flexibility of linguistic variables with applications. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems* 3:4, Pág.463-484, 1994.

[3] J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds). Accuracy improvements in linguistic fuzzy modeling. *Springer-Verlag*, 2003.

[4] J. Casillas, O. Cordón, M.J. del Jesús, F. Herrera. Genetic tuning of fuzzy rule deep structures preserving interpretability for linguistic modeling. *IEEE Trans. on Fuzzy Systems*, 2004, por aparecer.

[5] B. Carse, T.C. Fogarty, A. Munro, Evolving fuzzy rule based controllers using genetic algorithms. *Fuzzy Sets and Systems* 80, Pág.273-293, 1996.

[6] O. Cordón, F. Herrera. A three-stage evolutionary process for learning descriptive and approximate fuzzy logic controller knowledge bases from examples. *Int. J. of Approximate Reasoning* 17:4, Pág.369-407, 1997.

[7] O. Cordón, F. Herrera, L. Sánchez. Solving electrical distribution problems using hybrid evolutionary data analysis techniques. *Applied Intelligence* 10, Pág.5-24, 1999.

[8] L.J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. *Foundations of genetic Algorithms* 1, G.J.E. Rawlin (Ed). Morgan Kaufman, Pág.265-283, 1991.

[9] B. Filipic, D. Juricic, A genetic algorithm to support learning fuzzy control rules from examples. *Genetic Algorithms and Soft Computing*, F. Herrera, J.L. Verdegay (Eds). Physica-Verlag, Pág.403-418, 1996.

[10] F. Herrera, L. Martínez. A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Trans. Fuzzy Systems* 8:6, Pág.746-752, 2000.

[11] F. Herrera, M. Lozano, A.M. Sánchez. A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *Int. J. of Intelligent Systems* 18, Pág.309-338, 2003.

[12] L.X. Wang, J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Trans. on Systems, Man, and Cybernetics* 22:6, Pág.1414-1427, 1992.

METHOD: Un Enfoque Constructivo para Modelado Difuso Híbrido

Mercedes Valdés Vela **Antonio F. Gómez Skarmeta** **Juan A. Botía Blaya**

Departamento de Ingeniería de la Información y las Comunicaciones.
Universidad de Murcia. Campus de Espinardo.
E-30071. Murcia, España

Resumen

En los últimos años muchos autores han abordado el Modelado Difuso Conducido por datos (DDFM) combinando diferentes técnicas formando estrategias híbridas demostrando la validez y éxito de este enfoque para la construcción de sistemas de inferencia difusos a partir de datos entrada-salida. Sin embargo, estas combinaciones suelen hacerse ad hoc debido a la falta de mecanismos para combinar las técnicas de una manera sencilla y unificada. Este trabajo perfila el diseño de un marco llamado METHOD (Marco para la Especificación de Técnicas Híbridas de mOdelado Difuso) que ofrece las funcionalidades necesarias para combinar técnicas modulares con el objetivo de obtener estrategias híbridas de DDFM.

Palabras Clave: modelado difuso conducido por datos, sistemas de inferencia difusos, hibridación, esquemas de interacción.

1. INTRODUCCIÓN

El objetivo del Modelado Difuso Conducido por datos (DDFM) es encontrar un sistema de inferencia difuso (FIS) capaz de simular a un sistema de partida del cual conocemos observaciones entrada-salida. Esta clase de modelos se basa en la Teoría del Razonamiento Aproximado [10] para llevar a cabo las inferencias. El DDFM se divide en dos fases fundamentales. En primer lugar la fase de *identificación de la estructura* que consiste en el descubrimento de las relaciones entrada-salida que guían el comportamiento del sistema en forma de reglas difusas y, en segundo lugar, la fase de *optimización de los parámetros* de esas reglas. Por otro lado la identificación de la estructura consiste en: (1) la tarea de *selección de variables* que se refiere a la determinación de las variables de entrada que afectan a la salida del sistema, (2) la *identificación del orden* del modelo o, en otras palabras, el número de relaciones entrada-salida y (3) la *identificación de las propias relaciones entrada-salida.*

Debido, en parte, a esta naturaleza modular, en los últimos años han aparecido muchos enfoques híbridos para llevar a cabo tareas de DDFM. Se trata de estrategias híbridas en el sentido de que se componen de dos o más técnicas monolíticas (algoritmos de clustering, redes neuronales, algoritmos genéticos, etc.), cada una de ellas resolviendo una de las fases del DDFM [8, 2, 4]. Este enfoque híbrido ha demostrado ser una herramienta válida y exitosa para la identificación de sistemas. De hecho, tanto nuestras investigaciones al respecto como los trabajos de otros muchos autores confirman este resultado. Sin embargo, estas hibridaciones suelen hacerse ad hoc debido en parte a que no hay consenso en cuanto al número, propósito, definición (requerimientos y efectos) e interrrelaciones de las fases del DDFM.

Esta heterogeneidad conduce a la necesidad de herramientas para manejar la hibridación de técnicas de DDFM de una forma sencilla y unificada. Es por ello por lo que hemos diseñado METHOD, un Marco para la Especificación de Técnicas de mOdelado Difuso.

Este artículo se estructura como sigue: la sección 2 expone una visión general del marco. El resto del artículo se dedica a describir los elementos que lo componen. Así, en la sección 3 se describen los aspectos fundamentales sobre el universo de discurso de METHOD. La sección 4 se dedica a los conceptos de *sesión* e *instantánea* del DDFM. En la sesión 5 se describe la idea de *especificación METHOD de una técnica.* La sección 6 está dedicada a la hibridación de técnicas y al concepto de *esquema de hibridación.* Finalmente, la sección 7 presenta las conclusiones y trabajos futuros.

2. VISIÓN GENERAL DE METHOD

Desde una perspectiva práctica, las funcionalidades de METHOD se pueden resumir en cuatro roles principales que puede jugar un usuario. En primer lugar, el *integrador* es el usuario que toma un algoritmo implementado previamente susceptible de resolver alguna de las fases de DDFM, y le hace las modificaciones necesarias para incorporarlo al marco. El siguiente tipo de usuario es el *hibridador*, es decir, aquel que toma técnicas integradas en METHOD y las combina con el objetivo de crear estrategias capaces de resolver el proceso completo de DDFM. El *modelador* utiliza las técnicas integradas en METHOD para resolver tareas de DDFM. Finalmente, el *administrador* registra las técnicas incorporadas en METHOD para que estén disponibles para el *hibridador* y el *modelador*.

La figura 1 muestra los elementos fundamentales que forman el diseño de METHOD y cómo se relacionan. Cada técnica susceptible de resolver alguna de las fases de DDFM se encapsula en lo que llamamos su **especificación METHOD**. Este *envoltorio* explicita las capacidades que la técnica tiene en el contexto de DDFM empleando para ello términos pertenecientes al **universo de discurso** de METHOD. De esta manera, dichas técnicas se manejan como módulos que se pueden combinar de un modo sencilla. Para ello se pueden seguir distintos **esquemas de interacción**. Éstos determinan las relaciones entre los diferentes componentes de la estrategia híbrida y con el exterior. También determinan cómo se deriva la especificación METHOD de una técnica compuesta automáticamente a partir de las especificaciones METHOD de sus componentes, estableciéndose así un enfoque constructivo para la consecución de estrategias de DDFM.

Por otro lado, estas estrategias híbridas se aplican a **sesiones** de DDFM que se definen para resolver cierto problema de modelado y con ciertos objetivos. El estado de una sesión viene dado por la **instantánea** actual del proceso de DDFM que determina cuándo se han conseguido dichas metas. Además, esta instantánea proporciona el mecanismo mediante el que los distintos módulos de una estrategia compuesta comparten la información del proceso modelado.

Una vez dada esta visión general, en las siguientes secciones describimos cada uno de estos elementos.

3. UNIVERSO DE DISCURSO

Los algoritmos susceptibles de resolver alguna subtarea del DDFM se pueden considerar mecanismos que realizan transformaciones sobre un modelo de entrada obteniendo otro de salida. Estas transformaciones dan lugar a la existencia de varias representaciones para los

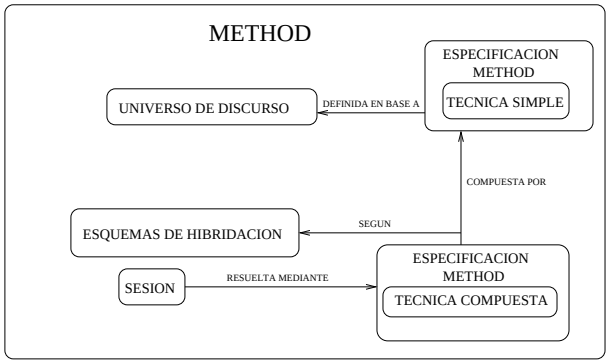


Figura 1: Visión general de METHOD.

modelos que constituyen una taxonomía. De hecho, un modelo es básicamente una forma de describir un *sistema* en base a *relaciones entrada-salida* que determinan su comportamiento. A partir de dichas relaciones y para una entrada dada, el modelo es capaz de inferir una salida simulando al sistema al que describe.

Las distintas representaciones de un modelo se diferencian en su *nivel organizativo*, es decir, en cómo se expresan las relaciones entrada-salida y en el modo en que se infieren las salidas a partir de las entradas. De hecho, nosotros vemos el proceso de DDFM como transformaciones sucesivas sobre el modelo, desde lo que consideramos su **fase más temprana como un conjunto de observaciones entrada-salida**, hasta obtener uno basado en reglas difusas y con las características deseadas. Esta idea se ilustra en la figura 2. La forma situada más a la izquierda, representa un modelo expresado como observaciones entrada-salida. A medida que el proceso de DDFM avanza, el modelo actual va "tomando forma" como consecuencia de la actuación de las técnicas que intervienen en la estrategia de DDFM empleada hasta llegar a ser un FIS (representado por el círculo).

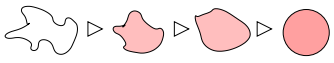


Figura 2: DDFM como transformaciones sucesivas de un modelo.

El estudio previo de los principales enfoques de DDFM nos ha encaminado a la identificación de varios tipos de modelos. Éstos se resumen en la tabla 1 y se organizan en una taxonomía.

Un conjunto de observaciones sobre un sistema objetivo se considera como un modelo de él. Cada punto del conjunto describe una relación entrada-salida. En cuanto a la capacidad de inferencia, un modelo de es-

Cuadro 1: Diferentes representaciones de un modelo

Modelo	Nivel Organizativo	Inferencia
observaciones	punto	sólo posible para ejemplos vistos
observaciones agrupadas en clusters difusos	definición por puntos	basada en distancias
FIS paramétrico	MFs paramétricas	FRM *a priori*
FIS y FRM paramétricos	MFs y operadores difusos paramétricos	FMs y FRM ajustables

te tipo solamente puede inferir salidas para entradas correspondientes a puntos del conjunto.

Un conjunto donde las observaciones están estructuradas en torno a clusters difusos en el espacio entrada-salida, se considera una definición por puntos de una relación entrada-salida. La salida inferida para una entrada dada es la suma ponderada de las salidas asociadas a cada cluster, donde los pesos son los grados de pertenencia de las distintas agrupaciones, los cuales dependen de la distancia a sus centroides.

En el siguiente nivel organizativo encontramos un FIS paramétrico en el que las relaciones entrada-salida son reglas $SI-ENTONCES$ con conjuntos difusos definidos como funciones de pertenencia (MF) paramétricas cuyos parámetros son ajustables. Su capacidad de inferencia depende de la interpretación concreta y *a priori* de los operadores difusos que intervienen en el mecanismo de razonamiento difuso FRM.

Finalmente un FIS puede venir dado en una forma totalmente paramétrica en la que tanto las MFs como el FRM son susceptibles de ajuste. De este modo, el FRM no tiene por qué ser seleccionado antes del proceso de DDFM sino que puede ser determinado durante el propio proceso mediante la optimización de los parámetros de sus operadores difusos.

Algunos algoritmos, además de obtener una nueva representación para un modelo, establecen características que el modelo tendrá en fases posteriores. Dichas "piezas", junto con la taxonomía de modelos constituyen el universo de discurso.

Un ejemplo de esta situación es el mecanismo de detección del número de reglas presentado en [7]. Obviando los detalles diremos que, este método, a partir de un conjunto de observaciones de entrada obtiene el número de reglas (relaciones entrada-salida) adecuado para el modelo. Es decir, además de obtener un modelo resultante (que es el mismo conjunto de observaciones inicial), descubre una característica que el modelo tendrá en posteriores fases, como es su número de reglas.

Vamos a denominar a estas "piezas" como *características discretas del modelo*. Esta idea se ilustra en la figura 3. En ella vemos como en el proceso de DDFM el modelo se transforma sucesivamente hasta obtener uno expresado en forma de FIS. Sin embargo, es posible que en cierto momento, se conozca alguna característica que tendrá el modelo en fases posteriores.



Figura 3: Proceso de DDFM en el que, además de transformar el modelo, se descubren características discretas

En consecuencia resulta de nuestro interés el poder "discretizar" los modelos en sus características discretas. De esta manera, **las técnicas se pueden definir en base a cuáles de estas características discretas son capaces de averiguar**, en lugar decir que resuelven cierta subtarea T procedente de un conjunto fijo de subtareas de DDFM (como selección de variables, identificación de estructura u optimización de parámetros). Esto proporciona mayor flexibilidad y un mayor rango de técnicas susceptibles de ser hibridadas.

Las características discretas forman una taxonomía teniendo en cuenta el tipo de modelo del que pueden ser extraídas. Las características extraíbles de cualquier modelo, sea del tipo que sea son: *el número de relaciones entrada-salida*, las propias *relaciones entrada-salida*, el *número de variables de entrada*, las propias *variables de entrada*, etc..

Sin embargo, para modelos particulares existen otras. Por ejemplo, para el tipo de modelo FIS se pueden extraer las siguientes características discretas: el tipo de función de pertenencia (MF) de los conjuntos difusos implicados, la interpretación de cada una de las conectivas, etc. Para un modelo basado en clusters difusos, podemos extraer el grado de borrosidad de la partición, el tipo de distancia utilizada, etc.. La importancia de las características discretas radica en que **constituyen el universo de discurso en base al que describir los requerimientos y efectos de las técnicas**.

4. EL ESTADO ACTUAL DEL PROCESO DE DDFM

Consideramos que, en un proceso de DDFM, toda la información asociada al modelo conseguido hasta el momento, constituye una unidad de información que condensa el "estado" del proceso de modelado. Supongamos que se lleva a cabo un proceso de DDFM. Supongamos también que podemos obtener instantáneas en distintos instantes del proceso. Estas instantáneas

reflejan información sobre cuál es el modelo conseguido hasta el momento, estructura, comportamiento actual (entendiendo con esto su capacidad aproximativa, de generalización, etc.), e incluso algunas propiedades que ya se han descubierto sobre el modelo en el futuro (cuál es el número adecuado de reglas para él, el número adecuado de variables, etc.). Esta información es la que llamamos *instantánea* del DDFM y la ilustramos en la figura 4.

Dado que nuestro objetivo fundamental es la formación de estrategias de modelado en las que intervengan varias técnicas, el propósito de la instantánea es servir como medio de transmisión de resultados entre los distintos componentes de la estrategia. De modo que la actuación de cada uno de ellos se refleja en la instantánea y los resultados parciales pueden ser utilizados como entradas de la siguiente componente de la estrategia.

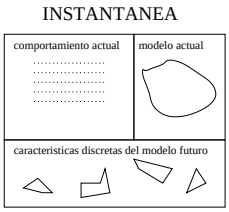


Figura 4: Concepto de instantánea del DDFM

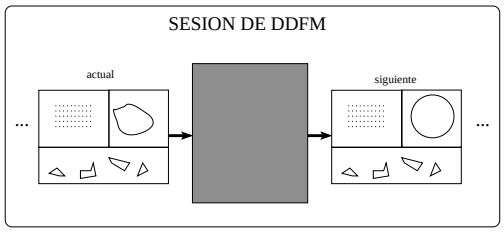


Figura 5: Nuestro enfoque modular de técnica de DDFM

.

Una sesión de DDFM contiene toda la información sobre el proceso de modelado: las observaciones entrada-salida del sistema a modelar, los objetivos (por ejemplo, cierta capacidad aproximativa mínima, cierto número de variables de entrada máximo, etc.). También contiene la instantánea actual, que determina si los objetivos han sido alcanzados o no. Desde la perspectiva de METHOD consideramos que el DDFM es el proceso mediante el cual, a partir de una sesión con una instantánea donde el modelo actual es simplemente un conjunto de observaciones sobre el sistema objetivo, se obtiene otra instantánea cuyo modelo cumple los requisitos especificados en los objetivos de la sesión.

5. ESPECIFICACIÓN METHOD DE UNA TÉCNICA

Consideramos que, desde el punto de vista de METHOD, un algoritmo susceptible de resolver alguna de las subtareas del DDFM, es básicamente una caja negra que "hace su parte" sobre una instantánea que toma como entrada, obteniendo otra como salida. Esta idea se ilustra en la figura 5.

Los algoritmos que intervienen en la estrategia de resolución de una sesión se pueden especificar en base al tipo de modelo y características discretas que requieren como entrada, así como al tipo de modelo y características discretas que obtienen como salida. Este "envoltorio" es lo que llamamos **especificación METHOD de una técnica** y constituye la base para poder manejar los algoritmos de DDFM como módulos para formar otros que resuelvan tareas de mayor nivel. Así, dado un algoritmo, esta especificación es todo lo que se necesita para saber si éste se puede utilizar en un momento dado del proceso de modelado. En lugar de decir que se trata de un algoritmo de selección de variables, o de generación de reglas, etc., queda definido mediante esos cuatro componentes como se muestra en la figura 6, los cuales, a su vez, se definen en base al universo de discurso de METHOD.

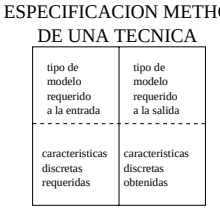


Figura 6: Resumen de la especificación METHOD de una técnica

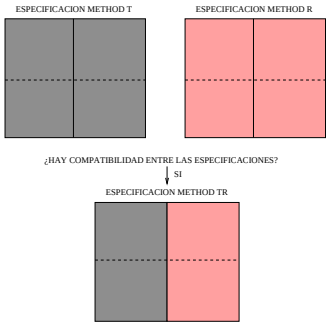


Figura 7: Las técnicas se pueden hibridar si se cumple el predicado de compatibilidad derivado del esquema de hibridación elegido.

Por otro lado, el algoritmo debe estar implementado de

forma tal que, dada una instantánea de entrada cuyo modelo actual cumple los requerimientos descritos en la especificación METHOD del algoritmo, obtenga una instantánea de salida coherente con los efectos que se describen también en la especificación.

6. HIBRIDACIÓN

Las técnicas de DDFM deben poder combinarse para resolver tareas de mayor nivel. Sin embargo, es necesario identificar los posibles modos en que interactúan los componentes de la combinación (*esquemas de interacción*). En nuestro contexto, un esquema de interacción establece:

- Las restricciones que las especificaciones METHOD de dos técnicas han de cumplir para que puedan combinarse.
- Cómo las capacidades de una técnica híbrida se definen automáticamente a partir de las capacidades de sus componentes. O dicho de otro modo, cómo se deriva la especificación METHOD de la estrategia compuesta a partir de las especificaciones METHOD de los componentes.
- A la hora de resolver una sesión de modelado, en qué orden actúan cada uno de las técnicas componentes sobre la instantánea actual.

Las dos primeras ideas se muestran en la figura 7.

En el trabajo [5] se identifican cuatro esquemas de interacción distintos para combinación neuro-simbólica. Extrapolando tales esquemas al contexto de DDFM, un análisis previo de métodos existentes en la literatura nos lleva a identificar, fundamentalmente, dos esquemas que se dan frecuentemente en las técnicas híbridas de DDFM. Dicho análisis se puede encontrar en [9].

En el **esquema de procesado en cadena** (figura 8 (a)), existe una técnica miembro que se encarga de resolver el problema, teniendo por tanto el rol *principal*. Además existe otra técnica (*filtro*) que actúa antes que el miembro principal. El rol filtro consiste en recoger la entrada desde el entorno, filtrarla y proporcionar la información necesaria al miembro que tiene el rol principal. Éste último se encarga de suministrar el resultado al entorno.

Como ejemplo, pongamos un módulo de selección de variables filtrando el conjunto de entrenamiento (es decir, el modelo temprano dado en forma de observaciones entrada-salida) para un subsiguiente componente que, a su vez, puede estar formado por una técnica de identificación de la estructura y otra que optimiza los parámetros. Este es el caso de la técnica propuesta en [4].

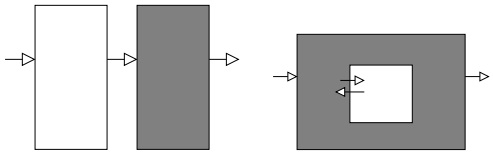


Figura 8: Esquemas de interacción: (a) en cadena; (b) por subprocesamiento.

El **esquema por subordinación** (figura 8 (b)) viene determinado por la existencia de un rol principal y un rol de subordinado. El miembro principal resuelve el problema mientras que el subordinado sólo actúa cómo y cuándo el principal lo requiere para resolver alguna subtarea que por sí solo el elemento principal no puede realizar. En cuanto a la relación con el entorno, es el miembro principal el que recoge las entradas y el que proporciona las salidas.

Consideramos que el esquema por subordinación se asemeja bastante a los enfoques llamados *wrapper* de selección de variables. El componente principal es un algoritmo generador de subconjuntos de variables candidatos. Para evaluar cada uno, el componente principal debe apoyarse en algún elemento subordinado de DDFM capaz de proporcionarle un modelo (basado en el subconjunto de variables candidado actual) cuya capacidad aproximativa le sirve para determinar la bondad del subconjunto. Este esquema se sigue, por ejemplo en [8].

Debemos puntualizar que los esquemas de interacción permiten que cualquiera de los miembros sea otra estrategia híbrida, dando lugar a varios niveles de anidamiento y, por tanto, a que se puedan combinar más de dos técnicas. Así, por ejemplo, podemos encadenar: (1) el algoritmo de clustering SCM propuesto en [3] que a partir de un modelo en forma de conjunto de observaciones obtiene otro en forma de observaciones agrupadas en clusters, (2) el mecanismo de optimización de clusters FCM propuesto en [1], que a partir de un modelo en forma de clusters obtiene otro optimizado, (3) el mecanismo de selección de variables propuesto por Chen y Linkens en [2], que a partir de un modelo de clusters obtiene un FIS proyectado en sus variables más relevantes y (4) la red neuro-difusa ANFIS [6] que optimiza los parámetros de un FIS. Esta combinación se muestra en la figura 9 y en [9] se muestra, tanto su factibilidad, como su especificación METHOD que se obtiene automáticamente a partir de las de sus componentes. No obstante, aquí podemos ver que existe coherencia entre los modelos de entrada y

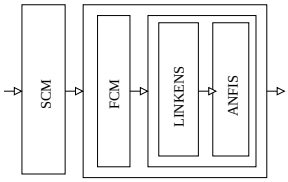


Figura 9: Ejemplo de estrategia híbrida basada en un esquema en cadena de tres niveles de anidamiento.

salida de componentes adyacentes y que esta estrategia híbrida es capaz de resolver el proceso completo de DDFM pues, a partir de observaciones entrada-salida obtiene un FIS.

7. CONCLUSIONES

El objetivo de este trabajo ha sido sentar las bases acerca del diseño de un marco para la especificación de técnicas híbridas de DDFM. Las dimensiones que conforman este marco son principalmente: (1) un mecanismo flexible para expresar las capacidades de las técnicas; (2) un universo de discurso que proporciona un lenguaje no ambiguo para describir estas capacidades; (3) unos esquemas de interacción en base a los cuales crear estrategias compuestas y que determinan, tanto cómo se derivan las capacidades de éstas automáticamente de las de sus componentes, como la factibilidad de las combinaciones; (4) una unidad de información (instantánea del DDFM) que sirve como medio de comunicación de resultados intermedios entre los miembros de una hibridación y que, a la vez, describe el estado actual de una sesión de modelado determinando cuándo se alcanzan los objetivos de ésta.

Los trabajos futuros se encaminan al diseño de una plataforma de agentes para DDFM ya que consideramos que las bases de conocimiento que guían el comportamiento de los agentes se pueden definir a partir de las especificaciones METHOD de las técnicas. Por un lado, los requerimentos y efectos que las forman, se pueden transformar en partes izquierdas y derechas de las reglas que dirigen a los agentes. Por otro, las instantáneas y sesiones de DDFM se pueden ver como hechos que disparan dichas reglas.

Por otra parte, hemos de remarcar que, si bien este marco se limita a la hibridación de técnicas para la obtencion de FISs, la misma filosofía empleada aquí se puede utilizar en base a otras formas de representación del conocimiento. Bastaría con extrapolar los conceptos de instantánea, sesión y esquemas de interacción al manejo de nuevos elementos en el universo de discurso.

AGRADECIMIENTOS

Este trabajo está financiado por el proyecto Fuzzy KIM (TIC2002-04021-C02-02) del Ministerio de Ciencia y Tecnología.

Referencias

[1] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum, New York, USA, 1981.

[2] M. Chen and D.A. Linkens. Input selection and partition validation for fuzzy modelling using neural network. *Fuzzy Sets and Systems*, 107(3):299–308, November 1999.

[3] S.L. Chiu. Fuzzy model identification based on cluster estimation. *Journal of the Intelligent and Fuzzy Systems*, 2(3):267–278, 1994.

[4] A. Gómez-Skarmeta and N. Bruno. Modelado difuso de sistemas aplicando soft computing. In *Congreso de la Asociación Española de Inteligencia Artificial*, pages 99–108, 1999.

[5] M. Hilario. *Connectionist-Symbolic Integration: From Unified to Hybrid Approaches*, chapter An Overview of Strategies for Neurosymbolic Integration. Kluwer Academic Publishers, 1996.

[6] J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing.* Matlab Curriculum. Prentice Hall, 1997.

[7] Y. Lin and G. Cunningham. A new approach to fuzzy-neural system modeling. *IEEE Transactions on Fuzzy Systems*, 3(2):191–198, May 1995.

[8] M. Sugeno and T. Yasukawa. A fuzzy logic based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems*, 1(1):7–31, August 1993.

[9] M. Valdés. *Definición de un Marco para la Especificación de Técnicas Híbridas para mOdelado Difuso.* PhD thesis, Departamento de Ingeniería de la Información y las Comunicaciones. Universidad de Murcia, España, Noviembre 2003.

[10] L. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions of Systems, Man and Cybernetics*, SMC-3:28–44, 1973.

Parámetros de control "dinámicos" en un Algoritmo Genético

J.A. Fernández Prieto
Departamento de Electrónica
Universidad de Jaén
C/ Alfonso X El Sabio, 28
23700 Linares (Jaén)
e-mail: jan@ujaen.es

Juan R. Velasco Pérez
Departamento Automática
Universidad de Alcalá de Henares
Campus Universitario
28871 Alcala de Henares (Madrid)
e-mail : juanra@aut.uah.es

Resumen

Uno de los problemas más importantes que existe a la hora de aplicar Algoritmos Genéticos es conocer los valores de los parámetros que usa. En distintos estudios se ha tratado de encontrar un conjunto de valores óptimos y "generales", es decir, un conjunto de valores que puedan ser aplicados a un gran número de problemas de optimización. Sin embargo, cada tipo de problema necesita de una configuración del algoritmo especifica para obtener un rendimiento satisfactorio. Además, distintos autores argumentan que estos valores no deben ser fijos durante la ejecución de un algoritmo sino que deben de variar en función del tiempo, deben ser "dinámicos".

En el presente documento se comparan los resultados obtenidos en un sistema borroso-genético que utiliza los valores de los parámetros fijos recomendados con los resultados obtenidos en ese mismo sistema haciendo uso, en este caso, de unos valores "dinámicos" previamente obtenidos con un Sistema Genético Adicional.

Palabras Clave: Algoritmos Genéticos, Controlador con Lógica Borrosa, parámetros de control.

1 INTRODUCCION

Cuando se diseña un Algoritmo Genético (AG) es necesario elegir sus componentes, los cuales son: una representación genética de las soluciones para el problema, un mecanismo para crear una población inicial de soluciones, una función de evaluación, operadores genéticos que alteren la composición genética de los nuevos individuos durante la reproducción y los valores de los parámetros que usa, fundamentalmente el tamaño de la población, la probabilidad de selección (P_s), probabilidad de entrecruzamiento (P_e) y la probabilidad de mutación (P_m). La elección de los valores de estos parámetros determinarán en gran medida que el algoritmo encuentre una solución óptima y de forma eficiente al problema dado.

La búsqueda de los valores de los parámetros que optimizan el comportamiento de un AG ha sido objeto de diferentes estudios (De Jong, 1975, Grefenstette 1986, Schaffer 1989, Bramlette 1991, Wu & Chow 1995; Eiben, Hinterding & Michalewicz 1999; Cicirello & Smith 2000). Uno de los primeros estudios fue realizado por De Jong [8]. Determinó experimentalmente que para una población comprendida entre 50 y 100 individuos los valores deberían ser: Pm=0.001 y Pe= 0.6

Posteriormente, Grefenstette [5] utilizó un AG como un meta-algoritmo para optimizar los valores obteniendo, para una población de 30 individuos, una Pm = 0.01 y una Pe = 0.95.

En 1989, el resultado obtenido por J.D. Schaffer [11] fue que, para una población comprendida entre 20 y 30 individuos, la Pm debe estar comprendida en el intervalo [0.005, 0.01] y la Pe en el intervalo [0.75, 0.95]

Por tanto, se ha tratado de encontrar un conjunto de valores óptimos y "generales", es decir, un conjunto de valores que puedan ser aplicados a un gran número de problemas de optimización. Los valores indicados anteriormente estaban basados en problemas de optimización de sencillas funciones y la codificación utilizada en el AG fue la binaria y probablemente su aplicación a otros tipos de problemas esté limitada.

En [3] se indica que cada tipo de problema necesita de una configuración del algoritmo especifica para obtener un rendimiento satisfactorio. Por tanto, un conjunto de valores óptimos para un AG no puede ser generalizado a todos los casos, no existiendo una regla general para elegir los apropiados. Por ello, la elección de dichos valores representa, en muchos de esos casos, un problema de prueba y error. Así, son necesarios métodos ó técnicas eficientes que ayuden a encontrar los valores óptimos para un problema dado.

Parámetros de control "dinámicos" en un Algoritmo Genético

J.A. Fernández Prieto
Departamento de Electrónica
Universidad de Jaén
C/ Alfonso X El Sabio, 28
23700 Linares (Jaén)
e-mail: jan@ujaen.es

Juan R. Velasco Pérez
Departamento Automática
Universidad de Alcalá de Henares
Campus Universitario
28871 Alcala de Henares (Madrid)
e-mail : juanra@aut.uah.es

Resumen

Uno de los problemas más importantes que existe a la hora de aplicar Algoritmos Genéticos es conocer los valores de los parámetros que usa. En distintos estudios se ha tratado de encontrar un conjunto de valores óptimos y "generales", es decir, un conjunto de valores que puedan ser aplicados a un gran número de problemas de optimización. Sin embargo, cada tipo de problema necesita de una configuración del algoritmo especifica para obtener un rendimiento satisfactorio. Además, distintos autores argumentan que estos valores no deben ser fijos durante la ejecución de un algoritmo sino que deben de variar en función del tiempo, deben ser "dinámicos".

En el presente documento se comparan los resultados obtenidos en un sistema borroso-genético que utiliza los valores de los parámetros fijos recomendados con los resultados obtenidos en ese mismo sistema haciendo uso, en este caso, de unos valores "dinámicos" previamente obtenidos con un Sistema Genético Adicional.

Palabras Clave: Algoritmos Genéticos, Controlador con Lógica Borrosa, parámetros de control.

1 INTRODUCCION

Cuando se diseña un Algoritmo Genético (AG) es necesario elegir sus componentes, los cuales son: una representación genética de las soluciones para el problema, un mecanismo para crear una población inicial de soluciones, una función de evaluación, operadores genéticos que alteren la composición genética de los nuevos individuos durante la reproducción y los valores de los parámetros que usa, fundamentalmente el tamaño de la población, la probabilidad de selección (P_s), probabilidad de entrecruzamiento (P_e) y la probabilidad de mutación (P_m). La elección de los valores de estos parámetros determinarán en gran medida que el algoritmo encuentre una solución óptima y de forma eficiente al problema dado.

La búsqueda de los valores de los parámetros que optimizan el comportamiento de un AG ha sido objeto de diferentes estudios (De Jong, 1975, Grefenstette 1986, Schaffer 1989, Bramlette 1991, Wu & Chow 1995; Eiben, Hinterding & Michalewicz 1999; Cicirello & Smith 2000). Uno de los primeros estudios fue realizado por De Jong [8]. Determinó experimentalmente que para una población comprendida entre 50 y 100 individuos los valores deberían ser: Pm=0.001 y Pe= 0.6

Posteriormente, Grefenstette [5] utilizó un AG como un meta-algoritmo para optimizar los valores obteniendo, para una población de 30 individuos, una Pm = 0.01 y una Pe = 0.95.

En 1989, el resultado obtenido por J.D. Schaffer [11] fue que, para una población comprendida entre 20 y 30 individuos, la Pm debe estar comprendida en el intervalo [0.005, 0.01] y la Pe en el intervalo [0.75, 0.95]

Por tanto, se ha tratado de encontrar un conjunto de valores óptimos y "generales", es decir, un conjunto de valores que puedan ser aplicados a un gran número de problemas de optimización. Los valores indicados anteriormente estaban basados en problemas de optimización de sencillas funciones y la codificación utilizada en el AG fue la binaria y probablemente su aplicación a otros tipos de problemas esté limitada.

En [3] se indica que cada tipo de problema necesita de una configuración del algoritmo especifica para obtener un rendimiento satisfactorio. Por tanto, un conjunto de valores óptimos para un AG no puede ser generalizado a todos los casos, no existiendo una regla general para elegir los apropiados. Por ello, la elección de dichos valores representa, en muchos de esos casos, un problema de prueba y error. Así, son necesarios métodos ó técnicas eficientes que ayuden a encontrar los valores óptimos para un problema dado.

Un enfoque que ha resultado efectivo en muchos contextos ha sido usar un AG como un meta-algoritmo para encontrar los valores. El gran inconveniente de este enfoque es su alto tiempo computacional.

Por otro lado, en [3], [4], [7], [10] se argumenta que la utilización de valores fijos durante la ejecución de un algoritmo es inapropiada, ya que se trata de un proceso intrínsecamente dinámico, adaptativo y por tanto el uso de parámetros fijos o "estáticos" que no cambian su valor en una ejecución va en contra de ese espíritu y puede conducir a un menor rendimiento del algoritmo.

La forma más sencilla de tratar este problema es usar parámetros que varíen en función del tiempo, es decir, cambiar un parámetro p por una función p(t), donde t es el contador de las iteraciones. Sin embargo, si encontrar los parámetros óptimos "estáticos" para un problema dado es una tarea difícil, no más fácil será encontrar una función p(t) .

En [2] Fogarty utilizó probabilidades de mutación variando en el tiempo, obteniendo un mayor rendimiento en el algoritmo (aunque no indica la función p(t) utilizada) . Hesser & Männer [10] determinaron la función $p_m(t) = \sqrt{\frac{\alpha}{\beta}} \frac{\exp(-\gamma \frac{t}{2})}{\lambda\sqrt{l}}$

donde α, β, y γ son constantes, λ es el tamaño de la población y l la longitud de los individuos.

En [3] se indica una función $p_m(t) = 1 - 0.9 * \frac{t}{T}$ donde t es el número de iteración y cuyo valor varía de 0 a T, el cual es el número máximo de iteraciones. Así, el valor de la probabilidad de la mutación se decrementará desde 1 (al comienzo de la ejecución) hasta 0.1.

En el presente documento se comparan los resultados obtenidos en un sistema borroso-genético (según el enfoque de Pittsburgh) que utiliza los valores de los parámetros fijos o "estáticos" recomendados por Grefenstette con los resultados obtenidos en ese mismo sistema haciendo uso, en este caso, de unos valores "dinámicos" previamente obtenidos con un meta-AG para el problema dado.

El documento está dividido en 6 secciones. En esta primera se ha realizado una introducción. En la segunda se describe brevemente el sistema genético utilizado. La tercera sección muestra la optimización de los valores de los parámetros. La cuarta describe la metodología seguida para realizar los experimentos. En la quinta se muestran los resultados obtenidos y por último, en la sexta se presentan las conclusiones.

2 SISTEMA GENETICO

En este apartado se describe brevemente el sistema genético (SG) ya que se encuentra descrito en [12]. La población del AG va a estar formada por 30 individuos, siendo cada uno de ellos una Base de Conocimiento (BC). Los individuos se han generado a partir de conocimiento humano, incorporándose éste a través de las reglas de actuación.

El sistema de evaluación utilizado es el descrito en [1]. Es un sistema de evaluación borroso, en el que los antecedentes son los estados del sistema a controlar y el consecuente es la evaluación instantánea del estado del sistema.

Durante la fase de reproducción se han utilizado los siguientes operadores:

- *Selección.* Se realiza la selección de un porcentaje de individuos de la población total, dado por la Probabilidad de Selección, P_S.

- *Entrecruzamiento.* El número de entrecruzamientos viene dado por la Probabilidad de Entrecruzamiento, P_E.

- *Mutación.* Existen tres tipos de mutaciones:

a) De puntos, que consiste en mover los puntos que definen un determinado conjunto borroso (CB). El número de mutaciones viene dado por la Probabilidad de Mutación de Puntos, P_{MP}.

b) De Conjuntos borrosos, que consiste en elegir un conjunto borroso de una variable que aparece en una regla de actuación de una base de conocimiento y se cambia por otro conjunto borroso de la misma variable. El número de mutaciones viene dado por la Probabilidad de Mutación de Conjuntos Borrosos, P_{MCB}.

c) De variables en las reglas, en donde se elige una variable del antecedente de una regla de actuación de una base de conocimiento y se cambia por otra variable de contexto. El número de mutaciones viene dado por la Probabilidad de Mutación de Variables, P_{MV}.

3 OPTIMIZACION DE LOS VALORES DE LOS PARAMETROS

El objetivo es encontrar los valores de los parámetros que optimizan el AG en un sistema borroso-genético, los cuales van a variar su valor en cada una de las iteraciones del algoritmo.

Vamos a utilizar una función p(t) que va a afectar a todos los valores de los parámetros del AG (P_S, P_E, P_{MP}, P_{CB} y $P_{MVU)}$ por igual. La función utilizada en nuestro caso es:

$$p(t) = p_0 * (1 - \frac{(Ln(t+1))^{1/a}}{(b \cdot Ln(T+1))^{1/a}}) \qquad a \in]0,2] \quad b \in [1,500]$$

donde p_0 es el valor inicial del parámetro, t es el número de iteración, cuyo valor varía de 0 a T, que es el número máximo de iteraciones. Además, existen 2 parámetros, a y b, que modelan la concavidad o convexidad de la curva y la atenuación del valor inicial respectivamente. La figura 1 muestra distintas curvas dependiendo de los valores de a y b para T=500 y $p_0 = 0.75$.

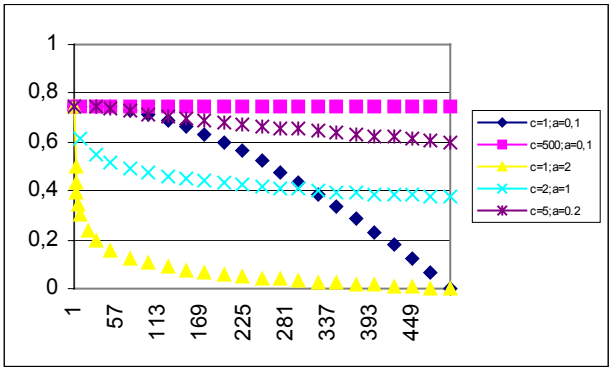


Figura 1. Efecto de la variación de los parámetros a y b.

Para la optimización de los valores de los parámetros del Sistema Genético así como de los valores de los parámetros a y b de la función p(t), se ha utilizado el esquema mostrado en la figura 2, en donde se ha añadido un Sistema Genético Adicional (SGA) [12] a un sistema de aprendizaje evolutivo de una base de conocimiento sobre un controlador borroso con enfoque de Pittsburg.

En la población del SGA, una solución candidata es CP_i (Conjunto de Probabilidades), i = 1,…,20, y representa el conjunto de valores que se aplican a los operadores genéticos que utiliza el SG junto con los parámetros a y b. Cada CP_i es un vector de números reales de la forma :

$$CP_i = \{P_S, P_E, P_{MP}, P_{CB}, P_{MVU,} a,b\}$$

Por tanto, representamos la población de 20 individuos por CP, y es un conjunto como :

$$CP = (CP_1,\ldots\ldots\ldots\ldots,CP_{20})$$

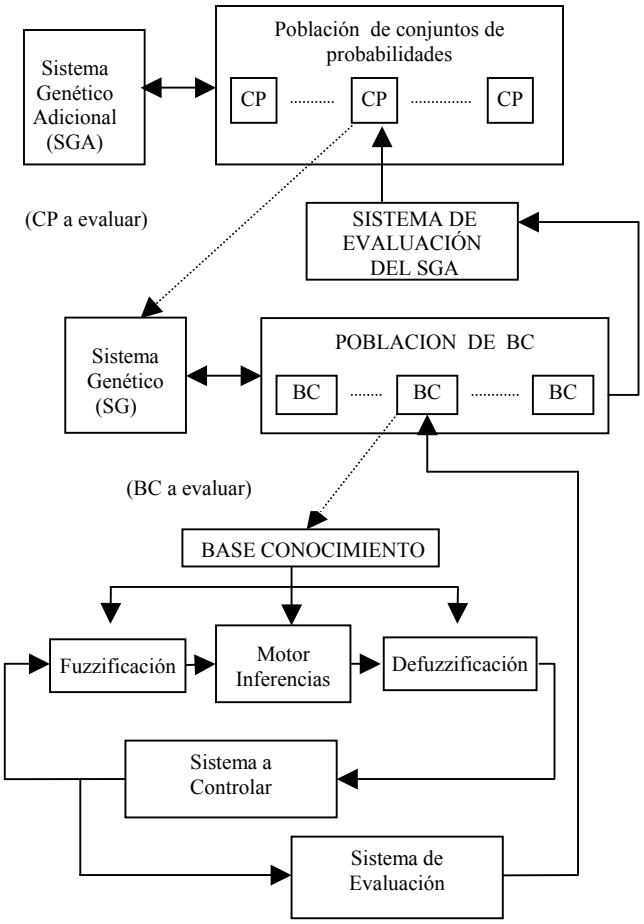


Figura 2: Aprendizaje evolutivo del Conjunto de Probabilidades de un sistema genético.

El proceso de aprendizaje de los valores de los parámetros del AG y de los valores de a y b consiste en :

1. Se parte de una población inicial CP(0), compuesta por CP_i (i = 1, … 20) individuos.
2. Evaluar la población inicial CP(0). Para evaluar cada CP_i de la población, se introduce en el SG y, utilizando sus valores, se realiza un proceso genético usando el enfoque de Pittsburg y trabajando sobre el controlador. Este proceso genético parte de la población inicial P(0) y tras n iteraciones obtiene la población P(n). Posteriormente, a partir de las evaluaciones obtenidas por cada BC de la población P(n), evaluamos CP_i.

   Para la evaluación de cada CP_i, el SG siempre parte de la misma población inicial, P(0).
3. Una vez evaluados todos los individuos de CP(x = 0), y mientras no se cumpla la condición de fin, se realiza:

(a) creación de una nueva población (CP(x+1)) aplicando los operadores genéticos del SGA a los individuos de CP(x)

(b) evaluación de la población CP(x+1). En la evaluación de cada CP_i de la población CP(x+1), el SG utiliza siempre la misma población inicial de BC, P(0).

(c) x = x+1

4. Stop.

4 METODOLOGÍA

La metodología que proponemos para realizar la comparación es la siguiente :

a) En primer lugar, se utilizará un sistema borroso-genético (como el resultante de eliminar el SGA en la figura 2), donde el sistema a controlar es un péndulo invertido, y se realizarán 35 experimentos utilizando los valores de los parámetros "estáticos" recomendados en [5]:

Tamaño de la población = 30
Probabilidad de entrecruzamiento = 0.95
Probabilidad de mutación de puntos = 0.01
Probabilidad mutación de conjuntos borrosos = 0.01
Probabilidad mutación variables en las reglas = 0.01

En cada uno de los experimentos, la población inicial de bases de conocimiento será la misma y el número de iteraciones en el AG será de 500. Al final de cada experimento se obtendrá una población diferente a la inicial, seleccionándose la BC que haya obtenido la mayor puntuación.

b) En segundo lugar, se añadirá el SGA. La población inicial de bases de conocimiento será idéntica a la utilizada anteriormente en a). El número de iteraciones del AG en el SG será de 500 y en el SGA de 50. Al final del experimento se obtendrá una población diferente a la inicial en el SGA, seleccionándose el CP_i que haya obtenido la mayor puntuación.

c) Por último, con el conjunto de probabilidades "dinámicas" (CP_i) obtenido en b) y utilizando el sistema borroso-genético de a) se volverán a realizar 35 experimentos.

De esta forma podemos comparar, en primer lugar, si las probabilidades obtenidas en b) ofrecen un mejor rendimiento en el algoritmo y en segundo lugar, se apreciará como es la variación temporal de los valores de los parámetros (si la atenuación debe ser elevada al principio, a mitad o al final de la ejecución o si la atenuación debe ser nula, con lo que los valores no tendrían una dependencia temporal).

5 RESULTADOS

Teniendo en cuenta la metodología indicada anteriormente, se han obtenido los siguientes resultados:

a) Utilizando los valores de los parámetros "estáticos" recomendados se han obtenido los siguientes resultados:

Nº experimentos	Media	Desviación Típica	Error tip. de la media
35	0,71511	0,026077	0,004408

La puntuación de la mejor BC obtenida ha sido de 0,755, mientras que la puntuación de la peor BC ha sido de 0,666.

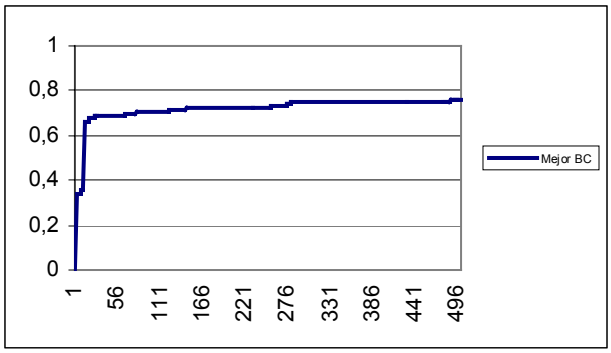


Figura 3. Evolución de la puntuación de la mejor BC.

b) Cuando se añadió el SGA, el Conjunto de Probabilidades "dinámicas" que obtuvo una mayor puntuación fue el siguiente:

Parámetro	Valor
P_S	0.8
P_E	0.49
P_{MP}	0.13
P_{MCB}	0.009
P_{MV}	0.016
a	0.097
b	6.95

Tabla 1. Mejor Conjunto de Probabilidades "dinámicas"

Por tanto, con los valores de a y b hallados, la función p(t) queda como:

$$p(t) = p_0 * (1 - \frac{(Ln(t+1))^{10.3}}{(6.95 \cdot Ln(T+1))^{10.3}})$$

En la figura 4 se muestra la forma de la curva p(t) (p.e., para p_o = 0.8 y T=500). Como puede apreciarse, los valores de los parámetros mantienen su valor inicial hasta aproximadamente la iteración 100, a partir de la cual sufren una moderada atenuación. Los valores finales de los parámetros (para T=500 iteraciones) son los mostrados en la tabla 2.

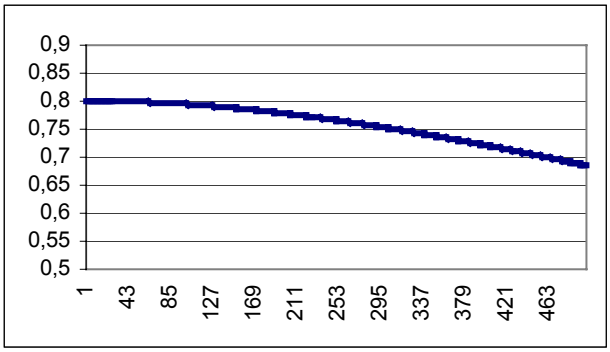


Figura 4. Representación de la función p(t) (p_o = 0.8 y T=500)

Parámetro	Valor final
P_S	0.685
P_E	0.419
P_{MP}	0.11
P_{MCB}	0.007
P_{MV}	0.013

Tabla 2. Valores finales de los parámetros para T=500

c) Utilizando las probabilidades "dinámicas" indicadas en el apartado anterior, los resultados obtenidos son los siguientes:

Nº experimentos	Media	Desviación Típica	Error tip. de la media
35	0,76897	0,031625	0,005346

La puntuación de la mejor BC obtenida ha sido de 0,826, mientras que la puntuación de la peor BC ha sido de 0,693.

En la figura 4 se muestra la evolución de la puntuación de la mejor BC obtenida.

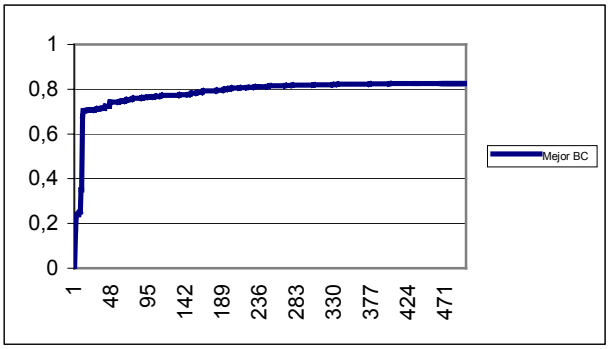


Figura 5. Evolución de la puntuación de la mejor BC.

Para comparar los resultados obtenidos se ha realizado un test estadístico, en concreto el test t-student y hemos utilizado el valor estándar de p-valor < 0.05 para la determinación de la significación. Se ha contrastado que la media obtenida con la utilización de los valores "dinámicos" es significativamente mayor que la media obtenida con los valores "estáticos" (p-valor < 0.001). Previamente se ha contrastado asimismo la normalidad de los datos mediante una prueba de Kolmogorov-Smirnoff (p-valor = 0.818) y su homocedasticidad (p-valor = 0.143).

Prueba de muestras independientes

		Prueba de Levene para la igualdad de varianzas		Prueba T para la igualdad de medias		
		F	Sig.	t	gl	Sig. (bilateral)
DATOS	Se han asumido varianzas iguales	2,198	,143	7,773	68	,000
	No se han asumido varianzas iguales			7,773	65,617	,000

6 CONCLUSIONES

A partir de los resultados obtenidos se pueden extraer las siguientes conclusiones:

1) Existe una dependencia del rendimiento de un AG con los valores de los parámetros que usa.

2) Al añadir el SGA se han encontrado unos determinados valores "dinámicos" de los parámetros, los cuales mantienen su valor inicial durante aproximadamente las primeras 100 iteraciones y posteriormente sufren una leve atenuación.

3) Con los valores "dinámicos" hallados con el SGA se ha obtenido un mejor rendimiento en el AG. En concreto, para los 35 experimentos realizados, el

aumento (en media) en el rendimiento ha sido del 5,38 %.

4) Todos los valores de los parámetros del AG se ven afectados por la función p(t) por igual, es decir, todos los valores van a variar en el tiempo de la misma forma. Sin embargo, no existe razón alguna para que esta variación sea la misma (en este caso se ha realizado así) y podría resultar interesante asignar a cada uno de los parámetros una función p(t) diferente. De esta forma se podría comprobar como es la evolución de cada parámetro de forma independiente.

Referencias

[1] J.C Bago and L. Magdalena. Knowledge Based Evaluation of Knowledge Bases. Eusflat-Estylf joint Conference, pages 83-86, Palma de Mallorca, September 1999

[2] T.C. Fogarty. Varying the probability of mutation in the genetic algorithm. In Proceeding of the Third International Conference on Genetic Algorithms and Their Applications, California, June 1989.

[3] A. E. Eiben, R. Hinterding, Z. Michalewicz. Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 3, pages 124-141, 1999

[4] T. Bäck. The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm. In Männer and Manderick [86], pages 85-94.

[5] J.J. Grefenstette. Optimization of control parameters for genetic algorithms. IEEE Transactions on Systems, Man and Cybernetics, 16, Nº 1, pages 122-128, 1986

[6] F. Herrera and L. Magdalena. Genetic Fuzzy Systems, Tatra Mountains Mathematical Publications Vol. 13, pages 93-121, 1997. R. Mesiar, B. Riecan (Eds.) Fuzzy Structures. Current Trends. Lecture Notes of the Tutorial: Genetic Fuzzy Systems. Seventh IFSA World Congress (IFSA97), Prage, June 1997.

[7] T. Bäck. Self-adaptation in genetic algorithms. In proceeding of the first European Conference on Artificial Life, December 11-13, Paris, France, 1991. (MIT Press, 1992)

[8] K. De Jong. An analysis of the behaviour of a class of genetic adaptive systems. PhD thesis, (University of Michigan, 1975)

[9] L. Magdalena and Juan R. Velasco. Fuzzy Rule-Based Controllers that Learn by Evolving their Knowledge Base. In F. Herrera and J.L. Verdegay, editors, Genetic Algorithms and Soft Computing. Physica-Verlag, pages 172-201, 1996.

[10] J.Hesser and R. Männer. Towards an optimal mutation probability in genetic algoritms. In H. P. Schwefel and R. Männer, eds., Parallel problem solving form nature, volume 496 of Lecture Notes in Computer Science, pages 23-32. (Springer, 1991)

[11] J.D. Schaffer, R.A. Caruana, L.J. Eshelman, and R. Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. Proceedings of the Third International Conference on Genetic Algorithms and Their Applications, San Mateo, California, pages 51-60, June 1989

[12] J.A Fernandez, Juan R. Velasco, Application of genetic algorithms in the research of the optimum probabilities of the genetic operators, 8th International Conference on Information Processing and Management of Uncertainty in knowledge-based systems, IPMU2000 pages 291-297, Madrid (Spain), July 2000

EXTRACCIÓN DE REGLAS DNF DIFUSAS EN UN PROBLEMA DE MARKETING

M. J. del Jesus, P. González
Departamento de Informática
Escuela Politécnica Superior
Universidad de Jaén
{mjjesus, pglez}@ujaen.es

F. Herrera
Dpto. de Ciencias de la Computación e IA
E.T.S.I. Informática
Universidad de Granada
herrera@decsai.ugr.es

M. Mesonero
Dpto. Organización y Marketing
Facultad Ciencias Empresariales
Mondragón Unibertsitatea
mmesoner@eteo.mondragon.edu

Resumen

En este trabajo se aborda la inducción descriptiva de reglas difusas en un problema de marketing mediante un enfoque evolutivo para extracción de reglas difusas en forma normal disyuntiva. Los resultados muestran que, pese a la dificultad del problema, se obtienen conjuntos de reglas con alto grado de interpretabilidad, de confianza, y para algunos valores de la variable objetivo, de completitud.

Palabras Clave: Minería de datos, inducción descriptiva, reglas difusas, algoritmos genéticos, lógica difusa.

1 INTRODUCCIÓN

La extracción de conocimiento se puede abordar desde dos perspectivas distintas: mediante un proceso de *inducción predictiva*, en el que se intenta obtener conocimiento para clasificación o predicción, o mediante un proceso *inducción descriptiva* cuyo objetivo fundamental es descubrir conocimiento de interés dentro de los datos. La inducción descriptiva se realiza bajo enfoques como la extracción de reglas de asociación [2], el descubrimiento de cláusulas [21], el descubrimiento de dependencias en bases de datos [8] o el descubrimiento de subgrupos [15][22][23], entre otros.

El problema del descubrimiento de subgrupos se puede definir de la siguiente forma: dado un conjunto de datos y una propiedad de los mismos que tiene interés para el usuario, encontrar (y describir) subgrupos relevantes respecto a esa propiedad o atributo. En un algoritmo de descubrimiento de subgrupos se extraen reglas o patrones de interés que representen de forma simbólica el conocimiento y que sean lo suficientemente sencillos y descriptivos como para ser reconocibles y utilizados por el usuario final.

Los Algoritmos Genéticos (AGs) [13][12] tienen un carácter de búsqueda global que hace que sean especialmente adecuados para resolver distintos problemas presentes en cualquier proceso de descubrimiento de conocimiento [9]. En procesos de extracción de reglas los AGs tratan de forma adecuada las interacciones entre atributos porque evalúan una regla como un todo mediante la función de adaptación en lugar de evaluar el impacto de añadir/eliminar una condición de una regla, como ocurre en los procesos de búsqueda local incluidos en la mayoría de los algoritmos de inducción de reglas y árboles de decisión.

En este trabajo se presenta un modelo basado en un AG híbrido para la obtención de un conjunto de reglas difusas que describen subgrupos. El algoritmo tiene capacidad para extraer reglas difusas o nítidas en función de que las variables del problema sean continuas o nominales y las reglas obtenidas tienen forma normal disyuntiva (DNF). La propuesta se aplica a un problema de extracción de conocimiento en certámenes feriales y se compara con otro modelo evolutivo de extracción de reglas difusas no DNF para estudiar las diferencias.

El artículo se organiza de la siguiente forma: En la Sección 2 se describe el problema de marketing a resolver y se realiza una breve revisión de algoritmos de inducción descriptivos de reglas difusas bajo enfoques no evolutivos y evolutivos. El modelo evolutivo para la extracción de reglas difusas DNF se describe en la Sección 3. En las Secciones 4 y 5 se muestra la experimentación realizada y las conclusiones obtenidas.

2 PRELIMINARES

2.1. EXTRACCIÓN DE CONOCIMIENTO EN CERTÁMENES FERIALES

En este trabajo se aborda un problema de marketing estudiado en el Departamento de Organización y

Marketing de la Universidad de Mondragón: La extracción de información útil sobre certámenes feriales.

Las empresas consideran los certámenes feriales un instrumento que facilita la consecución de objetivos comerciales tales como el contacto con los clientes actuales, la captación de nuevos clientes potenciales, la realización de pedidos o la mejora de la imagen corporativa entre otros [18]. Uno de los principales inconvenientes de este tipo de certámenes es la elevada inversión que suponen en términos tanto económicos como de tiempo. A esta inversión a veces se une una falta de planificación que enfatiza la sensación de que las ferias no son más que un "gasto" que las compañías han de afrontar por motivos varios (tradición, exigencia clientes, no dar la sensación de que las cosas van mal, etc.) [17]. Es conveniente, por tanto, la extracción automática de información sobre las variables implicadas que permita obtener datos desconocidos, determinantes en parte de la eficacia de los stands de un certamen.

En la Bienal de Máquina-Herramienta celebrada en Bilbao en Marzo de 2002, se recogió información sobre 104 variables de 228 expositores. De este conjunto de variables 7 de ellas son continuas y el resto son variables nominales, resultado de una discretización experta. Además, para cada uno de los expositores, en base a distintos criterios de marketing, se determinó la eficacia global de dicho stand en Eficacia Alta, Media o Baja en función del nivel de consecución de los objetivos planteados para el certamen.

El objetivo del proceso de extracción de conocimiento para este problema es determinar la aportación que las distintas variables de planificación ferial ejercen sobre los resultados obtenidos por el expositor.

2.2. INDUCCIÓN DESCRIPTIVA DE REGLAS DIFUSAS

La lógica difusa tiene una gran afinidad con la forma en que representamos el conocimiento humano; esto hace que su uso en reglas con variables cuantitativas facilite la interpretabilidad de las mismas, su diseño, la incorporación de conocimiento cualitativo sobre el problema, el tratamiento de valores perdidos y de clases con límites no muy bien definidos, así como el procesamiento del ruido en las variables que son resultado de medidas reales [24] [3]. En [4] se propone un algoritmo de extracción de reglas difusas de asociación con modificadores lingüísticos que hacen que el conocimiento extraído sea aún más comprensible y cercano al usuario, enriqueciendo la semántica de las reglas y haciendo que éstas sean más granulares.

Uno de los aspectos fundamentales cuando se trabaja con reglas difusas es la definición de las funciones de pertenencia asociadas a los conjuntos difusos utilizados. En el algoritmo de Kuok [16] para la extracción de reglas difusas es necesario que el experto aporte al algoritmo los conjuntos difusos de las variables continuas y sus correspondientes funciones de pertenencia. En este caso, la calidad de los resultados obtenidos por el algoritmo depende de la idoneidad de los conjuntos difusos. Para muchas aplicaciones es muy difícil saber a priori qué conjuntos difusos serán los más apropiados por lo que se han propuesto algoritmos como el descrito en [10] que determinan los conjuntos difusos y generan las funciones de pertenencia mediante técnicas de *clustering* o agrupamiento. No obstante, para incrementar la interpretabilidad de los resultados obtenidos en algunas propuestas como en [3] se introduce el conocimiento sobre el problema en la definición inicial de los conjuntos difusos, de forma que las reglas obtenidas estén definidas en base a esos conjuntos difusos. Nuestra propuesta se centra en esta línea.

Las reglas difusas utilizadas con mayor frecuencia representan en el antecedente una conjunción de proposiciones difusas o nítidas individuales (reglas canónicas). En problemas como el que abordamos aquí (en el que la mayoría de las variables son discretas y el número de valores perdidos es muy elevado), es adecuado adaptar el proceso para extraer reglas en forma normal disyuntiva, reglas DNF, en las que el antecedente está formado por una conjunción de atributos, cada uno de los cuales a su vez es una disyunción sobre los valores del atributo. Este tipo de reglas permiten representar el conocimiento de forma sencilla y comprensible, características básicas para el descubrimiento de subgrupos.

2.3. ALGORITMOS GENÉTICOS PARA EXTRACCIÓN DE REGLAS DIFUSAS

Se han desarrollado múltiples propuestas evolutivas para la extracción de reglas de distintos tipos, de clasificación, asociación o dependencias funcionales.

Con respecto al aspecto más determinante de cualquier AG, a la forma de codificar reglas dentro de una población de individuos, se siguen dos enfoques [6]:

- el enfoque "*Cromosoma = Regla*", en el que cada individuo codifica una única regla, y
- el enfoque "*Cromosoma = Base de Reglas*", o enfoque *Pittsburg*, en el que cada individuo codifica un conjunto de reglas.

A su vez, dentro del enfoque "*Cromosoma = Regla*" existen dos propuestas genéricas, el enfoque Michigan y el enfoque IRL (*Iterative Rule Learning*).

En [6] se puede encontrar una descripción completa de distintas propuestas bajo estos enfoques para la inducción evolutiva de reglas difusas.

3 UN ENFOQUE EVOLUTIVO PARA LA EXTRACCIÓN DE REGLAS DIFUSAS DNF DESCRIPTIVAS

Nuestra propuesta consiste en un modelo de extracción de reglas difusas DNF para la descripción de subgrupos. En él se distinguen dos componentes:

- Un AG híbrido para la extracción de una regla difusa DNF sencilla, interpretable y con un adecuado nivel de completitud y confianza.
- Un enfoque iterativo de extracción de reglas difusas DNF para la descripción de características de subgrupos soportados por zonas distintas (no necesariamente disjuntas) del espacio de ejemplo.

A continuación describimos ambos elementos.

3.1. MODELO DE EXTRACCIÓN DE REGLAS DIFUSAS DESCRIPTIVAS

El modelo de extracción de reglas difusas descriptivas sigue el enfoque IRL y su objetivo general es obtener para cada valor de la variable objetivo un número variable de reglas que expresen información sobre la mayoría de los ejemplos del conjunto de partida.

Para ello el AG híbrido para la generación de la mejor regla se incluye en un esquema iterativo que permite la obtención de varias reglas mientras las reglas generadas alcancen un nivel mínimo de confianza especificado previamente y describan información sobre zonas del espacio de búsqueda en las que aún quedan ejemplos no descritos por las reglas generadas en las iteraciones anteriores.

El esquema completo del algoritmo es el siguiente:

```
INICIO
  Cto_Reglas ← Ø
  REPETIR
    Ejecutar el AG obteniendo la regla R
    Búsqueda Local (R)
    CtoReglas ← CtoReglas + R
    Modificar el conjunto de ejemplos
  MIENTRAS confianza(R) >= confianza_min y
           R cubra ejemplos nuevos
FIN
```

El mecanismo de repetición potencia la generación de reglas distintas (en el sentido de que expresen información sobre grupos distintos de ejemplos). Esto se consigue penalizando -una vez obtenida una regla- el conjunto de ejemplos representados por la misma para la generación de futuras reglas. La penalización no impide la extracción de reglas solapadas que pueden ser de interés para el descubrimiento de subgrupos ya que descripciones redundantes de subgrupos pueden ser interesantes por describir propiedades de grupos de ejemplos desde diferente perspectiva.

Como se puede observar en el esquema del modelo de extracción, se exige en cada iteración que la confianza de la regla obtenida supere un valor mínimo especificado previamente. En los algoritmos de inducción descriptiva, uno de los problemas fundamentales y determinantes en parte de la calidad de los resultados obtenidos es la especificación del soporte mínimo (confianza mínima) exigido a las reglas a extraer. Este valor depende en gran medida del problema al resolver y su determinación es un problema aún no resuelto de forma completa para el que en [25] se describe un método basado en lógica difusa para la determinación del nivel mínimo de confianza.

3.2. CARACTERÍSTICAS DEL AG HÍBRIDO

Para la obtención de la mejor regla difusa DNF se utiliza un AG híbrido que tras la obtención evolutiva de la regla difusa aplica una etapa de post-procesamiento, un proceso de ascensión de colinas, para optimizar la regla obtenida con el objetivo de incrementar la completitud manteniendo el grado de confianza de la misma. Para ello, en cada iteración se determina la variable tal que, al eliminarla aumenta en mayor grado la completitud de la regla resultante, obteniendo así una regla más general. Finalmente, la regla optimizada sustituirá a la original sólo si supera la confianza mínima.

A continuación se describen los elementos del AG.

3.2.1. Esquema de representación

El AG tiene como objetivo descubrir reglas cuyo consecuente está formado por una variable definida a priori, es decir, el objetivo está prefijado. Para ello se codifica cada solución candidata mediante el enfoque "*Cromosoma = Regla*".

En el cromosoma sólo se representa el antecedente, con lo que todos los individuos de la población se asocian con el mismo valor de la variable objetivo. Esta forma de codificar la variable objetivo, utilizada en algoritmos de extracción de reglas de clasificación [14], obliga a realizar múltiples ejecuciones del algoritmo evolutivo para descubrir reglas de las distintas clases. En la bibliografía especializada se han considerado otras propuestas como la codificación dentro del genoma del individuo [7] o la elección determinística para cada regla del valor de la variable objetivo [11][19]. En problemas como el que presentamos, el enfoque adoptado es adecuado porque se necesita describir todos y cada uno de los valores de la variable objetivo, y las dos alternativas anteriores no aseguran la extracción de información sobre todas las clases.

Algunas de las variables del problema son variables continuas tratadas como variables lingüísticas con etiquetas lingüísticas. Los conjuntos difusos correspondientes a los términos lingüísticos vienen definidos por una partición difusa uniforme con funciones de pertenencia triangulares como la que se muestra en la Figura 1.

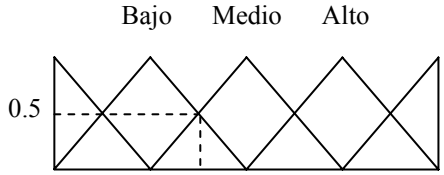


Figura 1: Ejemplo de partición difusa para una variable continua

En una regla DNF, el antecedente está compuesto por una conjunción entre atributos y varias disyunciones sobre los valores del atributo. Para representar este tipo de antecedente, se ha optado por un esquema de codificación binario, en el que por cada atributo se almacena un bit para cada uno de los valores que puede tomar, de forma que si el bit correspondiente tiene valor 0 indica que no pertenece a la condición y si tiene valor 1 que sí pertenece [7]. Si en un individuo todos los bits correspondientes a un atributo tienen valor 1 esto indica que dicha variable no es relevante para la información aportada en la regla (cualquier valor de la variable verifica la condición de la regla), por lo que esta variable se ignora. De esta forma, se almacena para cada variable una lista de enteros (tantos como valores posibles puede tomar la variable) que pueden tomar el valor 0 ó 1. En la última posición se guarda (a efectos de eficiencia) si la variable interviene o no en la regla (no interviene si están todos los valores, o no está ninguno).

Esto nos lleva a un esquema de codificación binaria con tantos genes por variable como valores posibles existan para la misma:

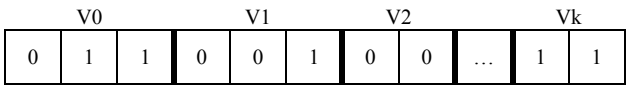


Figura 2: Codificación binaria de una regla DNF

El cromosoma descrito en la Figura 2 representa la regla:

> SI *Región* es *Centro* o *Levante* Y *Sector* es *Accesorios* Y …
>
> ENTONCES *Eficacia* es *Alta*

En el cromosoma anterior, no interviene la variable 2 (no tiene marcado ningún valor) ni la variable k (tiene todos los valores marcados, por lo que tampoco interviene).

3.2.2. Función de adaptación

En el proceso de descubrimiento de reglas se intentan conseguir reglas con capacidad predictiva alta, comprensibles e interesantes. Esto se puede alcanzar mediante una combinación lineal con pesos de estas tres medidas [19]. En nuestra propuesta, la función de adaptación tiene la expresión utilizada en [1] salvo que considera sólo dos factores, la confianza y completitud de la regla según la siguiente expresión:

$$fitness(c) = \frac{\omega_1 \cdot Completitud(c) + \omega_2 \cdot Confianza(c)}{\omega_1 + \omega_2}$$

- *Confianza.* Determina la precisión de la regla ya que refleja el grado con el que los ejemplos pertenecientes a la zona del espacio delimitado por el antecedente verifican la información indicada en el consecuente de la regla. Para el cálculo de este factor utilizamos una adaptación de la expresión de precisión aportada por Quinlan en [20] utilizada frecuentemente en la generación de reglas de clasificación difusas [5]: la suma del grado de pertenencia de los ejemplos de la clase a la zona determinada por el antecedente dividido entre la suma del grado de pertenencia de todos los ejemplos (independientemente de la clase a la que pertenezcan) a la misma zona. Para calcular estos grados de pertenencia se utilizan funciones de pertenencia triangulares y la t-norma mínimo. En el caso de reglas no difusas, los grados de pertenencia son los correspondientes a conjuntos clásicos, 0 ó 1.
- *Completitud*. Es una medida del grado de cobertura que la regla ofrece a los ejemplos de la clase. Se calcula como el cociente entre el número de nuevos ejemplos de la clase que cubre la regla y el número de ejemplos de la clase que quedaban por cubrir. Esta forma de medir la completitud toma sentido, al utilizar el AG dentro de un proceso iterativo, para potenciar la obtención de reglas distintas en cada ejecución del AG. A partir de la segunda iteración se penaliza indirectamente aquellas reglas que cubren ejemplos pertenecientes a las zonas delimitadas por reglas obtenidas previamente: el factor de completitud solo considera ejemplos no descritos por otras reglas ya obtenidas. No se utiliza ninguna función de distancia ya que se penalizan diferencias a nivel genotípico.

El objetivo global de la función de evaluación es orientar la búsqueda hacia reglas que maximicen la precisión, minimizando el número de ejemplos negativos y no cubiertos.

3.2.3. Esquema de reproducción y operadores

Se utiliza un esquema de reproducción de estado estacionario: la población original sólo se modifica mediante la sustitución de los peores individuos por los individuos resultantes de la recombinación (operadores de cruce y mutación).

La recombinación se realiza a través del operador de cruce multipunto y un operador de mutación uniforme sesgado con el que la mitad de las mutaciones realizadas tienen el efecto de eliminar la variable correspondiente, para incrementar la generalidad de las reglas.

4 EXPERIMENTACIÓN

La experimentación se ha realizado sobre los datos obtenidos en la Bienal de Máquina Herramienta celebrada en Bilbao en Marzo de 2002.

Para comprobar las características de nuestra propuesta DNF, se comparan sus resultados con los de un algoritmo evolutivo con las mismas características salvo que las reglas generadas tienen estructura canónica (no DNF).

La experimentación se ha realizado de la siguiente forma:

- Para ambos algoritmos se han utilizado poblaciones de 100 individuos, y 5000 evaluaciones.
- Se han ejecutado 3 veces ambos algoritmos para cada una de las 3 clases del atributo objetivo.
- Para la función de adaptación, se han utilizado como pesos asociados a los objetivos los siguientes: 0.5 para la completitud y 0.5 para la confianza.
- Se han elegido 3 etiquetas lingüísticas para las variables continuas.

En la Tabla 1 se muestran los resultados obtenidos con el algoritmo evolutivo de inducción descriptiva de reglas DNF y la versión con reglas no DNF. En la misma se muestra el nivel de confianza (soporte) mínimo exigido y la confianza y completitud de cada una de las reglas (ambos expresados como valores entre 0 y 100). Sobre ellos hay que destacar:

- Ambas propuestas permiten obtener conjuntos pequeños de reglas que describen cada valor del atributo objetivo.
- Los conjuntos de reglas tienen un nivel adecuado de confianza, aunque la completitud es un objetivo difícil de alcanzar en este problema para las clases 1 y 3.
- El algoritmo evolutivo de extracción de reglas DNF permite extraer más conocimiento de los datos y en ocasiones más general.
- El uso de reglas DNF facilita la extracción de reglas genéricas en problemas con variables nominales, permite flexibilizar la discretización realizada y cambiar la granularidad en el caso de variables tratadas como variables lingüísticas.
- El conocimiento extraído es interpretable por la estructura de la regla, que utiliza lógica difusa y permite seleccionar de forma local variables.
- No obstante, el bajo número de variables continuas de este problema limita el análisis de la propuesta aplicada a este problema.

Tabla 1: Resultados obtenidos con reglas DNF y no DNF.

Clase	Conf min	No DNF		DNF	
		Comp	Conf	Comp	Conf
1	0,8	7,895	100,000	5,263	100,000
2	0,6	22,297	68,750	6,757	100,000
		2,703	100,000	1,351	66,667
		9,459	73,684	1,351	100,000
		12,162	72,000	3,378	100,000
		0,676	100,000	6,757	90,909
		2,703	100,000	1,351	100,000
		6,081	81,818	1,351	66,667
		8,784	81,250	5,405	100,000
		66,216	67,586	2,027	100,000
				2,703	100,000
				3,378	100,000
				3,378	100,000
				7,432	100,000
				2,027	100,000
				1,351	100,000
				25,676	65,517
				2,703	100,000
				9,459	73,684
				62,162	68,657
3	0,6	4,762	100,000	7,143	100,000
		2,381	100,000		

5 CONCLUSIONES

En este trabajo se describe la aplicación de un modelo evolutivo para la inducción descriptiva de reglas difusas a un problema real de extracción de conocimiento en certámenes feriales.

Pese a las características del problema (elevado número de variables y valores perdidos, bajo número de ejemplos y pocas variables continuas) esta aproximación preliminar al problema permite obtener conjuntos de reglas fáciles de interpretar, con un nivel alto de confianza, y en el caso de la clase 2, un grado alto de soporte o completitud. Además, el análisis de la propuesta con reglas DNF y canónicas, permite observar un mejor comportamiento de la versión DNF para este problema.

Como trabajo futuro nos planteamos el estudio de una medida de interés adecuada para reglas DNF y el desarrollo de una versión evolutiva multiobjetivo que

considere los distintos objetivos presentes en un problema de extracción de reglas para descripción de subgrupos.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Ciencia y Tecnología y los fondos FEDER bajo los proyectos TIC-2002-04036-C05-01 y TIC-2002-04036-C05-04.

Referencias

[1] Aguilera, J.J., del Jesus, M.J., González, P., Herrera, F., Navío, M., Sáinz, J. "Extracción Evolutiva de Reglas de Asociación en un Servicio de Urgencias Psiquiátricas". Actas del II Congreso español sobre Metaheurísticas, Algoritmos evolutivos y bioinspirados. pp. 548-555. 2003.

[2] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, I. "Fast discovery of association rules", *Advances in Knowledge Discovery and Data Mining,* pp. 307-328, 1996.

[3] Au, W.H. & Chan, K.C.C. "An effective algorithm for discovering fuzzy rules in relational databases". *In Proceedings of IEEE International Conference on Fuzzy Systems* (Fuzz IEEE'98), pp.1314-1319, 1998.

[4] Chen, G. & Wei, Q. "Fuzzy association rules and the extended mining algorithms", *Information Sciences* 147, pp. 201-228, 2002.

[5] Cordón, O., del Jesus, M.J., Herrera, F., "Genetic Learning of Fuzzy Rule-based Classification Systems Co-operating with Fuzzy Reasoning Methods". International Journal of Intelligent Systems. Volumen 13. Número 10/11. pp. 1025-1053. 1998.

[6] Cordón, O., Herrera, F., Hoffmann, F., Magdalena, L. *Genetic Fuzzy Systems. Evolutionary tuning and learning of fuzzy knowledge bases*. World Scientific. 2001.

[7] De Jong, K.A., Spears, W.M., Gordon, D.F. "Using genetic algorithms for concept learning", Machine Learning. Volumen 13. pp. 161-188. 1993.

[8] Flach, P.A., Savnik, I. "Database dependency discovery: a machine learning approach", *AI Communications 12(3)*, pp. 139-160, 1999.

[9] Freitas, A.A., *Data Mining and Knowledge Disco–very with Evolutionary Algorithms*, Springer. 2002.

[10] Fu, A.W., Wong, M.H., Sze, S.C., Wong W.C., Wong W.L. & Yu, W.K. "Finding fuzzy sets for the mining of fuzzy association rules for numerical attributes". In *Proceedings of the First International Symposium on Intelligent Data Engineering and Learning* (IDEAL'98), pp. 263-268, 1998.

[11] Giordana, A., Neri, F. "Search-intensive concept induction". Evolutionary Computation. Volumen 3. Número 4. pp. 375-416.1995.

[12] Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley. 1989.

[13] Holland, J.H. *Adaptation in Natural and Artificial Systems.* University of Michigan Press. 1975.

[14] Janikow, C.Z. "A knowledge-intensive genetic algorithm for supervised learning". Machine Learning. Volumen 13. pp. 189-228. 1993.

[15] Klösgen, W. "Explora: a multipattern and multistrategy discovery assistant". In Fayyad, V., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (Eds.) *Advances in Knowledge Discovery and Data Mining,* MA: MIT Press. pp, 249-271. 1996

[16] Kuok, C., Fu, A., and Wong, M,. Mining fuzzy association rules in databases. *ACM SIGMOD Record*, 27: pp. 41–46, 1998.

[17] Miller, S. *Saque el máximo provecho de las ferias.* Traducido por Carlos Martinez Rueda, Ediciones Urano. 2003.

[18] Navarro, F. *Estrategias de Marketing Ferial*, ESIC Editorial. 2001

[19] Noda, E., Freitas, A.A., Lopes, H.S. "Discovering Interesting Prediction Rules with a Genetic Algorithm", in Proceedings of the Congress on Evo–lutionary Computation Vol. 2. pp. 1322-1329. 1999.

[20] Quinlan, J.R., *Generating production rules Machine Learning*. Morgan Kaufmann. 1987.

[21] Raedt, L. De, Dehaspe, L., « Clausal discovery", *Machine Learning 26*, pp. 99-146, 1997.

[22] Wrobel, S. "An algorithm for multi-relational discovery of subgroups". In *Proc. of the First European Conference on Principles of Data Mining and Knowledge Discovery,* pp. 78-87, 1997.

[23] Wrobel, S. "Inductive logic programming for knowledge discovery in databases". *Relational Data Mining,* pp. 47-101, 2001.

[24] Zadeh, L.A. "The concept of a linguistic variable and its applications to approximate reasoning, Parts I, II, III". Information Sciences Vol. 8-9. pp. 199-249, 301-357, 43-80, 1975.

[25] Zhang, S., Lu, J. Zhang, C., ''A fuzzy logic based method to acquire user threshold of minimum-support for mining association rules'', *Information ¨Sciences 164*: pp. 1-16. 2004.

APRENDIZAJE HÍBRIDO DE LA BASE DE CONOCIMIENTO DE UN SISTEMA BASADO EN REGLAS DIFUSAS MEDIANTE ALGORITMOS GENÉTICOS Y COLONIAS DE HORMIGAS

J. Casillas, O. Cordón, F. Herrera
Dept. Ciencias de la computación e inteligencia artificial
E.T.S. Ing. Informática
Universidad de Granada, 18071 - Granada
casillas,ocordon,herrera@decsai.ugr.es

P. Villar
Dept. Informática
E.S. de Ing. Informática
Universidade de Vigo, 32004 - Ourense
pvillar@uvigo.es

Resumen

En muchos ejemplos de modelado difuso lingüístico se hace necesario un aprendizaje automático de la base de conocimiento a partir de datos numéricos. En este trabajo, se propone un método para aprender la totalidad de la base de conocimiento (particiones difusas y reglas difusas), que combina un algoritmo genético y un algoritmo de optimización mediante colonias de hormigas. Estos dos mecanismos trabajan de forma coordinada para conseguir modelos con buen equilibrio entre precisión y interpretabilidad.

Palabras Clave: Aprendizaje automático, sistemas basados en reglas difusas, algoritmos genéticos, optimización mediante colonias de hormigas.

1. INTRODUCCIÓN

Actualmente, el modelado difuso lingüístico es una aplicación importante de los Sistemas Basados en Reglas Difusas (SBRDs). En este tipo de modelos, la interpretabilidad es un requisito fundamental, junto con la precisión. Para esta tarea, se utilizan los SBRDs Lingüísticos (SBRDLs), donde las reglas difusas toman valores de un conjunto de términos con una interpretación en el mundo real (variables lingüísticas).

La Base de Conocimiento (BC) de un SBRDL está formada por dos componentes: 1) la Base de Reglas (BR), constituida por el conjunto de reglas difusas, y 2) la Base de Datos (BD), que incluye las funciones de pertenencia de cada uno de los términos lingüísticos (o etiquetas) de las particiones difusas asociadas a las variables. La BC de un SBRDL depende del problema que se intente resolver, por tanto, se hace necesario algún mecanismo de derivación automática de dicha BC. Dentro de la literatura especializada existen numerosos métodos de aprendizaje de la BR a partir de una BD previamente definida.En la mayoría de esos métodos, se consideran particiones difusas uniformes con el mismo número de etiquetas para todas las variables. Sin embargo, la elección de la BD tiene una decisiva influencia en el comportamiento del SBRDL resultante[5].

Teniendo en cuenta la fuerte interrelación entre los dos componentes de la BC, sería muy aconsejable un mayor grado de cooperación entre las tareas de aprendizaje de la BR y la BD, que permitiese obtener modelos con un buen equilibrio entre interpretabilidad y precisión. Basándonos en esta idea, se propone un método de aprendizaje de la BC que mantiene una población de posibles definiciones de la BD al mismo tiempo que busca un conjunto compacto de reglas (BR). Más concretamente, para la derivación de las reglas difusas utilizaremos un mecanismo basado en la metodología COR junto con un algoritmo de optimización mediante colonias de hormigas (proceso COR-OCH). Para la definición de la BD se empleará un Algoritmo Genético que incluye sus principales componentes para cada variable: dominio, número de etiquetas y partición difusa. En este documento, describiremos primero el mecanismo de aprendizaje COR-OCH. A continuación, se presenta el método propuesto y finalmente, se muestran unos resultados experimentales y algunas conclusiones.

2. PROCESO DE APRENDIZAJE COR-OCH

El proceso seguido para integrar el aprendizaje de la base de reglas dentro de un meta-algoritmo de diseño de la base de datos debe ser rápido y eficaz. Para ello, basaremos el aprendizaje en la metodología COR (Cooperación entre Reglas) propuesta en [1] y exten-

[0]Trabajo parcialmente soportado por el Ministerio de Ciencia y Tecnología en el marco de los proyectos TIC2002-04036-C05-01, TIC2003-00877 (con fondos FEDER) y de la Red HEUR TIC2002-10866-E.

dida en [2] usando un algoritmo de optimización mediante colonias de hormigas (OCH) [7]. La reducción del espacio de búsqueda e inducción de cooperación entre reglas realizada por la metodología COR, junto con el uso de información heurística y de una búsqueda eficaz de la técnica OCH, permiten que el algoritmo propuesto sea rápido y genere bases de reglas con buena precisión y alta interpretabilidad. Las siguientes dos subsecciones describen la metodología de aprendizaje COR y el algoritmo de OCH basado en ella.

2.1. Metodología COR

En los últimos años se ha propuesto en la literatura especializada una familia de métodos simples y eficientes de extracción de reglas difusas guiados por criterios de cobertura de los datos del conjunto de ejemplos. Su simplicidad, además de su velocidad y fácil comprensión, hace a estos métodos muy adecuados para tareas de aprendizaje. Sin embargo, estos métodos generalmente buscan reglas difusas con el mejor comportamiento individual (por ejemplo, [11]) y, por tanto, no se considera la interacción global entre las reglas debida al razonamiento interpolado realizado por los sistemas difusos, provocando así una mala precisión en algunos casos.

Con la intención de paliar estos inconvenientes manteniendo las interesantes ventajas de estos métodos, en [1] se propone la metodología COR. En lugar de seleccionar el consecuente con el mejor comportamiento en cada subespacio, esta metodología considera la posibilidad de usar otro consecuente, diferente del mejor, si así se mejora el comportamiento global del sistema difuso gracias a tener una base de reglas con mejor cooperación.

COR se compone de los siguientes pasos: (1) *construcción del espacio de búsqueda*, se obtiene un conjunto de consecuentes candidatos para cada regla; y (2) *selección del conjunto de reglas más cooperativo*, se realiza una búsqueda combinatoria entre los conjuntos para encontrar la combinación de consecuentes con el mejor comportamiento global. En la figura 1 se muestra una descripción más detallada del proceso de generación de reglas basado en COR.

2.2. Algoritmo de OCH basado en COR

Dado que el espacio de búsqueda manejado en el paso 2. es grande, resulta necesario usar técnicas de búsqueda aproximadas. En [1] se obtuvieron modelos lingüísticos con buena precisión utilizando enfriamiento simulado. Sin embargo, dado que una de las restricciones de nuestro problema es que el proceso de aprendizaje debe ser rápido, en este artículo se propone el uso de OCH [7]. Esta es una técnica de búsqueda bio-inspirada que considera información heurística para permitir obtener buenas soluciones con rapidez. Esta sección describe los componentes del algoritmo propuesto.

Representación del problema Para aplicar OCH en la metodología COR es conveniente verla como un problema de optimización combinatoria con la capacidad de ser representado en un grafo ponderado. De esta forma, se puede afrontar el problema considerando un número fijo de subespacios e interpretando el proceso de aprendizaje como la forma de asignar vectores de consecuentes —es decir, etiquetas de las particiones difusas de las variables de salida— a estos subespacios según un criterio de optimalidad (esto es, siguiendo la metodología COR).

Por tanto, de acuerdo con la figura 1, cada nodo $S_h \in S^+$ se asigna a cada consecuente candidato $(B_1^{k_h}, \ldots, B_m^{k_h}) \in C(S_h)$ y al símbolo especial "no considerar" $(R_\emptyset)$ que indica la ausencia de reglas en el correspondiente subespacio.

Información heurística La información heurística sobre la preferencia potencial de elegir un vector de consecuentes específico, B^{k_h}, en cada combinación de antecedentes (subespacio) se determina de la siguiente forma para $B^{k_h} = (B_1^{k_h}, \ldots, B_m^{k_h}) \in C(S_h)$:

$$\eta_{hk_h} = \max_{e_l \in E^+(S_h)} Min\left(\mu_{A^h}(x^l), \mu_{B_j^{k_h}}(y^l)\right).$$

Si $B^{k_h} \notin C(S_h)$, hacer $\eta_{hk_h} = 0$.

Posteriormente, para el símbolo "no considerar", hacer lo siguiente:

$$\eta_{h,|\mathcal{B}_1|\cdot\ldots\cdot|\mathcal{B}_m|+1} = \frac{1}{\max\limits_{k_h \in \{1,\ldots,|C(S_h)|\}} \eta_{hk_h}}.$$

Inicialización de la feromona El valor inicial de la feromona para cada asignación se obtiene de la siguiente forma:

$$\tau_0 = \frac{1}{|S^+|} \sum_{S_h \in S^+} \max_{B^{k_h} \in C(S_h)} \eta_{hk_h}.$$

Así, la feromona inicial será el valor medio del camino construido tomando el mejor consecuente en cada regla de acuerdo con la información heurística (se trata, por tanto, de una asignación voraz).

Esquema de OCH: sistema de la mejor-peor hormiga Una vez diseñados los componentes previos, se debe definir un algoritmo OCH para resolver el problema. En esta contribución consideramos el algoritmo Sistema de la Mejor-Peor Hormiga (SMPH) [3], que ha demostrado tener un buen comportamiento en varios problema de optimización. Su esquema global se muestra en la figura 2, para más detalle refiérase a [3].

Entradas:

- *Un conjunto de datos entrada-salida* —$E = \{e_1, \ldots, e_l, \ldots, e_N\}$, donde $e_l = (x_1^l, \ldots, x_n^l, y_1^l, \ldots, y_m^l)$, $l \in \{1, \ldots, N\}$, siendo N el tamaño del conjunto de datos, y n (m) el número de variables de entrada (salida)— *que represente el comportamiento del problema a resolver.*
- *Una partición difusa de los espacios de las variables.* En nuestro caso, consideramos conjuntos difusos con distribución equidistante. Sea $\mathcal{A}_i$ el conjunto de términos lingüísticos de la i-ésima variable de entrada, siendo $i \in \{1, \ldots, n\}$, y $\mathcal{B}_j$ el conjunto de términos lingüísticos de la j-ésima variable de salida, siendo $j \in \{1, \ldots, m\}$, donde $|\mathcal{A}_i|$ ($|\mathcal{B}_j|$) es el número de etiquetas de la i-ésima (j-ésima) variable de entrada (salida).

Algoritmo:

1. *Construcción del espacio de búsqueda*:

   1.1. *Definir los subespacios de entrada difusos que contengan ejemplos positivos*: Para ello, debemos definir el conjunto de ejemplos positivos ($E^+(S_s)$) para cada subespacio de entrada difuso $S_s = (A_1^s, \ldots, A_i^s, \ldots, A_n^s)$, donde $A_i^s \in \mathcal{A}_i$ es una etiqueta, $s \in \{1, \ldots, N_S\}$, y $N_S = \prod_{i=1}^{n} |\mathcal{A}_i|$ es el número de subespacios de entrada difusos. En este artículo realizamos lo siguiente:

   $$E^+(S_s) = \{e_l \in E \ tal \ que \ \forall i \in \{1, \ldots, n\}, \forall A_i' \in \mathcal{A}_i, \ \mu_{A_i^s}(x_i^l) \geq \mu_{A_i'}(x_i^l)\},$$

   siendo $\mu_{A_i^s}(\cdot)$ la función de pertenencia asociada a la etiqueta A_i^s.
   Entre todos los N_S subespacios de entrada difusos posibles, considerar sólo aquellos que contengan al menos un ejemplo. Para ello, el conjunto de subespacios con ejemplos positivos se define como $S^+ = \{S_h \mid E^+(S_h) \neq \emptyset\}$.

   1.2. *Generar el conjunto de reglas candidatas en cada subespacio con ejemplos positivos*: En primer lugar se define el conjunto de consecuentes candidatos asociado con cada subespacio que contenga al menos un ejemplo, $S_h \in S^+$. En este artículo realizamos lo siguiente:

   $$C(S_h) = \{ \quad (B_1^{k_h}, \ldots, B_m^{k_h}) \in \mathcal{B}_1 \times \ldots \times \mathcal{B}_m \ tal \ que \\ \exists e_l \in E^+(S_h) \ donde \ \forall j \in \{1, \ldots, m\}, \forall B_j' \in \mathcal{B}_j, \ \mu_{B_j^{k_h}}(y_j^l) \geq \mu_{B_j'}(y_j^l) \ \ \}.$$

   A continuación se define el conjunto de reglas candidatas para cada subespacio como $CR(S_h) = \{R_{k_h} = [\text{SI } X_1 \text{ es } A_1^h \text{ y ... y } X_n \text{ es } A_n^h \text{ ENTONCES } Y_1 \text{ es } B_1^{k_h} \text{ y ... y } Y_m \text{ es } B_m^{k_h}] \ tal \ que \ (B_1^{k_h}, \ldots, B_m^{k_h}) \in C(S_h)\}$.
   Para permitir a la metodología COR reducir el número inicial de reglas difusas, se añade el elemento especial $R_\emptyset$ (que significa "no considerar") a cada conjunto de reglas candidatas, es decir, $CR(S_h) = CR(S_h) \cup R_\emptyset$. Si se selecciona este elemento, no se generará ninguna regla en el correspondiente subespacio de entrada difuso.

2. *Selección del conjunto de reglas difusas más cooperativo* — Esta fase se realiza ejecutando un algoritmo de búsqueda combinatoria para encontrar la combinación $RB = \{R_1 \in CR(S_1), \ldots, R_h \in CR(S_h), \ldots, R_{|S^+|} \in CR(S_{|S^+|})\}$ con el mejor comportamiento. Dado que el espacio de búsqueda manejado suele ser grande, se deben usar técnicas de búsqueda aproximadas.

   Se considera un índice $f(RB)$ para medir la calidad global del conjunto de reglas codificada en cada solución. En nuestro caso, el algoritmo usa la función de error global basada en el *error cuadrático medio* (ECM), definido como

   $$\text{ECM} = \frac{1}{2 \cdot N} \sum_{l=1}^{N} (\mathbf{F}(x_1^l, \ldots, x_n^l) - \mathbf{y}^l)^2,$$

   donde $\mathbf{F}(x_1^l, \ldots, x_n^l)$ es el vector de salida obtenido a partir del SBRD cuando se usa el ejemplo e_l como entrada, y $\mathbf{y}^l$ es el vector de salida deseado. Cuanto más cercano a cero sea la medida, mejor comportamiento global y, por tanto, mejor cooperación entre reglas.

Figura 1: Algoritmo COR

3. MÉTODO DE APRENDIZAJE DE LA BC DE UN SBRDL

En este apartado describiremos nuestra propuesta para el aprendizaje automático de la BC de un SBRDL, compuesta por dos mecanismos:

- Un proceso de aprendizaje de la BD mediante un Algoritmo Genético (AG) que nos va a definir:
  - Dominio de cada variable que interviene en el modelo.
  - Número de etiquetas de cada variable.
  - Semántica de cada etiqueta considerando particiones difusas no uniformes. Para cons-

1. Dar un valor inicial de la feromona, τ_0, para cada arco.
2. Mientras (*condición_terminación* no se cumpla) hacer:
   - a) Realizar el camino de cada hormiga mediante el **proceso de construcción de la solución**.
   - b) Aplicar el **mecanismo de evaporación de feromona**.
   - c) Aplicar el **proceso de búsqueda local** sobre la mejor solución actual.
   - d) Actualizar $S_{mejor\ global}$ y $S_{peor\ actual}$.
   - e) Aplicar la **regla de actualización de feromona Mejor-Peor**.
   - f) Aplicar la **mutación de feromona**.
   - g) Si (*condición_estancado* se cumple), aplicar **reinicialización**.

Figura 2: Algoritmo SMPH

truir dichas particiones se emplea una función de escala no lineal que define áreas con mayor o menor "sensibilidad".

- el algoritmo e aprendizaje de reglas COR-OCH, que deriva la BR tomando como punto de partida la BD previamente obtenida en el paso anterior. De esta manera, el método híbrido propuesto obtiene una definición completa de la BC mediante la acción cooperativa de ambos procesos.

Ya que todos los componentes de la BD irán evolucionando mediante un AG, sería aconsejable reducir la dimensión del espacio de búsqueda. Por tanto, el uso de una función de escala no lineal debería estar condicionado a escoger una función parametrizada con un reducido número de parámetros. Para este trabajo, hemos considerado la función propuesta en [8], que solamente tiene un parámetro denominado a ($a \in \mathbb{R}$). La función es: $f : [-1, 1] \rightarrow [-1, 1]$

$$f(x) = sign(x) \cdot |x|^a\,, \quad con\ \ a > 0$$

El resultado final es un valor entre $[-1, 1]$. La acción del parámetro a en la partición difusa depende de su valor: sensibilidad uniforme ($a = 1$), mayor sensibilidad en los valores centrales ($a > 1$), o mayor sensibilidad en los valores extremos ($a < 1$). En este trabajo se han considerado funciones de pertenencia triangulares, por tanto, la función de escala no lineal sólo se aplica a los tres puntos que definen la función de pertenencia. Es importante destacar que esa función produce efectos simétricos alrededor del punto central del dominio. Con una pequeña modificación se puede generar mayor sensibilidad únicamente en uno de los extremos (el proceso completo de construcción de la partición difusa se puede consultar en [4]). La figura 3 muestra una representación gráfica de esas cinco posibilidades para la partición difusa.

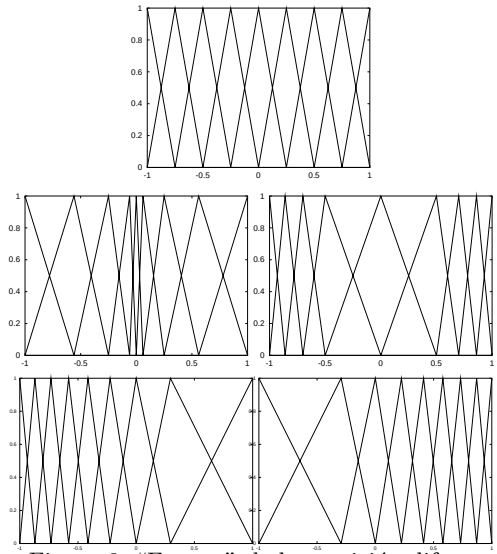


Figura 3: "Formas" de la partición difusa

A continuación, se describirá el proceso evolutivo que monitoriza el aprendizaje de la BC. Los AGs[9] son técnicas de optimización y búsqueda basadas en una formalización del proceso evolutivo de los seres vivos. Cuando se trabaja con AGs, los aspectos importantes son cómo codificar cada solución (en este caso, la BD de un SBRDL), como evaluar esa solución y como crear nuevas soluciones:

Codificación de la BD. Para un modelo con N variables, cada individuo o cromosoma está compuesto de cuatro partes:

- Número de etiquetas (C_1): codificados en un vector de valores enteros de longitud N. Para este trabajo, el rango considerado es $E \in \{2, \ldots, 7\}$.
- Parámetros de "forma": (C_2): codificados en un vector de enteros de longitud N, que puede tomar los valores $S \in \{0, 1, 2, 3, 4\}$, que representan las cinco posibilidades mostradas en la figura 3.
- Parámetros de sensibilidad (C_3): codificados en un vector de valores reales de longitud N, donde se almacena el valor del parámetro (a) de la función de escala no lineal para cada variable. El rango considerado es el intervalo $(0, 10)$.
- Dominios (C_4): codificados en un vector de $N \times 2$ valores reales que almacenan el dominio de cada variable ($[v_{min}, v_{max}]$). Si el dominio inicial de una variable es $[v^i_{inf}, v^i_{sup}]$, y d es la dimensión del intervalo ($d = v^i_{sup} - v^i_{inf}$), el rango para el límite inferior del dominio es $[v^i_{inf} - (1/4 * d), v^i_{inf}]$, y el rango para el límite superior es $[v^i_{sup}, v^i_{sup} + (1/4 * d)]$.

Una representación gráfica del cromosoma sería:

$$C_1 = (E_1, \ldots, E_N) \qquad C_2 = (S_1, \ldots, S_N)$$

$$C_3 = (a_1, \ldots, a_N) \quad C_4 = (v^1_{min}, v^1_{max}, \ldots, v^N_{min}, v^N_{max})$$

Evaluación de los cromosomas. El cálculo de la función de evaluación consta de tres pasos:

- Construir las particiones difusas de cada variable, utilizando la información contenida en el cromosoma. Primero, el dominio de cada variable (C_4) es normalizado al intervalo $[-1, 1]$. A continuación, se crea una partición difusa uniforme considerando su número de etiquetas (C_1). Después, la función de escala no lineal, con su "forma" (C_2) y su parámetro de sensibilidad (C_3) se aplica a los tres puntos que definen cada etiqueta. Finalmente, la partición es trasladada de nuevo a su dominio original (C_4).
- Generar la BR mediante COR-OCH, considerando la BD obtenida en el paso previo.
- Calcular el error cuadrático medio (ECM) sobre el conjunto de datos de entrenamiento usando la BC obtenida (BD + BR). Para mejorar la capacidad de generalización del SBRD resultante, intentando evitar el posible sobreaprendizaje, penalizaremos los SBRDLs con un número de reglas (NR) excesivo. Así, la función de evaluación es:

  $$F = \omega_1 \cdot ECM + \omega_2 \cdot NR$$

  siendo ω_1 y ω_2 porcentajes de peso de cada componente ($\omega_1 + \omega_2 = 1$). Para los experimentos, hemos considerado $\omega_1 = 0{,}9$.

Operadores genéticos. Como operador de selección, hemos utilizado el muestreo universal estocástico. Respecto a los operadores de recombinación genética, y dada la especial estructura de los cromosomas, con cuatro partes fuertemente relacionadas, es necesario diseñar operadores específicos que actúen de forma coordinada. Se han utilizado dos operadores de mutación, uno para codificación entera (C_1 y C_2) y otro para codificación real (C_3 y C_4). para el primer caso, se emplea el esquema propuesto por Thrift en [10] y para el segundo se utiliza el operador de mutación no uniforme de Michalewicz [9].

Respecto al operador de cruce, se emplean dos procesos diferentes dependiendo de las características de los cromosomas implicados. Si ambos progenitores tienen los mismos valores en C_1 y C_2 (para todas las variables), el AG ha localizado una zona del espacio de búsqueda que debe ser explotada. Para eso, aplicamos el operador de cruce *parent centric* [6] en C_3 y C_4 manteniendo los valores de C_1 y C_2 en los descendientes. En el otro caso, se utiliza un operador de cruce clásico en un punto sobre las cuatro partes del cromosoma. Es decir, se selecciona aleatoriamente un punto de cruce p ($p \in \{1, \ldots, N\}$) y los dos padres se cruzan por la variable p en C_1, C_2, C_3 y C_4, produciendo dos descendientes que mantienen la correlación que había entre los distintos componentes de cada variable en los progenitores.

4. EXPERIMENTOS

El método propuesto se ha probado con un problema de estimación del coste de mantenimiento de la red eléctrica de una ciudad, basado en datos simulados de la red óptima para 1059 ciudades, con cuatro variables de entrada y una de salida. Para los experimentos, se ha realizado una validación cruzada con 5 particiones distintas. Es decir, el conjunto de datos se divide en 5 subconjuntos de igual tamaño (aproximadamente) y el método es ejecutado 5 veces, dejando en cada una de ellas uno de esos subconjuntos como conjunto de datos de prueba y utilizando los otros cuatro como conjunto de datos de entrenamiento[1]. Para cada una de las cinco particiones de datos, se ha ejecutado el AG seis veces con diferentes valores para la semilla de inicio de la secuencia de números aleatorios. Los parámetros del AG son: tamaño de población= 64, probabilidad de cruce= 0.6, probabilidad de mutación= 0.1 y condición de parada= 2000 evaluaciones.

Los resultados obtenidos por el método propuesto (AG+COR-OCH) se han comparado con otros dos mecanismos: uno de ellos comprende el AG propuesto en este trabajo pero utilizando el método de Wang y Mendel [11] para aprender la BR (AG+WM) y el otro considera únicamente el método COR-OCH con una partición difusa uniforme y el mismo número de etiquetas en todas las variables (5 etiquetas, al ser el valor con el que se ha obtenido el modelo más preciso).

Los resultados se muestran en la tabla 1. Las columnas numeradas de 1 a 5 en la primera fila representan las cinco particiones utilizadas. En cada una de esas columnas aparecen los valores medios de las seis ejecuciones del AG realizadas. Las últimas tres columnas representan: la media aritmética ($\bar{x}$) sobre las cinco columnas anteriores, la varianza de esos mismos cinco valores ($\sigma_{\bar{x}_i}$) y la media aritmética de las varianzas obtenidas entre las seis ejecuciones de cada partición ($\bar{\sigma_{x_i}}$). Mientras que $\sigma_{\bar{x}_i}$ representa las diferencias entre las cinco particiones, $\bar{\sigma_{x_i}}$ representa las diferencias entre las distintas ejecuciones de una misma partición. Es decir, el primer valor es una medida de la dependencia que tiene el método de la partición considerada, y

[1] los conjuntos de datos utilizados se pueden obtener en `http://decsai.ugr.es/~casillas/FMLib/`

Tabla 1: Resultados obtenidos

		1	**2**	**3**	**4**	**5**	$\bar{x}$	$\sigma_{\bar{x}_i}$	$\bar{\sigma_{x_i}}$
COR-OCH	ECM_{tra}	38483	39106	39851	40746	40013	**39640**	966	566
	ECM_{tst}	46452	46050	40689	35850	39373	**41683**	4397	1599
	$\#R$	42.3	40.0	40.5	40.2	42.0	**41.0**	1.8	1.4
	$\#E$	25	25	25	25	25	**25**	-	-
AG+COR-OCH	ECM_{tra}	19834	20678	20631	20640	20018	**20360**	402	1561
	ECM_{tst}	25076	24653	20919	23605	19899	**22830**	2302	3259
	$\#R$	18.3	17.8	15.2	17.7	19.8	**17.8**	1.7	3.8
	$\#E$	22.3	22.8	20.2	21.5	22.5	**21.9**	1.1	1.8
GA+WM	ECM_{tra}	22765	22682	24147	21697	23778	**23014**	971	2143
	ECM_{tst}	25826	23942	23817	24319	22549	**24090**	1176	3667
	$\#R$	54.0	44.5	51.0	54.2	52.0	**51.1**	3.9	6.8
	$\#E$	26.3	25.0	26.0	27.2	26.5	**26.2**	0.8	1.5

el segundo mide cuán robusta es la metaheurística para obtener resultados similares independientemente de la semilla utilizada. Los cuatro valores que se presentan en cada columna para cada método son los ECM sobre entrenamiento y prueba ($\mathbf{ECM_{tra}}$, $\mathbf{ECM_{tst}}$), el número de reglas de la BR ($\#\mathbf{R}$) y la suma del número de etiquetas de cada variable ($\#\mathbf{E}$).

Como se puede observar en la tabla, el método propuesto mejora claramente a los otros dos, obteniendo valores de predicción más precisos (tanto en entrenamiento como en prueba) y modelos más interpretables (menor número de reglas y de etiquetas). Por otro lado, AG+COR-OCH y AG+WM son ligeramente menos dependientes a la partición utilizada ($\sigma_{\bar{x}_i}$) que COR-OCH, mientras que éste último es menos dependiente de la semilla de inicio ($\bar{\sigma_{x_i}}$) que los anteriores.

El principal inconveniente del mecanismo propuesto es el tiempo requerido por el AG. Cada vez que se genera un nuevo cromosoma, es necesaria una ejecución de COR-OCH. Aunque dicho proceso no es excesivamente costoso en tiempo, para algunos problemas complejos (con muchas variables), sí puede suponer un inconveniente. En trabajos futuros intentaremos ajustar la estructura y parámetros del AG para reducir su tiempo de ejecución manteniendo en lo posible su buen comportamiento.

Referencias

[1] J. Casillas, O. Cordón, F. Herrera, COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules, *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 32(4), 2002, pp. 526-537.

[2] J. Casillas, O. Cordón, F. Herrera. COR methodology: a simple way to obtain linguistic fuzzy models with good interpretability and accuracy. En J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.), Accuracy improvements in linguistic fuzzy modeling, (Studies in Fuzziness and Soft Computing, Springer, 2003), págs. 27–45.

[3] O. Cordón, F. Herrera, I. Fernández de Viana, L. Moreno, A new ACO model integrating evolutionary computation concepts: the best-worst ant system, *Proceedings of the 2nd International Workshop on Ant Algorithms*, 2000, pp. 22-29.

[4] O. Cordón, F. Herrera, L. Magdalena, P. Villar, A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base, *Information Sciences* 136, 2001, pp. 85-107.

[5] O. Cordón, F. Herrera, P. Villar, Analysis and guidelines to obtain a good uniform fuzzy partition granularity for fuzzy rule-based systems using simulated annealing, *International Journal of Approximate Reasoning* 25, 2000, pp. 187-215.

[6] K. Deb, D. Joshi, A. Anand, Real-Coded Evolutionary Algorithms with Parent-Centric Recombination, *Proc. of the 2002 congress on evolutionary computation*, IEEE press, 2002, pp. 61-66.

[7] M. Dorigo, G. Di Caro. The ant colony optimization meta-heuristic. En D. Corne, M. Dorigo, F. Glover (Eds.), New ideas in optimization (McGraw-Hill, 1999), págs. 11–32.

[8] L. Magdalena, Adapting the gain of an FLC with genetic algorithms, *International Journal of Approximate Reasoning* 17(4), 1997, pp. 327-349.

[9] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs (Springer-Verlag, 1996).

[10] P. Thrift, Fuzzy logic synthesis with genetic algorithms. Proceedings of Fourth International Conference on Genetic Algorithms (ICGA'91), 1991, pp. 509-513.

[11] L.X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6), 1992, pp. 1414–1427.

ANÁLISIS ESTADÍSTICO DEL USO DE REGLAS TAKAGI-SUGENO-KANG DE ORDEN ALTO EN EL DISEÑO DEL PROCESO DE INFERENCIA DIFUSA

L.J. Herrera, H. Pomares, I. Rojas, A. Guillén, J. González
Dpto. Arquitectura y Tecnología de los Computadores
Universidad de Granada
jherrera@atc.ugr.es

Resumen

El modelo TSK de inferencia difusa es bien conocido por representar una herramienta muy potente para problemas de aproximación funcional. En este artículo se revisa un nuevo modelo TSK, TaSe [7], que utiliza reglas con consecuente de orden alto basadas en el desarrollo en serie de Taylor. Dicho modelo proporciona una reducción considerable en el número de reglas, además de proporcionar interpretabilidad a la salida polinómica de orden alto de dichas reglas. Finalmente, se ha realizado un análisis estadístico ANOVA, en el que se ha estudiado el comportamiento de un sistema difuso al modificar los bloques funcionales de la inferencia difusa para diversas funciones ejemplo. Dicho análisis concluye que se pueden obtener mejores resultados utilizando reglas de orden alto, característica que añadida a las anteriores indica la conveniencia de utilizar el modelo TaSe en una gran diversidad de problemas.

Palabras Clave: Reglas TSK de orden alto, ANOVA, Interpretabilidad .

1 INTRODUCCIÓN

En la literatura sobre sistemas difusos existe una gran cantidad de posibilidades al seleccionar los operadores básicos cuando se procede a diseñar un proceso de inferencia difusa utilizando reglas tipo Takagi-Sugeno-Kang (TSK) [13]. Así, hay numerosas alternativas para el operador de agregación, para definir la T-Norma, T-Conorma, el tipo de función de pertenencia, etc [9]. Estas reglas se caracterizan porque los consecuentes son una función de las variables de entrada. El modelo TSK ha demostrado ser una poderosa herramienta para problemas de aproximación funcional gracias a la alta capacidad de modelar relaciones no lineales entre variables utilizando un reducido número de reglas. Usualmente en la literatura se utilizan reglas de tipo TSK de orden bajo; de orden cero o reglas de consecuente constante, o de orden uno en que la salida de la regla es función lineal de sus antecedentes. Sin embargo se puede generalizar la salida de las reglas TSK a un polinomio de cualquier orden función de los antecedentes de las reglas [3].

El modelo TSK sin embargo padece de la falta de interpretabilidad del modelo, característica por otra parte fundamental de los sistemas difusos. El modelo TSK modificado TaSe [7] solventa este inconveniente al permitir la interpretación de las salidas de las reglas como el desarrollo en serie de taylor de la salida del modelo respecto al centro de cada regla. Esta propiedad nos permite por tanto aumentar el orden de la salida de las reglas, conservando la interpretabilidad y obteniendo una reducción importante en el número de reglas necesarias para la aproximación.

Dadas estas características del modelo TaSe, en este trabajo se pretende realizar un análisis preciso, comparando la repercusión sobre la salida del sistema al modificar los operadores básicos que intervienen en la inferencia difusa, incluyendo la forma de los consecuentes de las reglas difusas tipo TSK. Para realizar este análisis, se ha utilizado una herramienta adecuada a este problema: el Análisis de la Varianza (ANOVA)[2] multifactorial, que consiste en una serie de técnicas estadísticas que permiten el análisis y la comparación de experimentoss mediante la descripción de las interacciones e interrelaciones entre las variables cuantitativas y cualitativas (factores) que forman el sistema neuro-difuso. Veremos que a igualdad de complejidad del sistema (medida como el número de parámetros que deben ser optimizados), los sistemas TSK con consecuentes de orden alto (que por tanto tienen un menor número de reglas) pueden obtener

mayor precisión en problemas de aproximación funcional.

El resto de este trabajo se organiza de la siguiente forma. La Sección 2 presenta el modelo TaSe para el proceso de inferencia difusa basado en reglas TSK de orden alto, que hace uso del Desarrollo en Serie de Taylor de una función en torno a un punto para aproximar la función usando un número reducido de reglas y para proporcionar interpretabilidad al modelo final obtenido. La Sección 3 describe brevemente la herramienta ANOVA de análisis estadístico para el problema del diseño del proceso de inferencia difusa y presenta las diferentes alternativas para cada uno de los factores considerados. La Sección 4 presenta los problemas experimentales considerados. En la Sección 5 se presentan y se discuten los resultados del análisis estadístico y finalmente, la Sección 6 presenta las principales conclusiones obtenidas.

2 MODELO TaSe

En esta sección describimos brevemente el Sistema Difuso Basado en el Desarrollo en Serie de Taylor (TaSe), que utiliza reglas de orden alto, además de proporcionar interpretabilidad al modelo TSK.

2.1 REGLAS CON CONSECUENTE POLINOMIAL DE ORDEN ALTO

Un sistema difuso TSK consta de un conjunto de K reglas IF-THEN que típicamente tienen la forma [3]

$$\begin{array}{c} R^k : IF\ x_1\ es\ A_1^k\ y\ \ldots\ y\ x_n\ es\ A_n^k \\ THEN\ y = Y_k\left(\vec{x}\right) \end{array} \quad (1)$$

donde los A_i^k son conjuntos difusos caracterizados por funciones de pertenencia $A_1^k\left(x_i\right)$, x_i son las variables de entrada y $Y_k\left(\vec{x}\right)$ es un polinomio en cualquier orden función de las variables de entrada.

Como ya hemos comentado, en la literatura se encuentran principalmente sistemas TSK donde las reglas tienen consecuente constante, habiéndose obtenido resultados razonables al aplicarse a numerosos problemas [11, 12]. El principal problema que presentan estos sistemas es el inmanejable número de reglas que surgen cuando se trabaja con problemas de complejidad media. Sobre todo cuando el espacio de entrada se encuentra particionado a modo de grid en los Sistemas Difusos Basados en Grid (GBFS). Siendo n el número de variables de entrada y mf_i el número de funciones de pertenencia de la variable i, el número total de reglas de un sistema difuso con consecuentes constantes en GBFSs es igual a

$$K = \prod_{i=1}^{n} mf_i \quad (2)$$

Se propone ahora por tanto utilizar reglas donde el orden del polinomio en el consecuente es mayor, con lo que obtenemos un número de reglas exponencialmente inferior en comparación, compensándose este decremento con un incremento en la expresividad de cada regla.

Tomando por ejemplo polinomios de orden dos en los consecuentes de las reglas, éstas quedarían en la forma

$$\begin{array}{c} R^k : IF\ x_1\ es\ A_1^k\ y\ \ldots\ y\ x_n\ es\ A_n^k \\ THENy = \lambda_0^k + \vec{\lambda}_1^k \cdot \vec{x} + \vec{x}^T \cdot \Lambda_2^k \cdot \vec{x} \end{array} \quad (3)$$

donde λ_0^k es el coeficiente de orden 0, $\vec{\lambda}_1$ es un vector de coeficientes de orden 1 y Λ_2^k es una matrix triangular de coeficientes de orden 2.

2.2 INTERPRETABILIDAD DE LAS REGLAS CON CONSECUENTE POLINÓMICO

Hemos visto la gran ventaja que puede suponer el utilizar reglas de consecuente polinómico de orden alto, que es la reducción exponencial en número de reglas (especialmente en GBFSs), al adquirir las reglas mayor expresividad para aproximar una región mayor del espacio.

Citaremos ahora otra gran ventaja de las reglas de consecuente polinomial de orden alto, que pese a que está fuera del ámbito del estudio ANOVA que realizamos en este trabajo, es de gran importancia a la hora de escoger este tipo de reglas para modelar un sistema de inferencia difuso.

Dicha propiedad es la posibilidad de proporcionar interpretabilidad a la salida polinómica de orden alto de las reglas difusas[7]. La salida de cada regla será interpretada como el desarrollo en serie de Taylor de la salida del sistema en torno al punto especificado en el antecedente de la regla

$$\vec{c} = \left(c_1^k, \ldots, c_n^k\right) \quad (4)$$

donde c_i^k es el centro del conjunto difuso A_i^k, caracterizado por la función de pertenencia $A_1^k\left(x_i\right)$. En problemas de aproximación funcional se interpretará como el desarrollo en serie de Taylor de la función en torno al punto 4 especificado en el antecedente de la regla. Para ello hay que utilizar un tipo especial de funciones de pertenencia que explicamos en la siguiente subsección.

2.2.1 Desarrollo en Serie de Taylor

El teorema de Taylor establece que si una función $F\left(\vec{x}\right)$ definida en un intervalo tiene derivadas de todos los órdenes, ésta puede ser aproximada en torno a un punto $\vec{x} = \vec{a}$ como el polinomio

$$F(\vec{x}) \simeq F(\vec{a}) + (\vec{x}-\vec{a})^T \left[\frac{\partial F}{\partial x_i}(\vec{a})\right]_{i=1\ldots n} + \frac{1}{2}(\vec{x}-\vec{a})^T \Lambda (\vec{x}-\vec{a}) + \ldots \quad (5)$$

donde Λ es una matriz de dimensiones $n \times n$ que tiene los valores

$$\left[\frac{\partial F}{\partial x_i \partial x_j}(\vec{a})\right]_{i=1\ldots n; j=1\ldots n} \quad (6)$$

Se puede observar que los consecuentes de las reglas con consecuente polinomial de orden alto 3 es similar a esta ecuación última 5. Como requerimiento para que la interpretabilidad pueda tener lugar, hay que exigir que la salida del sistema difuso sea continua y derivable tantas veces como el grado del polinomio en los consecuentes de las reglas.

2.2.2 Funciones de pertenencia que preservan la interpretabilidad

Para que la salida del sistema difuso sea continua y derivable un número de veces igual al grado de los polinomios de los consecuentes de las reglas, hay utilizarán un tipo especial de funciones de pertenencia que conservarán estas propiedades en la salida del sistema difuso.

Una base de funciones de pertenencia locales ordenadas de orden ν(OLMF de orden ν), resumidamente es un conjunto de funciones de pertenencia (F.P.) locales que recorren un dominio de una variable tales que, el centro de una F.P. coincide con los extremos de las F.P. adyacentes (forman un grid) y todas las F.P. que forman la base son continuas y ν-veces derivables y tienen ν-derivada igual a cero en sus centros y extremos.

Utilizando bases de funciones de pertenencia locales ordenadas de orden ν_a en los antecedentes del sistema difuso y reglas con consecuente polinómico de orden ν_c 3, se puede aplicar el teorema de Bikdash [1] que dice que: para $\nu_a \leq \nu_c$, todo los consecuentes polinómicos de las reglas del sistema difuso pueden ser interpretadas como el Desarrollo en Serie de Taylor Truncado de la salida del sistema en torno al centro 4 de la regla respectiva.

3 APLICACIÓN DE ANOVA A LOS OPERADORES BÁSICOS QUE INTERVIENEN EN LA INFERENCIA DIFUSA

La teoría y la metodología del Análisis de la Varianza (ANOVA) fue desarrollada principalmente primera vez por R.A. Fisher [6]. Consiste en una serie de técnicas tomadas de la Estadística que permiten analizar y comparar experimentos en los que existe una salida cuantitativa unidimensional descrita como función de una serie de variables, cuantitativas o cualitativas, que se denominan factores [2]. ANOVA se utiliza para estimar los efectos directos y de interacción que los factores tienen sobre la salida. El efecto directo es la relación directa entre un factor y la salida; los efectos de interacción son es la relación conjunta de varios factores sobre la salida.

Para la aplicación del análisis de la varianza, las observaciones de la variables respuesta se expresan como el resultado aditivo de una serie de componentes [10]. En general, si tenemos múltiples factores, la observación $Y_{a,b\ldots l,s}$, donde:

- $a, b \ldots l$ son las diferentes variables o factores del problema
- s es el número de observaciones cuando existe repetición de experimentos reales bajo las mismas condiciones o valores de los factores.

puede ser admitida, en una primera aproximación, que es consecuencia aditiva de los efectos de los factores $a, ..., l$. Por tanto la observación puede ser expresada como una suma lineal

$$Y_{a,b,\ldots,l,s} = a + b + \ldots + l + \epsilon \quad (7)$$

siendo ϵ el efecto llamado de causas no asignables o de azar. Si los factores $a, b, ..., l$ se mantienen constantes, las medidas de $Y_{a,b\ldots l,s}$ presentarán variaciones que deberán poder considerarse como atribuibles a un gran número de pequeñas causas indistinguibles entre sí, que es lo que se designa como variación aleatoria [5, 2]. Si disponemos un experimento en que midamos $Y_{a,b\ldots l,s}$ a niveles diferentes de uno o más factores, el conjunto de medidas obtenidas puede que no sea homogéneo, estando formado por dos o más grupos. El propósito de la técnica introducida por Fisher es precisamente contrastar esta heterogeneidad, para ver si tales factores son realmente causas asignables en la variación que se trata de estudiar, o bien, se debe atribuir dicha variación al efecto de azar. Es decir, en el análisis de la varianza se trata de separar las componentes de la variación que aparecen en un conjunto de datos estadísticos, determinando si la discrepancia *entre* las medias de los factores son mayores de lo que podría esperarse razonablemente de las variaciones que ocurren *dentro* de los factores.

Debido a la abundante bibliografía, a veces con resultados contradictorios sobre el tipo de orden del mod-

elo TSK ideal para ser utilizado en la lógica difusa, junto con otros parámetros como el tipo de función de pertenencia, tipo de T-norma, complejidad del sistema, etc., en este artículo se pretende abordar el problema desde otro punto de vista. Lo que se pretende es responder a la pregunta: ¿La elección del tipo de orden del modelo TSK para el diseño de un sistema difuso es más significativa para el comportamiento del sistema, que las propias operaciones (T-normas, tipo de operador de agregación, tipo de función de pertenencia, etc.) involucradas en su definición y en el proceso de inferencia? Es decir, se pretende averiguar si una vez elegido el orden del modelo TSK (teniendo en cuenta que a mayor orden, se requieren mayor número de parámetros y por tanto mayor complejidad), la selección de la T-norma para realizar las operaciones de conjunción, el tipo de función de pertenencia, el tipo de operador de agregación, la complejidad del sistema, y el tipo de función que se requiere aproximar, resulta ser una elección más transcendente que la propia decisión del orden TSK a utilizar. Si la respuesta es que el orden TSK repercute de forma muy significativa en la salida del sistema, se justificaría el interés por la investigación de las propiedades y comportamientos del orden TSK, como parte transcendental del algoritmo de inferencia. Si por el contrario, existen otras operaciones más influyentes en la obtención de la salida, no es en principio necesario un pre-análisis pormenorizado de qué orden o modelo TSK debe ser utilizado para un problema concreto. Se trata, por tanto, de informar y centrar la atención sobre qué operaciones deben de ser analizadas en mayor profundidad y cuales no necesitan tanta atención, cuando se pretende confeccionar un sistema difuso.

4 PROBLEMAS EJEMPLO CONSIDERADOS

El problema de estimar una función desconocida f a partir de muestras en la forma $\left\{\left(\vec{x}^k; z^k\right); k = 1, 2, \ldots, K \text{ con } z^k = f\left(\vec{x}^k\right) \in \Re, \vec{x}^k \in \Re^N\right\}$ (i.e. aproximación funcional a partir de un conjunto finito de puntos), es uno de los puntos clave de investigación en técnicas de soft-computing y de la teoría de la aproximación funcional. Una vez dicha función f ha sido estimada a partir de un conjunto de entrenamiento conocido, ésta puede ser utilizada para predecir valores de salida para nuevos valores de entrada.

Para este trabajo, se ha utilizado una batería bien conocida de funciones bidimensionales (compuesta de ejemplos tomados de otros trabajos anteriores), tomada del trabajo de Cherkassky [4]. Para las ocho funciones ejemplo consideradas (véase Tabla 1 con la definición de las funciones) se han generado 400 puntos aleatoriamente distribuidos en el espacio de entrada para la evaluación de los posibles sistemas según todas las alternativas que figuran en la Tabla 1.

5 RESULTADOS DEL ANOVA

La herramienta que se va a emplear para la resolución de la pregunta anteriormente planteada, va a ser un estudio estadístico mediante el análisis de la varianza, de los diferentes niveles de cada uno de los ordenes TSK, las T-normas empleadas, el tipo de operador de agregación, diferentes tipos de funciones para ser aproximadas, diversas formas de funciones de pertenencia y diversos tipos de complejidad. La Tabla 1 muestra los diferentes niveles de los factores utilizados en el estudio estadístico.

La variable de respuesta utilizada para realizar el análisis estadístico es el NRMSE (error cuadrático medio normalizado por la varianza), a su vez normalizado en el rango [0,1] para cada una de las funciones de ejemplo utilizadas, al objeto de poder comparar los resultados de múltiples ejemplos de forma simultanea. La Tabla 2 muestra el resultado del análisis de la varianza a seis-vías realizado para todo el conjunto de ejemplos del sistema neuro-difuso. Como se puede observar de dicha tabla, todos los factores principales son estadísticamente significativos, aunque es importante destacar que el tipo de TSK, junto con el tipo de operador de agregación, son los dos factores más relevantes en el diseño del sistema difuso. Para el único caso en que el Weighted Sum y el Weighted Average tienen el mismo comportamiento, es para la utilización de modelos difusos que utilicen una partición triangular de sus funciones de pertenencia, con el operador producto como T-norma (la suma de activaciones de las reglas del sistema es la unidad). La figura 1 muestra la diferencia en el error de ambos tipos de operador de agregación, siendo el Weight Average más preciso en términos del error medio, para el diseño de sistemas difusos.

Respecto al modelo TSK utilizado, es muy importante observar su comportamiento para los diferentes niveles. Para el modelo TSK de orden 0 (que sería el primer nivel de este factor), se obtienen en media errores que están entre el modelo TSK de orden 1 y del orden 2, con igual número de parámetros. La figura 2 ilustra los errores medios de los tres modelos TSK. Es importante volver a destacar, que a igualdad en el número de parámetros libres o a optimizar, el TSK de orden cero tendrá un número de funciones de pertenencia en las variables de entrada, y por consiguiente de reglas difusas, mayor que el TSK de orden uno y que el TSK de orden 2. El modelo TSK de orden uno presenta en media el mayor error, mientras que el orden 2,

con la metodología presentada en este artículo, obtiene los errores más bajos. Por lo tanto, y con un número de reglas difusas mucho menor que otros modelos TSK (aunque de mayor complejidad), se obtienen aproximaciones funcionales muy precisas. Este resultado confirma la utilidad e importancia de la creación de sistemas difusos que utilicen reglas de orden alto basadas en el desarrollo en serie de Taylor (TaSe), puesto que a igualdad de complejidades (número de parámetros), pero con un conjunto de reglas mucho más reducido, se obtienen excelentes resultados de aproximación.

El tercer parámetro importante en la construcción de un sistema difuso es sin lugar a dudas la complejidad (medida en términos de número de parámetros a optimizar, contabilizando tanto los parámetros que definen las funciones de pertenencia como los consecuentes de las reglas). Como era de esperar, cuanto mayor sea la complejidad del sistema, y por tanto, mayor el número de reglas difusas, menor será el error de aproximación (figura 3).

Con respecto a los restantes factores considerados en este estudio, aunque tienen significancia desde un punto de vista estadístico en el comportamiento del sistema difuso, su influencia no es tan relevante como los tres factores comentados anteriormente.

6 CONCLUSIONES

El modelo de Takagi-Sugeno-Kang (TSK) representa una herramienta muy potente para problemas de aproximación funcional, debido a la alta capacidad de modelar relaciones complejas no lineales entre las variables de entrada utilizando reglas cuyo consecuente es una función simple de las variables de entrada. Sin embargo este tipo de sistemas tiene dos inconvenientes principales que son la falta de interpretabilidad del modelo y el gran número de reglas que se necesitan (sobre todo para particionamiento del espacio de entrada basado en grids). En este artículo hemos revisado el modelo TaSe que disminuye ambas desventajas mediante la utilización de reglas de orden alto basadas en el desarrollo en serie de Taylor. Además, se ha realizado un estudio estadístico ANOVA, utilizando varias funciones ejemplo [4],en el que se demuestra la conveniencia de utilizar este tipo de sistemas en comparación con los sistemas tradicionales que utilizan reglas de orden cero o uno, a igualdad en el número de parámetros de los modelos.

Agradecimientos

Este trabajo ha sido realizado parcialmente al amparo del proyecto DPI2001-3219 financiado por el Ministerio de Ciencia y Tecnología de España.

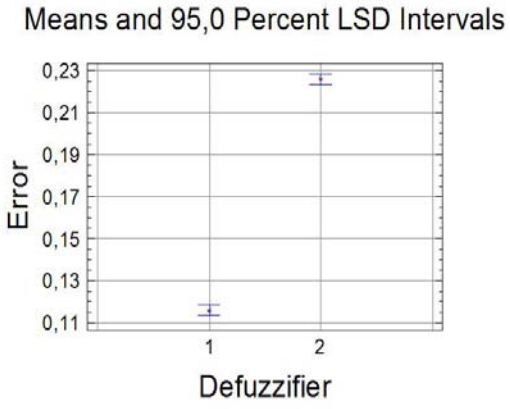


Figura 1: Evolución del error medio para diferentes niveles del factor Tipo de Operador de Agregación

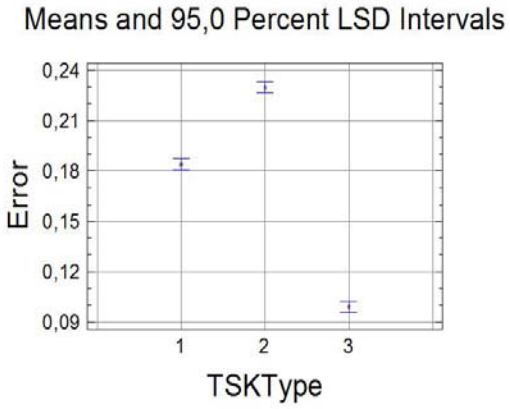


Figura 2: Evolución del error medio para diferentes niveles del factor Tipo de TSK

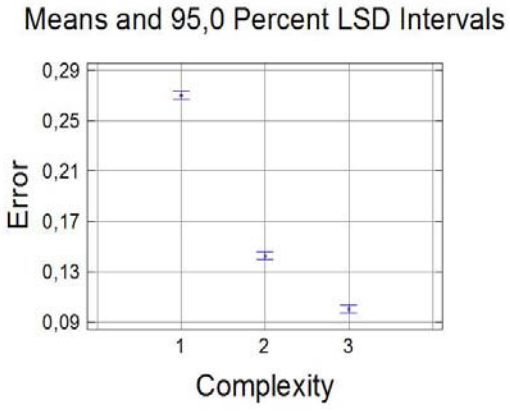


Figura 3: Evolución del error medio para diferentes niveles del factor Complejidad

Referencias

[1] M. Bikdash, A Highly Interpretable Form of Sugeno Inference Systems,*IEEE Trans. Fuz. Syst. Vol.7 No.6*,Pág.686-696. 1999

[2] G.E.P. Box, Estadística para investigadores. Introducción al diseño de experimentos, análisis de datos y construcción de modelos, *Editorial Reverté*, 1989

[3] J.J. Buckley, Sugeno-type Controllers are Universal Controllers, *Fuzzy Sets and Systems, Vol.25*, Pág. 299-303, 1993

[4] V. Cherkassky, D. Gchring, F. Mulier, Comparison of Adaptive Methods for Function Estimation from Samples, *IEEE Transactions on Neural Networks, Vol.7, N.4*, Pág. 969-984, 1996

[5] M.H. Degroot, Probabilidad y estadística, *Addison-Wesley Iberoamericana, Segunda edición*, 1988

[6] R.A. Fisher, Contribution to Mathematical Statistics, *New York, Wiley*, 1950

[7] L.J. Herrera, H. Pomares, I. Rojas, J.González, O. Valenzuela, Function Approximation through Fuzzy Systems Using Taylor Series Expansion-Based Rules: Interpretability and Parameter Tuning, *LNCS/LNAI Vol. 2972*, April 2004

[8] R. Mead, The design of experiments. Statistical principles for practical application, *Cambrigde University Press*, 1988

[9] J. M. Mendel, Fuzzy logic systems for engineering, *IEEE Proc. 83*, Pág. 345-377, 1995

[10] D.C. Montgomery, Diseño y análisis de experimentos,*Grupo Editorial Iberoamérica*, 1991

[11] H.Pomares, I.Rojas, J.González, A.Prieto, Structure Identification in Complete Rule-Based Fuzzy Systems, *IEEE Transactions on Fuzzy Systems, Vol.10 N. 3* Pág. 349-359, 2002

[12] I.Rojas, H.Pomares, J.Ortega, A.Prieto, Self-Organized Fuzzy System Generation from Training Examples, *IEEE Trans. Fuzz. Syst. Vol. 8, N. 1*, Pág. 23-36, 2000

[13] T.Takagi, M.Sugeno, Fuzzy identification of systems and its applications to modelling and control, *IEEE Transactions Systems Man and Cybernetics vol.15*,Pág. 116-132, 1985

Tabla 1: Variables utilizadas en el estudio estadístico

Niveles factores	**Op. Agr.**	**Tipo F. pertenencia**	**Complejidad**	**Tipo TSK**
Nivel 1	W. Aver.	OLMF	Baja (54 param.)	Orden 0
Nivel 2	W. Sum	Triangular	Media (120 param.)	Orden 1
Nivel 3		Gaussiana	Alta (180 param.)	Orden 2

Niveles factores	**T-Norm**	**Función ejemplo a aproximar**
Nivel 1	Prod.	$f_1 = sin(x_1 * x_2)$ X uniforme en [-2, 2]
Nivel 2	Min	$f_2 = exp(x_1 * sin(\pi * x_2))$ X uniforme en [-1, 1]
Nivel 3	Ham[1] $\lambda = 0.25$	$a = 40 * exp(8 * ((x_1 - .5)^2 + (x_2 - .5)^2))$ $b = exp(8 * ((x_1 - .2)^2 + (x_2 - .7)^2))$ $c = exp(8 * ((x_1 - .7)^2 + (x_2 - .2)^2))$ $f_3 = a/(b + c)$ X uniforme en [0, 1]
Nivel 4	Ham[1] $\lambda = 0.5$	$f_4 = \frac{(1+sin(2*x_1+3*x_2))}{(3.5+sin(x_1-x_2))}$ X uniforme en [-2, 2]
Nivel 5	Ham[1] $\lambda = 0.75$	$f_5 = 42.659(.1 + x_1(.05 + x_1^4 - 10x_1^2x_2^2 + 5x_2^2))$ X uniforme en [-.5, .5]
Nivel 6		$f_6 = 1.3356[1.5(1 - x_1)+$ $exp(2x_1 - 1)sin(3\pi(x_1 - .6)^2)+$ $exp(3(x_2 - .5))sin(4\pi(x_2 - .9)^2)]$ X uniforme en [0, 1]
Nivel 7		$f_7 = 1.9[1.35 + exp(x_1)sin(13(x_1 - .6)^2)$ $exp(-x_2)sin(7x_2)]$ X uniforme en [0, 1]
Nivel 8		$f_8 = sin(2\pi * sqrt(x_1^2 + x_2^2))$ X uniforme en [-1, 1]

Tabla 2: Tabla ANOVA para el análisis de los principales parámetros que intervienen en el diseño de un sistema difuso

	Suma de cuadr.	**Gr. lib.**	**Media cuadr.**	**F-Ratio**	**Sig. Level**
Factores Principales					
Op. Agregación	10,650	1	10,650	1721,40	0,00
Función ejemplo a aproximar	3,987	7	0,332	53,71	0,00
Tipo de función de pertenencia	2,672	2	1,336	216,01	0,00
T-Norm	13,473	4	3,368	544,45	0,00
Tipo de TSK	10,332	2	5,166	835,01	0,00
Complejidad	18,271	2	9,135	1476,62	0,00
Interacciones Significativas					
Tipo de TSK & complejidad	8,341	4	2,0854	337,08	0,00
Op. Agregación & Tipo de F.Pert	1,949	2	0,974	157,57	0,00

[1]Hamacher T-norm is expressed as: $\frac{\lambda \cdot a \cdot b}{1-(1-\lambda)(a+b-a \cdot b)}$

Fuzzy-XCS: An Accuracy-Based Fuzzy Classifier System *

Jorge Casillas
Dept. Computer Science & Artif. Intell.
University of Granada
Granada E-18071, Spain
casillas@decsai.ugr.es

Brian Carse, Larry Bull
Fact. Comput., Eng. & Mathematical Sci.
University of the West of England
Bristol BS16 1QY, UK
Brian.Carse, Larry.Bull@uwe.ac.uk

Abstract

The issue of rule generalization has received a great deal of attention in the discrete-valued learning classifier system field. In particular, the accuracy based XCS is the subject of extensive ongoing research.

However, the same issue does not appear to have received a similar level of attention in the case of the fuzzy classifier system. This may be due to the difficulty in extending the discrete-valued system operation to the continuous case.

The intention of this contribution is to propose an approach to properly develop a fuzzy XCS system.

Keywords: learning classifier systems, XCS, evolutionary algorithm, Michigan-style learning fuzzy systems, fuzzy implications.

1 Introduction

The fuzzy classifier system is a machine learning system which employs linguistic rules and fuzzy sets in its representation and an evolutionary algorithm (EA) for rule discovery. It therefore combines an easily understood representation (as opposed to, for example, neural networks approaches) with a general purpose search method. In order to exploit the fuzzy representing to the full, the ability to learn generalization is of great importance.

Generalized rules allow more compact rule bases, scalability to higher dimensional spaces, faster inference, and better linguistic interpretability. The issue of rule generalization, and the interplay between general and specific rules in the same evolving population, has received a great deal attention in the discrete-valued classifier system research community (e.g. [8]). The same issue does not appear to have received a similar level of attention in the case of fuzzy classifier systems [1, 2, 3, 4, 5, 6, 7].

Traditional Michigan-style classifier systems have been "strength-based" in the sense that a classifier accrues strength during interaction with the environment (through rewards and/or penalties). This strength can then be used for two purposes: resolving conflicts between simultaneously matched classifiers during learning episodes; and as the basis of fitness for the EA. A completely different approach can be taken in which a classifier's fitness, from the point of view of the EA, is based on its "accuracy," i.e. how well a classifier predicts payoff whenever it fires. Such an accuracy-based approach offers a number of advantages such as avoiding over-general classifiers, obtaining optimally general classifiers, and learning of a complete "covering map." This accuracy-based classifier system, called XCS, was proposed in [8] and it is currently of major interest to the research community in this field.

This work aims at proposing a new approach to achieve accuracy-based Michigan-style fuzzy classifier systems. The proposal, Fuzzy-XCS, is based on XCS but properly adapted to fuzzy systems. The fuzzy inference and the reinforcement component are changed to consider a different, competitive interaction among rules that allow an accuracy-based operation.

The paper is organized as follows. Section 2 introduces some difficulties to develop an accuracy-based fuzzy classifier system. Section 3 describes a competitive inference mechanism. Section 4 introduces the proposed Fuzzy-XCS system. Section 5 shows some experimental results. Section 6 concludes.

* Part of this research has been developed during a research stay of J. Casillas at the Faculty of Computing, Engineering and Mathematical Sciences of the University of the West of England (Bristol, UK) supported by the *Secretaría General de Universidades e Investigación de la Junta de Andalucía* (Spain)

2 Difficulties in Accuracy-Based Fuzzy Classifier Systems

Most of the difficulty in accuracy-based fuzzy classifier system is the fuzzy inference process, therefore, it is briefly explained in the next subsection. Then, the problems when using strength-based Michigan fuzzy classifier systems, the advantages of considering accuracy-based fitness, and the difficulties in doing that are introduced.

2.1 Cooperation-Based Fuzzy Inference

This is the most commonly used fuzzy inference method. Linguistic (or Mamdani-type) fuzzy rule-based systems are formed by linguistic fuzzy rules with the following structure:

...
also
R^{r-1}: **IF** X_1 is A_1^{r-1} and ... and X_n is A_n^{r-1}
THEN Y_1 is B_1^{r-1} and ... and Y_m is B_m^{r-1} ,
also
R^r: **IF** X_1 is A_1^r and ... and X_n is A_n^r
THEN Y_1 is B_1^r and ... and Y_m is B_m^r ,
also
...

with X_i and Y_j being input and output linguistic variables respectively, and with A_i and B_j being linguistic labels with associated fuzzy sets defining their meaning. These linguistic labels use a global semantic defining the set of possible fuzzy sets used for each variable.

The inference process, which obtains an output as response to a specific input, consists basically of the following steps: for each fired (or matched) fuzzy rule, firstly the conjunction (***and***) operator is used to obtain the *matching degree* of the rule; secondly, the implication (***then***) operator is applied to scale the output fuzzy set to a degree according to the matching; finally, the last step is to apply the aggregation (***also***) operator to reduce the combined output of all the rules acting together to a single fuzzy set.

If we need a real-valued output, a last stage (defuzzification), converts the output fuzzy set to a number. Center of gravity, center of sums, or mean of maxima are some possibilities to do that.

The most widely inference scheme used in linguistic fuzzy systems is that proposed by Mamdani and Assilian in the first fuzzy controller of 1975. It involves the combination of operators (*min*,*min*,*max*), i.e., minimum for conjunction, minimum for implication, and maximum for aggregation. This inference is usually named Max-Min. Alternatively, sometimes it uses another t-norm (e.g. product) to play the role of conjunction and/or implication, or another t-conorm (e.g. bounded sum) to the also operator.

This is the most commonly used approach (we could say even the only one) followed in engineering problems such as fuzzy control and fuzzy modeling. It is because of, with these operators, the fuzzy rules "cooperate" to generate the output, in the sense that an Interpolative Reasoning is performed to define the output in the zones where several rules work to a medium matching degree. Indeed, the fact of using a t-conorm to aggregate the information involves that we are using a *union* and, therefore, the effect of each rule is added to the final consensual output.

2.2 Problems with "Strength-Based" Michigan Fuzzy Classifier Systems

As mentioned in the introduction, the strength-based fuzzy classifier system is characterized by using the same parameter (i.e. strength) to resolve conflicts between matched classifiers and to compute their fitness. A number of problems arise from this dual use of classifier strength. These include:

1. The cooperation/competition problem. High-strength, potentially cooperative classifiers go on to compete under the action of the EA.

2. Over-general rules with relatively high (but inconsistent) payoff can come to dominate the population.

3. In some environmental states, the maximum payoff achievable (by performing the best possible action for that state) may be relatively low. Although a classifier might be the best that can exist for that state, it can be eradicated from the population by other classifiers that achieve higher rewards in other states. This results in gaps in the system's "covering map."

2.3 Advantages of using "Accuracy-Based" Fitness

A completely different approach can be taken in which a classifier's fitness, from the point of view of the EA, is based on its "accuracy," i.e. how well a classifier predicts payoff whenever it fires. Such an accuracy-based approach offers a number of advantages. Firstly, it can distinguish between accurate and over-general classifiers: an over-general classifier will have relatively low accuracy since payoff will vary according to the input states covered by the classifier. Indeed, it has been shown (in the discrete valued case) that the accuracy-based approach can lead to evolution of optimally general classifiers. Additionally it can maintain both consistently correct and consistently incorrect classifiers which allows learning of a complete "covering map."

A potential drawback of the accuracy-based approach is that it is likely to require larger populations of classifiers.

2.4 Difficulties in Moving to "Accuracy-Based" Fuzzy Classifier Systems

Firstly, in a traditional fuzzy classifier system several rules fire in parallel (this is how the system achieves interpolation); credit assignment is much more difficult in the fuzzy case and it may well be that apportioning credit in proportion to a fuzzy classifier's activation level is not appropriate. A further difficulty is measuring the accuracy of a rule's predicted payoff since (particularly early in the search) a fuzzy rule will fire with many different other fuzzy rules at different time-steps, giving very different payoffs. Yet another difficulty is that the payoff a fuzzy rule receives depends on the input vector — an active fuzzy rule will receive different payoffs for different inputs. This further complicates payoff predictions used as the basis for accuracy-based fitness.

3 Competitive Fuzzy Inference

The problems explained in the previous section are mainly due to the fact that, usually, all the matched rules *cooperate* to define the final solution in a interpolative behavior as explained in Section 2.1. If it is the problem, why not to change our approach and look for competitive interaction? This section introduces such an approach.

3.1 Why Use Competitive Inference?

Michigan-style learning classifier systems in general, and XCS in particular, do not consider the interaction of the different rules as a cooperative action, instead they consider that each rule competes with the rest to be the best one for a particular input vector. This involves that the action is only due to a set of rules that were the winners among the rules composing the match set. Extending this approach to fuzzy modeling, it makes sense to deal with competitive fuzzy rules, in which case, we need to know how to work with such rules.

In off-line fuzzy system learning, it seems that cooperative inference has some clear advantages that make it be more suitable for this problem. In this case, we known the whole data set *a priori* and our objective is to build a landscape of the output data. To do that, it seems a good approach to fill the gaps between data with interpolation, and this is the reason why cooperative inference is successful.

However, in on-line fuzzy system learning, why is it necessary to interpolate the actions of the rules? The objective in this problem should not be to define a landscape of the output, but a landscape of the reward. This means taking the best action in each state. For this purpose, a competitive action makes more sense since our objective is to optimize the rules individually to have the best reward.

With competitive fuzzy inference we propose a decision making process where the matched rules compete among themselves, and only one is the winner. The output finally obtained is mainly due to the winner rule. That is, loser rules do not have an important influence on the output. Nevertheless, competition does not involve a loss of interaction. There is an interaction, but with a selfish objective. To perform competitive inference we only need to change the roles of the inference operators.

3.2 Fuzzy Operators for Competitive Inference

First of all, since each rule competes with the rest, we should use an intersection (t-norm) as *also* operator, instead of the union approach followed by the cooperative inference. We use the *minimum* in this paper. However, this change does not allow us to use the same kind of implication operator.

Analyzing the classical Boolean implication $p \Rightarrow q$, equivalent to $\neg p \vee q$, it is true when p is false. That is, when we have not information about the certainty of the fact (antecedent), we assume the "optimistic" view of thinking that the implication is true.

However, in cooperative inference, a t-norm (intersection) is used as implication. It is a "pessimistic" view that assumes the falseness of the implication when the fact is false. It works if we use a union (t-conorm) to aggregate the effect of the different rules. This behavior could be interpreted that the belief of the rules resides in the set of them, instead of in each of them.

To develop a competitive inference where a intersection is considered as aggregation, the implication should follow the classical Boolean if-then view assuming that each rule has all the information about the relationship between fact and consequence. Thus, the belief develops separately for each rule. This family of implications are known as *logical* fuzzy implications, instead of the *engineering* fuzzy implications based on the t-norm. Table 1 includes some of the best known *logical* fuzzy implications.

To understand better the different behavior between cooperative and competitive inference, Figure 1 shows the output generated by two fuzzy systems that only

Table 1: Some *logical* fuzzy implications

Operator	$\mu_R(x,y)$
Kleene-Dienes	$max\{1-\mu_A(x), \mu_B(y)\}$
Zadeh	$max\{min\{\mu_A(x), \mu_B(y)\}, 1-\mu_A(x)\}$
Łukasiewicz	$min\{1, 1-\mu_A(x)+\mu_B(y)\}$
Dubois-Prade	$\begin{cases} 1-\mu_A(x) & \text{if } \mu_B(x)=0 \\ \mu_B(x) & \text{if } \mu_A(x)=1 \\ 1 & \text{otherwise} \end{cases}$
Gödel	$\begin{cases} 1 & \text{if } \mu_A(x) \leq \mu_B(x) \\ \mu_B(x) & \text{otherwise} \end{cases}$
Goguen	$min\{\mu_B(y)/max\{\mu_A(y), \mu_B(y)\}\}$

differ in the inference mechanism. In the cooperative inference, minimum, minimum, and maximum are used as conjunction, implication, and aggregation operators, respectively. In the competitive inference, minimum, Łukasiewicz, and minimum are used as conjunction, implication, and aggregation operators, respectively. The center of gravity is used as defuzzification process in both cases. Both systems consist of two input variables and one output variable, with 5 triangular fuzzy sets uniformly distributed in the universe of discourse [0,1] for each variable. Among the 25 fuzzy rules considered, only 7 have a consequent different from the medium linguistic term (with vertex at 0.5). As we can see, both approaches generate a smooth output but, while cooperative inference fills the gaps by interpolating, competitive inference isolates the effect of each rule.

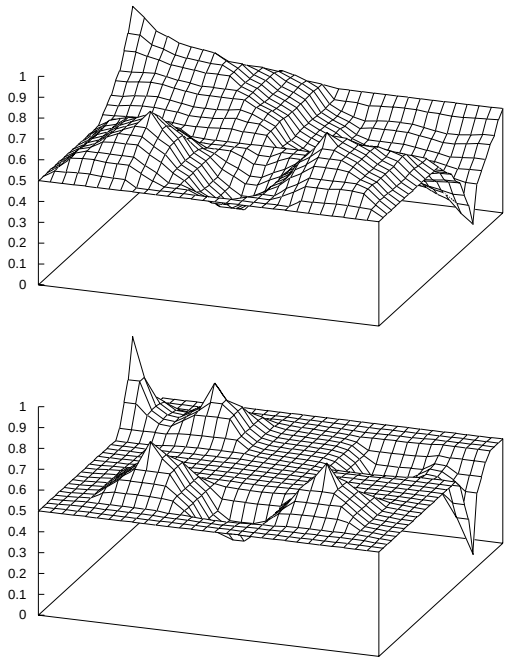


Figure 1: Output generated by *cooperative* (top) and *competitive* (bottom) inferences

On the other hand, Figure 2 illustrates some examples of competitive inference depending on the used implication operator. The *minimum* aggregation operator is used in all the cases. We can see how the action strength of the winner rule (the one with the highest matching degree) not only depends on its matching degree but also on the difference in the matching degree with respect to fuzzy rules with different actions (consequents), i.e., rival fuzzy rules. The more different these matching degrees, the higher the action strength of the selected rule. This involves that actually, the competition is performed among different linguistic actions. Therefore, we should say that competitive inference leads to competitive actions, but not necessarily competitive rules.

The competition among rules with the same action is only performed when using the minimum operator as aggregation, since in this case only the best rule of that action is considered in the inference. The use of other aggregation operators, such as the product, will lead the inference to a cooperation among rules with the same actions. According to XCS, this approach is not desired since it aims at obtaining the $X \times A \Rightarrow P$ map with the best classifier for each situation-action combination [8].

We can characterize slightly different interactions approaches among rules with different actions (i.e. rival rules) depending on the logical fuzzy implication operator used; these are shown in Table 2.

Table 2: Type of competition among rules with different actions depending on the used *implication* operator (a t-norm is used as *also* operator)

Implication	Type of competition
Kleene-Dienes	Rival rules reduce the importance of the winner rule in non-overlapped areas
Zadeh	Like Kleene-Dienes. No interaction among rules if the winner rule has a matching degree lower than 0.5
Łukasiewicz	Like Kleene-Dienes, but with higher preservation of overlapped areas than it
Dubois-Prade	Like Kleene-Dienes, but with the highest preservation of overlapped areas
Gödel	Output always in the overlapped area. No output with more than two different actions. If matchings greater than cross-point, output in the center of the overlapped area
Goguen	Like Gödel. Better with *mean of maximum* defuzzification

Finally, we should say that the consideration of competitive actions seems to fit properly with the XCS aim. Other fuzzy learning classifier systems, like Bonarini's work [1, 2], focus on the interaction among

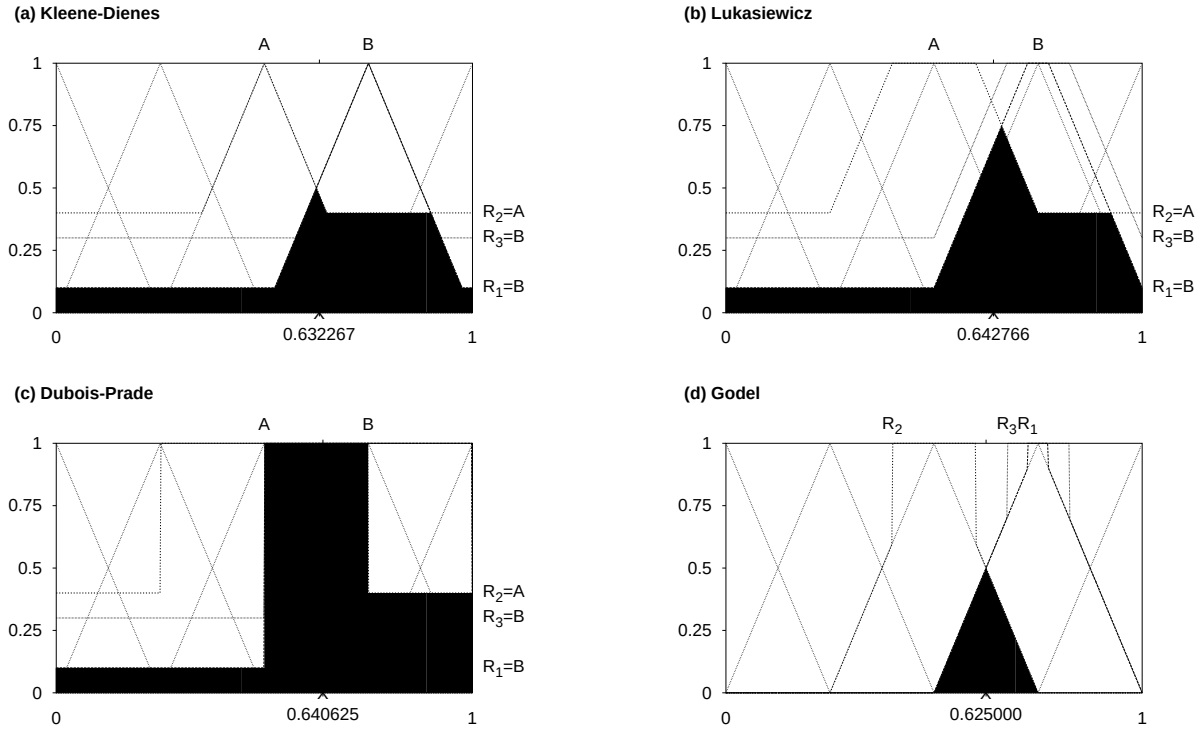


Figure 2: Results of four different fuzzy implications using minimum as *also* operator. Matching degrees: $\mu_{A_{R_1}} = 0.9$, $\mu_{A_{R_2}} = 0.6$, and $\mu_{A_{R_3}} = 0.7$

rules in the antecedent (state) instead of the consequent (action). He proposes a competition among rules with the same antecedent but a cooperation among rules with different ones. Besides, the implication and aggregation considered in his proposal leads implicitly to a cooperation between the consequents.

4 Fuzzy-XCS

The use of competitive fuzzy rules seems to solve the problems shown by the cooperative approach (Section 2.4). Let us now see how Fuzzy-XCS works.

4.1 Generalization Representation

First of all, a representation of fuzzy classifiers to allow proper generalization must be done. We propose the use of *disjunctive normal form* (DNF) fuzzy rules with the following structure:

IF X_1 is $\widetilde{A_1}$ and ... and X_n is $\widetilde{A_n}$ **THEN** Y is B

where each input variable X_i takes as a value a set of linguistic terms $\widetilde{A_i} = \{A_{i1} \vee \ldots \vee A_{il_i}\}$, whose members are joined by a disjunctive (t-conorm) operator, whilst the output variable remains a usual linguistic variable with a single label associated. We use the *bounded sum* ($min\{1, a + b\}$) as t-conorm.

This structure uses a more compact description that allows rules with different generalization degrees. Moreover, the structure naturally supports the absence of some input variables in each rule (simply making $\widetilde{A_i}$ be the whole set of linguistic terms).

In order to use this representation in Fuzzy-XCS, we propose a binary coding scheme for the antecedent of the fuzzy rule with size equal to the sum of the number of linguistic terms used in each input variable. The allele '1' means that the corresponding linguistic term is used in the corresponding variable. For the consequent of the rule, an integer coding scheme is used where each gene contains the index of the linguistic terms used for the corresponding output variable. For example, assuming we have three linguistic terms (S, M, and L) for each input/output variable, the fuzzy rule [IF X_1 is S and X_2 is {M or L} THEN Y_1 is M and Y_2 is L] is encoded as [100|011||23].

4.2 Performance Component

In XCS [8], the performance component consists of three stages: match set construction, prediction array computation, and action set selection. This process has the final objective of inferring an specific action from the set of classifiers that matches the current state. In Fuzzy-XCS the process is different, as described as follows:

• *Match set*: To avoid an excessive number of matched rules, only those rules with a matching degree greater or equal to a specific value (θ_M) are included in the match set.

• *Computation of candidate subsets*: This stage could be equivalent to the prediction array computation. It is necessary in discrete output systems like XCS to groups the matched classifiers according to the different actions, since it is not possible to select several actions. However, in real-valued output systems like Fuzzy-XCS, several "linguistic actions" (consequents) could be considered together. Thus, Fuzzy-XCS redefines the concept of prediction array computation. We can assume that what should not be accepted in our case is to have an action set with inconsistent rules, i.e. rules with the same antecedent and different consequent. Since DNF-type rules are considered, to have the same antecedent also involves rules where the antecedent of some of them are contained in others. Therefore, different groups of consistent fuzzy rule set (with the maximum number of rules in each group) are formed.

• *Action set selection*: The action set selection chooses the consistent classifier set with the highest mean prediction. In learning classifier systems the criterion used to select the action set alternately varies between exploration and exploitation approaches. According to this, our deterministic selection could be considered as a purely exploitation approach. Although we have experimented with several possibilities, the best results were obtained with this procedure.

4.3 Reinforcement Component

The p_j (prediction), ϵ_j (prediction error), and F_j (fitness) values are adjusted by the reinforcement learning standard techniques used in XCS (Q-learning, Widrow-Hoff, and MAM) for each fuzzy classifier C_j.

However, an important difference is considered in Fuzzy-XCS: the distribution among the classifiers must be made proportionally to the degree of contribution of each classifier to the obtained output. This is a crucial issue because the interaction among the classifiers (with competition actions in our case) is developed here.

The reinforcement performed in Fuzzy-XCS acts on the action set AS. The following subsections detail the reinforcement distribution process and the adjustment of the parameters.

4.3.1 Distribution of the Reinforcement

The reinforcement distribution among the classifiers of the action set AS is made by analyzing the contribution of each classifier to generate the aggregated output fuzzy set. Let B'_j be the scaled output fuzzy set generated by the fuzzy rule R_j:

$$B'_j = I(\mu_{A_{R_j}}(x), B_j), \tag{1}$$

where $\mu_{A_{R_j}}(x)$ is the matching degree of the rule R_j, I is the used logical fuzzy implication operator (see Table 1), and B_j the fuzzy set of the consequent of the rule R_j. Let $R_1 \in AS$ be the winner rule, i.e., $\mu_{A_{R_1}}(x) \geq \mu_{A_{R_j}}(x), \forall R_j \in AS - \{R_1\}$.

The process involves analyzing the area that the rival fuzzy rules (rules with lower matching degrees than R_1) "bite" into the area generated by the winner rule.

Thus, the weight of the winner rule is:

$$w_1 = \frac{\int \bigwedge_{j=1}^{|AS|} \mu_{B'_j}(y)\ dy}{\int \mu_{B'_1}(y)\ dy}, \tag{2}$$

with $|AS|$ being the action set size and $\wedge$ the t-norm used as aggregation operator (the minimum in our case), while the weights of the rival rules are computed as follows:

$$w_j = \frac{(1-w_1)\cdot\left(\int \mu_{B'_1}(y)\ dy - \int \mu_{B'_1}(y) \wedge \mu_{B'_j}(y)\ dy\right)}{\sum_{i=2}^{|AS|}\left(\int \mu_{B'_1}(y)\ dy - \int \mu_{B'_1}(y) \wedge \mu_{B'_i}(y)\ dy\right)} \tag{3}$$

This distribution is designed to take into account that competitive inference is being considered. To illustrate this behavior, we can see that, from the example shown in Figure 2(a), the competitive inference generates the weights $w_1 = 0.711$, $w_2 = 0.289$, and $w_3 = 0$, while a cooperative-based distribution proportional to the matching degrees generates the weights $w_1 = 0.409$, $w_2 = 0.318$, and $w_3 = 0.273$. Indeed, the selection pressure in the competitive inference is higher, thus allowing to discriminate between good and bad rules.

4.3.2 Adjustment of the Parameters

To adjust the parameters of each classifier, firstly the P (payoff) value is computed with the Q-learning technique as follows: $P = r + (\gamma \cdot \mu_{A_{R_1}}(x))$, with r being the external reward from the previous time-step, and $\gamma \in [0, 1]$ a constant decreasing factor.

Then, the following adjustment process is performed for each fuzzy classifier belonging to the action set:

1. Firstly, adjust the error values ϵ_j using the standard Widrow-Hoff delta rule with learning rate parameter β ($0 < \beta \leq 1$) toward $|P - p_j|$ considering the weights w_j computed in eqs. (2) and (3) to distribute the adjustment, i.e.

$$\epsilon_j \leftarrow \epsilon_j + \beta \cdot w_j \cdot (|P - p_j| - \epsilon_j). \tag{4}$$

The MAM (moyenne adaptive modifiee) technique is used by adjusting ϵ_j to the average of the $|P - p_j|$ values, instead of the above equation, during the first $1/\beta$ times that the corresponding classifier is adjusted.

2. Then, adjust prediction values

$$p_j \leftarrow p_j + \beta \cdot w_j \cdot (P - p_j). \tag{5}$$

Again, weights are considered to distribute the reinforcement. MAM technique is also used here during the $1/\beta$ first adjustments.

3. Finally, recalculate the fitness values F_j from the updated values of ϵ_j, i.e.

$$F_j \leftarrow F_j + \beta \cdot (k'_j - F_j), \tag{6}$$

with

$$k'_j = \frac{k_j}{\sum_{R_i \in AS} k_i}, \qquad k_j = \begin{cases} (\epsilon_j/\epsilon_0)^{-\nu} & \epsilon_j > \epsilon_0 \\ 1 & \text{otherwise.} \end{cases} \tag{7}$$

No weights w_j are considered to update the fitness since it depends on the prediction error instead of the received payoff. Again, MAM technique is used.

4.4 Discovery Component

The EA for Fuzzy-XCS acts only on the action set. It selects two classifiers with probabilities proportional to their fitness, applies crossover and mutation operators with probabilities χ and μ_{chom} (per chromosome), respectively, and inserts the offspring in the population. If the population contains the maximum number of fuzzy classifiers allowed, two individuals are deleted to make room. They are randomly selected proportionally to the prediction error weighted by the mean action set sizes where each fuzzy classifier was involved.

A simple two-point crossover operator that only acts on the antecedent part of the chromosomes (binary coding scheme) is considered. Prediction, prediction error, and fitness values of offspring are initialized to the mean values of the parents.

The mutation randomly selects an input/output variable of the rule. If an input variable is selected, one of the three following possibilities is applied: *expansion*, which flips to 1 a gene of the selected variable; *contraction*, which flips to 0 a gene of the selected variable; or *shift*, which flips to 0 a gene of the variable and flips to 1 the gene immediately before or after it. The selection of one of these mechanisms is made randomly among the available choices (e.g., contraction can not be applied if only a gene of the selected variable has the allele 1). If an output variable is selected, the mutation operator simply increases or decreases the integer value. Prediction, prediction error, and fitness values of mutated classifiers are not changed.

An EA subsumption is performed. Thus, if the offspring is logically contained by either of its parents and this parent is sufficiently experienced (it has been updated a threshold θ_{GA} number of times), the offspring is not added to the pool but the parent's numerosity is incremented.

When no fuzzy rules cover the state with the highest matching degree, a covering mechanism is used to include a fuzzy classifier with the input linguistic term set that best matches the state and a random action.

5 Experimental Results

Experiments have been performed to test the behavior of Fuzzy-XCS. We have developed a laboratory problem to play a similar role as the multiplexer problem for discrete-valued classifier systems. Thus, we have generated a example data set from a previously defined rule base with different degrees of generalization. Two input variables and one output variable are considered. A total of 576 examples uniformly distributed in the input space (24×24) were generated. Five linguistic terms are considered for each variable. Uniformly distributed triangular-shaped membership functions are used. The rule base considered to generate the data set is shown in Table 3. The reward depends inversely on the difference between the inferred and the desired output in a non-linear way. The objective is to obtain the set of rules that best approximate the data with the highest degree of generalization, i.e., a rule base as accurate and compact as possible.

Table 3: Rule base used to generate the data set

	X_1					X_2					Y				
	VS	S	M	L	VL	VS	S	M	L	VL	VS	S	M	L	VL
R_1	x	x				x	x				x				
R_2	x	x						x	x	x		x			
R_3			x	x	x	x							x		
R_4			x	x	x		x							x	
R_5			x	x	x			x	x	x					x

The used parameter values are the following: maximum number of classifiers = 200, $\beta = 0.2$, $\nu = 3$, $\epsilon_0 = 0.05$, $\alpha = 0.1$, $\theta_{Del} = 30$, $\theta_{GA} = 30$, $\chi = 0.8$, $\mu_{chrom} = 0.1$, $\theta_{timestep} = 20$, $\theta_M = 0.3$, *also* operator = min, and *then* operator = Łukasiewicz. γ parameter is not considered since it is a simple-step problem.

Figure 3 shows the average behavior of 10 runs of Fuzzy-XCS with competitive inference and reinforcement. The upper figure depicts the relative numerosity of the five optimum fuzzy classifiers. It shows the capability of the algorithm to find and keep the op-

timum solution. The bottom figure depicts the mean approximation error of the last 50 iterations. It shows the capability of the algorithm to provide the appropriate action (output) to the corresponding state (input). Note that the optimum solution is properly found.

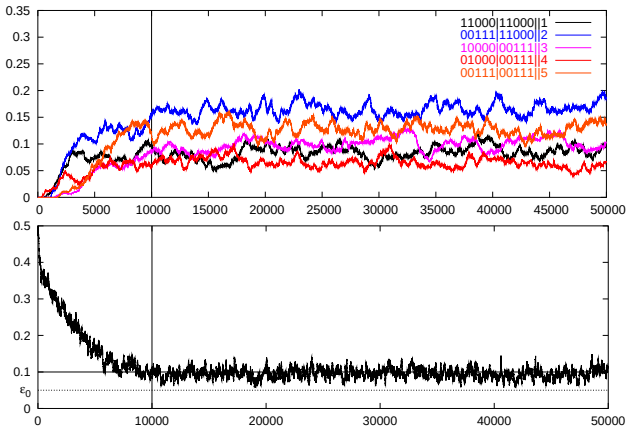


Figure 3: Results of Fuzzy-XCS with *competitive* inference and reinforcement approach

On the other hand, Figure 4 shows the behavior of an algorithm following a cooperative inference and reinforcement. To do that, the only changes made to the algorithm proposed in this paper are the use of *minimum* as implication and *maximum* as aggregation in the inference engine, and a distribution of the reward proportional to the matching degrees in the reinforcement component. This algorithm is tested with a data set generated with the same cooperative inference (Max-Min) to avoid biasing the experiment. Notice that the cooperative approach is unable to find the solution.

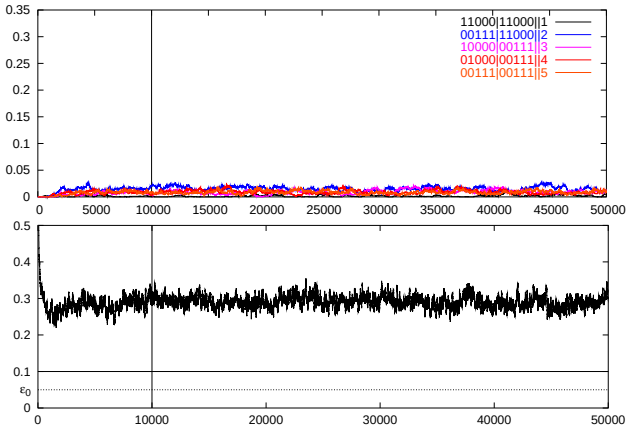


Figure 4: Results of Fuzzy-XCS with *cooperative* inference and reinforcement approach

6 Concluding Remarks

The paper has presented a proposal to properly develop an accuracy-based fuzzy classifier system. It is mainly based on a different inference approach that considers the interaction among fuzzy classifiers (rules) from a competitive point of view. The reinforcement component, based on XCS, is adapted to allow this behavior. The approach has the advantage of performing a higher selection pressure that results in a proper discrimination between good and bad fuzzy classifiers.

Promising results of the proposal have been obtained in a simple laboratory problem. Current and future work involves investigating the behavior of the proposal in multi-step and real-world problems.

References

[1] A. Bonarini. Evolutionary learning of fuzzy rules: competition and cooperation. In W. Pedrycz (Ed.), *Fuzzy modelling: paradigms and practice*, pages 265–284. Kluwer Academic, Norwell, MA, USA, 1996.

[2] A. Bonarini and V. Trianni. Learning fuzzy classifier systems for multi-agent coordination. *Information Sciences*, 136:215–239, 2001.

[3] H. Ishibuchi, T. Nakashima, and T. Murata. Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Trans. Syst., Man, and Cybern.—Part B: Cybern.*, 29:601–608, 1999.

[4] A. Parodi and P. Bonelli. A new approach to fuzzy classifier systems. In *Proc. 5th Int. Conf. Genetic Algorithms*, pp. 223–230, San Mateo, CA, USA, 1993. Morgan Kaufmann Publishers.

[5] M. Valenzuela-Rendón. The fuzzy classifier system: a classifier system for continuously varying variables. In *Proc. 4th Int. Conf. Genetic Algorithms*, pp. 346–353, San Mateo, CA, USA, 1991. Morgan Kaufmann Publishers.

[6] M. Valenzuela-Rendón. Reinforcement learning in the fuzzy classifier system. *Expert Systems with Applications*, 14:237–247, 1998.

[7] J.R. Velasco. Genetic-based on-line learning for fuzzy process control. *Int. J. Intell. Syst.*, 13:891–903, 1998.

[8] S.W. Wilson. Classifier fitness based on accuracy. *Evolutionary Computation*, 3(2):149–175, 1995.

Reducción del número de iteraciones en el algoritmo Fuzzy C-Means

Jesús Soto Espinosa
UCAM*
jsoto@pdi.ucam.edu

Antonio Flores-Sintas
UCAM
aflores@pdi.ucam.edu

Javier Palarea Albaladejo
UCAM
jpalarea@pdi.ucam.edu

M. Isabel Vigo Aguiar
Dpt. de Matemática Aplicada
Universidad de Alicante

Resumen

El algoritmo más utilizado en clasificación difusa es el *fuzzy c-means* (FCM). Este algoritmo depende de varios factores para su utilización, que deben tenerse en cuenta antes de su ejecución. En el presente trabajo vemos cómo podemos variar el algoritmo para que, al tiempo que reducimos las consideraciones previas, reduzcamos el número de iteraciones en su ejecución.

Palabras Clave: Clasificación difusa, FCM.

1 INTRODUCIÓN

Desde que Dunn [2] formalizase el FCM y Bezdek [1] lo generalizó demostrando su convergencia, la mayoría de los algoritmos utilizados en clasificación difusa siguen el mismo propósito de minimizar una función objetivo.

Como lo expresa Xinbo[8], el problema de la clasificación de una muestra $X = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^F$ en c grupos mediante una c-partición difusa, $U = \{\mu_{ik} | 1 \leqslant i \leqslant c, 1 \leqslant k \leqslant n\}$, siendo μ_{ik} el grado de pertenencia de la observación x_k al grupo difuso i, puede formalizarse mediante la obtención de una c-partición difusa óptima mediante la solución de un problema de programación no lineal con restricciones,

$$\min_{(U^*, B^*)} \left\{ \begin{array}{l} J_m = \sum_{i=1}^{c} \sum_{i=k}^{n} (\mu_{ik})^m \cdot D(X_k, \beta_i) + \\ \quad + \zeta, \text{ sujeto a } \ f(\mu_{ik}) \in \mathcal{C} \end{array} \right\} \quad (1)$$

donde ζ es un elemento de penalización, $f(\mu_{ik}) \in \mathcal{C}$ la condición de restricción, m el exponente de "fuzzificación", $B = \{\beta_i | 1 \leqslant i \leqslant c\}$ los prototipos representativos de cada grupo y $D(\cdot)$ el criterio de similaridad o disimilaridad, expresado habitualmente como la distancia entre x_k y el prototipo β_i. La función objetivo está determinada completamente por el conjunto de parámetros $\{U, D(\cdot), B, m, X\}$.

Si elegimos el número c de grupos, $m \in \mathbb{R}_{>1}$, y ε como la precisión exigida para parar el algoritmo, la obtención de la c-partición difusa óptima se realiza mediante el proceso iterativo:

- Inicializar $U^{(0)} = \{\mu_{ij} = \{0, 1\}\}, j = 0$
- Determinar

$$\beta_i^{(0)} = \frac{\sum_{x \in X} \mu_{ik}^m x}{\sum_{x \in X} \mu_{ik}^m} \quad (2)$$

- Repetir
  - Incremento $j = j + 1$,
  - Computar $U^{(j)}$
  - Determinar $\beta_i^{(j)}$
  - Si $||U^{(j-1)} - U^{(j)}|| \leqslant \varepsilon$ parar.

Figura 1: Algoritmo *fuzzy c-means*.

Este algoritmo se conoce con el nombre de *fuzzy c-means*. La forma de calcular $U^{(j)}$ difiere en la literatura existente [1, 5, 3, 4, 6], incluso algunos autores cambian la condición de parada por:

$$\text{Si } \sum_{i=1}^{c} ||\beta_i^{(j-1)} - \beta_i^{(j)}|| \leqslant \varepsilon \text{ parar} \quad (3)$$

$$\text{Si } |J_m^{(j-1)} - J_m^{(j)}| \leqslant \varepsilon \text{ parar.} \quad (4)$$

Como hemos dicho el proceso iterativo es convergente, sin embargo, a la hora de realizar los cómputos

*Departamento de Informática. Universidad Católica San Antonio de Murcia

numéricos también debemos tener en cuenta otros parámetros. Así, según la métrica que se utilice, la muestra X deba normalizarse o no, además habrá que tener en cuenta la precisión que deseemos en el cálculo de

$$||\Gamma^{(j-1)} - \Gamma^{(j)}|| \leqslant \varepsilon,\ \Gamma = \{\beta_i, U, J_m\}, \qquad (5)$$

la cual está relacionada con los elementos de X. Otro criterio para detener el algoritmo es el número de iteraciones realizadas, con el fin de evitar un exceso de computación. En este trabajo pretendemos mostrar cómo podemos reducir el número de iteraciones del algoritmo expuesto en la figura 1. Al tiempo, veremos que la modificación del algoritmo no requiere tener en cuenta tantas consideraciones previas.

2 Problemas detectados

Consideraremos el algoritmo expresado en (2), que suele denominarse *fcm*, con las siguientes expecificaciones: exponente para matriz de la partición U (p.d.: 2.0),máximo número de iteraciones (p.d.: 100), valor de ε (p.d.: 10^{-5}), utilizando además la métrica eucliana y la condición (4). Si aplicamos el algoritmo *fcm* al conocido conjunto de datos de 75 elementos de Ruspini obtenemos el resultado mostrado en la figura 2, donde hemos reflejado en ordenadas el valor de $J_m^{(j)}$ y en abscisas el número de iteraciones necesarias para que se verifique la condición (4).

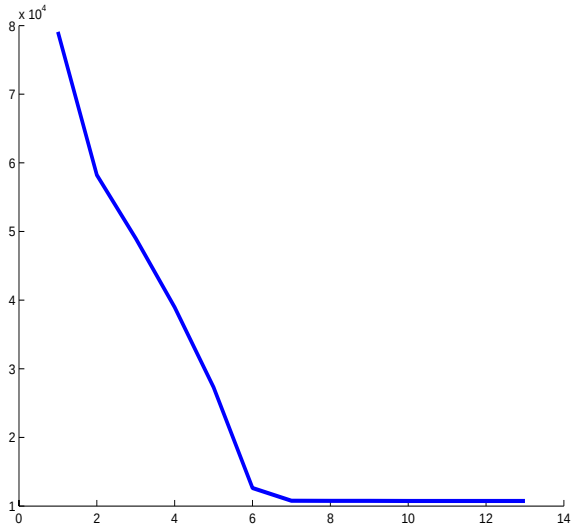


Figura 2: Iteraciones para la muestra Ruspini.

Si escalásemos los datos $X = X \times 1000$ el número de iteraciones sería el de la figura 3, en la que observamos que este aumenta, sin embargo las particiones que da lugar el algoritmo no difieren prácticamente de las obtenidas para los datos sin escalar.

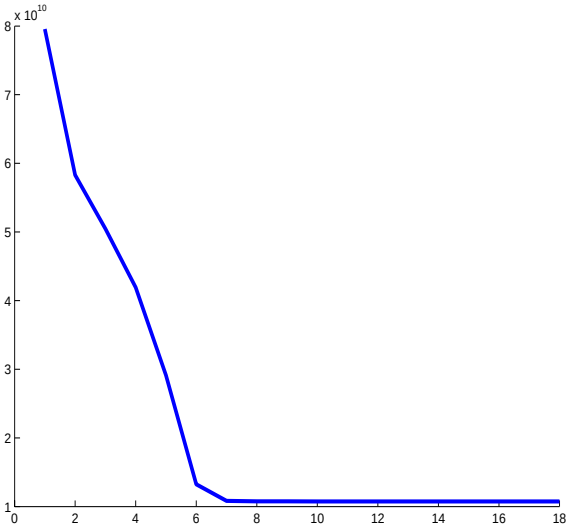


Figura 3: Iteraciones para la muestra Ruspini $\times 1000$.

Conociendo que los datos están escalados sería suficiente con reducir $\varepsilon = 10^{-3}$, para conseguir resultados más cercanos a los de la figura 2 y particiones prácticamente iguales. Entonces, ¿deberíamos saber si los datos están escalados?. No es habitual tener esta información.

Analicemos otro ejemplo. Sea una muestra $X \in \mathbb{R}$ de 16000 registros consistentes en el saldo bancario en pesetas de clientes distintos de una entidad bancaria. Supongamos que deseamos realizar una clasificación en 3 grupos. La diferencia de un cliente a otro en menos de 1.000.000 pesetas no se considera relevante, por tanto fijemos $\varepsilon = 10^4$. Si aplicamos el *fcm* para clasificarlos en los tres grupos obtenemos que el número de iteraciones necesarias es de 97.

Viendo las gráficas de la figura 4 nos preguntamos: ¿cuánto error cometeríamos si hubiésemos detenido el proceso en la iteración 40?, ¿y en la 50?, ¿y en la 60?, ... En este caso el tamaño del dato $x_k \in X$ parece relevante para el número de iteraciones. Es intuitivo observar la relación directa entre el tamaño de $x_k \in X$ y el de ε. Si probásemos otros algoritmos seguiríamos obteniendo conclusiones similares: un estudio previo de los datos puede reducir el número de iteraciones. Sin embargo, nuestro propósito es encontrar un procedimiento mediante el cual no tengamos que cambiar considerablemente las condiciones del algoritmo cada vez que cambiemos la muestra a clasificar.

3 Reducción de las iteraciones

Consideremos la siguiente sucesión $\sigma_j = J_m^{(j)} \in \mathbb{R}$. Sabemos por la convergencia del algoritmo *fuzzy c-*

means [1] que

$$\lim_{j\to\infty} \frac{\sigma_j}{\sigma_{j-1}} = 1, \tag{6}$$

por lo que

$$\lim_{j\to\infty} \left(\frac{\sigma_j}{\sigma_{j-1}} - \frac{\sigma_{j-1}}{\sigma_{j-2}} \right) = 0. \tag{7}$$

Consideremos la sucesión

$$\psi_j = \left(\frac{\sigma_j}{\sigma_{j-1}} - \frac{\sigma_{j-1}}{\sigma_{j-2}} \right), \tag{8}$$

ahora podemos sustituir la condición (4) por

$$\text{If } |\psi_j| \leqslant \varepsilon \text{ stop} \tag{9}$$

en el algoritmo *fuzzy c-means*.

Para probar la condición (9) hemos utilizado los datos anteriores con las mismas especificaciones para todas las muestras: exponente para matriz de la partición U (p.d.: 2.0),máximo número de iteraciones (p.d.: 100), valor de ε (p.d.: 10^{-5}), y métrica euclidiana. Las gráficas de la figura 5 muestran como el nuevo procedimiento reduce las iteraciones, de manera considerable en el último ejemplo. Es más, si consideramos los prototipos obtenidos mediante este procedimiento en los ejemplos de Ruspini y Ruspini×1000, la diferencia con los dados por el *fuzzy c-means* con la condición (4) es de orden menor de 10^{-6}. Y por el caso de la muestra Saldos, como se observa en la tabla 1, la diferencia no

Tabla 1: Prototipos de la muestra Saldos.

fuzzy c-means	Prototipos
condición (4)	$10^7 \times [0.7720,\ 6.5413,\ 2.1060]$
condición (9)	$10^7 \times [0.7703,\ 6.4762,\ 2.0896]$

es significativa.

Además, si los datos estuviesen dados en euros, sería lo mismo que considerar Saldos/166.386, en cuyo caso el nuevo procedimiento mantendría las iteraciones con pequeñas diferencias (±2 en las pruebas) y los prototipos obtenidos

$$10^5 \times [0.4629,\ 3.8920,\ 1.2558]$$

varían muy poco del resultado

$$10^7 \times [0.7703,\ 6.4762,\ 2.0896]/166.386. \tag{10}$$

Otro ejemplo puede verse con los datos dados en Pima Indians Diabetes Database [7], una base de datos de 768 registros con 8 campos clave. En este ejemplo repetimos 5 veces el proceso de calcular una partición de cuatro conjuntos difusos de los 768 registros. Aquí la complejidad radica en el número de variables de cada dato: 8. Bajo las mismas especificaciones que las expuestas anteriormente, hemos calculado la media de las iteraciones, y la media de la diferencia al cuadrado entre los prototipos:

$$\sum_{1}^{4}\sum_{1}^{8}(Prot_cond(4) - Prot_cond(9))^2 \tag{11}$$

El resultado puede verse en la tabla 2.

Tabla 2: Comparativa del ejemplo Pima.

fuzzy c-means	Iteraciones
condición (4)	71
condición (9)	34
Diferencia de prototipos(11)	
4.0593	

4 Conclusión

Hemos visto que el algoritmo *fuzzy c-means* con la condición (4) necesita especificar una serie de parámetros a la hora de aplicarlo a una muestra. Considerando que la métrica utilizada, ya sea euclidea ([2, 1]), de Mahalanobis ([5, 3]), o de otro tipo ([4, 6]), y la función de pertenencia que utiliza el algoritmo están elegidas previamente, son las opciones exponente para matriz de la partición U, máximo número de iteraciones y valor de ε las que tendremos en cuenta. Habitualmente el exponente para la matriz de la partición U es 2 y el máximo número de iteraciones podemos fijarlo sin perdida de generalidad. Es por tanto el parámetro ε el que puede modificar los resultados dependiendo de como se presenten los datos en las muestras, como vimos en la sección anterior.

En este trabajo hemos mostrado que si sustituimos la condición (4) por

$$\text{If } |\psi_j| \leqslant \varepsilon \text{ stop}, \tag{12}$$

reduce el número de iteraciones del algoritmo, y problemas como el de escalabilidad no afecta considerablemente a los resultados, ofreciendo resultados igualmente aceptables para distintas muestras sin necesidad de modificar ε.

Referencias

[1] J.C. Bezdek. A convergence theorem for the fuzzy isodata clustering algorithms. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-2:1–8, 1980.

[2] J. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well separated cluster. *J. Cybern.*, 3:32–57, 1974.

[3] A. Flores-Sintas, J.M. Cadenas, and F. Martin. Membership functions in the fuzzy c-means algorithm. *Fuzzy Sets and Systems*, 101:49–58, 1999.

[4] I Gath and A.B. Geva. Unsupervised optimal fuzzy clustering. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):773–781, 1989.

[5] E.E. Gustafson and W.C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *Proc. IEEE Conf. on Decision and Control*, pages 761–6, January 1979.

[6] M. Ménard and M. Eboveya. Extrema physical information and objetive function in fuzzy clustering. *Fuzzy Sets and Systems*, 128(3), 2002.

[7] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R.S. Johannes. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Symposium on Computer Applications and Medical Care*, pages 261–265. IEEE Computer Society Press, 1988.

[8] Gao Xinbo and Xie Weixin. Advances in theory and applications of fuzzy clustering. *Chinese Science Bulletin*, 45(11):961, June 2000.

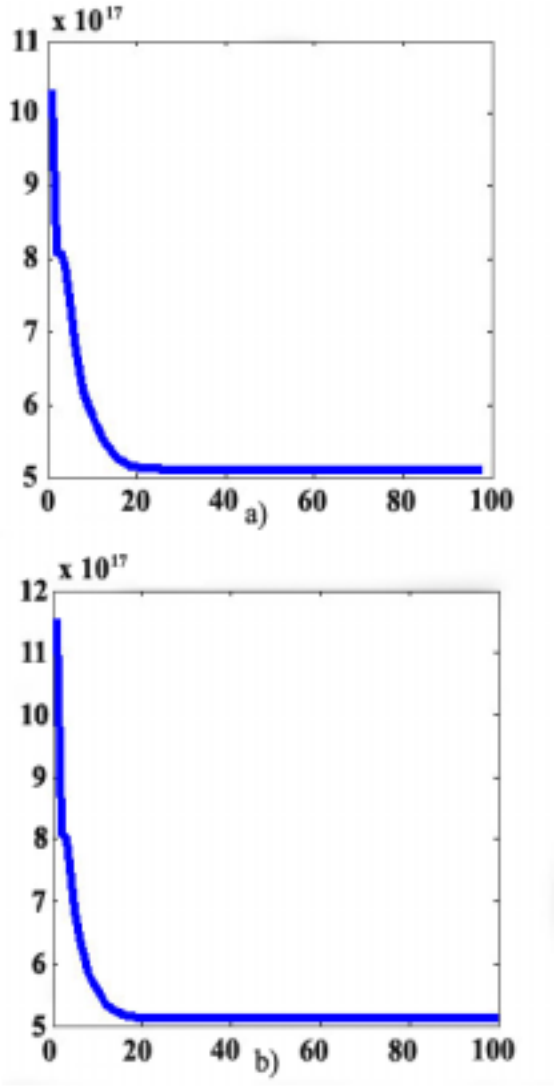


Figura 4: Iteraciones para la muestra de saldos. La gráfica a) muestra los resultados cuando $\varepsilon = 10^4$. En b) consideramos $\varepsilon = 10^{-5}$. Observar que los resultados son muy similares.

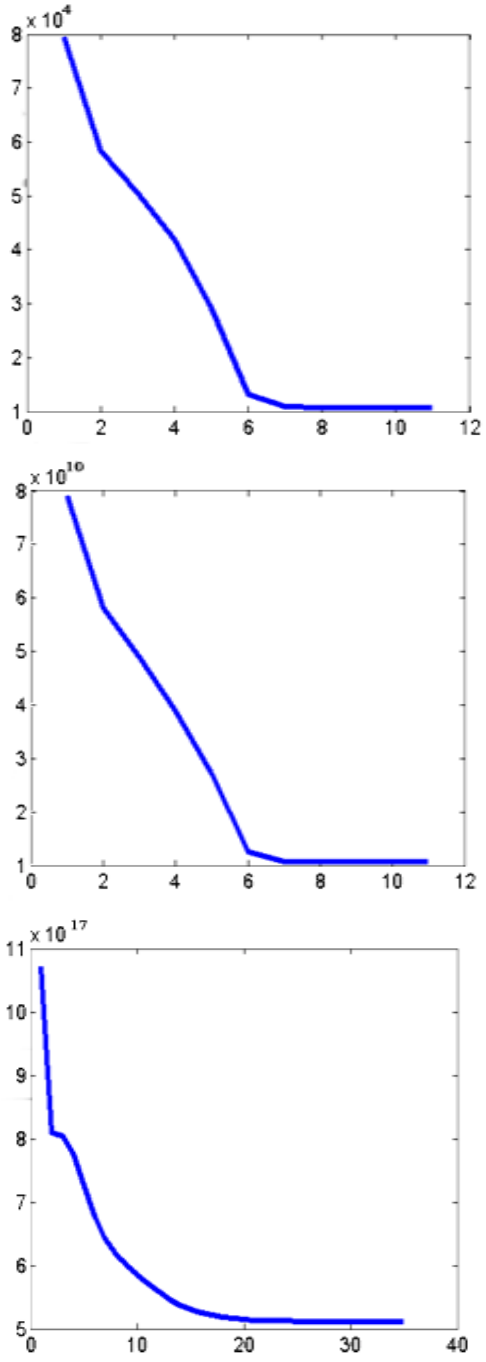


Figura 5: Iteraciones con la condición (9) correspondientes a los datos de Ruspini, Ruspini×1000 y Saldos.

DETECCIÓN DE PUERTAS EN ROBÓTICA MÓVIL MEDIANTE REGLAS TEMPORALES BORROSAS

P. Cariñena
Grupo de Sistemas Inteligentes
Dep. Electrónica y Computación
Univ. Santiago de Compostela
puri@dec.usc.es

C.V. Regueiro
Dep. Electrónica y Sistemas
Universidad da Coruña
cvazquez@udc.es

A. Bugarín **S. Barro**
Grupo de Sistemas Inteligentes
Dep. Electrónica y Computación
Univ. Santiago de Compostela
{alberto,senen}@dec.usc.es

Resumen

Este trabajo describe un nuevo mecanismo de detección de puertas a partir de los datos de ultrasonidos de un robot móvil que se desplaza a lo largo de un pasillo. Se ha utilizado el paradigma de reglas temporales borrosas para modelar el conocimiento necesario para la detección considerando la variación temporal de las señales de los sensores. Usando los robots *Nomad 200* y *Pioneer 2AT* en entornos reales se han obtenido unos elevados porcentajes de puertas detectadas, superiores al 79 %. El mecanismo utilizado se puede extender fácilmente para identificar otros tipos de marcas.

Palabras Clave: Detección de marcas, Robótica móvil, Reglas Temporales Borrosas.

1. INTRODUCCIÓN

En muchas tareas en robótica móvil resulta de interés la detección de las distintas características del entorno, bien para elaborar un mapa, bien para situar al robot. Una característica muy frecuente en el interior de un edificio son las puertas que, además de usarse para localizar el robot, también sirven para la planificación y ejecución de rutas [7, 8] (por ejemplo, "seguir la pared hasta la tercera puerta").

En robótica se han utilizado diferentes tipos de sensores para la detección de marcas en general y de puertas en particular. Sin embargo, prácticamente todos los robots comerciales están equipados con sensores de ultrasonidos, que proporcionan una medida de la distancia entre ellos y los objetos en el entorno. Su amplia utilización se debe principalmente a su bajo coste, su simplicidad, fácil instalación y bajo consumo. Además, son capaces de cubrir todo el espacio que rodea al robot y la información que proporcionan puede ser procesada en tiempo real.

La principal desventaja de estos sensores es el ruido y la incertidumbre de las medidas, debido a su baja resolución angular, al "cross-talking" entre sensores, a reflexiones especulares y a la recepción de falsos ecos. Por todos estos motivos, la incertidumbre en las mediciones sensoriales del robot es un aspecto fundamental que no se puede soslayar, lo que ha hecho de las reglas borrosas un mecanismo de gran utilidad dentro de este ámbito, en la medida que permiten modelar adecuadamente el conocimiento en situaciones de imprecisión como las indicadas.

En el caso de la detección de puertas, la complejidad del conocimiento a modelar hace que los modelos convencionales de reglas borrosas no ofrezcan suficiente flexibilidad. Por ello, nos hemos planteado la utilización del paradigma de Reglas Temporales Borrosas (RTBs) introducido en [1, 2]. Este modelo (en [1] puede verse una descripción detallada) contempla la posibilidad de establecer relaciones complejas de dependencia entre la ocurrencia de eventos, operadores que dotan a las proposiciones de una mayor riqueza expresiva que la habitual y permite manejar persistencia dentro de las referencias temporales.

1.1. TRABAJOS RELACIONADOS

No existe apenas bibliografía relacionada con la tarea de detección de puertas utilizando únicamente señales de ultrasonidos. En [4] se propone un método basado en las marcas de las señales de ultrasonidos en la interfaz gráfica del robot. Para la aplicación de este mecanismo de detección se ha diseñado un sistema experto que trata de reproducir el modo de operación de un humano cuando detecta visualmente una puerta en la interfaz gráfica. En este sistema un cambio en la morfología a detectar obligaría a rediseñar el sistema.

En [6] se realiza una detección de puertas analizando la morfología de las señales de ultrasonidos, mediante un

modelo de Perfiles Temporales Borrosos Multivariable. La presencia de una puerta se corresponde con determinados perfiles en un par de señales de ultrasonidos (las situadas a 45° respecto a la pared en que se encuentra la posible puerta). En este modelo la búsqueda de una morfología diferente únicamente requeriría una redefinición de las restricciones entre los puntos temporales que definen cada perfil. Sin embargo, las variaciones en el perfil ideal (por ejemplo, en puertas abiertas o cuando gira el robot) pueden hacer fracasar el proceso de detección.

2. MÉTODO DE DETECCIÓN DE PUERTAS

En primer lugar introduciremos el conocimiento involucrado en el sistema de detección, y mostraremos cómo puede ser modelado mediante RTBs.

2.1. DESCRIPCIÓN DEL PROBLEMA

Para centrar la discusión, supongamos la situación descrita en la Fig. 1[1]. A medida que el robot avanza a lo largo del pasillo (de derecha a izquierda), los dos sensores situados aproximadamente a 45° con respecto a la perpendicular a la pared a cada lado (2 y 6 para la pared de la izquierda, 10 y 14 para la derecha) miden una distancia aproximadamente constante mientras el robot se mantenga paralelo a la pared.

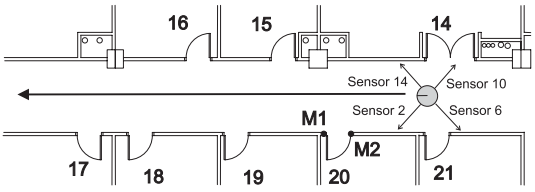


Figura 1: Detección de una puerta mediante los sensores de ultrasonidos.

Sin embargo, cuando el robot se aproxima a una puerta en, por ejemplo, la pared de la izquierda (puerta 20) el sensor delantero (sensor 2) deja de detectar la pared y comienza a detectar el marco M1; esto produce una subida brusca en la señal de ultrasonidos en el instante t_r (ver Fig. 2). La altura del pico depende de la anchura de la puerta (alrededor de 70 cm para una puerta estándar). A este pico le sigue una caída suave a medida que el robot se aproxima al marco M1, regresando la señal a su valor original cuando pasa a ser reflejada de nuevo por la pared.

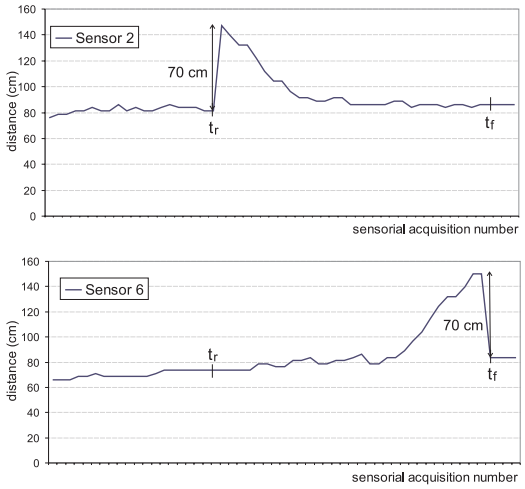


Figura 2: Patrón producido en las señales de ultrasonidos cuando el robot pasa paralelo a una puerta.

Poco después ocurre una situación simétrica con el sensor posterior (sensor 6): deja de detectar la pared y empieza a ser reflejado por el marco M2, produciéndose una subida suave a medida que el robot se aleja de M2. Cuando detecta de nuevo la pared se produce un descenso brusco en la señal en el instante t_f (Fig. 2) recuperando el valor inicial. Una descripción más detallada de este proceso puede verse en [3].

Este comportamiento permite detectar de manera automática las puertas en el recorrido del robot: la detección de una subida brusca en la señal del sensor 2 (S2), seguida poco después de una caída brusca en la señal del sensor 6 (S6), indica que hay una puerta en la pared de la izquierda del robot. De la misma forma se pueden detectar las puertas a la derecha del robot, utilizando los sensores simétricos (10 y 14).

Otras situaciones también pueden producir caídas o subidas bruscas en las señales de ultrasonidos (giros del robot, columnas, bancos, etc.), aunque generalmente no verificarán la relación temporal "poco después". Cuando sí la cumplan podemos tratar de eliminar los falsos positivos atendiendo a la forma que presentan las señales de ultrasonidos en presencia de una puerta. Así, en la Fig. 2 se observa que tanto la caída brusca como la subida brusca se producen respecto a un valor de referencia: la distancia a la que se detecta la pared. Analíticamente esta relación se puede expresar como:

$$S2(t_r) \approx S6(t_f) \approx S6(t_r) \approx S2(t_f)$$

[1]Pasillo en el departamento de Electrónica y Computación de la Universidad de Santiago de Compostela. La numeración de las puertas se corresponde con los números reales de despachos y laboratorios, y serán éstos los que utilizaremos para identificar cada una de las puertas.

2.2. MODELADO CON RTBs

Este método de detección de puertas requiere el modelado de la relación temporal entre detecciones consecutivas de los eventos correspondientes y de la relación de valor entre las señales, aspectos cubiertos dentro del modelo de RTBs [1]. Traduciendo el conocimiento involucrado en la detección a este modelo obtenemos las reglas para detectar las puertas a la izquierda y a la derecha del robot. En ellas se toma como relación temporal entre los eventos "subida brusca" y "bajada brusca" la relación "pocos instantes antes", pues se trata de razonar a partir de un valor actual en uno de los sensores y valores previos en el otro. Utilizar la relación simétrica "pocos instantes después" supondría la necesidad de valores futuros, aún no disponibles si la regla se ejecuta en el instante actual.

La regla para detectar las **puertas a la izquierda** es:

SI $S2$ *sube bruscamente pocos instantes antes de que* $S6$ *baje bruscamente*
Y $S6$ *justo después de que* $S6$ *baje bruscamente es similar a* $S2$ *justo antes de que* $S2$ *suba bruscamente*
Y $S6$ *justo antes de que* $S2$ *suba bruscamente es similar a* $S2$ *justo después de que* $S6$ *baje bruscamente*
Y $S2$ *justo después de que* $S6$ *baje bruscamente es similar a* $S2$ *justo antes de que* $S2$ *suba bruscamente*
ENTONCES *hay una puerta en la pared a la izquierda del robot.*

La regla para detectar puertas a la derecha es análoga, cambiando $S2$ por $S14$ y $S6$ por $S10$.

Los eventos *subida brusca* y *bajada brusca* se definen por diferencias entre el valor de las señales en un punto temporal y el valor dos muestras antes. Las relaciones "*justo después*" y "*justo antes*" se definen de manera nítida como los puntos temporales situados una unidad por delante o por detrás, respectivamente.

La Fig. 3 muestra la la distribución de posibilidad asociada a la relación espacial "*similar a*", en la que se asume que las diferencias entre los valores de las señales no deben superar los 30 cm para ser considerados similares: si la diferencia es mayor, el patrón detectado se descarta como correspondiente a una puerta.

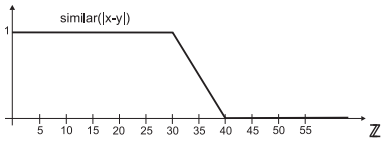


Figura 3: Definición borrosa de la relación espacial "*similar a*".

La distancia temporal entre los patrones correspondientes a los dos marcos de cada puerta depende de la velocidad del robot. Puesto que ésta no es constante, sino que se va adaptando a las características del entorno [5], tomamos como velocidad de referencia la que lleva el robot en el momento de aproximarse a la puerta. La velocidad instantánea del robot presenta importantes oscilaciones, por lo que resulta más adecuado considerar la velocidad media en las proximidades de la puerta que se pretende detectar. El modelo de RTBs también contempla la inclusión de operadores borrosos sobre valores de señal; así, la proposición temporal borrosa que describe esta situación es:

El valor medio de la velocidad del robot en los instantes previos a la bajada brusca de S.

siendo $S = S6$ para las puertas a la izquierda, y $S = S14$ para las puertas a la derecha.

La restricción temporal "*en los instantes previos*" se define mediante la distribución de posibilidad mostrada en la Fig. 4, por lo que a la hora de aplicar el operador "valor medio" deberá considerarse un modelado borroso del mismo. Una vez evaluada la proposición sobre la velocidad se define la relación temporal "*pocos instantes antes*" como función de la velocidad media.

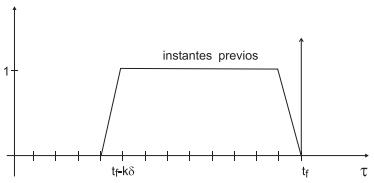


Figura 4: Definición borrosa de la restricción temporal "*en los instantes previos*".

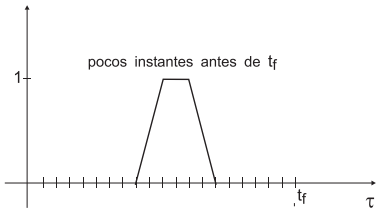


Figura 5: Definición borrosa de la relación temporal "*pocos instantes antes*".

De esta forma, el proceso de detección de una puerta comienza con la detección del posible final de la misma. Una vez detectado éste (mediante una bajada brusca en la señal del sensor correspondiente), se busca en el histórico de valores del otro sensor una subida brusca que verifique la relación temporal "pocos instantes antes". Si se encuentra esta subida entonces se habrá detectado una puerta.

3. RESULTADOS

3.1. AJUSTE DE PARÁMETROS

Con objeto de calibrar el modelo y establecer los parámetros para las definiciones de las relaciones *similar a*, *pocos instantes antes* o *en los instantes previos*, se usaron las lecturas de ultrasonidos tomadas en una situación "ideal", con recorridos en los que se mantuvo relativamente baja la velocidad del robot (15 cm/s). El control de la trayectoria se realizó de manera manual, intentando que en todo momento fuera lo más recta y centrada posible a lo largo del pasillo (es decir, que la distancia a las paredes derecha e izquierda fuera la misma: aproximadamente 100 cm). De esta forma, se detectan con igual precisión las puertas situadas a ambos lados del pasillo.

En esta situación, los patrones correspondientes a las puertas se ajustan al patrón "ideal" mostrado en la Fig. 2, lo que facilita el ajuste de los valores de los parámetros. En situaciones reales de movimiento del robot los patrones, como veremos en el apartado siguiente, son más ruidosos, con lo que el ajuste heurístico de los parámetros sería mucho más dificultoso.

3.2. VALIDACIÓN

Se han realizado pruebas con dos robots diferentes (*Nomad 200*, *Pioneer 2AT*), a lo largo de distintos tramos de pasillo con objeto de cubrir un abanico amplio de situaciones: puertas juntas o separadas, puertas dobles, puertas cerca de esquinas y giros, etc. Debido al gran ancho del cono de emisión de las ondas de ultrasonidos, la detección de objetos es tanto más precisa cuanto más próximos se encuentren. Por ello, se han realizado pruebas en las que el robot ejecuta una tarea de seguir contorno [5], de forma que la distancia a la pared que se sigue sea la mínima posible, mejorando la detección de las puertas en esa pared. Por supuesto, ello conlleva un empeoramiento en los resultados para la pared opuesta. Se han realizado las pruebas a distintas velocidades, e incluyendo situaciones de giros y cambios de sentido del robot.

A modo de ejemplo, la Fig. 6 ilustra un recorrido en el que el robot sigue el contorno derecho. La Fig. 7 muestra la traza del robot, así como la posición de las puertas detectadas en la pared de la derecha y de la izquierda, marcadas a través de su punto inicial (x) y final (o). En la Fig. 8 se muestran las señales de ultrasonidos procedentes de los sensores 10 y 14, sobre la pared a la derecha del robot, y de los sensores 2 y 6, sobre la pared de la izquierda, para este recorrido. Sobre las señales se indican los patrones correspondientes a cada una de las puertas.

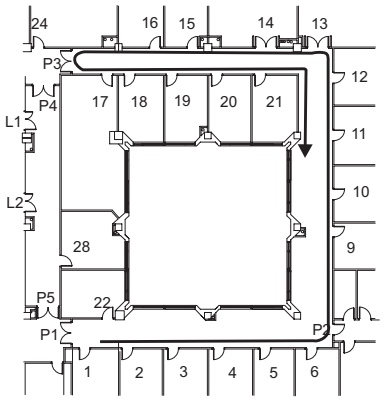


Figura 6: Recorrido siguiendo el contorno derecho.

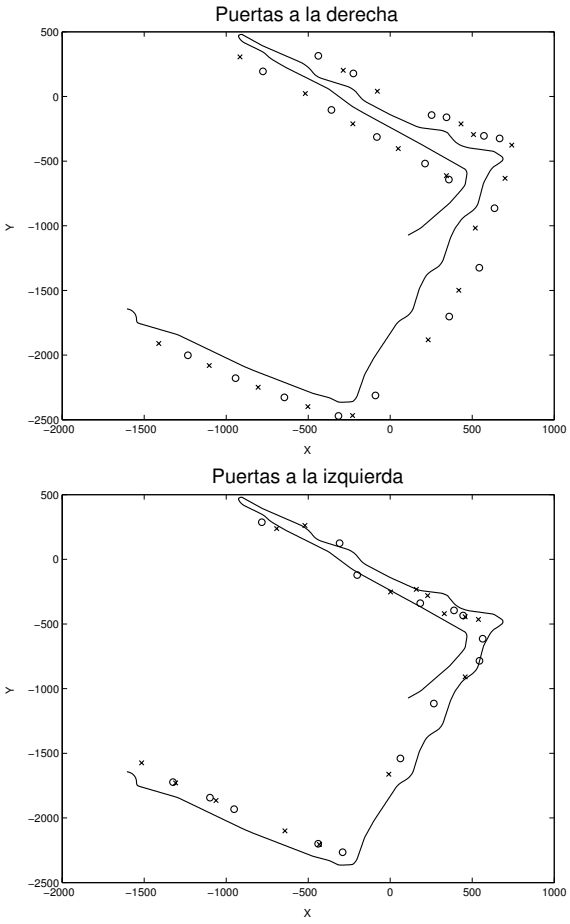


Figura 7: Traza del robot y posición de las puertas detectadas a la derecha (mismo lado) y a la izquierda (lado contrario), para el recorrido de la Fig. 6.

Como puede apreciarse, no todas las puertas producen el patrón "ideal" mostrado en la Fig. 2. Puede suceder que el eco del ultrasonido no se refleje completamente en los marcos, debido a cambios en la orientación entre

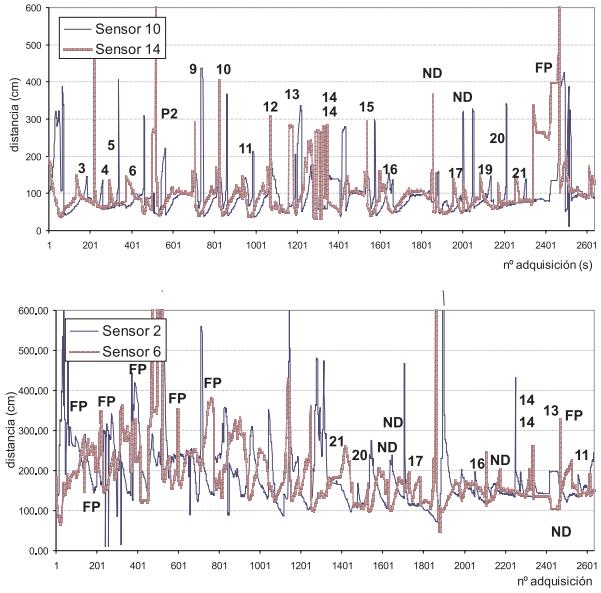


Figura 8: Puertas detectadas en el recorrido de la Fig. 6: sensores 10 y 14 (puertas a la derecha), sensores 2 y 6 (puertas a la izquierda).

robot y puerta (de forma que el ángulo entre sensor y puerta es mayor que 45°) o a que la puerta está abierta. Ello origina picos mayores que los 70 cm asumidos para una puerta estándar (por ejemplo, puertas 9 y 10). Sin embargo, esto no representa un problema ya que los picos serán igualmente detectados.

En la pared lejana (la izquierda, parte inferior de la Fig. 8) los 6 primeros falsos positivos se corresponden a un tramo de pasillo en el que no hay puertas y la pared es de cristal. El cambio de material, unido a la mayor distancia, hace que los sensores de ultrasonidos proporcionen medidas mucho más ruidosas y que el sistema de detección falle. En concreto, este pasillo es el más ancho de los tres tramos en este recorrido (unos 3 m) por lo que la distancia al contorno contrario es cercana a los 2 m, incrementándose la incertidumbre debida a la imprecisión angular de los ultrasonidos.

3.3. DISCUSIÓN

La tabla 1 recoge el número de puertas reales, las puertas detectadas, los aciertos, los falsos positivos y las puertas no detectadas (puertas reales menos aciertos) para las pruebas realizadas con los robots *Nomad 200* y *Pioneer 2AT*, respectivamente, relativos al contorno que se está siguiendo. El porcentaje de aciertos se calcula sobre el número de puertas reales y el porcentaje de falsos positivos sobre el número de puertas detectadas. En la tabla 2 se resumen los datos sobre la detección en el contorno opuesto para el *Nomad 200*.

Tabla 1: Detección de puertas durante la tarea de seguir contorno: contorno de referencia.

Robot	PR	PD	DC	ND	FP	%DC	%FP
Nomad	107	115	97	10	18	91	16
Pioneer	95	96	75	21	22	79	22

PR: Puertas Reales; PD: Puertas Detectadas; DC: Detecciones Correctas; ND: Puertas No Detectadas; FP: Falsos Positivos

Tabla 2: Detección de puertas durante la tarea de seguir contorno: contorno opuesto (*Nomad 200*).

Robot	PR	PD	DC	ND	FP	% DC	% FP
Nomad	65	99	51	14	48	78	48

PR: Puertas Reales; PD: Puertas Detectadas; DC: Detecciones Correctas; ND: Puertas No Detectadas; FP: Falsos Positivos

Analizando estos resultados podemos observar cómo sobre el contorno que se está siguiendo se detectan prácticamente todas las puertas reales (el porcentaje global de puertas detectadas correctamente es del 91 % en el *Nomad 200* y del 79 % en el *Pioneer 2AT*), con un bajo número de falsos positivos en relación al número total de puertas detectadas (16 % y 22 %, respectivamente). En la tabla 2 puede verse cómo el porcentaje de acierto disminuye al considerar el contorno contrario (78 %). También aumenta de forma significativa el número de falsos positivos, que sube a un 48 %.

La principal causa de esta bajada en el rendimiento es que el haz de las señales de ultrasonidos forma un cono de unos 30°, por lo que a una cierta distancia el frente de onda puede ser mucho más ancho que la puerta que se quiere detectar, lo que hace imposible su detección. Así, cuando el robot sigue un contorno lo hace a una distancia de entre 50 y 70 cm, mientras que la distancia a la pared opuesta puede variar entre 100 y 200 cm (dependiendo del ancho del pasillo). Otra causa relativamente frecuente del incremento de los falsos positivos son las paredes compuestas de varios materiales (por ejemplo, paredes acristaladas del primer pasillo en la Fig. 6) que generan una gran cantidad de ruido que no siempre se puede filtrar adecuadamente.

Para comprobar la robustez del método hemos realizado alguno de los recorridos con los dos robots. La tabla 3 muestra los resultados obtenidos en un recorrido de ida y vuelta sobre el pasillo de la Fig. 1 (8 puertas reales). Puede apreciarse que los resultados son

prácticamente independientes del robot, dependiendo más de aspectos como la velocidad y la realización con mayor o menor rigor de la tarea de seguir contorno.

Tabla 3: Detección de puertas en el pasillo de la Fig. 1.

Robot	Rec.	PD	DC	ND	FP
Nomad	A	9	7	1	2
Nomad	B	8	6	2	2
Nomad	C	8	7	1	1
Pioneer	D	8	7	1	2
Pioneer	E	9	8	0	1

Rec.: Recorrido; PD: Puertas Detectadas; DC: Detecciones Correctas; ND: Puertas No Detectadas; FP: Falsos Positivos

4. CONCLUSIONES

En este trabajo se ha desarrollado un sistema de detección de puertas en robótica móvil para entornos del interior de edificios. El sistema es muy preciso cuando el robot se desplaza a una distancia no muy elevada de la pared sobre la que se pretende detectar las puertas, lo cual se produce con frecuencia, principalmente en el seguimiento de contornos. Las pruebas experimentales realizadas en entornos reales con dos robots comerciales (*Pioneer 2AT* y *Nomad 200*) han arrojado unos porcentajes de acierto en la detección de las puertas en el contorno que sigue el robot muy altos (entre el 79 % y el 91 % según el robot empleado).

Además de la distancia a la pared en donde se encuentra la puerta existen otros aspectos que influyen en la calidad de la detección. En primer lugar, la velocidad del robot: a mayor velocidad, mayor porcentaje de falsos positivos. Por otra parte, las oscilaciones en la trayectoria del robot (es decir, las trazas sinuosas) también dificultan el proceso de detección, ya que nos separamos de la premisa inicial de que el robot se debe desplazar casi paralelo a la puerta a detectar.

Si la detección de puertas se incluye en una tarea de construcción de un mapa del entorno, entonces se debería manejar adecuadamente tanto los falsos positivos como las puertas no detectadas. En este sentido la presencia de incertidumbre y ambigüedad en la construcción de mapas nos lleva a pensar que el modelo de RTBs también puede ser apropiado para abordar esta tarea de alto nivel.

Una línea inmediata de avance a partir de este método de detección es la consideración de otras marcas en el entorno, por ejemplo las columnas o las esquinas. En cualquier caso, es evidente que las limitaciones de los sensores de ultrasonidos podrían hacer necesarios otros sensores más precisos, como láser o visión artificial.

Agradecimientos

Los autores agradecen el apoyo de la Comisión Interministerial de Ciencia y Tecnología a través de los proyectos TIC-2000-0873-C02-01 y TIC2003-09400-C04-03.

Referencias

[1] P. Cariñena, *Un modelo de Reglas Temporales Borrosas para el razonamiento sobre sistemas dinámicos*, Tesis Doctoral, Universidad de Santiago de Compostela, 2003.

[2] P. Cariñena, A. Bugarín, M. Mucientes, F. Díaz-Hermida y S. Barro, *Technologies for Constructing Intelligent Systems 2. Tools, Studies in Fuzziness and Soft Computing*, cap. "Fuzzy Temporal Rules: A Rule-based Approach for Fuzzy Temporal Knowledge Representation and Reasoning", pp. 237–250, Springer-Verlag, 2002.

[3] P. Cariñena, C.V. Regueiro, A. Otero, A. Bugarín y S. Barro, "Landmark detection in mobile robotics using fuzzy temporal rules", *IEEE Trans. Fuzzy Systems*. Special Issue on Advances of Fuzzy Systems in Robotics, Agosto, 2004.

[4] M.R. Masliah y R.W. Albrecht, "The mobile robot surrogate method for developing autonomy," *IEEE Trans. Robotics and Automation* **14** (2), pp. 314–320, 1998.

[5] M. Mucientes, R. Iglesias, C.V. Regueiro, A. Bugarín y S. Barro, "A fuzzy temporal rule-based velocity controller for mobile robotics," *Fuzzy Sets and Systems* **134** (3) Special Issue: Fuzzy Set Techniques for Intelligent Robotic Systems, pp. 83–99, 2003.

[6] A. Otero, P. Félix, C.V. Regueiro, M. Rodríguez y S. Barro, "A model to perform knowledge-based temporal abstraction over multiple signals," *Proc. TIME-ICTL 2003*, pp. 128–136, IEEE Press.

[7] C.V. Regueiro, M. Rodríguez, J. Correa, D.L. Moreno, R. Iglesias y S. Barro, "A control architecture for mobile robotics based on specialists," in *Control and Electric Power Systems*, C.T. Leondes, Ed., vol. 6 of *Intelligent Systems: Technology and Applications*, pp. 337–360. CRC Press, 2003.

[8] P. Veelaert y H. Peremans, "Flexibility maps: A formalisation of navigation behaviours," *Robotics and Autonomous Systems*, vol. 27, pp. 151–169, 1999.

Detección de puertas mediante visión y lógica difusa *

Eugenio Aguirre, Miguel García-Silvente, Antonio González, Rafael Muñoz
Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S. de Ingeniería Informática. Universidad de Granada
18071 - Granada tlfno.: +34.958.240807 fax: +34.958.243317
e-mail: {eaguirre, mgs, gonzalez, salinas}@decsai.ugr.es

Resumen

Uno de los métodos existentes para desarrollar la navegación de un robot en un entorno de interiores, consiste en la utilización de un mapa topológico del mismo y una serie de comportamientos de navegación adecuados. Un elemento muy común en este tipo de entornos son las puertas, cuya detección nos facilitará enormemente tanto conocer su estructura, como explorarlo y llevar a cabo una navegación eficiente. En este trabajo utilizamos la información suministrada por la cámara de un robot para que éste pueda establecer un nivel de creencia sobre la existencia de un marco de una puerta. Para ello se definen las características que deben tener los segmentos detectados en la imagen para pertenecer al marco. Este proceso se realiza con reglas expresadas mediante lógica difusa. Esto nos permite tanto poder medir la creencia del robot como manejar imágenes en las que las puertas se ven desde diferente perspectiva y con distintos tamaños y condiciones de luminosidad. Se muestran algunos experimentos para validar la propuesta.

Palabras Clave: Robots móviles, visión artificial, modelos perceptuales difusos.

1. Introducción

En el área de la robótica móvil autónoma, el robot debe ser capaz de extraer información útil a partir de su sistema sensorial a pesar de la existencia de ruido e imprecisión en los datos de los sensores. Esta información debe ser adecuada para ayudar al robot a realizar la tarea que tiene encomendada. En el caso de la navegación por entornos de interior el robot debe alcanzar un destino concreto a partir de su situación original. Una de las formas de resolver este problema es mediante la utilización de un mapa topológico [13] y el empleo de una arquitectura de navegación basada en comportamientos [1, 2, 4]. Para ello es necesario contar con modelos perceptuales que permitan al robot detectar determinados lugares distinguidos que formarán parte del mapa topológico, ya sea para construir dicho mapa o para llevar a cabo la fase de navegación propiamente dicha. En trabajos anteriores [3] hemos desarrollado un modelo perceptual basado en ultrasonidos mediante el cual el robot puede establecer estas creencias en ciertos objetos como paredes, esquinas, pasillos o puertas. Este modelo utiliza la lógica difusa como herramienta para el manejo de la imprecisión e incertidumbre así como para la propia definición de estos objetos.

En este trabajo nos proponemos utilizar la información que nos suministra una cámara situada en nuestro robot Nomad 200, para poder establecer visualmente un nivel de creencia en la presencia del marco de una puerta en un entorno de interiores. En posteriores trabajos se integrará este nuevo valor de creencia, obtenido mediante visión, con otros valores que el robot puede obtener aplicando el modelo perceptual anteriormente comentado.

El problema de la detección visual de puertas ha sido abordado por diversos autores empleando distintas técnicas. En el trabajo desarrollado por [7], dos redes neuronales son utilizadas para la detección de puertas en imágenes en color. Una red es utilizada para detectar los marcos laterales, mientras que la otra se utiliza para detectar el marco superior. Las redes reciben como entrada ventanas de 18x18 píxeles centradas en cada píxel de la imagen con las componentes de brillo y saturación. En total cada red cuenta con una capa de entrada de 648 elementos, una intermedia y una de salida que indica el grado en que el píxel analizado pertenece al marco de una puerta. Tras este proceso de clasificación de cada píxel de la imagen, se realiza un proceso de agrupación para determinar si hay puerta en la imagen. El sistema es capaz de detectar puertas en condiciones de oclusión parcial y desde distintos ángulos, pero presenta el problema de requerir un alto coste computacional.

Basándose en el concepto de usabilidad, en [9] se presenta un método para la detección de puertas usando un sistema trinocular de visión. El concepto *puerta* es definido como una U invertida que puede ser atravesada por personas. Para ello analizan los segmentos presentes en las imágenes buscando formas rectangulares. Si cumplen una serie de criterios de altura y anchura que establecen sobre las puertas de su entorno son clasificadas como tales. Este sistema cuenta con el

*Este trabajo ha sido parcialmente financiado por el proyecto TIC2003-04900 del MCYT

inconveniente del coste del equipo perceptual además de que no se realiza un análisis de cómo puede influir la perspectiva en la detección.

En el trabajo propuesto por [12] se utiliza un método para la detección de puertas limitado a pasillos teniendo en cuenta las dimensiones del mismo. La detección se realiza buscando las líneas verticales del pasillo y agrupándolas en puertas. La técnica limita la detección de las puertas a un entorno muy concreto y no tiene en cuenta las deformaciones debidas a los cambios de perspectiva.

En [11] se propone un método para la detección de puertas, usando una única cámara, basado en la extracción de segmentos con redes neuronales de la Transformada de Hough. Además, usando la proyección de los puntos inferiores de la puerta y conociendo las dimensiones de la misma así como la orientación de la cámara, son capaces de determinar la posición relativa del robot a la puerta. Sin embargo hay aspectos poco claros en diversos puntos del trabajo. Por un lado no se explica como se realiza el reconocimiento de la puerta a partir de las líneas, se dice que se usa una red neuronal pero no se indican cual es la información de entrada a la misma. Tampoco se indica ninguna información sobre el conjunto de datos que se ha utilizado para llevar a cabo el entrenamiento y no queda muy claro como se ha realizado el proceso de validación.

En [10] la detección visual de puertas sirve de guía a un robot móvil para cruzar puertas. La detección de los marcos se realiza en las inmediaciones de la misma como columnas de ancho mayor de 35 píxeles. Esta detección permite indicar al robot el centro de la puerta. Usando la información del sistema visual el robot es capaz de cruzar las puertas de su entorno de trabajo. Sin embargo, el mecanismo utilizado para la detección visual es poco elaborado y difícilmente puede ser extendido a otras situaciones.

En la línea del trabajo anterior, Cañas y otros [5] diseñan un sistema que integra ultrasonidos y visón activa para detectar las puertas en el entorno del robot. La detección visual se basa exclusivamente en la localización de líneas verticales. Como en el caso anterior la detección visual de la puerta es poco flexible frente a cambios de perspectiva sufridos por movimientos de la cámara en el eje vertical. Esto limita las condiciones en que el método puede ser aplicado.

En este trabajo presentamos un nuevo método para la detección visual de puertas basado en la extracción de los segmentos de la imagen y en la definición, mediante lógica difusa, de las características que presentan dichos segmentos al formar parte del marco. Es decir la puerta se detecta buscando los segmentos en la imagen que constituyen el marco de la misma. En primer lugar, detectaremos los bordes en la imagen usando el detector de fronteras Canny [6]. Sobre esta imagen aplicaremos las Transformada de Hough [8] con objetivo de extraer los segmentos. Luego, utilizando lógica difusa analizaremos las relaciones entre los segmentos para determinar la presencia de posibles marcos de puertas en la imagen. Se definen el concepto de *marco simple* y *marco doble* para poder diferenciar aquellas situaciones en las que se detecte sólo una de las fronteras del marco o las dos. El método es capaz de detectar las puertas más comúnmente encontradas en los entornos de interior en condiciones de iluminación variable. Además es robusto frente a cambios de escala y a la deformación debida a la perspectiva.

2. Detección visual de puertas

En esta sección explicaremos cómo se realiza el proceso de detección visual de las puertas a partir de los marcos. Nuestro enfoque se basa en la detección de segmentos en la imagen resultado de un extractor de fronteras. De esta manera, buscaremos los segmentos de la imagen que forman el marco de una puerta. Realmente, según hemos comprobado en nuestra experimentación, en algunos casos se puede detectar solamente una de las fronteras del marco, correspondiente a uno de sus bordes (el interior o exterior según las imágenes). En otras ocasiones se detectan ambas fronteras del marco, la interior y la exterior. En el segundo caso, se puede considerar que la evidencia visual de la presencia de la puerta es más fuerte que en el primero. Para tratar esta diferencia se va a definir el concepto *marco simple* para referirnos al primer caso y el concepto *marco doble* para el segundo caso. En concreto, un *marco simple* se define como una pareja de segmentos verticales (correspondientes a las zonas laterales de la puerta) unidas en su zona superior por un segmento horizontal (correspondiente a la zona superior). Por otra parte, un *marco doble* está formado por dos marcos simples que guardan cierta relación de paralelismo y cercanía por lo que podemos suponer que son las dos fronteras de un mismo marco de una puerta. En las imágenes de la figura 2 en los dos primeros casos se detecta un marco simple, mientras que en el último se detecta un marco doble.

El proceso de detección comienza aplicando el detector de fronteras de Canny sobre la imagen en niveles de grises. La imagen resultado $I(x, y)$ es una imagen binaria donde los píxeles a valor *verdadero* corresponden a bordes en la imagen original. Supondremos para facilitar la notación, que la imagen $I(x, y)$ es cuadrada de tamaño NxN. Con los píxeles de contorno de $I(x, y)$ aplicamos la Transformada de Hough. A partir de la Transformada de Hough se obtienen los segmentos de la imagen que denotaremos $S = \{S^0, S^1, ..., S^n\}$. Cada segmento está compuesto de dos puntos en el plano de la imagen, $S^i = \{p_0^i, p_1^i\}$ donde $p_j^i = (x_j^i, y_j^i)$. A su vez, cada segmento tiene una orientación que se extrae de la Transformada de Hough $S_\phi^i \in [-\frac{\pi}{2}, \frac{\pi}{2}]$.

Una vez obtenidos los segmentos se realiza el proceso de análisis que se explica a continuación para ver si cumplen las propiedades de pertenecer a posibles *marcos simples* de puertas.

2.1. Selección y clasificación de segmentos candidatos

El conjunto de segmentos total obtenido en una imagen es alto y deseamos reducirlo sólo a aquellos que puedan pertenecer al marco de una puerta. Recordamos que estamos buscando los segmentos verticales del lateral del marco y el segmento más o menos horizontal de la parte superior del mismo. Dado que deseamos poder realizar la detección desde diferentes perspectivas, la proyección de los segmentos laterales y superior del marco no serán necesariamente paralelos al plano

de la imagen, pudiéndose por tanto proyectar como segmentos inclinados que no mantienen la perpendicularidad que tienen en la realidad. Esta circunstancia nos obliga a entender el concepto de *"segmento lateral y horizontal perteneciente al marco de una puerta"* de una forma flexible. Por otra parte, debido a que suelen aparecer una gran cantidad de segmentos al analizar una imagen se hace necesario realizar una selección de los mismos para considerar que efectivamente forman parte de un posible marco. Para afrontar ambas tareas se va a tener en cuenta que el robot en el proceso de detección visual de las puertas, va a establecer unos niveles de creencia sobre la detección de dichos segmentos laterales y horizontales. Para expresar estos niveles de creencia acudiremos a las técnicas proporcionadas por la lógica difusa como se muestra a continuación.

Si analizamos los entornos de interior podremos observar como las puertas son objetos relativamente altos. Por tanto, los segmentos laterales deben tener un tamaño grande en la imagen y proyectarse con un aspecto más o menos vertical. Haciendo uso de las características *inclinación* y *tamaño* de un segmento, el robot establecerá un nivel de creencia en el intervalo $[0, 1]$ sobre el hecho de que un segmento S^i pueda pertenecer al lateral de un marco. La primera característica se medirá mediante una variable lingüística que llamaremos $Inclinacion(S^i)$ que presenta dos posibles valores, *vertical* y *horizontal* representados por los conjuntos difusos de la figura 1(a). El valor de entrada para esta variable será la inclinación del segmento calculada como el $\sin(S_\phi)$. De esta manera la inclinación es *vertical* cuando el seno toma valores cercanos a 0 y *horizontal* en valores próximos a 1.

La segunda característica la mediremos mediante la variable lingüística $TamanoSL(S^i)$ que tiene también dos posibles valores *pequeño* y *apropiado* representados por los conjuntos difusos de la figura 1(b). El valor de entrada para esta variable será el tamaño de un segmento, $\text{tamano}(S^i)$, que acotaremos en el intervalo $[0, 1]$ según la ecuación 1. Donde $dist(p_0^i, p_1^i)$ es la distancia euclídea entre los dos extremos del segmento y $N\sqrt{2}$ la distancia máxima en la imagen.

$$tamano(S^i) = \frac{dist(p_0^i, p_1^i)}{N\sqrt{2}} \tag{1}$$

A partir de estas variables lingüísticas definimos el nivel de creencia, $SegmentoL(S^i)$, como la posibilidad de que S^i sea un segmento lateral perteneciente al marco de una puerta. Esta posibilidad viene dada por la siguiente regla:

$$\begin{array}{l} Si\ Inclinacion(S^i)\ es\ Vertical \wedge \\ TamanoSL(S^i)\ es\ Apropiado \\ entonces\ S^i\ es\ Segmento\ lateral. \end{array} \tag{2}$$

Entonces, $SegmentoL(S^i)$ se define como el grado de cumplimiento de la regla anterior. A la hora de aplicar esta regla usaremos el operador mínimo para llevar a cabo la intersección.

En lo concerniente al segmento superior de la puerta, su proyección en la imagen se verá afectada en mayor medida por la perspectiva. Esto se debe a que el grado de libertad que se permite en el plano horizontal de visión (cuando rodeamos la puerta) es mayor que en el vertical (cuando la vemos desde abajo hacia arriba). Por ello se podrá proyectar en un amplio rango de tamaños y orientaciones en torno a la horizontal. Pero se puede decir que debería estar en posiciones superiores de la imagen. El robot asignará un grado de creencia en el intervalo $[0, 1]$ sobre hecho de que un segmento S^i pueda ser un segmento perteneciente al marco superior de la puerta, basándose en las características de *inclinación*, *tamaño* y *posición* del segmento. La primera característica se medirá usando la variable lingüística anteriormente explicada $Inclinacion(S^i)$. La característica tamaño será evaluada mediante la variable lingüística $TamanoSH(S^i)$ que tiene dos posibles valores *pequeño* y *apropiado* representados por los conjuntos difusos de la figura 1(c). El valor de entrada de esta variable se calculará usando la ecuación (1).

Finalmente la posición del segmento en la imagen será medida por la variable lingüística $PosicionY(S^i)$ que tiene dos posibles valores *alta* y *baja*, definidos en la figura 1(d). El valor de entrada, $posY(S^i)$, será la posición del punto medio del segmento en el eje vertical de la imagen acotado en el intervalo $[0, 1]$ que se define como:

$$posY(S^i) = \frac{y_0^i + y_1^i}{2N}. \tag{3}$$

Valores bajos indican que el segmento tiene su punto medio en la zona superior de la imagen y viceversa.

Con estas tres variables definimos el nivel de creencia $SegmentoH(S^i)$, como la posibilidad de que S^i sea un segmento horizontal correspondiente a la zona superior del marco de una puerta mediante la siguiente regla:

$$\begin{array}{l} Si\ Inclinacion(S^i)\ es\ Horizontal \wedge \\ TamanoSH(S^i)\ es\ Apropiado \wedge \\ PosicionY(S^i)\ es\ Alta \\ entonces\ S^i\ es\ Segmento\ horizontal. \end{array} \tag{4}$$

De forma similar al caso anterior, $SegmentoH(S^i)$ se define como el grado de cumplimiento de esta regla.

Estos valores de creencia nos van a servir para definir el concepto de *marco simple* como veremos en la siguiente sección pero además también los utilizaremos para acelerar el proceso de búsqueda de los posibles marcos. Para ello, se realiza una selección de segmentos candidatos en base a los grados de creencia $SegmentoL(S^i)$ y $SegmentoH(S^i)$. Si alguno de ellos supera un determinado umbral α_1, el segmento S^i será candidato a formar parte de un marco simple y pasará a la siguiente fase. En ese caso S^i será considerado lateral si $SegmentoL(S^i) > SegmentoH(S^i)$ y horizontal en caso contrario. En lo que sigue denotaremos al conjunto de segmentos laterales candidatos como $L = \{L^0, ..., L^n\}$ y al de horizontales $H = \{H^0, ..., H^n\}$.

Con la flexibilización de estos conceptos de esta manera, si bien es necesario tener la puerta a cierta distancia para poder detectarla, la detección se hace relativamente independiente a cambios de escala y perspectiva. Además podemos asumir, como dijimos anteriormente, que en entornos de interior las puertas son

elementos relativamente grandes y que por lo general cumplirán estas restricciones.

2.2. Definición del concepto de marco simple

El siguiente paso es analizar si en el conjunto de segmentos extraído hay algún marco simple. Un marco simple se define como un par de segmentos laterales atravesado por uno horizontal en su zona superior. Para ello tomaremos por cada segmento horizontal H^i un par de segmentos laterales L^i, L^j y asignaremos un grado de creencia perteneciente al intervalo [0,1] de ser un trío candidato a ser un posible marco simple, basándonos en la distancia mínima de los puntos extremos de los segmentos.

Definimos la variable lingüística $DistanciaSegmentos(S^i, S^j)$ para medir la distancia entre dos segmentos en la imagen. Esta variable puede tener tres posibles valores representados por los conjuntos difusos de la figura 1(e). Como valor de entrada toma:

$$distS(S^i, S^j) = min\{dist(p_0^i, p_0^j), dist(p_0^i, p_1^j), dist(p_1^i, p_0^j), dist(p_1^i, p_1^j)\}/N\sqrt{2} \quad (5)$$

Esta ecuación expresa la distancia entre dos segmentos como la mínima distancia entre sus puntos extremos dividida por la máxima distancia. Los valores están acotados al intervalo $[0, 1]$, dónde 0 indica que ambos elementos no tienen separación y 1 indica separación máxima.

Para evaluar el grado de creencia de que el trío $\{H^i, L^j, L^k\}$ forme un marco simple, llamémosle $MarcoSimple(H^i, L^j, L^k)$, evaluaremos el nivel de cumplimiento de la siguiente regla:

$$\begin{aligned} &Si\ H^i\ es\ SegmentoH \wedge L^j\ es\ SegmentoL \wedge \\ &L^k\ es\ SegmentoL \wedge \\ &DistanciaSegmentos(H^i, L^j)\ es\ baja \wedge \\ &DistanciaSegmentos(H^i, L^k)\ es\ baja \wedge \\ &DistanciaSegmentos(L^j, L^k)\ es\ alta\ entonces \\ &\{H^i, L^j, L^k\}\ es\ Marco\ simple. \end{aligned} \quad (6)$$

En este momento, el robot ya puede calcular un grado de creencia en la existencia de un marco de una puerta $MarcoSimple(H^i, L^j, L^k)$ utilizando el proceso descrito. Este valor lo utilizará como una información perceptual más que se tendrá que fusionar dentro del modelo perceptual existente y que le ayudará en la toma de las decisiones oportunas en la navegación por el entorno.

Ahora es necesario definir el segundo concepto, ya que dependiendo de la imagen que tengamos de la puerta, puede ser que se hayan detectado los dos bordes del marco de la puerta. Hasta ahora estos dos bordes serían considerados como dos marcos simples cuando realmente pertenecen a la misma puerta. Por tanto necesitamos agrupar aquellos marcos que pertenecen a la misma puerta y obtener un único valor de creencia final para estos casos, que se verá reflejado en el concepto de *marco doble*. Para realizar el análisis siguiente nuevamente llevamos a cabo un selección de candidatos con el objetivo de acelerar el proceso. Si el grado de creencia $MarcoSimple(H^i, L^j, L^k)$ supera un determinado umbral α_2 ese trio es considerado candidato a formar parte de un marco doble y se tendrá en cuenta en la siguiente fase. Notaremos dicho trío de segmentos como $MS^i = \{L_l^i, Sup^i, L_r^i\}$. Donde Sup^i=H^i, L_l^i es lateral y está en la imagen situado a la izquierda de L_r^i que es el otro segmento lateral.

2.3. Concepto de marco doble

Dado un conjunto de marcos simples candidatos $MS = \{MS^0, ..., MS^n\}$, estos se deben analizar para ver si forman parte de la misma puerta formando por tanto lo que hemos llamado un marco doble. Para ello asignaremos un grado de creencia perteneciente al intervalo [0,1] en el hecho de que dos posibles marcos simples pertenezcan a la misma puerta, basándonos en las características de distancia y paralelismo entre ellos. Definimos la variable lingüística $DistanciaMarcos(MS^i, MS^j)$ para medir la distancia entre dos marcos simples MS^i, MS^j. Dicha variable puede tomar tres valores representados por los conjuntos difusos de la figura 1(f). El valor de entrada se calcula mediante:

$$distMS(MS^i, MS^j) = \frac{distS(L_l^i, L_l^j)}{3} + \frac{distS(Sup^i, Sup^j)}{3} + \frac{distS(L_r^i, L_r^j)}{3} \quad (7)$$

Este valor, está acotado en el intervalo $[0, 1]$ y mide la distancia entre dos marcos simples como la media de las distancias entre los segmentos homónimos en ambos marcos. La distancia entre dos segmentos $distS(S^i, Sj)$ se mide con la ecuación (5).

Por otro lado, la variable lingüística $Paralelismo(MS^i, MS^j)$ mide el grado de paralelismo entre dos posibles marcos simples. Los conjuntos difusos que representan los valores definidos para esta variable se reflejan en la figura 1(g). La variable toma como valor de entrada:

$$paralelismo(MS^i, MS^j) = \frac{cos(L_{l\phi}^i - L_{l\phi}^j)}{3} + \frac{cos(Sup_\phi^i - Sup_\phi^j)}{3} + \frac{cos(L_{r\phi}^i - L_{r\phi}^j)}{3} \quad (8)$$

En esta expresión, $cos(S_\phi^a - S_\phi^b)$ mide el grado de perpendicularidad de S^a, S^b en el intervalo $[0, 1]$. Si el coseno es 0 se debe a que los segmentos forman un ángulo $\frac{\pi}{2}$ y toma valor 1 en caso de que no exista diferencia entre sus ángulos. De esta manera (8) expresa la media del paralelismo entre los segmentos homónimos en ambos marcos simples.

Finalmente, asignaremos un grado de creencia al hecho de que dos posibles marcos simples MS^i, MS^j formen parte realmente de la misma puerta constituyendo un marco doble, al que denominaremos $MarcoDoble(MS^i, MS^j)$, evaluando el cumplimiento de la siguiente regla:

$$\begin{aligned} &Si\ MS^i\ es\ MarcoSimple \wedge MS^j\ es\ MarcoSimple \\ &\wedge\ DistanciaMarcos(MS^i, MS^j)\ es\ no(alta) \wedge \\ &Paralelismo(MS^i, MS^j)\ es\ alto \\ &entonces\ \{MS^i, MS^j\}\ es\ Marco\ doble. \end{aligned} \quad (9)$$

De esta forma se podrá calcular el grado de creencia $MarcoDoble(M^i, M^j)$ que luego en el proceso de fusión con el modelo perceptual anterior se podrá tener en cuenta con más peso. La razón es que podemos suponer que el detectar los dos bordes del marco nos ofrece una evidencia visual mayor en la existencia de la puerta.

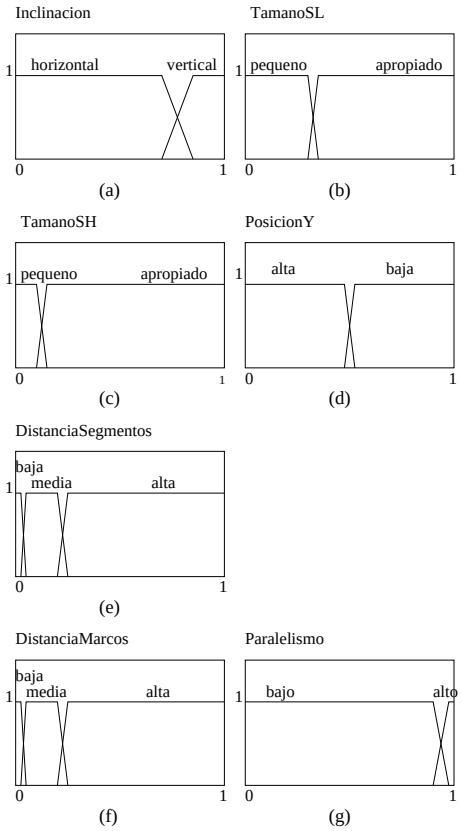


Figura 1: Variables lingüísticas empleadas. a) Inclinación b) TamanoSL c) TamanoSH d) PosicionY e) DistanciaSegmentos f)DistanciaMarcos g)Paralelismo

Para finalizar la descripción del proceso, queremos comentar que las funciones de pertenencia de los conjuntos difusos implicados y los valores concretos para los parámetros α_1 y α_2 han sido determinados a partir de la experimentación. Puede ser interesante la utilización de algún procedimiento automatizado para su determinación, lo que nos plantearemos en próximos trabajos.

3. Experimentación

En esta sección mostramos algunos resultados de la experimentación realizada con nuestro método. Para validar el método hemos realizado dos tipos de pruebas. Mediante la primera se pretende determinar el éxito en la detección de puertas en imágenes que efectivamente muestran puertas. Con la segunda se pretende comprobar que el método no detecta falsas puertas en imágenes donde no hay ninguna (falsos positivos). En el primer caso se han usado un total de 105 imágenes de distintos tipos de puertas tomadas desde diferentes perspectivas y a una distancia máxima de 4 metros, obteniéndose un éxito del 87 %. En la segunda prueba se han usado 50 imágenes y el sistema no detecta ningún falso positivo demostrando un comportamiento robusto. Con el objeto de mejorar el porcentaje de éxito en la primera prueba se pretende introducir alguna técnica automática de ajuste de los parámetros del sistema en la que actualmente estamos trabajando.

La figura 2 muestra en la primera fila imágenes de diferentes edificios en las que aparece una puerta. En la segunda línea se muestra el conjunto total de segmentos que obtiene nuestro detector basado en la Transformada de Hough. La tercera línea muestra los segmentos preseleccionados por el método explicado en 2.1. Finalmente en la última línea de imágenes se muestran las puertas que se han detectado a partir de la búsqueda de los marcos de las mismas. Como se puede observar en las dos primeras puertas comenzando por la izquierda, se han detectado solamente el concepto marco simple. En el tercer caso, sin embargo se han detectado dos marcos simples que corresponden realmente a la misma puerta formando un marco doble. En los tres ejemplos el nivel de creencia alcanzado en los conceptos marco simple y marco doble ha sido de 0.9 sobre un valor máximo de 1.

Las imágenes muestran como el método desarrollado es capaz de detectar puertas de diferentes tipos, a partir de imágenes tomadas desde diferentes puntos de vista y con condiciones de iluminación variables. La primera imagen muestra una puerta vista desde dentro de la habitación en una posición muy cercana y bajo ella. En esta imagen se observa claramente la inclinación de los segmentos debida a la perspectiva. En el resto, la visión de la puerta se realiza desde posiciones más alejadas en las que los marcos de las mismas adquieren mayor tamaño, siendo igualmente detectados de forma correcta.

4. Conclusiones y trabajo futuro

Hemos desarrollado un método para la detección visual de puertas basado en el análisis de los segmentos detectados en la imagen y que forman parte del marco de la misma. Para diferenciar la situación en la que se detecta una frontera del marco o las dos, se definen los conceptos de marco simple y doble utilizando expresiones mediante lógica difusa. En este sentido, el método hace uso de lógica difusa para definir las características apropiadas según las cuales un segmento puede ser considerado como un segmento lateral u horizontal de un marco de una puerta, estableciendo un cierto grado de creencia en ambos casos. Luego, tomando en cuenta los segmentos que cumplen esa condición en cierto grado, se analizan sus relaciones espaciales para determinar si pueden formar un marco simple. Posteriormente se analizan las relaciones entre los marcos simples para detectar posibles marcos dobles. El empleo de la lógica difusa en todo este proceso nos permite, por un lado, limitar el espacio de búsqueda sobre el conjunto de posibles segmentos y definir de manera natural los conceptos de marco simple y doble. Por otro lado, también nos da la posibilidad de analizar un conjunto de segmentos con diferente grado de inclinación y tamaño lo que nos proporciona una gran flexibilidad a

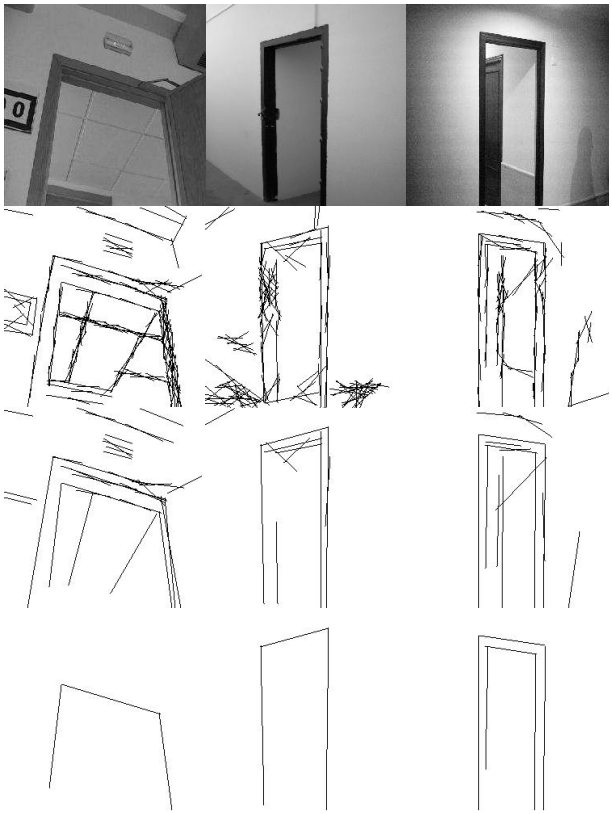


Figura 2: De arriba a abajo. Imagen original. Todos los segmentos detectados. Segmentos candidatos seleccionados. Puertas detectadas con alto nivel de creencia

la hora de manipular la información contenida en las imágenes. Esta flexibilidad es necesaria para contemplar las posibles diferentes proyecciones del marco de una puerta en la imagen según la situación de la cámara del robot respecto de la puerta, haciéndolo robusto frente a cambios de perspectiva y escala. Por último, gracias a la lógica difusa podemos expresar el grado de creencia del robot en que los segmentos detectados visualmente pertenezcan al marco de la puerta (simple o doble), de manera que este nivel de creencia puede ser fusionado con otros valores aportados por otros sensores como los ultrasonidos. Con ello se pretende que unos sensores complementen la información proporcionada por otros.

La experimentación realizada muestra que el método propuesto es capaz de detectar de forma efectiva las puertas que se encuentran en las imágenes como se ha mostrado en el apartado 3. Pensamos también, que el método es válido para aquellas puertas de tipo similar al que encontramos en la mayoría de los entornos de interiores.

En lo que respecta a trabajos futuros, además de llevar a cabo la integración con el modelo perceptual anterior, hay que tener en cuenta que tal y como ha sido definido, el método explicado requiere que los tres segmentos del marco estén presentes en la imagen. Sin embargo, en las pruebas realizadas hemos observado que en ocasiones alguno de los segmentos puede estar oculto. Por ello como trabajo futuro habrá que modificar este método para poder contemplar la posibilidad de oclusión de parte del marco de la puerta en las observaciones. Para ello se tendrá en cuenta tanto el papel activo que puede adoptar el robot, como elemento que se puede desplazar por el entorno, como la información complementaria que pueda aportar el modelo perceptual ya desarrollado.

Referencias

[1] E. Aguirre, M. Gómez, R. Muñoz, y C. Ruiz. Un sistema multi-agente que emplea visión activa y ultrasonidos aplicado a navegación con comportamientos difusos. En *Actas del IV Workshop en Agentes Físicos (WAF'2003)*, páginas 63–74, Alicante, 2003.

[2] E. Aguirre y A. González. Fuzzy behaviors for mobile robot navigation: Design, coordination and fusion. *International Journal of Approximate Reasoning*, 25:255–289, 2000.

[3] E. Aguirre y A. González. A fuzzy perceptual model for ultrasound sensors applied to intelligent navigation of mobile robots. *Applied Intelligence*, 19(3):171–187, 2003.

[4] R.C. Arkin. *Behavior-Based Robotics*. The MIT Press, 1998.

[5] J.M. Cañas, R. Simmons, y M. C. García-Alegre. Detección probabilística de puertas con visión monocular activa. En *Actas del II WorkShop Hispano-Luso en Agentes Físicos (Waf'2001)*, páginas 113–128, Madrid, 2001.

[6] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analsysis and Machine Intelligence*, PAMI-8:679–698,1986.

[7] G. Cicirelli, T. D'orazio, y A. Distante. Target recognition by component for mobile robot navigation. *Journal of Experimental and Theoretical Artificial Intelligence*, 15(3):281–297, 2003.

[8] P.V.C. Hough. Method and means for recognizing complex patterns. *U.S. Patent 3069654*, 1962.

[9] D. Kim y R.Ñevatia. A method for recognition and localizatión of generic objects for indoor navigation. *Image and Vision Computing*, 16(11):729–743, 1998.

[10] I. Monasterio, E. Lazkano, I. Rañó, y B. Sierra. Learning to traverse door using visual information. *Mathematics and Computer in Simulation*, 60:347–356, 2002.

[11] O.L. Cetin y A. Erden. Improvement of an image processing system for autonomous mobile robot navigation. En *Proc. 5th Int. Conf. on Mechatronics Design and Modelling*, 2001.

[12] S. A. Stoeter, F. L. Mauff, y N. P. Papanikolopoulos. Real-time door detection in cluttered enviroments. En *Proceeding of the 15th IEEE International Symposium on Intelligent Control (ISIC 200)*, páginas 187–191, 2000.

[13] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

ALGORITMO EVOLUTIVO PARA EL CONTROL DIFUSO DEL ATRAQUE DE UN AGV

José Miguel Lucas
Dept. Ingeniería de la Información y las Comunicaciones
Universidad de Murcia
jmlucas@um.es

Humberto Martínez
Dept. Ingeniería de la Información y las Comunicaciones
Universidad de Murcia
humberto@um.es

Juan A. Botía
Dept. Ingeniería de la Información y las Comunicaciones
Universidad de Murcia
juanbot@um.es

Fernando Jiménez
Dept. Ingeniería de la Información y las Comunicaciones
Universidad de Murcia
fernan@dif.um.es

Resumen

Algunos robots móviles necesitan realizar maniobras de atraque con precisión. En el dominio de aplicación en el que se encuadra este trabajo, una carretilla autónoma debe realizar con frecuencia maniobras de atraque para depositar pallets en cintas transportadoras. En dichas maniobras, el movimiento de robot debe ser controlado con extrema precisión a medida que se aproxima al destino. En este artículo proponemos un algoritmo evolutivo para obtener el controlador correspondiente a la maniobra de atraque, encargado de seguir una trayectoria generada de forma on-line por el AGV. El objetivo consiste en realizar el atraque con la máxima precisión y en el menor tiempo posible.

Palabras Clave: Seguimiento de Trayectorias, Robots Autónomos, Algoritmos Evolutivos, Control Difuso.

1 INTRODUCCIÓN

En algunos entornos los robots móviles necesitan realizar maniobras de atraque con precisión. Una maniobra de atraque puede verse como el movimiento desde una posición actual a otra posición y orientación deseada, mientras se sigue una trayectoria segura [2]. La posición y orientación final deben ser las adecuadas para las tolerancias requeridas para una tarea en particular. Un robot móvil necesita realizar una maniobra de atraque cuando tiene que manipular o inspeccionar un objeto, por ejemplo, en nuestro entorno real, una carretilla autónoma realiza maniobras de atraque para depositar pallets en cintas transportadoras [8].

La tarea de atraque puede dividirse en tres subproblemas distintos [1]: (a) movimiento rápido hacia la ubicación del objeto mientras no haya peligro de colisión; (b) movimiento preciso para aproximarse a la configuración requerida para el atraque; (c) movimiento hacia la configuración final con la precisión requerida o realizar alguna operación de atraque final.

El objetivo es realizar el atraque con la máxima precisión y en el menor tiempo posible siguiendo una trayectoria generada de forma on-line por el vehículo autonómo guiado (AGV). Para ello se ha desarrollado un algoritmo evolutivo para llevar a cabo un proceso de aprendizaje mediante la evolución de la base de reglas de un controlador difuso. Los objetivos para abordar la tarea de atraque se han codificado en la función de fitness.

El proceso de aprendizaje se ha realizado usando el sistema de simulación de robots móviles, ThinkingCap-II [11]. Los nuevos controladores evolucionados se simulan en dicho entorno permitiendo la adquisición de los datos necesarios para valorar la bondad de dicho controlador. En el entorno real se han llevado a cabo una serie de experimentos con la carretilla industrial para evaluar el rendimiento de los controladores aprendidos.

El artículo está organizado de la siguiente forma: en la sección 2 se introducen los algoritmos evolutivos y su aplicación al control difuso y se describe el algoritmo evolutivo propuesto. La sección 3 muestra los resultados obtenidos en este trabajo, y finalmente, la sección 4 resume las conclusiones más importantes que hemos obtenido y futuros trabajos.

2 DESCRIPCIÓN DEL ALGORITMO EVOLUTIVO

2.1 ALGORITMOS EVOLUTIVOS Y CONTROL DIFUSO

Los Algoritmos Evolutivos (AE) son mecanismos de optimización robustos inspirados en la Teoría de la

Evolución Natural de Darwin [3, 5]. La idea fundamental de los algoritmos evolutivos consiste en simular el proceso de selección y evolución de las especies. Estos algoritmos han sido aplicados a numerosos problemas como los de diseño y optimización automática de controladores difusos [6, 7, 10]. En [4] se aplica un algoritmo genético para diseñar y optimizar de forma automática un controlador difuso para que un tren siga una trayectoria con un control suave y ciertas restricciones de velocidad.

Los AE permiten la posibilidad de tratar con entornos en los que se tiene un conocimiento impreciso e incompleto como en el caso de la robótica móvil. La idea es aplicar un algoritmo evolutivo para evolucionar la base de reglas difusas de un controlador con el objetivo de conseguir que el robot realice una tarea de forma más precisa y eficiente. La bondad o valor de fitness de cada controlador nuevo evolucionado se mide evaluando dicho controlador en un entorno simulado. En nuestro caso, evolucionamos el controlador difuso encargado de llevar a cabo la maniobra de atraque para una carretilla industrial autónoma.

2.2 CARACTERÍSTICAS DEL ALGORITMO EVOLUTIVO

Las principales características del AE son las siguientes:

- **Representación de las soluciones:** La representación usada es una tupla de números reales de longitud fija en donde cada individuo representa un controlador difuso de N reglas. Cada regla está representada por un conjunto difuso trapezoidal que representa el error angular en la trayectoria (variable de entrada) y por dos variables de salida para las velocidades angular y lineal representadas en punto flotante. Las reglas del controlador difuso son del tipo:

  $$Si \quad \delta \in A \quad entonces \quad \omega = c0, \quad \nu = c1$$

  La variable de entrada δ que corresponde con el error angular en la trayectoria consta de siete conjuntos difusos trapezoidales (NL, NM, NS, Z, PS, PM, PL). Cada función de pertenencia trapezoidal se representa con cuatro parmetros a, b, c y d. Estos valores reales para cada uno de los conjuntos son los que componen la primera parte de la tupla que representa la solución. Debido a la simetría del controlador sólo se representa la mitad de los conjuntos difusos. Las variables de salida como la velocidad angular ω tiene siete valores en punto flotante (NL, NM, NS, Z, PS, PM, PL) y la velocidad lineal ν tiene cuatro (NF, NM, NS, NZ). Estos valores se concatenan con los valores de la variable de entrada formando la representación de la solución completa. Con esta tupla de valores reales se construyen las reglas del controlador.

- **Población inicial:** La población inicial se obtiene de forma híbrida mediante la inclusión de controladores generados por un experto y a través de variaciones aleatorias de éstos en su dominio.

- **Mecanismo de selección, muestreo y sustitución generacional:** El método empleado para seleccionar los mejores individuos es el de selección proporcional y muestreo estocástico con reemplazamiento (rueda de ruleta) [5]. En nuestra implementación, los individuos con menor valor de fitness tendrán más probabilidad de ser seleccionados al ser un problema de minimización. La sustitución generacional corresponde a un *steady-state* con la mitad de la población en la que los nuevos individuos reemplazan a sus padres. La otra mitad de la nueva población corresponde a los individuos de la antigua población que no han sido seleccionados. El elitismo se lleva a cabo manteniendo siempre el mejor individuo de la población en la siguiente.

- **Operadores de variación:** Los operadores de cruce que han sido empleados son: uniforme y aritmético [9]. Tenemos dos operadores de mutación: el primer operador realiza una alteración en un valor real dentro de un rango determinado; el segundo añade al valor original un ruido gaussiano con media cero para realizar una perturbación controlada por el valor de la desviación estándar [3].

Para el cálculo de la función de fitness se emplea el entorno de simulación de robots móviles ThinkingCap-II (TC)[11] para adquirir los datos necesarios tras el proceso de simulación del individuo.

Finalmente, el algoritmo evolutivo termina cuando un número máximo determinado de generaciones se realizan o en otro caso podría acabar cuando se alcanza un nivel de fitness específico.

2.3 DESARROLLO DE LA FUNCIÓN DE FITNESS

Un aspecto importante en el diseño de un algoritmo evolutivo es la definición de una función de fitness adecuada ya que determina la dirección de la búsqueda. En la mayoría de los casos, el objetivo es la maximización de la función de fitness pero en nuestro caso es mejor considerar la minimización de dicha función ya que el principal objetivo es minimizar una serie

de factores para realizar la maniobra de atraque de forma precisa en el menor tiempo posible. Uno de los objetivos principales es minimizar el error angular (heading) respecto a la trayectoria generada. El error heading se denota por δ. Asumimos una trayectoria generada de forma on-line para realizar la maniobra de atraque. El cálculo del valor δ es el siguiente:

- Path.look-ahead.x = coordenada x del punto de look-ahead en la trayectoria.
- Path.look-ahead.y = coordenada y del punto de look-ahead en la trayectoria.
- Dx = Path.look-ahead.x - Posicion Actual Robot.x
- Dy = Path.look-ahead.y - Posicion Actual Robot.y
- Heading = arco tangente de Dy / Dx en el rango -π a π

$$\delta = Heading - OrientacionActualdelRobot$$

donde δ está normalizado.

Para definir la función de fitness o de adecuación hay que tener en cuenta cuatro factores: el primero es el tiempo de ejecución (número de acciones, expresión 1); el segundo es el error angular medio en toda la trayectoria (expresión 2); el tercero es el error angular medio en las últimas acciones (p.ej. cuando el robot está alcanzando el punto de destino al final del recorrido de la trayectoria, expresión 3) y el último factor es el error final en la distancia entre la posición final del robot y el punto de atraque o de destino (expresión 4). De hecho, es importante que el robot móvil siga la trayectoria de forma precisa en los últimos pasos mientras se aproxima al punto de atraque, por lo que se consideran los errores angulares en las acciones finales con el propósito de reducirlos a cero.

Estos objetivos a minimizar se combinan en la función de fitness agregada:

$$TiempoEjecucion = NumeroAccionesRealizadas \quad (1)$$

$$ErrorAngularMedio = \frac{\sum_{i=1}^{T} |Delta_i|}{TiempoEjecucion} \quad (2)$$

donde $T = TiempoEjecucion$.

$$UltimoErrorAngularMedio = \frac{\sum_{i=1}^{L} |Delta_i|}{L} \quad (3)$$

donde L = Número de los últimos errores angulares (porcentaje del total de errores angulares a lo largo de la trayectoria, p.ej. 10%).

$$UltimoErrorDistancia = \sqrt{ErrorX^2 + ErrorY^2} \quad (4)$$

donde este error es la distancia entre la posición final del robot y la posición del punto destino o atraque.

En cada evaluación en un punto de atraque se calcula la siguiente función de fitness agregada en la que se combinan los objetivos a minimizar:

$$\begin{aligned} Fatraque &= w_{l,1} \cdot TiempoEjecucion & (5)\\ &+ w_{l,2} \cdot ErrorAngularMedio & (6)\\ &+ w_{l,3} \cdot UltimoErrorAngularMedio & (7)\\ &+ w_{l,4} \cdot UltimoErrorDistancia & (8) \end{aligned}$$

donde $w_{l,1}$,$w_{l,2}$, $w_{l,3}$ y $w_{l,4}$ son los pesos que ponderan la importancia relativa de los objetivos de la función y cuya suma debe ser uno.

Los sumandos de la función de fitness se normalizan usando un escalado lineal donde los valores máximos establecen los límites que se desean aceptar. En cuanto a los valores mínimos para la mayoría de los sumandos es cero ya que es un problema de minimización, excepto para el tiempo de ejecución al cual se le otorga cierto margen.

En principio, el conjunto de reglas difusas evolucionadas de un controlador podría evaluarse en tres puntos de atraque distintos con diferentes niveles de dificultad: fácil, medio, y díficil. De hecho, también se puede considerar los tres puntos de atraque con el mismo nivel de dificultad. En cualquier caso, la función de fitness para tres puntos de atraque es la siguiente:

$$\begin{aligned} F_{TresPuntosAtraque} &= w_{g,1} \cdot Fatraque_{facil} & (9)\\ &+ w_{g,2} \cdot Fatraque_{medio} & (10)\\ &+ w_{g,3} \cdot Fatraque_{dificil} & (11) \end{aligned}$$

donde $w_{g,1}$,$w_{g,2}$ y $w_{g,3}$ son los pesos cuya suma debe ser uno.

Con los coeficientes de ponderación tenemos la posibilidad de ajustar la función de fitness para dar importancia a los factores o aspectos que se deseen y así poder guiar el proceso de búsqueda. Un problema que surge es la dificultad para obtener un conjunto de pesos que escale adecuadamente los objetivos con un conocimiento escaso e impreciso del problema.

3 RESULTADOS EXPERIMENTALES

3.1 PROCESO DE APRENDIZAJE

Con el objetivo de añadir nuevas características de aprendizaje al sistema de simulación de robots móviles ThinkingCap-II (TC-II), ha sido desarrollada una arquitectura para un sistema de aprendizaje. Estas nuevas propiedades nos permiten aplicar algoritmos de aprendizaje como los evolutivos para llevar a cabo una serie de simulaciones con el robot móvil y así poder evaluar el rendimiento de un algoritmo de aprendizaje. Durante el proceso de aprendizaje se evalúan los controladores evolucionados mediante el cálculo de la función de fitness adquiriendo para ello los datos necesarios de las simulaciones realizadas en el TC-II.

Tras un proceso previo de experimentación con distintos valores de probabilidad de cruce y mutación, los que han dado mejores resultados han sido para la probabilidad de cruce 0.8, pero no tiene que ser estricta puede variar en el rango [0,7..0,8] y para la mutación el valor elegido ha sido 0.05 del rango orientativo [0,001..0,1]. Al tener varios tipos de operadores de cruce se seleccionan con igual probabilidad, aunque puede pensarse en usar con más frecuencia operadores de tipo aritmético para tener una mayor variedad de valores diferentes a la de los progenitores. Lo mismo ocurre con los tipos de operadores de mutación que son elegidos con igual probabilidad.

Respecto a la función de fitness, los coeficientes de ponderación son parámetros de entrada para el algoritmo. Estos coeficientes explicados en el apartado anterior son importantes a la hora de ponderar la importancia de los subobjetivos de la función de fitness. En dicha función de fitness para un punto de atraque (expresiones de la 5 a la 8), tenemos los coeficientes de ponderación $w_{l,1}$,$w_{l,2}$, $w_{l,3}$ y $w_{l,4}$. El primer coeficiente $w_{l,1}$ corresponde al tiempo de ejecución cuyo valor varía en el rango [0,05..0,1]. El segundo $w_{l,2}$ pondera el error angular medio de toda la trayectoria que puede variar en el rango [0,2..0,4]. El tercero $w_{l,3}$ corresponde al último error angular medio que puede variar en el rango [0,4..0,7], a éste se le otorga un poco más de importancia que a los demás debido a que necesitamos que realice la maniobra de forma más precisa cuando se aproxime al punto de atraque para conseguir finalmente una posición final satisfactoria. El coeficiente $w_{l,4}$ pondera el último error en la distancia entre la posición final y de atraque del robot, cuyo rango puede ser [0,05..0,1]. La elección de los coeficientes de ponderación se ha llevado a cabo de forma inicial intentando dar una mayor importancia a unos objetivos sobre otros. Tras varias simulaciones y pruebas se han determinado unos rangos orientativos. El cálculo de la función de fitness para tres puntos de atraque distintos (expresiones de la 9 a la 11) se lleva a cabo calculando en cada punto la función de fitness para un solo punto de atraque (expresiones de la 5 a la 8). Los coeficientes de ponderación para dicha función de fitness en tres puntos, $w_{g,1}$, $w_{g,2}$ y $w_{g,3}$, se consideran en principio con el mismo valor de 1/3 para dar más flexibilidad al proceso de aprendizaje y debido también a la complejidad de estimar el nivel de dificultad de cada punto de atraque.

Los procesos de aprendizaje se han realizado usando siete reglas para el controlador difuso y tamaños de población entre 16 y 40 individuos. En cambio, el número de generaciones depende de la convergencia o del estancamiento del algoritmo al tener una población con individuos que no sean muy buenos. Por otro lado, la estrategia elitista es usada en las simulaciones con el fin de intentar preservar la mejor solución alcanzada en cada momento.

El tiempo de aprendizaje es relativo debido a que cuando no se produce cruce y/o mutación, el nuevo individuo no se vuelve a evaluar, pero dicho tiempo de aprendizaje es linealmente dependiente del tamaño de la población y del número de generaciones. En la Tabla 1 se muestra un ejemplo de varios procesos de aprendizaje con sus respectivos tiempos.

Tabla 1: Tiempos de Aprendizaje.

Tamaño	**Generaciones**	**Tiempo**
20	230	70 hrs.
24	70	48 hrs.
40	250	106 hrs.

3.2 EXPERIMENTOS EN UN ENTORNO REAL

Varios experimentos se han realizado en un entorno real para evaluar los controladores difusos aprendidos. La idea es evaluar el rendimiento de un controlador que se obtuvo tras realizar un proceso de aprendizaje mediante la ejecución del algoritmo evolutivo implementado junto con el sistema de simulación de robots ThinkingCap-II. Para realizar una comparación se ha probado el controlador original (obtenido de forma manual) y el aprendido usando una carretilla industrial en un entorno físico. Las pruebas se han desarrollado usando la plataforma i-Fork [8] que está basada en una carretilla paletizadora autónoma que permite realizar tareas de carga y descarga de pallets en una planta industrial agrícola. En dicha planta se establecieron dos puntos de almacenaje para realizar las maniobras de atraque. En ellos se llevaron a cabo diez experimentos

con cada controlador para medir su rendimiento empleando la medida obtenida del láser del AGV y calculando las diferencias entre la posición y orientación deseada y la final tras la maniobra de atraque.

Los resultados obtenidos en las pruebas se resumen en las tablas 2 y 3, para el controlador original y el aprendido respectivamente. En las tablas se indican los valores medios de los experimentos y las desviaciones estándares para los factores que se desean optimizar. Los errores en x, y y en el *ángulo* indicados en las tablas 2 y 3 corresponden al error de la posición final del robot respecto a la posición objetivo y permiten estimar el rendimiento de la maniobra de atraque.

Tabla 2: Controlador Difuso Original.

Tiempo Ejecución (seg)	43,55
Error en X (m)	0,006
Error en Y (m)	0,017
Error en Distancia (m)	0,019
Desviación Estándar Error Distancia	0,014
Error Angular (grados)	0,646
Desviación Estándar Error Angular	0,328

Tabla 3: Controlador Difuso Aprendido.

Tiempo Ejecución (seg)	35,98
Error en X (m)	0,006
Error en Y (m)	0,013
Error en Distancia (m)	0,015
Desviación Estándar Error Distancia	0,009
Error Angular (grados)	0,464
Desviación Estándar Error Angular	0,249

Hemos representado las trayectorias seguidas por la carretilla autónoma para el controlador original y el aprendido en dos gráficas (figuras 1 y 2). En cada una de ellas se muestran dos curvas: la primera con línea discontinua es una curva de tipo B-Spline que corresponde con la trayectoria generada de forma online; y la segunda con línea continua es la trayectoria seguida por el robot.

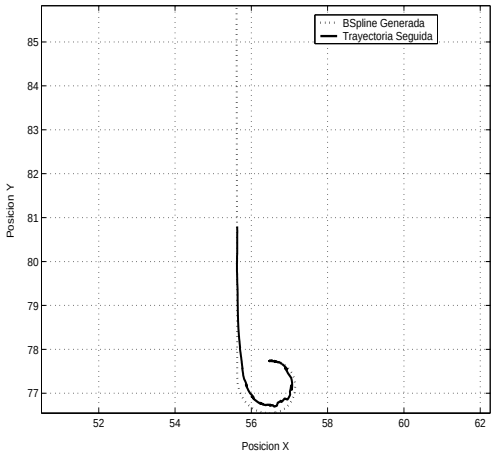


Figura 1: Trayectoria seguida por el controlador difuso original.

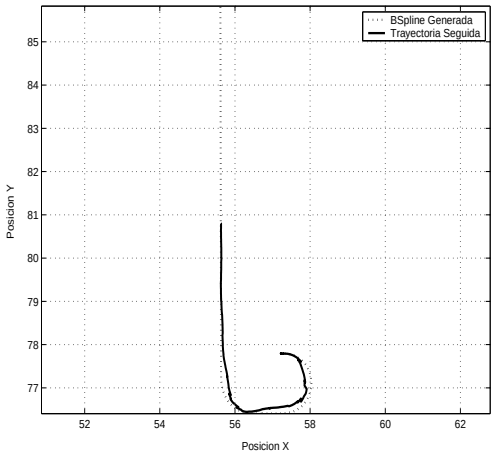


Figura 2: Trayectoria seguida por el controlador difuso aprendido.

4 CONCLUSIONES

En un entorno dinámico con un conocimiento impreciso e incompleto, la definición de la función de fitness o de adecuación para un algoritmo evolutivo es un desafío, ya que tenemos que abordar varios aspectos y valorar cuáles de ellos son los más importantes. En nuestro caso, la función de fitness es una función agregada en la que los subobjetivos son combinados en una suma ponderada. La elección de los coeficientes de ponderación es difícil, pero es la base para evaluar el rendimiento de un controlador difuso después de la simulación en el proceso de aprendizaje. Una cuestión importante es que el proceso de aprendizaje se realiza con un sistema de simulación evaluando cada controlador evolucionado lo que permite tener ventajas como la posibilidad de realizar numerosas evaluaciones para medir la bondad de cada controlador. Aunque el proceso de aprendizaje es costoso ya que necesita tiempo para realizar una gran cantidad de simulaciones, sería imposible llevar a cabo dichas evaluaciones en un entorno real. En cambio hay algunas diferencias entre una simulación y una prueba en un entorno real, así que alguno de los resultados obtenidos de una simulación pueden variar con respecto a los obtenidos en un experimento real. En los resultados obtenidos, varios de los aspectos considerados han sido reducidos como

el tiempo total necesario para realizar la maniobra de atraque, el error angular respecto a la posición final y la deseada del robot y por último también se ha reducido el error en la distancia al punto de atraque con respecto a la posición final del robot. Sin embargo, hay que tener en cuenta que al incrementar de forma considerable la velocidad de la carretilla podría desestabilizarse durante el seguimiento de la trayectoria. De hecho, esta cuestión se observa mejor en el entorno real que en el simulado, ya que en el primero hay que considerar otros factores como la respuesta de la carretilla respecto a los cambios de dirección y velocidad.

En las gráficas de las trayectorias (ver figuras 1 y 2) se puede observar que el controlador aprendido sigue la trayectoria previamente generada aproximándose más que el controlador original y empleando un tiempo menor.

Un objetivo importante que guiará futuros procesos de aprendizaje es que un controlador debe ser rápido pero sin causar finalmente maniobras de atraque imprecisas, por lo que se considera más crítico reducir los errores angulares respecto a la orientación tanto al final como durante la trayectoria y el error en la distancia entre la posición final y la deseada o de destino.

El sistema de aprendizaje implementado es muy útil a la hora de evaluar nuevos controladores difusos y en definitiva es una potente herramienta para generar, validar y ajustar de forma automática nuevos controladores, así como para optimizar el controlador que se obtuvo inicialmente de forma manual.

En este trabajo hemos identificado varios objetivos del problema que han sido integrados en uno solo a optimizar. Un trabajo futuro consiste en diseñar un AE multiobjetivo para capturar en una única ejecución del algoritmo, múltiples soluciones no dominadas del problema, de forma que podamos elegir la solución más adecuada al entorno de decisión existente sin necesidad del volver a ejecutar el algoritmo. En el entorno en el que trabajamos, esto es deseable cuando manejamos, por ejemplo, mercancías frágiles o peligrosas, en las cuáles se prefiere una solución de mayor precisión aunque el tiempo de la maniobra sea mayor.

Agradecimientos

Este trabajo ha sido financiado por el proyecto CICYT TIC 2001-0245-C02-01 del Ministerio de Ciencia y Tecnología de España.

Referencias

[1] R.C. Arkin, D. MacKenzie. Temporal Coordination of Perceptual Algorithms for Mobile Robot Navigation. *IEEE Trans. on Robotics and Automation*, 10(3):276–286, 1994.

[2] R.C. Arkin, R.R. Murphy. Autonomous Navigation in a Manufacturing Environment. *IEEE Trans. on Robotics and Automation*, 6(4):445–454, 1990.

[3] T. Bäck, David B. Fogel. Handbook of Evolutionary Computation. , Oxford, NY, 1997.

[4] P. Bonissone, P. Khedkar, Y. Chen. Genetic Algorithms for Automated Tuning of Fuzzy Controllers: A Tranportation Application. *IEEE Conference on Fuzzy Systems (FUZZ-IEEE'96)*, Pág. 674-680, 1996.

[5] D. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. *Addison-Wesley*, U.S.A., 1989.

[6] F. Hoffmann. Evolutionary Algorithms for Fuzzy Control System Design. *Proceedings of the IEEE*, vol. 89, no. 9, Pág. 1318-1333, 2001.

[7] F. Hoffmann, G. Pfister. Learning of a Fuzzy Control Rule Base Using Messy Genetic Algorithms. *Genetic Algorithms and Soft Computing*, Pág. 279-305, 1996.

[8] H. Martínez-Barberá, J.P. Cánovas, M.A. Zamora, A.F. Gómez-Skarmeta. i-Fork: a Flexible AGV System using Topological and Grid Maps. *Procs. of the IEEE Intl. Conf. on Robotics and Automation*, Pág. 2147-2152, Taipei, Taiwan, 2003.

[9] Z. Michalewicz. Genetic Algorithms, Numerical Optimization and Constraints. *Proceedings of the 6th International Conference on Genetic Algorithms*, Pág. 151-158, Pittsburgh, 1995.

[10] K. Ng, Y. Li. Design of Sophisticated Fuzzy Logic Controllers Using Genetic Algorithms. *IEEE World Congress on Computational Intelligence*, Pág. 1708-1712, 1994.

[11] ThinkingCap-II: http://ants.dif.um.es/∼humberto/robots/tc2

Generación Eficiente de un Controlador Difuso con alta Interpretabilidad para Navegación de Robots Móviles*

Manuel Mucientes
Dept. Electrónica y Computación
Universidad de Santiago de Compostela
15782 Santiago
manuel@dec.usc.es

Jorge Casillas
Dept. Ciencias de la Computación e I.A.
Universidad de Granada
18071 Granada
casillas@decsai.ugr.es

Resumen

El trabajo presenta el diseño de un controlador difuso para el comportamiento de seguir contornos en robótica móvil utilizando la metodología COR con optimización mediante colonias de hormigas. El sistema ha sido evaluado en diferentes entornos simulados, y se ha comparado con otro controlador basado en algoritmos genéticos. La aproximación propuesta obtiene una base de conocimiento altamente interpretable en un tiempo reducido.
Palabras Clave: robótica móvil, seguir contornos, control difuso, interpretabilidad, colonias de hormigas.

1. Introducción

El campo de la robótica móvil se caracteriza por la gran cantidad de incertidumbre presente en los entornos reales. Además, la información que proporcionan los sensores es ruidosa y poco fiable. La lógica difusa ha demostrado ser una herramienta útil para tratar con toda esta incertidumbre, y ha sido ampliamente utilizada para el diseño de comportamientos en robótica [7]. El diseño de sistemas basados en reglas difusas requiere un profundo conocimiento de la tarea a controlar, y su ajuste requiere una gran cantidad de tiempo [5]. Por estos motivos, en los últimos años se ha generalizado el uso de métodos de aprendizaje para el diseño de controladores difusos.

En este artículo se presenta el diseño de un controlador difuso para el seguimiento de contornos en robótica móvil usando la metodología COR (Cooperación entre Reglas) [1, 2]. Las principales ventajas de la aproximación propuesta son la facilidad de diseño, la rapidez en la obtención de la base de conocimiento, y también la alta interpretabilidad de la misma. Finalmente, la utilización de una función que puntúa la acción del controlador sobre cada uno de los ejemplos del conjunto de entrenamiento permite que la calidad del comportamiento aprendido no dependa del entorno.

El artículo se organiza como sigue: en la sección 2 se explica el comportamiento de seguir contornos, y en la 3 la metodología utilizada. La sección 4 muestra los resultados y, finalmente, se exponen las conclusiones.

2. Aprendizaje del comportamiento de seguir contornos

El comportamiento de seguir contornos se utiliza habitualmente cuando el robot se encuentra explorando un área desconocida, o cuando se mueve entre dos puntos en un mapa. Un buen controlador para este comportamiento se debe caracterizar por tres aspectos: mantener una distancia adecuada con el contorno que se sigue, moverse a la máxima velocidad posible y, finalmente, tratar que los movimientos sean suaves y progresivos. En lo que sigue, se asumirá que el robot sigue el contorno situado a su derecha (para seguir el otro contorno basta con realizar un intercambio en las entradas sensoriales).

Las variables de entrada del sistema de control son la distancia derecha (DD), el cociente de distancias (CD) (muestra la posición relativa del robot en el entorno), la velocidad lineal del robot (VL), y la orientación del robot con respecto al contorno que sigue. Las variables de salida son la aceleración lineal y la velocidad angular.

$$CD = \frac{distancia\,izquierda}{DD} \tag{1}$$

Se ha escogido un conjunto de ejemplos (23085) para el aprendizaje de la base de conocimiento. Estos ejemplos cubren todo el universo de discurso de todas las

*Trabajo financiado en parte por el Ministerio de Ciencia y Tecnología mediante los proyectos TIC2002-04036-C05-01, TIC2003-00877 y TIC2003-09400-C04-03

variables de la parte antecedente de la regla, con una precisión p_n, donde n es la variable. La función SF, que puntúa la acción de la base de reglas sobre un ejemplo, se define como:

$$SF\left(RB\left(e^l\right)\right) = \frac{1}{\alpha_1 + \alpha_2 + \alpha_3 + 1} \quad (2)$$

donde α_1, α_2, y α_3 son respectivamente:

$$\alpha_1 = 100\frac{|DD - distancia\ de\ referencia|}{p_{DD}} \quad (3)$$

$$\alpha_2 = 10\frac{|velocidad\ máxima - VL|}{p_{VL}} \quad (4)$$

$$\alpha_3 = \frac{|orientación|}{p_{orientación}} \quad (5)$$

y p_{DD}, p_{VL}, y $p_{orientación}$ son las precisiones de las respectivas variables de entrada. Las precisiones son utilizadas en estas ecuaciones con el fin de evaluar las desviaciones en los valores de las variables respecto a los deseados, pero medidos en unidades de p_n. Esto hace que las desviaciones de las distintas variables sean comparables, con lo que es posible asignar un peso a cada desviación. Estos pesos (100, 10 y 1 para (3), (4), y (5)) han sido obtenidos de forma heurística.

El índice que mide la calidad global de cada conjunto de reglas es:

$$f(RB) = \frac{1}{2 \cdot NE}\sum_{l=1}^{NE}\left\{\left\{1 - \frac{SF\left(RB\left(e^l\right)\right)}{\text{máx}\left(SF\left(e^l\right)\right)}\right\} \cdot \omega\right\}^2 \quad (6)$$

donde NE es el número de ejemplos, ω es un factor de escala que toma el valor 1000, y máx $\left(SF\left(e^l\right)\right)$ es la máxima puntuación que una acción puede obtener para el ejemplo e^l. Estos valores han sido obtenidos antes del comienzo del algoritmo, puntuando todas las posibles acciones sobre cada ejemplo.

3. Metodología de aprendizaje basada en COR

El proceso seguido para aprender el controlador difuso se basa en la metodología COR (Cooperación entre Reglas) propuesta en [1] y extendida en [2]. Hemos seleccionado este proceso debido a sus buenas propiedades para obtener bases de conocimiento rápidamente con gran interpretabilidad. Los siguientes apartados describen la metodología de aprendizaje, un análisis de sus principales propiedades y el algoritmo propuesto basado en dicha metodología.

3.1. Metodología COR

En los últimos años se ha propuesto en la literatura especializada una familia de métodos simples y eficientes de extracción de reglas difusas guiados por criterios de cobertura de los datos del conjunto de ejemplos. Su simplicidad, además de su velocidad y fácil comprensión, hace a estos métodos muy adecuados para tareas de aprendizaje. Sin embargo, estos métodos generalmente buscan reglas difusas con el mejor comportamiento individual (por ejemplo, [8]) y, por tanto, no se considera la interacción global entre las reglas debida al razonamiento interpolado realizado por los sistemas difusos, provocando así una mala precisión en algunos casos.

Con la intención de paliar estos inconvenientes manteniendo las interesantes ventajas de estos métodos, en [1] se propone la metodología COR. En lugar de seleccionar el consecuente con el mejor comportamiento en cada subespacio, esta metodología considera la posibilidad de usar otro consecuente, diferente del mejor, si así se mejora el comportamiento global del sistema difuso gracias a tener una base de reglas con mejor cooperación.

COR se compone de los siguientes pasos: (1) *construcción del espacio de búsqueda*, se obtiene un conjunto de consecuentes candidatos para cada regla; y (2) *selección del conjunto de reglas más cooperativo*, se realiza una búsqueda combinatoria entre los conjuntos para encontrar la combinación de consecuentes con el mejor comportamiento global. En la figura 1 se muestra una descripción más detallada del proceso de generación de reglas basado en COR.

3.2. Ventajas de la Metodología COR para Aprender Controladores Difusos en Navegación con Robots Móviles

La metodología descrita anteriormente tiene varias ventajas interesantes que la hacen muy útil para aprender controladores difusos para navegación con robots móviles. En los siguientes apartados destacamos algunas de ellas.

3.2.1. Reducción del espacio de búsqueda

La metodología COR reduce el espacio de búsqueda basándose en información heurística. Este hecho diferencia a COR de otros métodos de aprendizaje de reglas difusas y le permite ser más rápido y realizar una mejor exploración de las soluciones. Este es un aspecto importante en el aprendizaje de controladores difusos, donde se suele utilizar un alto número de ejemplos. En el comportamiento de seguimiento de contornos presentado en este artículo, donde se consideran 23.085 ejemplos, la metodología propuesta empleó tan solo 20 minutos para obtener el controlador. A diferencia de esto, una solución basada en algoritmos genéticos con el mismo número de ejemplos tardaría varias horas.

Entradas:

- *Un conjunto de datos entrada-salida* —$E = \{e_1, \ldots, e_l, \ldots, e_N\}$, donde $e_l = (x_1^l, \ldots, x_n^l, y_1^l, \ldots, y_m^l)$, $l \in \{1, \ldots, N\}$, siendo N el tamaño del conjunto de datos, y n (m) el número de variables de entrada (salida)— *que represente el comportamiento del problema a resolver.*
- *Una partición difusa de los espacios de las variables.* Sea $\mathcal{A}_i$ el conjunto de términos lingüísticos de la i-ésima variable de entrada, siendo $i \in \{1, \ldots, n\}$, y $\mathcal{B}_j$ el conjunto de términos lingüísticos de la j-ésima variable de salida, siendo $j \in \{1, \ldots, m\}$, donde $|\mathcal{A}_i|$ ($|\mathcal{B}_j|$) es el número de etiquetas de la i-ésima (j-ésima) variable de entrada (salida).

Algoritmo:

1. *Construcción del espacio de búsqueda*:

   1.1. *Definir los subespacios de entrada difusos que contengan ejemplos positivos*: Para ello, dado el conjunto de ejemplos positivos $E^+(S_s)$ para cada subespacio de entrada difuso $S_s = (A_1^s, \ldots, A_i^s, \ldots, A_n^s)$ (con $A_i^s \in \mathcal{A}_i$):

   $$E^+(S_s) = \{e_l \in E \; tal \; que \; \forall i \in \{1, \ldots, n\}, \forall A_i' \in \mathcal{A}_i, \; \mu_{A_i^s}(x_i^l) \geq \mu_{A_i'}(x_i^l)\}, \tag{7}$$

   el conjunto de subespacios con ejemplos positivos se define como $S^+ = \{S_h \mid E^+(S_h) \neq \emptyset\}$.

   1.2. *Generar el conjunto de reglas candidatas en cada subespacio con ejemplos positivos*: En primer lugar se define el conjunto de consecuentes candidatos asociado con cada subespacio $S_h \in S^+$:

   $$\begin{aligned} C(S_h) = \{ \quad & (B_1^{k_h}, \ldots, B_m^{k_h}) \in \mathcal{B}_1 \times \ldots \times \mathcal{B}_m \; tal \; que \\ & \exists e_l \in E^+(S_h) \; donde \; \forall j \in \{1, \ldots, m\}, \forall B_j' \in \mathcal{B}_j, \; \mu_{B_j^{k_h}}(y_j^l) \geq \mu_{B_j'}(y_j^l) \quad \}. \end{aligned} \tag{8}$$

   A continuación se define el conjunto de reglas candidatas para cada subespacio como $CR(S_h) = \{R_{k_h} = [\text{SI } X_1 \text{ es } A_1^h \text{ y ... y } X_n \text{ es } A_n^h \text{ ENTONCES } Y_1 \text{ es } B_1^{k_h} \text{ y ... y } Y_m \text{ es } B_m^{k_h}] \; tal \; que \; (B_1^{k_h}, \ldots, B_m^{k_h}) \in C(S_h)\}$. Para permitir a la metodología COR reducir el número inicial de reglas difusas, se añade el elemento especial $R_\emptyset$ (que significa "no considerar") a cada conjunto de reglas candidatas, es decir, $CR(S_h) = CR(S_h) \cup R_\emptyset$. Si se selecciona este elemento, no se generará ninguna regla en el correspondiente subespacio de entrada difuso.

2. *Selección del conjunto de reglas difusas más cooperativo* — Esta fase se realiza ejecutando un algoritmo de búsqueda combinatoria para encontrar la combinación $RB = \{R_1 \in CR(S_1), \ldots, R_h \in CR(S_h), \ldots, R_{|S^+|} \in CR(S_{|S^+|})\}$ con el mejor comportamiento.

   Se considera un índice $f(RB)$ para medir la calidad global del conjunto de reglas codificada en cada solución. Para obtener soluciones con gran interpretabilidad, se modifica la función original para penalizar excesivo número de reglas:

   $$f'(RB) = f(RB) + \beta \cdot f(RB_0) \cdot \frac{\#RB}{|S^+|} \tag{9}$$

   siendo $\beta \in [0, 1]$ un parámetro que regula la importancia del número de reglas, $\#RB = |\{R_h \in RB \; tal \; que \; R_h \neq R_\emptyset\}|$ el número de reglas usado en la solución a evaluar y RB_0 la base de reglas inicial.

Figura 1: Algoritmo COR

Esta reducción del espacio de búsqueda se realiza imponiendo dos restricciones:

1. *Número máximo de subespacios de entrada difusos*: El número máximo de subespacios de enterada difusos, y por tanto el número máximo de reglas difusas, se limita mediante el conjunto de ejemplos positivos. Las restricciones impuestas para construir $E^+(S_s)$ — ecuación (7)— dividen el espacio de entrada en una rejilla crisp mediante los puntos de corte entre los conjuntos difusos de las etiquetas y, por consiguiente, cada ejemplo contribuye a generar una sola regla. Esto es una selección del conjunto de espacios conservadora que genera el menor número posible de reglas que garantice una cobertura completa de los ejemplos.

En nuestro problema, dado que el conjunto de ejemplos está uniformemente distribuido en el espacio de entrada completo (como se describe en la sección 2), no se realiza una reducción del número máximo de subespacios de entrada difusos. Sin embargo, el hecho de asignar cada ejemplo a un solo subespacio implica reducir el número de consecuentes candidatos, ya que el conjunto de ejemplos positivos se reduce.

2. *Conjunto de reglas candidatas en cada subespacio*: Se realiza una segunda reducción del espacio de búsqueda restringiendo el conjunto de consecuentes posibles para cada combinación de antecedentes. De nuevo, usamos una condición restrictiva para construir $C(S_h)$ —ecuación (8)— que genera un menor número de reglas candidatas.

Para ilustrar el efecto de esta reducción del espacio de búsqueda podemos comprobar que, a partir del conjunto de datos de ejemplos propuesto en la sección 2, y usando el siguiente número de términos lingüísticos para cada variable de entrada/salida, $|\mathcal{A}_1| = 5$, $|\mathcal{A}_2| = 2$, $|\mathcal{A}_3| = 5$, $|\mathcal{A}_4| = 2$, $|\mathcal{B}_1| = 9$, $|\mathcal{B}_2| = 9$, nuestra metodología genera un espacio de búsqueda de $\prod_{S_h \in S^+} |C(S_h)| = 9{,}8\text{e}{+}89$ combinaciones, mientras que el total de posibles combinaciones (considerando los $|S^+| = 100$ subespacios de entrada analizados) es $(|\mathcal{B}_1| \cdot |\mathcal{B}_2|)^{100} = 7{,}1\text{e}{+}190$.

3.2.2. Propiedades de interpretabilidad

La metodología propuesta tiene también algunas ventajas interesantes desde el punto de vista de la interpretabilidad del modelo difuso obtenido. Este es un aspecto importante en control difuso para navegación de robots móviles ya que permite que las acciones del robot sean fácilmente compresibles y puedan ser comunicadas a otros módulos que supervisen el comportamiento de la arquitectura de control. Básicamente, podemos destacar dos aspectos:

• *La estructura del modelo y las funciones de pertenencia permanecen invariables para una excelente interpretabilidad*: La metodología COR es un esfuerzo por explotar la capacidad de precisión de los sistemas difusos lingüísticos centrándose exclusivamente en el diseño de la base de reglas. En este caso, las funciones de pertenencia y la estructura del modelo se mantienen invariables, resultando así en la máxima interpretabilidad.

• *Reducción de la base de reglas para mejorar la interpretabilidad y la precisión*: Al definir una base de reglas, es común incluir reglas redundantes o reglas que generan conflictos con otras en ciertas ocasiones. Además, una base con un alto número de reglas es difícil de interpretar, incluso cuando se considera una estructura de regla difusa lingüística.

Para afrontar este problema se suele realizar un *post-procesamiento*. Cuando se consideran restricciones de interpretabilidad, las reglas pueden ser fusionadas [6], generando así una estructura dispersa (no basada en una rejilla) donde cada regla usa diferentes conjuntos difusos para cada variable.

Otra posibilidad si queremos obtener reglas difusas lingüísticas con gran interpretabilidad es realizar un proceso de selección para obtener un subconjunto de la base de reglas original. Sin embargo, este enfoque no parece apropiado para generar un conjunto de reglas preciso ya que no se considera la interdependencia existente entre las tareas de aprendizaje y reducción. Es decir, es seguro que después de la reducción del conjunto de reglas, el nuevo conjunto de reglas que mejor cooperan será diferente.

La metodología COR realiza el proceso de reducción al mismo tiempo que el aprendizaje con la intención de mejorar la precisión (eliminando reglas conflictivas y erróneas) y la interpretabilidad. Este proceso se realiza añadiendo la regla nula ($R_\emptyset$) a cada conjunto de reglas candidato correspondiente a cada subespacio, como se muestra en el paso 1.2. de la figura 1. Además, se considera una función objetivo modificada, ecuación (9), para penalizar un alto número de reglas y favorecer aún más la reducción.

3.3. Metodología COR con Optimización Basada en Colonias de Hormigas

Dado que el espacio de búsqueda manejado en el paso 2. de la metodología COR es grande, resulta necesario usar técnicas de búsqueda aproximadas. En [1] se obtuvieron modelos lingüísticos con buena precisión utilizando enfriamiento simulado. Sin embargo, dado que una de las restricciones de nuestro problema es el manejo de una función de evaluación computacionalmente costosa, en este artículo se propone el uso de optimización mediante colonias de hormigas (OCH) [4]. Esta es una técnica de búsqueda bio-inspirada que considera información heurística para permitir obtener buenas soluciones con rapidez. Esta sección describe los componentes del algoritmo propuesto.

3.3.1. Representación del problema basado en COR

Para aplicar OCH en la metodología COR es conveniente verla como un problema de optimización combinatoria con la capacidad de ser representado en un grafo ponderado. De esta forma, se puede afrontar el problema considerando un número fijo de subespacios e interpretando el proceso de aprendizaje como la forma de asignar vectores de consecuentes —es decir, etiquetas de las particiones difusas de las variables de salida— a estos subespacios según un criterio de optimalidad (esto es, siguiendo la metodología COR).

Por tanto, de acuerdo con la figura 1, cada nodo $S_h \in S^+$ se asigna a cada consecuente candidato $(B_1^{k_h}, \ldots, B_m^{k_h}) \in C(S_h)$ y al símbolo especial "no considerar" ($R_\emptyset$) que indica la ausencia de reglas en el correspondiente subespacio.

3.3.2. Información heurística

La información heurística sobre la preferencia potencial de elegir un vector de consecuentes específico, B^{k_h}, en cada combinación de antecedentes (subespacio) se determina de la siguiente forma para $B^{k_h} =$

$(B_1^{k_h}, \ldots, B_m^{k_h}) \in C(S_h)$:

$$\eta_{hk_h} = \max_{e_l \in E^+(S_h)} Min\left(\mu_{A^h}(x^l), \mu_{B_j^{k_h}}(y^l)\right). \quad (10)$$

Si $B^{k_h} \notin C(S_h)$, hacer $\eta_{hk_h} = 0$.

Posteriormente, para el símbolo "no considerar", hacer lo siguiente:

$$\eta_{h,|\mathcal{B}_1|\cdot\ldots\cdot|\mathcal{B}_m|+1} = \frac{1}{\max\limits_{k_h \in \{1,\ldots,|C(S_h)|\}} \eta_{hk_h}}. \quad (11)$$

3.3.3. Inicialización de la feromona

El valor inicial de la feromona para cada asignación se obtiene de la siguiente forma:

$$\tau_0 = \frac{1}{|S^+|} \sum_{S_h \in S^+} \max_{B^{k_h} \in C(S_h)} \eta_{hk_h}. \quad (12)$$

Así, la feromona inicial será el valor medio del camino construido tomando el mejor consecuente en cada regla de acuerdo con la información heurística (se trata, por tanto, de una asignación voraz).

3.3.4. Esquema de OCH: sistema de la mejor-peor hormiga

Una vez diseñados los componentes previos, se debe definir un algoritmo OCH para resolver el problema. En esta contribución consideramos el algoritmo sistema de la mejor-peor hormiga (SMPH) [3]. Su esquema global se muestra en la figura 2, para más detalle refiérase a [3].

1. Dar un valor inicial de la feromona, τ_0, para cada arco.
2. Mientras (*condición_terminación* no se cumpla) hacer:
   - a) Realizar el camino de cada hormiga mediante el **proceso de construcción de la solución**.
   - b) Aplicar el **mecanismo de evaporación de feromona**.
   - c) Aplicar el **proceso de búsqueda local** sobre la mejor solución actual.
   - d) Actualizar $S_{mejor\ global}$ y $S_{peor\ actual}$.
   - e) Aplicar la **regla de actualización de feromona Mejor-Peor**.
   - f) Aplicar la **mutación de feromona**.
   - g) Si (*condición_estancado* se cumple), aplicar **reinicialización**.

Figura 2: Algoritmo SMPH

4. Resultados

El controlador aprendido ha sido probado en cuatro entornos simulados diferentes utilizando el software del robot Nomad 200. Estos entornos incluyen todo el abanico de situaciones a las que el robot puede enfrentarse durante la navegación. Ninguno de los entornos de prueba ha sido utilizado durante el aprendizaje, puesto que en esta etapa se ha empleado un conjunto de ejemplos escogidos de forma que se cubra todo el espacio de entradas con una precisión adecuada. Estas condiciones garantizan que la calidad del controlador obtenido no dependa de los entornos en los que se ha entrenado el sistema, y además que el robot sea capaz de enfrentarse a cualquier situación.

La figura 3 muestra el camino seguido por el robot a lo largo de uno de los entornos de prueba (entorno D). La trayectoria del robot se representa mediante círculos. Una mayor concentración indicará menor velocidad. El controlador aprendido tiene 77 reglas y ha sido obtenido en menos de 20 minutos. Si en una determinada situación no se dispara ninguna regla, se seleccionará aceleración lineal y velocidad angular nulas. La máxima velocidad del robot es de 61 cm/s, y la distancia de referencia a la que se debe seguir la pared es de 51 cm. Se han llevado a cabo 10 pruebas para cada uno de los entornos de test. En la tabla 1 se muestran los valores medios de algunos parámetros que reflejan el rendimiento del controlador. Estos parámetros son la distancia media a la pared derecha (el contorno que se sigue), la velocidad lineal media, el tiempo empleado en recorrer el camino, y el cambio medio en velocidad entre dos ciclos consecutivos (refleja la suavidad del comportamiento).

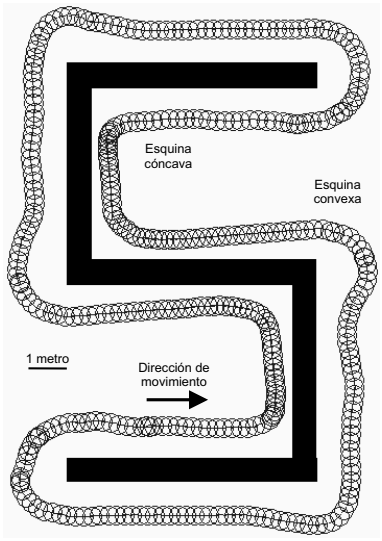


Figura 3: Recorrido del robot en el entorno D.

Con el fin de mostrar la calidad del controlador se va a describir el recorrido del robot en el entorno D (figura 3). Este entorno es bastante complejo, pues tiene cuatro esquinas cóncavas y seis convexas a lo largo de sus 59 metros. En estas situaciones el robot deja de

detectar la pared correctamente en algunos instantes, y además debe reducir la velocidad significativamente. A pesar de ello la velocidad media obtenida ha sido alta, y la distancia a la que se ha seguido el contorno se encuentra próxima a la distancia de referencia. La diferencia es debida al elevado número de esquinas de este entorno, ya que en ellas la orientación del robot es mala, y se da prioridad al giro rápido antes que a una distancia correcta.

Ent.	DD (cm)	Vel. (cm/s)	Cambio vel. (cm/s)	Tpo (s)
A	65	55	5.73	62
B	57	52	5.75	103
C	55	50	4.10	84
D	55	54	4.27	112

Tabla 1: Valores medios de algunos parámetros (77 reglas lingüísticas).

Para evaluar el controlador obtenido, se ha realizado una comparación con otro diseñado utilizando algoritmos genéticos [6]. El diseño en [6] constaba de dos etapas: aprendizaje de la base de datos y una base de reglas general (aproximación Pittsburgh), y reducción de la base de reglas generada mediante la fusión de reglas. Esta reducción provoca una pérdida de interpretabilidad en la base de conocimiento final, pues las etiquetas usadas en cada regla no son generales, sino dependientes de la misma. Este controlador tiene 46 reglas (en la tabla 2 se muestran los valores medios de algunos parámetros para los entornos de prueba). El tiempo empleado en el aprendizaje es elevado (varias horas) si se compara con los menos de 20 minutos necesarios para la obtención del controlador presentado en este trabajo.

Ent.	DD (cm)	Vel. (cm/s)	Cambio vel. (cm/s)	Tpo (s)
A	59	46	8.69	72
B	54	43	9.70	120
C	54	36	7.75	114
D	54	46	7.18	127

Tabla 2: Valores medios de algunos parámetros para el sistema descrito en [6] (46 reglas difusas)

El controlador descrito en este trabajo presenta un mayor número de reglas (77 vs. 46), pero mejora los resultados obtenidos en [6]. La velocidad media es mayor en todos los entornos, reduciendo el tiempo de navegación. La distancia derecha media es similar en ambos controladores y, finalmente, el cambio medio en velocidad se reduce bastante en este controlador, lo que refleja la calidad del comportamiento obtenido en comparación con [6]. Por otra parte, la interpretabilidad de las reglas es alta, en contraposición a [6].

5. Conclusiones

En este trabajo se ha presentado el diseño de un controlador difuso para el seguimiento de contornos utilizando la metodología COR. Las principales ventajas de la aproximación propuesta son la facilidad de diseño, la rapidez en la obtención de la base de conocimiento, el alto grado de interpretabilidad de la misma y, finalmente, que la calidad del comportamiento obtenido no depende del conjunto de entrenamiento.

El controlador ha sido probado en varios entornos simulados mostrando valores adecuados en los valores medios de algunos parámetros que reflejan la calidad del comportamiento. El sistema ha sido también comparado con un diseño previo basado en algoritmos genéticos, incrementando la calidad e interpretabilidad del mismo.

Referencias

[1] J. Casillas, O. Cordón, F. Herrera. COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules. *IEEE Trans. Syst., Man, Cybern. B*, 32(4):526–537, 2002.

[2] J. Casillas, O. Cordón, F. Herrera. COR methodology: a simple way to obtain linguistic fuzzy models with good interpretability and accuracy. In J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.), *Accuracy improvements in linguistic fuzzy modeling*. Springer, Heidelberg, Germany, 2003.

[3] O. Cordón, F. Herrera, I. Fernández de Viana, L. Moreno. A new ACO model integrating evolutionary computation concepts: the best-worst ant system. En *Proc. 2nd Int. Workshop on Ant Algorithms*, 22–29, Brussels, Belgium, 2000.

[4] M. Dorigo, G. Di Caro. The ant colony optimization meta-heuristic. En D. Corne, M. Dorigo, F. Glover (Eds.), *New ideas in optimization*, 11–32. McGraw-Hill, New York, USA, 1999.

[5] M. Mucientes, R. Iglesias, C. V. Regueiro, A. Bugarín, S. Barro. A fuzzy temporal rule-based velocity controller for mobile robotics. *Fuzzy Sets and Systems*, 134:83–99, 2003.

[6] M. Mucientes, D. L. Moreno, C. V. Regueiro, A. Bugarín, S. Barro. Design of a fuzzy controller for the wall-following behavior in mobile robotics with evolutionary algorithms. En *Proc. IPMU'2004*, aceptado, 2004.

[7] A. Saffiotti. The uses of fuzzy logic in autonomous robot navigation. *Soft Computing*, 1(4):180–197, 1997.

[8] L.-X. Wang, J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Trans. Syst., Man, Cybern.*, 22(6):1414–1427, 1992.

RELACIONES BORROSAS ASOCIADAS A TRIÁNGULOS DE PESOS*

R.Fuentes-González‡, G. Mayor†

‡Universidad Pública de Navarra. 31006 Pamplona
rfuentes@unavarra.es
†Universidad de las Islas Baleares. 07122 Palma de Mallorca
gmayor@uib.es

Resumen

Se introduce el concepto de relación de distribución de un triángulos de pesos $\triangle$ como cierta relación borrosa $F_{\triangle} \in [0,1]^{\mathbb{N}\times\mathbb{N}}$, demostrando que caracteriza al triángulo $\triangle$. Se utiliza ese concepto para proponer un método general de extensión de relaciones binaria nítidas en $[0,1]$ al conjunto $\mathcal{TRI}$ de triángulos $\triangle$ de pesos. Se introduce una estructura de retículo acotado $(\mathcal{TRI}, \preccurlyeq, \curlywedge, \curlyvee, \triangle_{(0)}, \triangle_{(1)})$ en el conjunto de triángulos de pesos, en el que se sumerge la cadena $([0,1], \leq, \max, \min, 0, 1)$. Utilizando extensiones a $[0,1]^{\mathbb{N}\times\mathbb{N}}$ de operaciones monarias y binarias en $[0,1]$, se propone en $\mathcal{TRI}$ extensiones de las mismas. Finalmente, se obtiene la expresión del índice de "orness" de un triángulo de pesos en función de su relación de distribución.

Palabras clave: Triángulo de pesos, funciones de agregación extendidas.

1 INTRODUCCIÓN

Desde hace algún tiempo el problema de la agregación/fusión de la información es un área de estudio importante para un buen grupo de investigadores. Clases diferentes de operadores útiles en el campo de la agregación han sido introducidos y estudiados de forma exhaustiva, pero la mayor parte de ellos tienen carácter n-dimensional, es decir el número de inputs a agregar es constante e igual a n. La existencia de operadores fundamentales intrínsecamente multidimensionales hace que se plantee el estudio de la agregación multidimensional [5], [6], en particular es de interés el tratamiento de los operadores tipo OWA y media ponderada en su formato multidimensional [2], [3], [4]. Así, por ejemplo, un operador OWA multidimensional viene definido por un *triángulo de pesos* [1] sobre el que quedan reflejadas las condiciones exigidas al operador, en particular la monotonía que resulta de poder comparar listas de inputs de diferente longitud.

Los triángulos de pesos están asociados a sucesiones del tipo

$$\triangle = (1, (w_1^2, w_2^2), (w_1^3, w_2^3, w_3^3), ..., (w_1^n, w_2^n, ...w_i^n, ..., w_n^n), ...)$$

en las que los números $w_i^n \in [0,1]$ son tales que la suma de los componentes de cada elemento $(w_1^n, w_2^n, ..., w_n^n)$ de la sucesión verifica: $\sum_{i=1}^{n} w_i^n = 1 \ \ \forall n$. Estas sucesiones pueden representarse en disposición triangular:

$$\triangle = \begin{array}{ccccccc} & & 1 & & & & \\ & & w_1^2 & w_2^2 & & & \\ & & w_1^3 & w_2^3 & w_3^3 & & \\ & & & \cdots & & & \\ & & w_1^n & w_2^n & w_3^n & w_4^n & \cdots & w_n^n \\ & & & \cdots & & & \end{array},$$

disposición que los autores [1] llaman *triángulo de pesos*. Representamos aquí la clase de estos triángulos de pesos por $\mathcal{TRI}$. Ejemplos de elementos $\triangle \in \mathcal{TRI}$ son:

(a) Para $\alpha \in [0,1]$, consideremos la sucesión $\triangle_{(\alpha)}$:

$$(1, (\alpha, 1-\alpha), (\alpha, 0, 1-\alpha), \cdots, (\alpha, \underbrace{0, \ldots, 0}_{n-2}, 1-\alpha) \cdots).$$

Por ejemplo, los triángulos de pesos $\triangle_{(0)}, \triangle_{(\alpha)}, \triangle_{(1)}$ tienen la forma:

$$\begin{array}{ccccc} 1 & & & & \\ 0 & 1 & & & \\ 0 & 0 & 1 & & \\ 0 & 0 & 0 & 1 & \\ & \cdots & & & \end{array}, \quad \begin{array}{ccccc} 1 & & & & \\ \alpha & 1-\alpha & & & \\ \alpha & 0 & 1-\alpha & & \\ \alpha & 0 & 0 & 1-\alpha & \\ & \cdots & & & \end{array}, \quad \begin{array}{ccccc} 1 & & & & \\ 1 & 0 & & & \\ 1 & 0 & 0 & & \\ 1 & 0 & 0 & 0 & \\ & \cdots & & & \end{array}.$$

(b) Para cada $\alpha \in]0,1[$, consideremos ahora el triángulo de pesos

$\triangle_{p(\alpha)} := (w_m^n = \binom{n-1}{m-1} \alpha^{n-m} (1-\alpha)^{m-1}, n \in \mathbb{N}, m \leq n)$.

En particular, $\triangle_{p(\frac{1}{2})} := (w_m^n = \frac{\binom{n-1}{m-1}}{2^{n-1}}, n \in \mathbb{N}, m \leq n)$ es el triángulo de Pascal normalizado [1]:

$$\triangle_{p(\frac{1}{2})} = \begin{array}{ccccc} & & 1 & & \\ & 1/2 & & 1/2 & \\ 1/4 & & 2/4 & & 1/4 \\ 1/8 & & 3/8 & & 3/8 & & 1/8 \\ & & \cdots & & \end{array}.$$

(c) $\triangle_{median} := \begin{cases} w_m^{2k} = \frac{1}{2} & si \quad m \in \{k, k+1\} \\ w_m^{2k-1} = 1 & si \quad m = k \\ 0 & en \quad otro\ caso \end{cases}$

*Financiado parcialmente por el proyecto de investigación TIC2003-08693 del Ministerio de Ciencia y Tecnología.

$$\triangle_{med} := (w_m^n = \tfrac{1}{n},\, n \in \mathbb{N}, m \le n):$$

$$\triangle_{median} = \begin{matrix} 1 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 0 & 1 & 0 & \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ & \cdots & & \end{matrix} \qquad \triangle_{med} = \begin{matrix} 1 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ & \cdots & & \end{matrix} .$$

(d) Los valores $w_m^n \in [0,1]$ definidos para $(n,m) \in \mathbb{N}\times\mathbb{N}$ con $m \le n$ por:

$$w_m^n = \begin{cases} 1 & si \quad 1 = n = m \\ 0 & si \quad 1 = m < n \\ \frac{1}{3n} & si \quad 1 < m < n \\ \frac{2(n+1)}{3n} & si \quad 1 < m = n \end{cases}$$

verifican: $\sum_{i=1}^{n} w_i^n = 1 \quad \forall n > 1$, luego constituyen el triángulo de pesos $\triangle_d$:

$$\triangle_d = \begin{matrix} 1 & & & \\ 0 & 1 & & \\ 0 & 1/9 & 8/9 & \\ 0 & 1/12 & 1/12 & 10/12 \\ & \cdots & & \end{matrix} .$$

Estamos interesados aquí en los siguientes conceptos asociados a triángulos de pesos:

El *triángulo de pesos inverso* [1]:

$$\triangle^r = (1, (u_1^2, u_2^2), ..., (u_1^n, u_2^n, ...u_j^n, ..., u_n^n), ...)$$

asociado a $\triangle = (1, (w_1^2, w_2^2), ..., (w_1^n, w_2^n, ...w_i^n, ..., w_n^n), ...)$ es tal que: $u_j^n = w_{n-j+1}^n$ para $n > 1$ y $j \in \{1, 2, ..., n\}$. Por ejemplo:

$$\triangle_{(\alpha)}^r = \triangle_{(1-\alpha)}, \triangle_{p(\alpha)}^r = \triangle_{p(1-\alpha)}, \triangle_{median}^r = \triangle_{median} .$$

El operador $^r : \mathcal{TRI} \to \mathcal{TRI}$ que asocia a $\triangle$ su inverso $\triangle^r$, es involutivo: $(\triangle^r)^r = \triangle \;\; \forall \triangle \in \mathcal{TRI}$.

Un triángulo de pesos $\triangle \in \mathcal{TRI}$ es *regular* [1] si para $n > 1$ y $p = 1, 2, ..., n$ se verifica:

$$\sum_{i=1}^{p} w_i^{n+1} \le \sum_{i=1}^{p} w_i^n \le \sum_{i=1}^{p+1} w_i^{n+1}.$$

Sea $\mathcal{TRIR} \subset \mathcal{TRI}$ la clase de triángulos de pesos regulares.

2 RELACIÓN DE DISTRIBUCIÓN $F_\triangle$ ASOCIADA A UN TRIÁNGULO DE PESOS $\triangle$

En este trabajo caracterizamos en primer lugar cada triángulo de pesos $\triangle \in \mathcal{TRI}$ por medio de una relación binaria borrosa $F_\triangle \in [0,1]^{\mathbb{N}\times\mathbb{N}}$ que llamaremos *relación de distribución* de $\triangle$. Utilizamos ese concepto para proponer un método general de extensión de relaciones binaria nítidas en $[0,1]$ al conjunto $\mathcal{TRI}$. En particular, la extensión $\mathcal{TRI}$ del orden parcial $\le$ habitual en $[0,1]^{\mathbb{N}\times\mathbb{N}}$proporciona una estructura de retículo acotado $(\mathcal{TRI}, \preccurlyeq, \curlywedge, \curlyvee, \triangle_{(0)}, \triangle_{(1)})$, (con $\mathcal{TRIR}$ como subretículo), en el que se sumerge la cadena $([0,1], \le, \max, \min, 0, 1)$. Utilizando extensiones a $[0,1]^{\mathbb{N}\times\mathbb{N}}$ de operaciones monarias y binarias en $[0,1]$, (negaciones, normas e implicaciones entre otras), proponemos en $\mathcal{TRI}$ y $\mathcal{TRIR}$ operaciones análogas, recuperando como una de ellas el operador inverso ^{r} [1]. Finalmente, se obtiene la expresión del índice de "orness" de un triángulo de pesos en función de su relación de distribución.

2.1 Definición. Sea $\mathbb{N} = \{1, 2, \cdots\}$. Dado el triángulo de pesos

$\triangle = (1, (w_1^2, w_2^2), ..., (w_1^n, w_2^n, ...w_i^n, ..., w_n^n), ...)$, llamaremos *relación de distribución* de $\triangle$, a la relación borrosa $F_\triangle : \mathbb{N}\times\mathbb{N} \longrightarrow [0,1]$ tal que

$$F_\triangle(n,m) = \begin{cases} w_1^n & si \quad m = 1 \\ \sum_{i=1}^{m} w_i^{n+m-1} & si \quad m > 1 \end{cases} ,$$

Esta relación está bien definida, ya que, para cualquier $(n,m) \in \mathbb{N}\times\mathbb{N}$:

$$0 \le \sum_{i=1}^{m} w_i^{n+m-1} \le \sum_{i=1}^{n+m-1} w_i^{n+m-1} = 1 \Longrightarrow F_\triangle(n,m) \in [0,1]$$

En particular, $\forall m \in \mathbb{N} : F_\triangle(1,m) = \sum_{i=1}^{m} w_i^m = 1$.

Podemos disponer los elementos iniciales de la relación $F_\triangle$ en forma de tabla no limitada:

1	1	1	$\cdots$	1	$\cdots$
$F_\triangle(2,1)$	$F_\triangle(2,2)$	$F_\triangle(2,3)$	$\cdots$	$F_\triangle(2,m)$	$\cdots$
$F_\triangle(3,1)$	$F_\triangle(3,2)$	$F_\triangle(3,3)$	$\cdots$	$F_\triangle(3,m)$	$\cdots$
	$\cdots$				
$F_\triangle(n,1)$	$F_\triangle(n,2)$	$F_\triangle(n,3)$	$\cdots$	$F_\triangle(n,m)$	$\cdots$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	.

2.2 Ejemplos. Escribiendo F_S en lugar de $F_{\triangle_s}$, las expresiones y los primeros valores de las relaciones de distribución asociadas a los triángulos de pesos que aparecen en *(a)*, *(b)*, *(c)* y *(d)* del apartado anterior son:

(a') $F_{(\alpha)}(n,m) = \begin{cases} 1 & si \quad n = 1 \\ \alpha & si \quad n > 1 \end{cases} \quad \forall m \in [0,1]$, en particular:

$F_{(0)}(n,m) = 0$ si $n > 1$,

$F_{(1)}(n,m) = 1 \;\; \forall (n,m) \in \mathbb{N}\times\mathbb{N}$:

$$F_{(0)}=\begin{matrix}1&1&1&1\\0&0&0&0\\0&0&0&0\\0&0&0&0\\&\dots&&\end{matrix}\quad F_{(1)}=\begin{matrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\\&\dots&&\end{matrix}\quad\dots\quad(1)$$

(b') $F_{p(\frac{1}{2})}(n,m)=\frac{1}{2^{n+m-2}}\sum_{i=1}^{m}\binom{n+m-2}{i-1}$:

$$F_{p(\frac{1}{2})}=\begin{matrix}1&1&1&1\\1/2&3/4&7/8&15/16\\1/4&6/8&11/16&26/32&\dots\\1/8&10/16&16/32&42/64\\&\dots&&\end{matrix}.$$

(c') $$F_{median}(n,m)=\begin{cases}0 & si \quad m\leq n-2\\ \frac{1}{2} & si \quad m=n-1\\ 1 & si \quad m\geq n\end{cases},$$

$F_{med}(n,m)=\frac{m}{n+m-1}$:

$$F_{median}=\begin{matrix}1&1&1&1\\\frac{1}{2}&1&1&1\\0&\frac{1}{2}&1&1\\0&0&\frac{1}{2}&1\\&\dots&&\end{matrix}\quad..\quad F_{med}=\begin{matrix}1&1&1&1\\\frac{1}{2}&\frac{2}{3}&\frac{3}{4}&\frac{4}{5}\\\frac{1}{3}&\frac{2}{4}&\frac{3}{5}&\frac{4}{6}\\\frac{1}{4}&\frac{2}{5}&\frac{3}{6}&\frac{4}{7}\\&\dots&&\end{matrix}\quad..$$

(d') La expresión de la relación de distribución F_d del triángulo de pesos $\triangle_d$ es:

$$F_d(n,m)=\begin{cases}1 & si \quad n=1\\ \frac{m-1}{3(n+m-1)} & si \quad n>1\end{cases}\ \forall(n,m)\in\mathbb{N}\times\mathbb{N}:$$

$$F_d=\begin{matrix}1&1&1&1\\0&1/9&2/12&3/15\\0&1/12&2/15&3/18&..\\0&1/15&2/18&3/21\\&\dots&&\end{matrix}.$$

La relación $F_\triangle$ verifica las siguientes propiedades:

2.3 Proposición. *Para todo $\triangle\in\mathcal{TRI}$, se verifica:*
(i) $F_\triangle(1,m)=1\ \ \forall m\in\mathbb{N}$.
(ii) $F_\triangle(n+1,m)\leq F_\triangle(n,m+1)\ \ \forall(n,m)\in\mathbb{N}\times\mathbb{N}$.
Si el triángulo de pesos es regular, se encuentra un valor entre los dos anteriores:
(ii') *Si* $\triangle\in\mathcal{TRIR}$ *entonces* $\forall(n,m)\in\mathbb{N}\times\mathbb{N}$:
$F_\triangle(n+1,m)\leq F_\triangle(n,m)\leq F_\triangle(n,m+1)$.

Y los valores w_k^n del triángulo $\triangle$ quedan a su vez determinados por $F_\triangle$ como sigue:

2.4 Proposición. *Para todo triángulo de pesos* $\triangle=(1,(w_1^2,w_2^2),...,(w_1^n,w_2^n,...w_i^n,...w_n^n),...)$ *se verifica:*
$w_1^n=F_\triangle(n,1)\ \ \forall n\in\mathbb{N}$ y
$w_k^n=F_\triangle(n-k+1,k)-F_\triangle(n-k+2,k-1)$ si $k>1$.

Por otra parte, si $\mathcal{DT}\subset[0,1]^{\mathbb{N}\times\mathbb{N}}$ y $\mathcal{DTR}\subset\mathcal{DT}$ representan respectivamente las clases de relaciones borrosas que cumplen *(i)* , *(ii)* y *(i)* , *(ii')* de la proposición 2.3:

$$\mathcal{DT}=\{G\in[0,1]^{\mathbb{N}\times\mathbb{N}}/(G(1,m)=1)\&\\(G(n+1,m)\leq G(n,m+1))\},\qquad(2)$$

$$\mathcal{DTR}=\{G\in[0,1]^{\mathbb{N}\times\mathbb{N}}/(G(1,m)=1)\&\\(G(n+1,m)\leq G(n,m)\leq G(n,m+1))\},\qquad(3)$$

entonces se cumple el siguiente resultado:

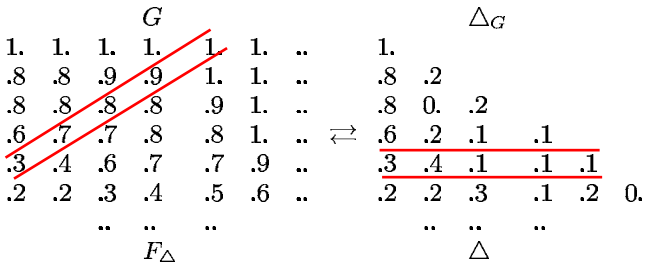
2.5 Proposición. *(i) Sea* $G\in\mathcal{DT}$ *y sea* $\triangle_G=(1,(w_1^2,w_2^2),...,(w_1^n,w_2^n,...w_i^n,...,w_n^n),...)$ *la sucesión construida a partir de* G *por:*

$$w_k^n=\begin{cases}G(n,k), & k=1\\ G(n-k+1,k)-G(n-k+2,k-1), & k>1\end{cases}.$$

Entonces $\triangle_G$ *es un triángulo de pesos cuya relación de distribución es* $F_{\triangle_G}=G$.
(ii) Si $G\in\mathcal{DTR}$ *entonces el triángulo* $\triangle_G$ *es regular.*

3 ESTRUCTURAS INDUCIDAS EN LA CLASE DE TRIÁNGULOS $\mathcal{TRI}$ POR OPERACIONES EN $\mathcal{DT}$

Los resultados de la sección anterior muestran que un triángulo de pesos $\triangle$ perteneciente a $\mathcal{TRI}$ (o a $\mathcal{TRIR}$) queda determinado por una relación borrosa $F_\triangle$, (su relación de distribución), perteneciente a $\mathcal{DT}\subset[0,1]^{\mathbb{N}\times\mathbb{N}}$ (o a $\mathcal{DTR}\subset\mathcal{DT}$). Recíprocamente, una relación borrosa $G\in[0,1]^{\mathbb{N}\times\mathbb{N}}$ que verifique las propiedades de los apartados *(i)* y *(ii)* o *(i)* y *(ii')* de la proposición 2.3, determina un triángulo de pesos. Obtenemos tanto triángulos de pesos $\triangle_G$ partiendo de una relación borrosa G, como relaciones borrosas $F_\triangle$ a partir de triángulos de pesos $\triangle$:



Además, justificaremos en lo que sigue que la interpretación de los triángulos como relaciones borrosas proporciona un método para definir leyes monarias o binarias nítidas entre triángulos utilizando leyes análogas en

$[0,1]^{\mathbb{N}\times\mathbb{N}}$ que sean cerradas en $\mathcal{DT}$ (o en $\mathcal{DTR}$). También podemos definir relaciones nítidas entre triángulos partiendo de relaciones entre elementos de $\mathcal{DT}$. Estas definiciones responden a los esquemas siguientes:

Si $\gamma : \mathcal{DT} \to \mathcal{DT}$ y $\eta : \mathcal{DT} \times \mathcal{DT} \to \mathcal{DT}$ son leyes de composición para relaciones F, se definen las análogas $\widehat{\gamma} : \mathcal{TRI} \to \mathcal{TRI}$ y $\widehat{\eta} : \mathcal{TRI} \times \mathcal{TRI} \to \mathcal{TRI}$ para triángulos de pesos $\triangle$ por:

$$\widehat{\gamma}(\triangle) = \triangle_{\gamma(F_{\triangle})} \tag{4}$$

$$\widehat{\eta}(\triangle_1, \triangle_2) = \triangle_{\eta(F_{\triangle_1}, F_{\triangle_2})} \quad . \tag{5}$$

Si $R \subseteq \mathcal{DT} \times \mathcal{DT}$ es una relación (nítida), entonces $\widehat{R} \subseteq \mathcal{TRI} \times \mathcal{TRI}$ es la nueva relación:

$$\triangle_1 \widehat{R} \triangle_2 \Longleftrightarrow F_{\triangle_1} R F_{\triangle_2} \quad . \tag{6}$$

El siguiente resultado se utilizará para introducir un orden parcial entre triángulos de pesos.

3.1 Proposición. *(i) Los conjuntos parcialmente ordenados $(\mathcal{DT}, \leq)$ y $(\mathcal{DTR}, \leq)$ determinados por las relaciones borrosas (2), (3), con la restricción del orden usual $\leq$ en $[0,1]^{\mathbb{N}\times\mathbb{N}}$ son subretículos del retículo $([0,1]^{\mathbb{N}\times\mathbb{N}}, \leq, \wedge, \vee)$, con las operaciones $\wedge$ (infimo) y $\vee$ (supremo) definidas puntualmente.*

(ii) Los elementos mínimo y máximo de $(\mathcal{DT}, \leq, \wedge, \vee)$ y $(\mathcal{DTR}, \leq, \wedge, \vee)$ son las relaciones borrosas $F_{(0)}$ y $F_{(1)}$ especificadas en (1) del ejemplo (a') de este trabajo.

(iii) Si $\mathcal{F}$ representa el subconjunto: $\mathcal{F} = \{F_{(\alpha)}/\alpha \in [0,1]\} \subset \mathcal{DT}$ de las relaciones de distribución consideradas en (a'), entonces la cadena $(\mathcal{F}, \leq)$ incluida en $(\mathcal{DTR}, \leq)$ es isomorfa a la cadena $([0,1], \leq)$.

En consecuencia, según (6) y (5), se obtienen los retículos de triángulos de pesos $(\mathcal{TRI}, \preccurlyeq, \curlywedge, \curlyvee)$ y $(\mathcal{TRIR}, \preccurlyeq, \curlywedge, \curlyvee)$ en los que el orden $\preccurlyeq$ y las operaciones $\curlywedge$ (infimo) y $\curlyvee$ (supremo) son:

3.2 Proposición. *(i) La relación $\preccurlyeq$ definida en $\mathcal{TRI}$, (y su restricción a $\mathcal{TRIR}$), por*

$$\triangle_1 \preccurlyeq \triangle_2 \Longleftrightarrow F_{\triangle_1} \leq F_{\triangle_2}, \tag{7}$$

es de orden y $(\mathcal{TRI}, \preccurlyeq, \curlywedge, \curlyvee)$, $(\mathcal{TRIR}, \preccurlyeq, \curlywedge, \curlyvee)$ son retículos con elementos mínimo $\triangle_{(0)}$ y elemento máximo $\triangle_{(1)}$ en los que las operaciones inf ($\curlywedge$) y sup ($\curlyvee$) verifican:

$$\triangle_1 \curlywedge \triangle_2 = \triangle_{F_{\triangle_1} \wedge F_{\triangle_2}}, \tag{8}$$

$$\triangle_1 \curlyvee \triangle_2 = \triangle_{F_{\triangle_1} \vee F_{\triangle_2}}. \tag{9}$$

(ii) Se puede considerar la cadena $([0,1], \leq)$ sumergida en $(\mathcal{TRIR}, \preccurlyeq)$ identificando $\alpha \in [0,1]$ con el triángulo $\triangle_{(\alpha)}$.

3.3 Nota. La expresión de la relación de orden $\preccurlyeq$ (7) en función de los pesos de los triángulos es la siguiente:

si $\triangle_1 = (1, (w_1^2, w_2^2), ..., (w_1^n, w_2^n, ...w_i^n, ..., w_n^n), ...)$ y $\triangle_2 = (1, (u_1^2, u_2^2), ..., (u_1^n, u_2^n, ...u_j^n, ..., u_n^n), ...)$, teniendo en cuenta la definición 2.1:

$$\triangle_1 \preccurlyeq \triangle_2 \Longleftrightarrow (\sum_{i=1}^{m} w_i^{n+m-1} \leq \sum_{i=1}^{m} u_i^{n+m-1} \; \forall (n,m) \in \mathbb{N}^2).$$

3.4 Ejemplo. Utilizando (8), (9), la definición 2.1, la proposición 2.6 y los triángulos $\triangle_{median}$ y $\triangle_{med}$ incluidos en el apartado 1 de este trabajo, obtenemos los triángulos $\triangle_{\curlywedge} = \triangle_{median} \curlywedge \triangle_{med}$ y $\triangle_{\curlyvee} = \triangle_{median} \curlyvee \triangle_{med}$, algunos de cuyos valores aparecen en las tablas no limitadas siguientes :

$$\triangle_{\curlywedge} = \begin{array}{cccccc} & & 1 & & & \\ & \frac{1}{2} & & \frac{1}{2} & & \\ 0 & & \frac{2}{3} & & \frac{1}{3} & \\ 0 & & \frac{1}{2} & & \frac{1}{4} & \frac{1}{4} \\ & & \dots & & & \end{array} \quad , \qquad \triangle_{\curlyvee} = \begin{array}{cccccc} & & 1 & & & \\ & \frac{1}{2} & & \frac{1}{2} & & \\ \frac{1}{3} & & \frac{2}{3} & & 0 & \\ \frac{1}{4} & & \frac{1}{4} & & \frac{1}{2} & 0 \\ & & \dots & & & \end{array} \quad .$$

En los retículos $(\mathcal{TRI}, \preccurlyeq, \curlywedge, \curlyvee)$ y $(\mathcal{TRIR}, \preccurlyeq, \curlywedge, \curlyvee)$ se puede introducir negaciones utilizando las extensiones de otras en $([0,1], \leq)$ a $(\mathcal{DT}, \leq)$ y $(\mathcal{DTR}, \leq)$ como se expone en los siguientes resultados:

3.5 Proposición. *(i) Si $' : [0,1] \to [0,1]$ es una negación y $F \in \mathcal{DT}$, entonces la función $F^* \in [0,1]^{\mathbb{N}\times\mathbb{N}}$ definida por:*

$$F^*(n,m) = \begin{cases} 1 & si \quad n = 1 \\ (F(m+1, n-1))' & si \quad n > 1 \end{cases} \tag{10}$$

pertecece también a $\mathcal{DT}$.

(ii) Si $F \in \mathcal{DTR}$ entonces $F^* \in \mathcal{DTR}$.

3.6 Proposición. *(i) Dada una negación $'$ en $[0,1]$, la aplicación $^* : \mathcal{DT} \to \mathcal{DT}$ tal que hace correspoder a F la función F^* definida en (10) y su restricción a $\mathcal{DTR}$ son también negaciones en los retículos $(\mathcal{DT}, \leq)$ y $(\mathcal{DTR}, \leq)$.*

(ii) Si la aplicación $'$ es una negación fuerte en $[0,1]$, su negación F^* también lo es en $\mathcal{DT}$.

(iii) Para toda negación $'$, se verifica: $(F_{(\alpha)})^* = F_{(\alpha')}$. En particular, $(F_{(0)})^* = F_{(1)}$ y $(F_{(1)})^* = F_{(0)}$.

3.7 Ejemplo. Si $'$ es la negación de Zadeh, entonces, según (10), la negación asociada * viene dada por

$$F^*(n,m) = \begin{cases} 1 & si \quad n = 1 \\ 1 - F(m+1, n-1) & si \quad n > 1 \end{cases}, \tag{11}$$

y en este caso de negación de Zadeh, las relaciones de los ejemplos *(b')* y *(c')* siguientes resultan ser puntos fijos: $F^*_{P(\frac{1}{2})} = F_{P(\frac{1}{2})}$, $F^*_{median} = F_{median}$ y $F^*_{med} = F_{med}$.

Estamos en condiciones de extender las negaciones a triángulos de pesos utilizando la proposición 3.6 y (4):

3.8 Proposición. *(i) Dada una negación $'$ en $[0,1]$, entonces la aplicación $^* : \mathcal{TRI} \to \mathcal{TRI}$ definida por:*

$$\triangle^* = \triangle_{F^*_\triangle} \tag{12}$$

es una negación. También lo es su restricción a $\mathcal{TRIR}$.

(ii) Si $'$ es una negación fuerte entonces la negación * definida en (12) también lo es.

3.9 Ejemplo. Si $\alpha' = \frac{1-\alpha}{1+2\alpha}$ obtenemos por (10):

$$F^*_{med}(n,m) = \frac{m}{3n+m-3} \quad \forall (n,m) \in \mathbb{N}^2,$$

y en consecuencia, por la proposición 2.5, la negación de $\triangle_{med}$ es el triángulo de pesos:

$\triangle^*_{med} = (1, (w^2_1, w^2_2), ..., (w^n_1, w^n_2, ...w^n_i, ...w^n_n), ...)$ con

$$w^n_i = \frac{3n}{(3n-2i)(3n-2i+2)} \quad \forall (n,i) : n \geq i.$$

El siguiente resultado muestra el carácter distinguido de la negación de Zadeh para triángulos de pesos.

3.10 Proposición. *Si $\alpha' = 1-\alpha$, entonces la negación de un triángulo de pesos coincide con su inverso:*

$$\triangle^* = \triangle^r.$$

Finalmente, podemos definir otros tipos de operaciones entre triángulos utilizando el siguiente resultado:

3.11 Proposición. *Sea L la cadena $[0,1]$ o el retículo de relaciones de distribución $\mathcal{DT}$. Sean 0 y 1 los elementos mínimo y máximo de L. Consideramos una ley de composición $\Psi : L^2 \to L$ y las siguientes propiedades que puede cumplir:*

(1) $\Psi(1,\alpha) = \alpha$.

(2) Ψ *es creciente para sus dos argumentos.*

(2') Ψ *es decreciente para el primer argumento y creciente para el segundo.*

(3) $\Psi(0,0) = 1$.

(4) Ψ *es conmutativa.*

(5) Ψ *es asociativa.*

Entonces

(i) Si $L = [0,1]$ y Ψ es una ley de composición interna en $[0,1]$ que verifica (1) y (2), entonces la extensión puntual $\widehat{\Psi}$ definida por

$$\overline{\Psi}(F,G)(n,m) = \Psi(F(n,m), G(n,m)) \quad \forall (n,m) \in \mathbb{N}^2, \tag{13}$$

$\forall (F,G) \in \mathcal{DT}^2$, es ley interna ahora en $L = \mathcal{DT}$ y satisface también (1) y (2). Además, si Ψ satisface (4) o (5), entonces $\overline{\Psi}$ también.

(ii) Si $L = [0,1]$ y Ψ es una ley de composición interna en $[0,1]$ que verifica (1) y (2') entonces la aplicación $\widetilde{\Psi}$ definida por

$$\widetilde{\Psi}(F,G)(n,m) = \begin{cases} 1 & n = 1 \\ \Psi(F(m+1,n-1), G(n,m)) & n > 1 \end{cases} \tag{14}$$

$\forall (F,G) \in \mathcal{DT}^2$, es ley interna en $L = \mathcal{DT}$ y satisface también (1) y (2'). Si Ψ satisface (3) entonces $\widetilde{\Psi}$ también satisface la propiedad (3).

(iii) Si F y G pertenecen a $\mathcal{DTR} \subset \mathcal{DT}$, entonces $\overline{\Psi}(F,G)$ y $\widetilde{\Psi}(F,G)$ pertenecen también a $\mathcal{DTR}$.

Las expresiones (13) y (14) junto a la definición (5), pueden utilizarse para extender operaciones Ψ del tipo t-norma, implicación, etc., en $[0,1]$ a operaciones $\widehat{\overline{\Psi}}$, $\widehat{\widetilde{\Psi}}$ en el conjunto $\mathcal{TRI}$ de triángulos de pesos:

$$\widehat{\overline{\Psi}}(\triangle_1, \triangle_2) = \triangle_{\overline{\Psi}(F_{\triangle_1}, F_{\triangle_2})} \text{ y}$$

$$\widehat{\widetilde{\Psi}}(\triangle_1, \triangle_2) = \triangle_{\widetilde{\Psi}(F_{\triangle_1}, F_{\triangle_2})}.$$

3.12 Ejemplos. *(e)* Si Ψ_1 es la t-norma producto $\Psi_1(\alpha,\beta) = \alpha \cdot \beta$ entonces la aplicación definida por (13) para las relaciones del ejemplo *(c')* es:

$$\overline{\Psi}_1(F_{median}, F_{med})(n,m) = \begin{cases} 0 & si \quad m \leq n-2 \\ \frac{1}{2} & si \quad m = n-1 \\ \frac{m}{n+m-1} & si \quad m \geq n \end{cases},$$

y por tanto $\widehat{\overline{\Psi}}_1(\triangle_{median}, \triangle_{med})$ es el triángulo con pesos cuya expresión general es, para $m \leq n$:

$$w^n_m = \begin{cases} 0 & si \quad n \geq 2m+1 \\ \frac{1}{2} & si \quad n = 2m \\ \frac{n+1}{2n} & si \quad n = 2m-1 \\ \frac{1}{n} & si \quad n \leq 2m-2 \end{cases},$$

es decir:

$$\widehat{\overline{\Psi}}_1(\triangle_{median}, \triangle_{med}) = \begin{array}{ccccccc} & & 1 & & & & \\ & 1/2 & & 1/2 & & & \\ 0 & & 2/3 & & 1/3 & & \\ 0 & & 1/2 & & 1/4 & & 1/4 \\ 0 & & 0 & & 3/5 & & 1/5 & & 1/5 \\ & & \cdots & & \cdots & & & & \end{array}.$$

(f) Si Ψ_2 es la implicación de Łuckasiewicz $\Psi_2(\alpha,\beta) = 1 \wedge (1-\alpha+\beta)$, entonces la aplicación definida por (14) para las relaciones del ejemplo *(c')* es:

$$\widetilde{\Psi}_2(F_{median}, F_{med})(n,m) = \begin{cases} 1 & si \quad n \leq m+1 \\ \frac{m}{n+m-1} & si \quad n \geq m+2 \end{cases},$$

en consecuencia, la implicación $\widehat{\widetilde{\Psi}}_2(\triangle_{median}, \triangle_{med})$ entre los triángulos de pesos del ejemplo *(c)* incluido en el apartado 1 de este trabajo, viene determinado por:

$$w_m^n = \begin{cases} 1 & m=1, \quad n \leq 2 \\ 0 & n \leq 2m-2, \quad 1 < m \leq n \\ \frac{n-m+1}{n} & 1 < n = 2m-1 \quad \acute{o} \quad n = 2m \\ \frac{1}{n} & n \geq 2m+1 \end{cases},$$

es decir:

$$\widehat{\widetilde{\Psi}}_2(\triangle_{median}, \triangle_{med}) = \begin{array}{ccccc} & & 1 & & \\ & 1 & 0 & & \\ 1/2 & 2/3 & 0 & & \\ 1/4 & 3/4 & 0 & 0 & \\ 1/5 & 1/5 & 3/5 & 0 & 0 \\ & \cdots & \cdots & & \end{array}.$$

4 EXPRESIÓN DEL ÍNDICE DE "ORNESS" DE $\triangle$ Y $\triangle^r$ EN FUNCIÓN DE $F_\triangle$

Dado un triángulo $\triangle \in \mathcal{TRI}$, si existe el límite:

$$\lim_{n\to+\infty} \frac{1}{n-1} \sum_{i=1}^{n} (n-i) w_i^n,$$

se llama *índice de "orness"* de $\triangle$ [1] y se representa por $orness(\triangle)$.

Los resultados siguientes proporcionan una nueva expresión para el cálculo de $orness(\triangle)$ y $orness(\triangle^r)$ de un triángulo $\triangle \in \mathcal{TRI}$ y de su inverso $\triangle^r$.

4.1 Proposición. *Dado un triángulo de pesos $\triangle$, si existe $orness(\triangle)$, entonces:*

$$orness(\triangle) = \lim_{n\to+\infty} \frac{1}{n-1} \sum_{j=1}^{n-1} F_\triangle(n-j+1, j). \quad (15)$$

4.2 Proposición. *Si $F \in \mathcal{DT}$ y F^* es su negación de Zadeh dada por la expresión* (11) *en el ejemplo 3.7 y si existe el límite* (15)*, se verifica:*

$$\lim_{n\to+\infty} \frac{1}{n-1} \sum_{j=1}^{n-1} F(n-j+1, j)$$
$$= 1 - \lim_{n\to+\infty} \frac{1}{n-1} \sum_{j=1}^{n-1} F^*(n-j+1, j).$$

4.3 Corolario. *Si existe $orness(\triangle)$ entonces:*

$$orness(\triangle^r) = 1 - orness(\triangle).$$

4.4 Ejemplos. *(g)* Como hemos visto en *(d')*, el triángulos de pesos $\triangle_d$ del ejemplo *(d)*, tiene por relación de distribución

$$F_d(n,m) = \begin{cases} 1 & si \quad n=1 \\ \frac{m-1}{3(n+m-1)} & si \quad n>1 \end{cases} \forall (n,m) \in \mathbb{N}^2,$$

luego:
$$orness(\triangle_d) = \lim_{n\to+\infty} \frac{1}{n-1} \sum_{j=1}^{n-1} \frac{j-1}{3n}$$
$$= \lim_{n\to+\infty} \frac{1}{(n-1)3n} \frac{(n-2)(n-1)}{2} = \frac{1}{6}$$

y el índice de su inverso es: $orness(\triangle_d^r) = 1 - \frac{1}{6} = \frac{5}{6}$.

(h) Para la representación $\triangle_{(\alpha)}$ de $\alpha \in [0,1]$:

$$orness(\triangle_{(\alpha)}) = \lim_{n\to+\infty} \frac{1}{n-1} \sum_{j=1}^{n-1} F_{(\alpha)}(n-j+1, j)$$
$$= \lim_{n\to+\infty} \frac{1}{n-1} (n-1)\alpha = \alpha.$$

Referencias

[1] T. Calvo, G. Mayor, J. Suñer, M. Mas and C. Carbonell, Generation of weighting triangles associated with aggregation functions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems Vol. 8 No. 4 (2000) 417-451

[2] T. Calvo, M. Carbonell y P. Canet, Medias Casiliniales Ponderadas Extendidas. Proc. Estylf'97. Tarragona 1997 33-38

[3] T. Calvo and G. Mayor, Remarks on two types of Extended Aggregation Functions. Tatra Mount. Math. Publ. 16-1 (1999) 103-115

[4] M. Carbonell, M. Mas and G. Mayor, On a class of Monotonic multidimensional OWA Operators. Proc. Sixth IEEE Int. Conference on Fuzzy Systems (Barcelona) 1997, pp. 1695-1700

[5] M. Mas, G. Mayor, J. Suñer and J. Torrens, Generación de funciones de Agregación Extendidas. Proc. Estylf'97. Tarragona 1997 39-44

[6] G. Mayor and T. Calvo, On Extended Aggregation Functions. Proc. Seventh IFSA Congress (Prague), Vol. I 1997 pp 281-285

On Rational Weak Nilpotent Minimum Logics

Francesc Esteva
IIIA-CSIC
08193 Bellaterra, Spain
esteva@iiia.csic.es

Lluís Godo
IIIA-CSIC
08193 Bellaterra, Spain
godo@iiia.csic.es

Carles Noguera
IIIA-CSIC
08193 Bellaterra, Spain
cnoguera@iiia.csic.es

Abstract

In this paper we investigate some extensions of Weak Nilpotent Minimum logic, a weaker logic than both Gödel and Nilpotent Minimum logics, by adding rational truth-values as truth constants in the language and by adding corresponding book-keeping axioms for the truth-constants. Weak and strong standard completeness of these logics are studied in general and when we restrict ourselves to formulas of the kind $\overline{r} \to \varphi$, where r is a rational in [0, 1] and φ is a formula without rational truth-constants.

Keywords: Gödel Logic, Weak Nilpotent Minimum, Rational Pavelka Logic, Rational Gödel logic, Rational Weak Nilpotent Minimum logic.

1 Introduction

As Pavelka pointed out in [12], it seems natural to introduce truth values in the language in order to be able to deal with *partial truth*. With this aim, he built a many-valued logical system over Lukasiewicz logic by adding into the language as many truth constants as truth values, i.e. a truth constant $\overline{r}$ for each real $r \in [0, 1]$, and a number of additional axioms. Although this Lukasiewicz logic extended with truth-constants, PL, is not strong complete (like Lukasiewicz logic), Pavelka proved that it is complete in a weaker sense. Indeed, he introduced a weaker notion of strong completeness based on the degrees of provability and truth of a formula φ in an arbitrary theory T. The truth degree of φ in T is defined as

$$|| \varphi ||_T = \inf\{e(\varphi) \mid e \text{ evaluation model of } T\}$$

and the degree of provability of φ in T as

$$| \varphi |_T = \sup\{r \mid T \vdash_{PL} \overline{r} \to \varphi\}.$$

Pavelka proved that these degrees coincide. This kind of completeness, which strongly relies in the continuity of Lukasiewicz logic truth functions, is usually known as Pavelka-style completeness. Moreover he also proved that Pavelka-style completeness is preserved if and only if the language is extended with any connective whose corresponding truth-function on the real unit interval is a continuous (real) function.

Later, Hájek [11] proved that Pavelka's logic PL could be significantly simplified while keeping the completeness results. Namely, Hájek's system is an extension of Lukasiewicz logic by only a countable number of truth-constants, $\overline{r}$ for each *rational* $r \in [0, 1]$, and by two additional axiom schemata to deal with the truth-constants, called book-keeping axioms:

$$\overline{r} \& \overline{s} \leftrightarrow \overline{r * s}$$
$$\overline{r} \to \overline{s} \leftrightarrow \overline{r \Rightarrow s}$$

where $*$ and $\Rightarrow$ are the t-norm of Lukasiewicz and its residuum respectively. He denoted this new system as RPL, for Rational Pavelka Logic, and proved the same results that Pavelka proved for his system with continuously many truth-constants. Moreover, in [11] it is proved that RPL is strong complete for finite theories. Remark that the semantics of RPL is kept on the *real* unit interval [0, 1].

Similar *rational* extensions for other popular fuzzy logics can be obviously defined, but Pavelka-style completeness cannot be obtained since Lukasiewicz is the only fuzzy logic with continuous truth-functions in the real unit interval [0, 1]. For instance, in [11] Hájek defines an extension of G_Δ, the extension of Gödel logic with Baaz's Delta operator, with a finite number of rational truth-constants. Later, in [4] the authors define logical systems obtained by adding (rational) truth-constants to $G_\sim$ (Gödel logic with an involutive negation) and to Π (Product logic) and $\Pi_\sim$ (Product logic with an involutive negation). For the first system, $RGL_\sim$, usual strong completeness is proved for finite theories, while for the second systems, $R\Pi L$ and

$R\Pi L_\sim$, it is possible to prove Pavelka-style completeness provided an infinitary inference rule is added to overcome the problem that the residuum of the product t-norm is not continuous at the point $(0, 0)$. Finally also notice that in [1] standard completeness of Gödel logic with rational truth-constants is stated. Although the result holds true (see Section 3), the proof given there is not correct.

Another different approach to reasoning with partial degrees of truth is the framework of abstract fuzzy logics developed by Gerla [8] based on the notion of fuzzy consequence or deduction operators over fuzzy sets of formulas, where the membership degree of formulas are interpreted as lower bounds on their truth degrees.

In this paper we extend preliminary results reported in [6] by investigating first some particular extensions of the Weak Nilpotent minimum logic WNM, and second their expansion with rational truth-constants, in the style of RPL. WNM was introduced in [3] as the axiomatic extension of MTL by the following axiom,

$$\text{(WNM)} \qquad (\varphi\&\psi \to 0) \vee (\varphi \wedge \psi \to \varphi\&\psi)$$

and proved to be standard complete with respect to the family of WNM t-norms and their residua. WNM t-norms are left-continuous t-norms defined from a weak negation function n and the minimum t-norm in the following way

$$x *_n y = \begin{cases} 0, & \text{if } x \leq n(y) \\ \min(x, y), & \text{otherwise} \end{cases}$$

Well-known particular cases of WNM t-norms are the minimum t-norm (when n is Gödel negation) and Fodor's nilpotent t-norm (when $n(x) = 1-x$). The paper is structured as follows. In next section we give a general formal account of algebraic semantics for the expansions of usual fuzzy logics with rational truth-constants. This is done by means of Blok and Pigozzi's theory of algebraization of propositional logics. In the first part of Section 3 we consider three different WNM logics and prove their standard completeness. The rest of the section is devoted to several standard completeness results for the rational expansions of these logics. We conclude with some final remarks.

2 Preliminaries

Our general logical framework for this section will be that of MTL and its axiomatic extensions. MTL logic was defined in [3] as a propositional logic in the language $\mathcal{L} = \{\&, \to, \wedge, \overline{0}\}$. In the frame of MTL extensions, other usual connectives are definable, in particular $\overline{1}$ is $\varphi \to \varphi$, $\neg\varphi$ is $\varphi \to \overline{0}$, $\varphi \vee \psi$ is $((\varphi \to \psi) \to \psi) \wedge ((\psi \to \varphi) \to \varphi)$, and $\varphi \leftrightarrow \psi$ is $(\varphi \to \psi) \wedge (\psi \to \varphi)$. We will let $\mathcal{RL} = \mathcal{L} \cup \{\overline{r} : r \in \mathbb{Q} \cap (0, 1]\}$ be the expansion of $\mathcal{L}$ with new constant symbols for every rational in $(0, 1]$.

In [3] an algebraic semantics for MTL was given, based on the notion of MTL-algebras, i.e. bounded residuated lattices satisfying the prelinearity equation: $(x \to y) \vee (y \to x) \approx 1$. Let $\mathbb{MTL}$ be the variety of all MTL-algebras.

Definition 1. *Given $\Gamma \cup \{\varphi\} \subseteq Fm_{\mathcal{L}}$, we define:*
(i) If $\mathcal{A} \in \mathbb{MTL}$, $\Gamma \vDash_{\mathcal{A}} \varphi$ iff, for all evaluations v in $\mathcal{A}$, we have $v(\varphi) = 1$ whenever $v(\psi) = 1$ for all $\psi \in \Gamma$.
(ii) $\Gamma \vDash \varphi$ iff for all $\mathcal{A} \in \mathbb{MTL}$ we have $\Gamma \vDash_{\mathcal{A}} \varphi$

Then, one can prove the next theorem of strong completeness for MTL logic.

Theorem 2. *If $\Gamma \cup \{\varphi\} \subseteq Fm_{\mathcal{L}}$, $\Gamma \vDash \varphi$ iff $\Gamma \vdash_{MTL} \varphi$.*

But this result can be improved by means of the equational consequence:

Definition 3. *Let $Eq_{\mathcal{L}}$ be the set of $\mathcal{L}$-equations and let $\Pi \cup \{\varphi \approx \psi\} \subseteq Eq_{\mathcal{L}}$. We define the equational consequence $\Pi \vDash_{\mathbb{MTL}} \varphi \approx \psi$ iff, for all $\mathcal{A} \in \mathbb{MTL}$ and for all evaluation v in $\mathcal{A}$, we have $v(\varphi) = v(\psi)$ whenever $v(\alpha) = v(\beta)$ for all $\alpha \approx \beta \in \Pi$.*

Theorem 4. *The relation of derivability in the system MTL and the equational consequence in the variety $\mathbb{MTL}$ are mutually translatable:*

1. *For every $\Gamma \cup \{\varphi\} \subseteq Fm_{\mathcal{L}}$, $\Gamma \vdash_{MTL} \varphi$ iff $\{\psi \approx 1 : \psi \in \Gamma\} \vDash_{\mathbb{MTL}} \varphi \approx 1$.*
2. *For every $\Pi \cup \{\varphi \approx \psi\} \subseteq Eq_{\mathcal{L}}$,*
   $\Pi \vDash_{\mathbb{MTL}} \varphi \approx \psi$ iff
   $\{\alpha \leftrightarrow \beta : \alpha \approx \beta \in \Pi\} \vdash_{MTL} \varphi \leftrightarrow \psi$

In addition, each one of these translations is the inverse of the other, that is:

3. *$\varphi \approx \psi \vDash_{\mathbb{MTL}} \varphi \leftrightarrow \psi \approx 1$, $\varphi \leftrightarrow \psi \approx 1 \vDash_{\mathbb{MTL}} \varphi \approx \psi$*
4. *$\varphi \vdash_{MTL} \varphi \leftrightarrow \overline{1}$, $\varphi \leftrightarrow \overline{1} \vdash_{MTL} \varphi$.*

Therefore, MTL is an algebraizable logic in the sense of Blok and Pigozzi (see [2]) whose equivalent algebraic semantics is the variety $\mathbb{MTL}$. Thus, using the general theory of [2], all axiomatic extensions of MTL are also algebraizable in this strong sense. Namely, if L denotes the extension of MTL by a given set of axiom schemata Σ, the equivalent algebraic semantics of L is the subvariety of $\mathbb{MTL}$ defined by the translation of the formulas in Σ into equations. We will refer to the algebras of this subvariety as L-algebras.

There is another useful kind of completeness result for MTL, completeness with respect to the totally ordered algebras (we will call them 'chains'):

Theorem 5. *[3] Each MTL-algebra is isomorphic to a subdirect product of MTL-chains.*

As a consequence, Theorem 2 remains valid if the logical consequence $\models$ in Definition 1 is restricted to evaluations over MTL-chains. This is also true for every axiomatic extension of MTL. Next we will consider the issue of algebraization for fuzzy logics with constant symbols for the rationals.

Definition 6. *Let L be MTL or any axiomatic extension of MTL and let $*$ be a left-continuous t-norm and $\Rightarrow$ its residuum such that $[0,1]_* = \langle [0,1], *, \Rightarrow, \min, \max, 0, 1\rangle$ is an L-algebra. By RL($*$) we will denote the propositional logic in the language $\mathcal{RL}$ obtained by adding to L the so-called 'book-keeping axioms':*

$$\overline{r}\&\overline{s} \leftrightarrow \overline{r * s}$$
$$\overline{r} \wedge \overline{s} \leftrightarrow \overline{\min(r,s)}$$
$$\overline{r} \rightarrow \overline{s} \leftrightarrow \overline{r \Rightarrow s}$$

for every $r, s \in \mathbb{Q} \cap [0,1]$.

RL($*$)-algebras are structures $\mathcal{A} = \langle A, \&, \rightarrow, \wedge, \vee, \{\overline{r}^{\mathcal{A}} : r \in \mathbb{Q} \cap [0,1]\}\rangle$ such that:
1. *$\langle A, \&, \rightarrow, \wedge, \vee, \overline{0}^{\mathcal{A}}, \overline{1}^{\mathcal{A}}\rangle$ is a L-algebra,*
2. *for every $r, s \in \mathbb{Q} \cap [0,1]$:*

$$\overline{r}^{\mathcal{A}} \& \overline{s}^{\mathcal{A}} = \overline{r * s}^{\mathcal{A}}$$
$$\overline{r}^{\mathcal{A}} \wedge \overline{s}^{\mathcal{A}} = \overline{\min(r,s)}^{\mathcal{A}}$$
$$\overline{r}^{\mathcal{A}} \rightarrow \overline{s}^{\mathcal{A}} = \overline{r \Rightarrow s}^{\mathcal{A}}$$

Given $\Gamma \cup \{\varphi\} \subseteq Fm_{\mathcal{RL}}$, we define $\Gamma \models_{\mathcal{A}} \varphi$ iff for all evaluations e in $\mathcal{A}$ (i.e. such that $e(\overline{r}) = \overline{r}^{\mathcal{A}}$), we have $e(\varphi) = \overline{1}^{\mathcal{A}}$ whenever $e(\psi) = \overline{1}^{\mathcal{A}}$ for all $\psi \in \Gamma$.

When $A = [0,1]$ and $\overline{r}^{\mathcal{A}} = r$ for all $r \in \mathbb{Q} \cap [0,1]$, we say that $\mathcal{A}$ is the standard RL($*$)-algebra[1].

Using [2], it is easy to prove that RL($*$) is an algebraizable logic whose equivalent algebraic semantics is the variety of RL($*$)-algebras. We obtain also completeness with respect to linearly ordered algebras.

Proposition 7. *The following conservativeness results hold:*

- *If L is strong standard complete for arbitrary theories w.r.t. $[0,1]_*$, then RL($*$) is a conservative extension of L.*
- *If L is strong standard complete for finite theories w.r.t. $[0,1]_*$, then RL($*$) is a conservative extension of L for finite theories.*
- *If L is weak standard complete w.r.t. $[0,1]_*$, then RL($*$) is a conservative extension of L for theorems.*

[1]Notice that in such a case & and $*$ necessarily coincide (hence $\rightarrow$ and $\Rightarrow$ as well), since they coincide on the rationals and are left-continuous t-norms.

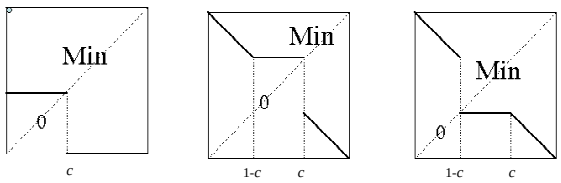


Figure 1: WNM t-norms $*_c$,$\star_c$ and $\circ_c$.

Proof: We shall prove the first claim as an example. Let $\Gamma \cup \{\varphi\} \subseteq Fm_{\mathcal{L}}$ be such that $\Gamma \vdash_{RL(*)} \varphi$. Since RL($*$)-algebras are an algebraic semantics, we have $\Gamma \models_{\mathcal{A}} \varphi$ where $\mathcal{A}$ is the standard L($*$)-algebra. Hence, using that these formulas are written in $\mathcal{L}$, $\Gamma \models_{[0,1]_*} \varphi$, so $\Gamma \vdash_L \varphi$. □

3 Main results

As already mentioned, WNM logic is standard complete with respect to the family of WNM t-norms, but it is still not known which is the structure of subvarieties of WNM. There is not a general result giving, for each WNM t-norm $*$, the axiomatic characterization of the logic WNM_* (the logic complete with respect to $*$ and its residuum). The lack of this axiomatics makes impossible to define in general its corresponding rational logic, $RWNM_*(*)$ using the notation introduced in the previous section (for simplicity we will denote it $RWNM_*$ hereafter). However there are two well known exceptions, Gödel logic (G) and Nilpotent Minimum logic (NM), both extensions of WNM logic, and standard complete with respect to the minimum t-norm and to the nilpotent minimum t-norm respectively. So in these particular cases we can indeed consider their rational expansions.

3.1 Characterization of three new subvarieties of WNM

We now introduce three classes of WNM t-norms. Let $*_c$ for any $c \in [0,1)$, and $\star_c$ and $\circ_c$ for any $c \in [\frac{1}{2}, 1)$ be the WNM t-norms defined respectively by the negation functions on [0, 1] n_{*_c}, $n_{\star_c}$ and $n_{\circ_c}$ given in Figure 1 and defined by,

$$n_{*_c}(x) = \begin{cases} 1, & \text{if } x = 0 \\ c, & \text{if } x \leq c \\ 0, & \text{otherwise} \end{cases}$$

$$n_{\star_c}(x) = \begin{cases} 1-x, & \text{if } x \in [0, 1-c] \cup (c, 1] \\ c, & \text{otherwise} \end{cases}$$

$$n_{\circ_c}(x) = \begin{cases} 1-x, & \text{if } x \in [0, 1-c] \cup [c, 1] \\ 1-c, & \text{otherwise} \end{cases}$$

Notice that for $c = 0$, $*_c = \min$, and for $c = 1/2$, we have $\star_c = \circ_c$ and coincide with the nilpotent minimum t-norm.

Next we define three corresponding axiomatic extensions of WNM logic. WNM_* is the axiomatic extension of WNM by adding the axiom

$$(\varphi \to \varphi^2) \to ((\varphi \wedge \psi) \to (\varphi \& \psi)),$$

$WNM_\star$ is the extension of WNM by adding the axiom

$$(\neg\neg\varphi \to \varphi) \vee (\neg\neg\varphi \leftrightarrow \neg\varphi),$$

and $WNM_\circ$ is the extension of WNM by the axioms:

$$(\neg\neg\varphi \to \varphi) \vee (\neg\neg\psi \to \psi) \vee ((\neg\varphi \leftrightarrow \neg\psi) \wedge (\neg\varphi \to \varphi)),$$
$$q(\varphi) \vee (q(\psi) \to (p(\varphi) \to p(\psi))),$$

where denotes $p(\chi)$ denotes $\chi \vee \neg\chi$ and $q(\chi)$ denotes $\neg\neg p(\chi) \to p(\chi)$.

Next we will prove that, for any $0 < c < 1$, WNM_* is the logic of the t-norm $*_c$, and for any $1/2 < c < 1$, $WNM_\star$ and $WNM_\circ$ are the logics of $\star_c$ and $\circ_c$ respectively.

Lemma 8. *Let $\mathcal{A}$ be a WNM-chain . Then:*

(i) $\mathcal{A}$ is a WNM_*-chain iff there exists an element $c \in A$ such that the negation in $\mathcal{A}$ is like n_{*_c}, i.e. $\neg x = c$ for $\overline{0}^{\mathcal{A}} < x \leq c$ and $\neg x = \overline{0}^{\mathcal{A}}$ for $x > c$.

(ii) $\mathcal{A}$ is a $WNM_\star$-chain iff it has a fixpoint c and it is such that the negation is involutive except for a segment ending in c in which $\neg$ is constantly c.

(iii) $\mathcal{A}$ is a $WNM_\circ$-chain iff there exists an element d such that $d \leq \neg d$ and that $\neg$ is involutive in the segments $[\overline{0}^{\mathcal{A}}, d]$ and in $[\neg d, \overline{1}^{\mathcal{A}}]$ and $\neg x = d$ for $x \in (d, \neg d]$.

Theorem 9. *If $+$ denotes $*$, $\star$ or $\circ$, the logic WNM_+ is complete with respect to each one of the following sets of chains:*
(1) the linearly ordered WNM_+-chains.
(2) the WNM_+-chains over [0,1].
(3) the WNM_+-chain over [0,1] defined by a t-norm $+_c$, where $0 < c < 1$ for $+$ being $*$ and $1/2 < c < 1$ for $+$ being $\star$ or $\circ$.

Proof. We sketch the proof for WNM_*, the proofs for $WNM_\star$ and $WNM_\circ$ are similar. The first result (1) is a particular case of the general result about all axiomatic extensions of MTL (see preliminaries). The proof of (2) is actually completely analogous with the obvious changes to the proof of standard completeness of WNM given in [3]. Finally, from the last lemma, the only WNM_*-chains on [0, 1] are those defined by the WNM t-norms $*_c$, for $c \in [0, 1)$. But it is easy to prove that all t-norms $*_c$ with $c > 0$ are isomorphic and that $*_0$ (= min) is isomorphic to the subalgebra of any of them defined on the subset $\{0\} \cup (c, 1]$. Thus any WNM_*-chain over [0,1] defined by a t-norm $*_c$ with $c > 0$ generates the whole variety of WNM_*-algebras. Hence (3) is proved. □

3.2 Weak Standard completeness of logics $RWNM_*$, $RWNM_\star$ and $RWNM_\circ$

Now we will consider rational expansions of the logics WNM_*, $WNM_\star$ and $WNM_\circ$. To do so, we add to the language as many truth-constants as rationals in [0,1], and we add to the logics the book-keeping axioms corresponding to any t-norm of their families. Let $RWNM_{*_c}$, $RWNM_{\star_c}$ and $RWNM_{\circ_c}$ be these logics respectively, where c is any suitable parameter defining the t-norms $*_c$, $\star_c$ or $\circ_c$.

The following lemma describes how rational constants can appear in the linearly ordered algebras of these logics.

Lemma 10. *Let $+$ denote $*$, $\star$ or $\circ$, and let c any suitable parameter defining the t-norms $*_c$, $\star_c$ or $\circ_c$. For any $RWNM_+(+_c)$-chain $\mathcal{A} = \langle A, \&, \to, \wedge, \vee, \{\overline{r}^{\mathcal{A}} : r \in \mathbb{Q} \cap [0,1]\}\rangle$ the following conditions hold:*

(i) The set $R_1 = \{r \in \mathbb{Q} \mid \overline{r}^{\mathcal{A}} = \overline{1}^{\mathcal{A}}\}$ is a right-closed interval with 1 as upper bound.

(ii) The set $R_0 = \{r \in \mathbb{Q} \mid \overline{r}^{\mathcal{A}} = \overline{0}^{\mathcal{A}}\}$ coincides with $n_{+_c}(R_1)$.

(iii) $\overline{r}^{\mathcal{A}} < \overline{s}^{\mathcal{A}}$ for any rationals $r < s$ not belonging to $R_1 \cup R_0$.

Next result establishes the (weak) standard completeness of the logics with respect any WNM t-norm of their corresponding families.

Theorem 11. *If $+$ denotes $*$, $\star$ or $\circ$, and c any suitable parameter defining the t-norms $*_c$, $\star_c$ or $\circ_c$, then $\vdash_{RWNM_{+_c}} \varphi$ if, and only if, $\vDash_{[0,1]_{+_c}} \varphi$.*

Proof: We will prove it for $RWNM_{*_c}$ as an example, the other cases are provable in a similar way. Without loss of generality and for simplicity we will assume $c = 1/2$ but we will omit its reference in the notations. The soundness part is trivial as usual. To prove completeness, suppose $\nvdash_{RWNM_*} \varphi$, then by completeness of $RWNM_*$ w.r.t. the class of $RWNM_*$-chains, there exists such a countable chain $\mathcal{C}$ and an evaluation e over $\mathcal{C}$ such that $e(\varphi) < \overline{1}^{\mathcal{C}}$. We have to show that there is an evaluation e' on an standard algebra $[0,1]_{*_{1/2}}$ such that $e'(\varphi) < 1$. Let $X = \{e(\psi), n_C(e(\psi)), n_C^2(e(\psi)) \mid \psi$ subformula of $\varphi\} \cup \{0^{\mathcal{C}}, 1^{\mathcal{C}}\}$. Consider first the case that $\alpha = \inf\{r \mid e(\overline{r}) = \overline{1}^{\mathcal{C}}\} > c$. Now we can take an order-preserving

injection $g : \langle X, \leq_C \rangle \hookrightarrow \langle [0,1], \leq \rangle$ which is strictly increasing and such that $g(\overline{r}^{\mathcal{C}}) = r$ for all rational $\overline{r}^{\mathcal{C}} \in X$. Then we define and $RWNM_*$-evaluation e' on the standard $RWNM_*$-algebra $[0,1]$ as follows: for all propositional variable p, $e'(p) = g(e(p))$ if p appears in φ and $e'(p) = 1$ otherwise (and, of course, with $e'(\overline{r}) = r$), i.e., $e'(x) = g(e(x))$ for all x being a propositional variable or a truth-constant belonging to X. Then e' is extended to $RWNM_*$-formulas as usual. We want to show that $e'(\varphi) < 1$. Actually, one can prove, by induction on the complexity, that for any formula ψ the following statement holds: "if $e(\psi) = \overline{1}^{\mathcal{C}}$ then $e'(\psi) \geq \alpha$, and if $e(\psi) = e(x) < \overline{1}^{\mathcal{C}}$ for some atom (propositional variable or truth-constant) x in ψ then $e'(\psi) = e'(x) < \alpha$". We omit details here. In particular, then from $e(\varphi) < \overline{1}^{\mathcal{C}}$ we can derive $e'(\varphi) < 1$, and the theorem is proved. Consider now the case $\alpha = c$, i.e. $\overline{r}^{\mathcal{C}} = \overline{1}^{\mathcal{C}}$ for all $r \in \mathbb{Q} \cap (c,1]$. In such a case we can build a new $RWNM_*$-chain $\mathcal{C}_1$ and an evaluation e_1 over $\mathcal{C}_1$ such that $e_1(\varphi) < \overline{1}^{\mathcal{C}_1}$ and $\inf\{r \mid e(\overline{r}) = \overline{1}^{\mathcal{C}_1}\} > 0$ and then we will be able to apply the proof of the previous case. To build this chain let $s = \min\{r \in \mathbb{Q} \cap (c,1] \mid \overline{r} \text{ appears in } \varphi\}$. Without loss of generality we can assume $C \cap \mathbb{Q} = \emptyset$. Then take $C_1 = C \cup ([0,s) \cap \mathbb{Q})$ and let $\mathcal{C}_1$ be the $RWNM_*$-chain defined over C_1 by the meet operation:

$$x \wedge_{\mathcal{C}_1} y = \begin{cases} x \wedge_{\mathcal{C}} y, & \text{if } x, y \in C \\ x \wedge_{\mathbb{Q}} y, & \text{if } x, y \in \mathbb{Q} \\ x, & \text{if } x \in \mathbb{Q}, y \in C \\ y, & \text{if } x \in C, y \in \mathbb{Q} \end{cases}$$

and define the constants $\overline{r}^{\mathcal{C}_1}$ as follows:

$$\overline{r}^{\mathcal{C}_1} = \begin{cases} \overline{1}^{\mathcal{C}}, & \text{if } r \in [s,1] \\ r, & \text{if } r \in [0,s) \end{cases}$$

Now define e_1 as the $RWNM_*$-evaluation over $\mathcal{C}_1$ such that $e_1(x) = e(x)$ for each propositional variable x. Obviously e_1 is an $RWNM_*$-evaluation over the countable $RWNM_*$-chain $\mathcal{C}_1$ where $\inf\{r \mid e(\overline{r}) = \overline{1}^{\mathcal{C}_1}\} = s > c$ such that $e_1(\varphi) < \overline{1}^{\mathcal{C}_1}$ as required. □

3.3 On finite strong completeness for $RWNM_*$, $RWNM_\star$ and $RWNM_\circ$

For all the WNM logics studied in this section, the following form of the deduction theorem is valid: $\varphi \vdash \psi$ iff $\vdash (\varphi \& \varphi) \to \psi$. From this it is easy to see that these loics are neither strong standard complete nor Pavelka complete.

Example 1. Let $T = \{\overline{r} \vee p\}$, where $0 < r < 1$ and p is a propositional variable. It is clear that T is satisfiable for any evaluation e such that $e(p) = 1$, and that $T \models_{[0,1]_{+_c}} p$. But $T \not\vdash p$ in any of these logics, since if so, by the deduction and weak completeness theorems for RG, it should also be true that $\models_{[0,1]_{+_c}} (\overline{r} \vee p) \to p$, which is false (take e such that $e(p) = 0$ and $r > c$). For Pavelka-style completeness notice that similar arguments lead to $|| p ||_T = 1$ but $| p |_T \leq n_{+_c}(r)$. □

Nevertheless, as we will prove, these logics are strong standard complete if we restrict ourselves to formulas of type $\overline{r} \to \varphi$ where φ is a formula of the corresponding WNM logic (without rational truth constants). This kind of formulas will be also denoted as (φ, r). We want to prove the following equivalence (where + denotes $*$, $\star$ or $\circ$):

$$\{(\psi_i, r_i) \mid i = 1,2,..,n\} \vdash_{RWNM_{+_c}} (\varphi, s)$$
if, and only if,
$$\{(\psi_i, r_i) \mid i = 1,2,..,n\} \models_{[0,1]_{+_c}} (\varphi, s)$$

To prove this result we need some previous results and lemmas. Actually, due to the fact that RWNM enjoys the (syntactical) deduction theorem and to the weak standard completeness, one can easily notice that proving the above restricted (finite) strong standard completeness for any RWNM logic amounts to prove the following semantical version of the deduction theorem:

$$\{(\psi_i, r_i) \mid i = 1,2,..,n\} \models_{[0,1]_{+_c}} (\varphi, s)$$
if, and only if,
$$\models_{[0,1]_{+_c}} (\bigwedge_{i=1,2,..,n} (\psi_i, r_i)) \to (\varphi, s).$$

Lemma 12. *Let $a \in (c,1]$ and define a mapping $f^a : [0,1] \to [0,1]$ as follows:*

$$f^a(x) = \begin{cases} 1, & \text{if } x \geq a \\ 0, & \text{if } x \leq n_{+_c}(a) \\ x, & \text{otherwise} \end{cases}$$

Then f^a is a morphism with respect to the operations of the algebra $[0,1]_{+_c}$. Therefore, if e is a WNM_{+_c}-evaluation of formulas, then $e^a = f^a \circ e$ is another WNM_{+_c}-evaluation.

Theorem 13.
Let $r_1, \ldots, r_n \in (c,1]$ and $s \in [0,1]$. Then:
$\{(\varphi_1, r_1), \ldots, (\varphi_n, r_n)\} \models_{[0,1]_{+_c}} (\psi, s)$ *iff* $\models_{[0,1]_{+_c}} (\&_{i=1}^n (\varphi_i, r_i))^2 \to (\psi, s)$.

Proof: One direction is easy. As for the difficult one, it is enough to prove that if there is an evaluation e which is not a model of $(\&_{i=1}^n (\varphi_i, r_i))^2 \to (\psi, s)$, then we can find another evaluation e' which is model of $\{(\varphi_1, r_1), \ldots, (\varphi_n, r_n)\}$ and not of (ψ, s).

Let e be an evaluation such that $e((\&_{i=1}^n (\varphi_i, r_i))^2 \to (\psi, s)) < 1$, i.e. $e((\&_{i=1}^n (\varphi_i, r_i))^2) > e((\psi, s))$. This means that:
(i) $e((\&_{i=1}^n (\varphi_i, r_i))^2) > 0$, hence this is also valid for each i and thus $e((\varphi_i, r_i)^2) = e((\varphi_i, r_i)) > c$ and

$e((\&_{i=1}^n(\varphi_i, r_i))^2) = \min_{i=1}^n e((\varphi_i, r_i)) > c$;
(ii) $e((\psi, s)) < 1$, hence $s > e(\psi)$ and $e((\psi, s)) = \max(1 - s, e(\psi))$.
Therefore we are assuming an evaluation e such that $\min_{i=1}^n e((\varphi_i, r_i)) > \max(1 - s, e(\psi))$, with $s > e(\psi)$.

If e is a model of every (φ_i, r_i) for $i = 1, ..., n$, then we can take $e' = e$ and the problem is solved. Otherwise, there exists some $1 \leq j \leq n$ for which $r_j > e(\varphi_j)$ and thus $e((\varphi_j, r_j)) = \max(1 - r_j, e(\varphi_j)) = e(\varphi_j) < 1$, the last equality due to the fact that we are assuming $\max(1 - r_j, e(\varphi_j)) > c$ and $r_j > c$.

Let $J = \{j \mid r_j > e(\varphi_j)\}$ and let $a = e((\&_{i=1}^n(\varphi_i, r_i))^2) = \min\{e(\varphi_j) \mid j \in J\}$. Then the $RWNM_+$-evaluation e' such that $e' = e^a$ over the propositional variables does the job. Namely, by the corresponding transformation of Lemma 12, over $RWNM_+$-formulas, e' is a model of (φ_i, r_i)'s for all i.

On the other hand, since $s > e(\psi)$ and $e(\psi) < a$, it turns out, due to the corresponding translation of Lemma 12, that $e'(\psi) = e^a(\psi) \leq e(\psi)$ and therefore $s > e'(\psi)$ as well. So, $e'((\psi, s)) < 1$ and the proof is completed. □

After this last theorem, the announced result of finite strong completeness of $RWNM_*$, $RWNM_\star$ and $RWNM_\circ$ when restricted to formulas of the kind (φ, r) comes as an easy consequence. Next corollary sumarize these three results (being + any one of the three symbols $*$, $\star$ or $\circ$) with the restriction for the values r_i as in the corresponding theorem.

Corollary 14. *$\{(\psi_i, r_i) \mid i = 1, 2, .., n\} \vdash_{RWNM_{+_c}} (\varphi, s)$ if, and only if, $\{(\psi_i, r_i) \mid i = 1, 2, .., n\} \models_{[0,1]_{+_c}} (\varphi, s)$.*

4 Conclusions and further work

Book-keeping axioms can be defined only when we have chosen a t-norm and its residuum. Thus the process to create a new logic by adding truth constants has only sense if the initial logic is "the" logic of a left continuous t-norm and its residuum. RPL, RG (rational Gödel), RNM (rational Nilpotent Minimum) and the examples of RWNM studied in the paper, have sense since Ł, G and NM and the considered axiomatic extensions of WNM are "the" logics of Lukasiewicz t-norm, of minimum t-norm, nilpotent minimum t-norm and of the t-norms $*_c$, $\star_c$ and $\circ_c$ (and their residua) respectively. For example, RBL has no sense since Hájek's BL, is the logic of a family of t-norms (the continuous ones) and thus it is not determined what t-norm can be used in the book-keeping axioms.

In [5], for each continuous t-norm $*$, the extension of BL, noted BL($*$), which is standard complete with respect to the BL-algebra in $[0, 1]$ defined by the continuous t-norm $*$ and its residuum, is defined. Moreover an algorithm to obtain a finite axiomatization of them is also given. This group of logics are suitable to be extended with rational truth-constants by adding the corresponding book-keeping axioms. In this way an interesting matter of future research could be the definition and study the rational extensions of the logics BL($*$).

Acknowledgments

The authors acknowledge partial support of the Spanish project LOGFAC, TIC2001-1577-C03-01.

References

[1] B. De Baets, F. Esteva, J. Fodor and L. Godo. Systems of Ordinal Fuzzy Logic with application to Preference Modelling. *Fuzzy Sets and Systems* 124(3), 2001, pp. 353-360.

[2] W. J. Blok and D. Pigozzi. Algebraizable logics. Mem. Amer. Math. Soc. 396, vol 77, 1989.

[3] F. Esteva and L. Godo. Monoidal t-norm based Logic: Towards a logic for left-continuous t-norms. *Fuzzy Sets and Systems* 124 (2001) 271–288.

[4] F. Esteva, L. Godo, P. Hájek, M. Navara. Residuated fuzzy logic with an involutive negation. *Archive for Mathematical Logic* 39 (2000) 103-124.

[5] F. Esteva, L. Godo and F. Montagna. Axiomatization of any residuated fuzzy logic defined by a continuous t-norm. In: *Fuzzy Sets and Systems - IFSA 2003*, LNAI 2175, Springer, pp. 172-179, 2003.

[6] F. Esteva, L. Godo, C. Noguera. On rational Gödel and Nilpotent Minimum logics. To appear in *Proc. of IPMU'04*, Perugia (Italy), 2004.

[7] J. Fodor. Nilpotent minimum and related connectives for fuzzy logic. Proc. of FUZZ–IEEE'95, pp. 2077–2082, 1995.

[8] G. Gerla. *Fuzzy Logic: Mathematical Tools for Approximate Reasoning*, Kluwer, 2001.

[9] S. Gottwald. *A Treatise on Many-valued Logics*, Studies in Logic and Computation 9, Research Studies Press Ltd., Baldock, UK, 2001.

[10] S. Jenei On the structure of rotation-invariant semigroups. *Archive for Mathematical Logic*, 42 (2003), 489-514.

[11] P. Hájek. *Metamathematics of Fuzzy Logic*, Kluwer, 1998.

[12] J. Pavelka. On Fuzzy Logic I, II, III. *Z. Math. Logic Grunlag. Math* 25 (1979) 45-52, 119-134, 447-464.

[13] E. Trillas. Sobre funciones de negación en la teoría de conjuntos difusos. *Stochastica*, Vol.I, (1979), 47-60.

[14] San-Min Wang, Bao-Shu Wang and Fang Ren. NMŁ, a schematic extension of F.Esteva and L.Godo's logic MTL. *Fuzzy Sets and Systems*, in press.

CONJUNTOS DIFUSOS CON ONDAS ASOCIADAS

C.J. Mantas, J. Portilla, J.M. Sánchez Álvarez
Dpto de Ciencias de la Computación e I.A.
Universidad de Granada, 18071
Granada, Spain
{cmantas, javier, jmsa}@decsai.ugr.es

Resumen

En este trabajo presentamos una extensión del concepto de conjunto difuso. Consiste en asociar un valor de onda a la pertenencia de cada elemento de un conjunto difuso.

Esta contribución no afecta a las propiedades, conceptos y operaciones definidas previamente a partir de la función de pertenencia estándar de un conjunto difuso. Sin embargo, sí aporta a los conjuntos difusos nuevas características adicionales provenientes de las propiedades de las ondas.

De estas nuevas características adicionales, se ha analizado la suma de conjuntos difusos con ondas asociadas de la misma frecuencia y diferente fase. Este análisis nos ha llevado a una nueva forma de agregar conclusiones difusas adecuada para ciertas situaciones prácticas.

Palabras Clave: Lógica Difusa, Algoritmos Difusos, Cuántica, Ondas.

1 INTRODUCCIÓN

La lógica difusa y su concepto asociado de conjunto difuso [2,5] se han convertido en herramientas indispensables para el tratamiento de información imprecisa. El razonamiento con este tipo de información ha tenido una enorme aplicación en el campo de control, dando lugar a lo que se ha denominado controladores difusos [6,7, 8].

Sin embargo, la dinámica de los controladores difusos consiste en un solo paso de razonamiento con información imprecisa modelada mediante conjuntos difusos. Cuando hay que realizar varios pasos de razonamiento difuso [9, 11,13] la imprecisión sobre la información que estamos tratando crece por encima de los límites de lo considerado aceptable.

Esta limitación del razonamiento difuso parece insalvable con las herramientas matemáticas de las que dispone actualmente la lógica difusa. Este hecho nos ha animado a buscar la ampliación del concepto de conjunto difuso. Para llevar a cabo esta tarea hemos buscado conceptos parecidos en otros campos de la ciencia, en concreto, en el área de la física cuántica [1,4] hemos encontrado un concepto interesante en el manejo de la imprecisión sobre la localización espacial y el momento de una "partícula" subatómica (por ejemplo, un electrón).

En física cuántica la información sobre la posición y el momento está contenida en una función que posee unas propiedades particulares. Éstas permiten estudiar a un electrón como si fuera una onda o como si fuera una partícula, según interese en cada momento.

Llevando esta idea al campo de la lógica difusa, hemos definido el concepto de conjunto difuso con una onda asociada. Manejando la información imprecisa con este nuevo concepto podemos, por un lado, tener todas las características que disponíamos con los conjuntos difusos estándar y, por otro lado, adquirir nuevas propiedades procedentes de las ondas. Como ejemplo de estas nuevas características presentamos un nuevo operador sobre conjuntos difusos: Suma de conjuntos difusos con una onda asociada. Este operador tiene la peculiaridad de poseer distintos comportamientos en función del tipo de las ondas asociadas a los conjuntos difusos.

2 CONJUNTO DIFUSO CON UNA ONDA ASOCIADA

Habitualmente un conjunto difuso A se define mediante el conjunto de tuplas que relaciona a cada elemento x_0 de un universo de discurso X con su grado de pertenencia $\mu_A(x_0)$. De esta manera, se ha llegado al acuerdo de identificar a un conjunto difuso A con su función de pertenencia μ_A.

En este apartado extenderemos la definición del conjunto difuso A asociándole una onda (obteniendo $A^{\wr}$). Así, para cada x_0 no sólo tendremos su valor de pertenencia, sino que también tendremos una dirección y un sentido de este

grado de pertenencia. De esta manera, ya no será equivalente el concepto de conjunto difuso A^ζ y el de su función de pertenencia $\mu_{A\zeta}(x)$.

Veamos más formalmente la definición de conjunto difuso con una onda asociada.

Definición 1

Sea A un conjunto difuso estándar sobre el universo de discurso X y $\mu_A(x)$ su función de pertenencia. Entonces podemos definir un **conjunto difuso A^ζ con una onda asociada** como:

$$A^\zeta = \{(\mu_{A\zeta}(x), x) / x \in X\},$$

con

$$\mu_{A\zeta}(x) = \left| A^\zeta(x) \right|$$

y

$$A^\zeta(x) = \mu_A(x) \times e^{i(\omega x + \theta)}.$$

■

Proposición 1

La función de pertenencia $\mu_A(x)$ de un conjunto difuso estándar A y la función de pertenencia $\mu_{A\zeta}(x)$ del conjunto difuso con una onda asociada A^ζ son equivalentes, esto es,

$$\mu_{A\zeta}(x) = \mu_A(x).$$

Demostración (Proposición 1)

$$\mu_{A\zeta}(x) = \left| \mu_A(x) \times e^{i(\omega x + \theta)} \right| =$$

$$= \left| \mu_A(x) \times \left(\cos(\omega x + \theta) + i \sin(\omega x + \theta) \right) \right| =$$

$$= \left| \mu_A(x) \right| \times \left| \left(\cos(\omega x + \theta) + i \sin(\omega x + \theta) \right) \right| =$$

$$= \left| \mu_A(x) \right| \times 1.0 = \mu_A(x).$$

■

De acuerdo a la proposición anterior, todas las propiedades, conceptos y operaciones definidas en la teoría de conjuntos difusos de acuerdo al concepto de función de pertenencia [2,5], son también válidas sobre conjuntos difusos con una onda asociada.

Además, al incluir la función $A^\zeta(x)$ en la definición de conjunto difuso, aportamos las propiedades de las ondas complejas a la hora de manipular la imprecisión con lógica difusa.

En el anexo I analizamos la equivalencia, en una máquina con precisión finita, entre un conjunto difuso estándar y un conjunto difuso con una onda real asociada (*no* compleja).

2.1. REPRESENTACIÓN GRÁFICA DE A^ζ

Si representáramos gráficamente el valor $A^\zeta(x_0)$ (Figura 1) podríamos apreciar que en vez de tener un único valor real correspondiente al grado de pertenencia de x_0 en A^ζ, lo que tenemos es un vector. Este vector contiene en su módulo la información del grado de pertenencia de x_0. Además de esto, $A^\zeta(x_0)$ nos indica una dirección de x_0 en A^ζ.

¿Estas nuevas características son útiles para la manipulación de información imprecisa?.

En la siguiente sección, aprovechando el parámetro fase θ de la onda asociada a un conjunto difuso, definiremos una nueva operación sobre conjuntos difusos con ondas asociadas. Tras su análisis presentaremos varios casos prácticos donde puede ser útil su uso.

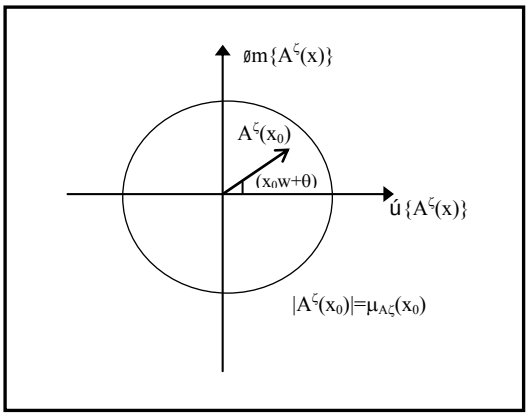


Figura 1: Representación gráfica del valor $A^\zeta(x_0)$.

3 SUMA DE CONJUNTOS DIFUSOS CON UNA ONDA ASOCIADA

El hecho de disponer de una onda en la definición de la función de pertenencia $\mu_{A\zeta}(x)$, nos proporciona nuevos parámetros como son la frecuencia ω y la fase θ de la onda. Estos parámetros nos permiten plantear nuevas operaciones sobre conjuntos difusos que podrían aportar más flexibilidad para manejar situaciones reales con información imprecisa.

Un ejemplo de operación es la *suma* de conjuntos difusos con una onda asociada. Limitaremos a que las ondas tengan la misma frecuencia para facilitar el análisis de los posibles resultados de la operación "*suma*".

Definición 2

Sean A^{ζ}_1 y A^{ζ}_2 dos conjuntos difusos con ondas asociadas

$$A^{\zeta}{}_1(x) = \mu_{A_1}(x) \times e^{i(\omega x + \theta_1)}$$

$$A^{\zeta}{}_2(x) = \mu_{A_2}(x) \times e^{i(\omega x + \theta_2)}$$

cuyas funciones de pertenencia son:

$$\mu_{A^{\zeta}_1}(x) = \left| A^{\zeta}{}_1(x) \right| = \mu_{A_1}(x)$$

$$\mu_{A^{\zeta}_2}(x) = \left| A^{\zeta}{}_2(x) \right| = \mu_{A_2}(x)$$

Entonces definiremos la función de pertenencia de la *suma* de conjuntos difusos con ondas asociadas ($A^{\zeta}_1 + A^{\zeta}_2$) como:

$$\mu_{A^{\zeta}_1 + A^{\zeta}_2}(x) = \left| A^{\zeta}{}_1(x) + A^{\zeta}{}_2(x) \right| =$$

$$= \sqrt{\mu_{A_1}(x)^2 + \mu_{A_2}(x)^2 + 2\,\mu_{A_1}(x)\mu_{A_2}(x)\cos(\theta_2 - \theta_1)}$$

Como podemos apreciar en la definición anterior, el comportamiento de la operación *suma* de conjuntos difusos con una onda asociada es diferente en función de la diferencia de fase de las ondas $\theta_{dif} = \theta_2 - \theta_1$ (o $\theta_{dif} = \theta_1 - \theta_2$, ya que cos(x)=cos(-x)) .

Podemos destacar tres tipos de comportamientos (con $n \in \mathbb{N}$):

a) $\theta_{dif} = 2\Pi n \Rightarrow \cos(\theta_{dif}) = 1 \Rightarrow$ ***Interferencia constructiva***:

$$\mu_{A^{\zeta}_1 + A^{\zeta}_2}(x) = \mu_{A_1}(x) + \mu_{A_2}(x).$$

En las zonas de intersección de los conjuntos difusos A^{ζ}_1 y A^{ζ}_2 se produce la suma de sus funciones de pertenencia.

b) $\theta_{dif} = 2\Pi n + \Pi/2$ (ó $2\Pi n + 3\Pi/2$) $\Rightarrow \cos(\theta_{dif}) = 0 \Rightarrow$ ***Interferencia neutra***:

$$\mu_{A^{\zeta}_1 + A^{\zeta}_2}(x) = \sqrt{\mu_{A_1}(x)^2 + \mu_{A_2}(x)^2}.$$

En este caso, si saturamos a uno la función de pertenencia del conjunto difuso ($A^{\zeta}_1 + A^{\zeta}_2$) nos encontramos con un caso particular (p=2) de la familia de t-conormas de Yager:

$$\text{t}-\text{conorma}\,(x, y) = \min\left(1, \sqrt[p]{x^p + y^p}\right), p > 0.$$

c) $\theta_{dif} = 2\Pi n + \Pi \Rightarrow \cos(\theta_{dif}) = -1 \Rightarrow$ ***Interferencia destructiva***:

$$\mu_{A^{\zeta}_1 + A^{\zeta}_2}(x) = \left| \mu_{A_1}(x) - \mu_{A_2}(x) \right|.$$

En este caso, en la zona de intersección de dos conjuntos difusos (frontera no nítida) se anulan mutuamente las funciones de pertenencia de cada conjunto. En los siguientes apartados veremos la utilidad de este comportamiento.

3.1 ANÁLISIS DE LA SUMA DE CONJUNTOS DIFUSOS CON UNA ONDA ASOCIADA

En este apartados analizaremos como el uso del operador suma al agregar conclusiones de varias reglas difusas da lugar a diferentes comportamientos del sistema difuso. En concreto, encontraremos el comportamiento de los sistemas difusos aproximativos o TSK [10], de los sistemas difusos descriptivos o de Mamdani [6] y a un nuevo comportamiento ilustrado con varios ejemplos. Todos estos comportamientos diferentes se obtienen solamente variando el tipo de ondas asociadas a los conjuntos difusos.

Para llevar a cabo este análisis, supongamos las siguientes reglas:

R_i: Si antecedente_difuso$_i$ $\Rightarrow$ consecuente$_i$, i=1, ..., n.

a) Si las reglas R_i son reglas TSK [10] el *consecuente$_i$* tiene el siguiente formato:

$f_i(\bar{x}) = a_0 + a_1x_1 + ... + a_nx_n, \qquad a_i \in \Re.$

Si tenemos dos reglas R_j y R_k que dan el mismo valor de salida para la entrada $\bar{x}_0$ ($f_j(\bar{x}_0) = f_k(\bar{x}_0)$), el valor final del sistema difuso TSK es igual a:

$$y_{final} = \frac{\sum_{i=1}^{n} f_i(\vec{x_0}) \cdot \alpha_i}{\sum_{i=1}^{n} \alpha_i} = \frac{\left(\sum_{i \in \{1,\ldots,n\} - \{j,k\}} f_i(\vec{x_0}) \cdot \alpha_i \right) + (\alpha_j + \alpha_k) \cdot f_j(\vec{x_0})}{\sum_{i=1}^{n} \alpha_i},$$

donde α_i es el grado en que la entrada $\bar{x}_0$ activa la regla R_i.

Observando la ecuación de la salida y_{final} podemos apreciar que la salida de un sistema difuso TSK es equivalente a defuzzificar con el método del centro de gravedad el resultado de sumar, con **interferencia constructiva**, los conjuntos difusos salida de cada regla TSK (α_i ,f_i ($\bar{x}_0$)).

b) Si R_i son reglas difusas de Mamdani [6], los resultados de cada regla difusa se agregan usando una t-conorma. En este caso, podemos usar la t-conorma resultante de sumar conjuntos difusos con ondas asociadas produciendo **interferencia neutra**.

c) Si para agregar la salida de las reglas difusas usamos el operador suma produciendo **interferencia destructiva**, encontramos un nuevo comportamiento del sistema difuso. Esta peculiaridad podría ser útil en algunos casos prácticos tal como se ilustra con algunos ejemplos en el siguiente apartado.

3.1.1 EJEMPLOS DE USO DE INTERFERENCIA DESTRUCTIVA

A continuación presentamos dos ejemplos en los que es interesante asociar ondas con interferencia destructiva a dos conjuntos difusos con intersección diferente del vacío.

a) ***Consecuentes de reglas incompatibles***.

Supongamos:

- Dos términos difusos de la variable "Tiempo de cocción" con intersección distinta del vacío:

  Tiempo de cocción Bajo.
  Tiempo de cocción Medio.

- Dos términos con intersección no nula de la variable difusa "Dureza de arroz cocido":

  Dureza Alta.
  Dureza Normal.

- Las dos siguientes reglas difusas:

  1) Si Tiempo de cocción es Bajo ⇒ Dureza Alta.
  2) Si Tiempo de cocción Medio ⇒ Dureza Normal.

A la hora de inferir con este sistema difuso nos podemos encontrar un Tiempo de cocción que pertenezca a la vez al término "Bajo" y al término "Medio".

Esto implica activar las reglas 1) y 2) cada una con su grado. Obteniendo dos consecuencias difusas, las cuales se combinarían con una t-conorma si hemos usado una t-norma como operador de implicación.

Esta agregación de consecuentes nos puede llevar a un resultado difuso parecido al que se presenta en la Figura 2.

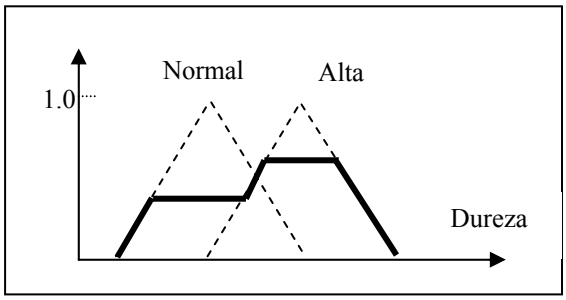


Figura 2: Posible resultado del sistema difuso de cocción de arroz.

Este conjunto difuso pierde parte de su interpretación semántica (problema de la aproximación lingüística [3,12]). Además, si quisiéramos obtener un valor exacto de dureza tendríamos que defuzzificar el conjunto difuso de la Figura 2. Esto nos llevaría a perder el tratamiento formal de la imprecisión.

Al defuzzificar es bastante probable que obtengamos un valor de dureza perteneciente a la vez a dureza "Media" y a dureza "Alta".

Este parece ser un resultado válido a no ser que los consecuentes de estas dos reglas sean incompatibles.

Esta incompatibilidad podría presentarse, por ejemplo, si usáramos la información suministrada por el sistema difuso para tomar una decisión sobre el arroz cocido: nos lo comemos o lo tiramos a la basura, o sea:

Si Dureza es Alta ⇒ Tirar el arroz.

Si Dureza es Normal ⇒ Comer el arroz.

El resultado del sistema difuso (Figura 2) no me facilita la tarea de tomar una decisión.

En conclusión, al aplicar en esta situación una t-conorma para agregar resultados de reglas difusas hemos encontrado dos problemas:

1.- Hemos obtenido un conjunto difuso como resultado (Figura 2) difícil de entender.

2.- Al defuzzificar, el sistema difuso no nos aclara la siguiente acción a realizar.

Analicemos ahora que ocurriría si:

- asociamos ondas con interferencia destructiva a los conjuntos difusos de los términos "Dureza Alta" y "Dureza Normal" y,
- al agregar conclusiones de reglas difusa usamos el operador *suma* definido en el apartado anterior.

En este caso el resultado que nos proporcionaría el sistema difuso para un Tiempo de cocción entre "Bajo" y "Medio" podría ser el reflejado en la Figura 3.

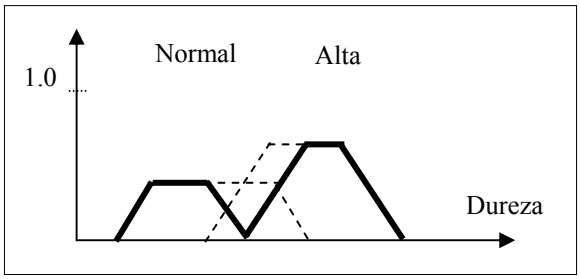


Figura 3: Posible resultado del sistema difuso donde los conjuntos difusos "Normal" y "Alta" tienen ondas asociadas con interferencia destructiva.

El conjunto difuso obtenido parece más fácil de entender que el reflejado en la Figura 2. Se aprecian claramente dos áreas correspondientes a cada valor de Dureza.

El usuario puede decidir si se come o no el arroz con un simple vistazo a la gráfica que ilustra la conclusión del sistema difuso. Elegiría la opción con mayor área.

b) ***Descrición imprecisa de rutas aéreas.***

Supongamos que una ruta aérea está descrita de forma imprecisa de la siguiente manera:

"Para ir del aeropuerto de la ciudad A al aeropuerto de la ciudad C hay que pasar por la "Zona Media" de la cordillera".

El término "Zona Media" describe un área geográfica imprecisa de una cordillera de montañas por donde los pilotos de los aviones saben que pueden pasar.

De un día para otro, abren un nuevo aeropuerto en la ciudad B y describen la siguiente ruta aérea:

"Para ir del aeropuerto de la ciudad B al aeropuerto de la ciudad C hay que pasar por la "Zona Alta" de la cordillera".

Además, las áreas geográficas descritas por "Zona Media" y "Zona Alta" tienen una intersección no nula.

A la hora de calcular los pasillos aéreos que atraviesan la cordillera habría que preguntarse si es interesante asociar ondas con interferencia destructiva a los términos difusos "Zona Media" y "Zona Alta".

Describiendo los pasillos aéreos sin interferencia destructiva, originaría dos corredores aéreos ($A \rightarrow C$ y $B \rightarrow C$) con una zona de intersección no nula (paso común por la cordillera). Esto podría dar lugar a la colisión de dos aviones, uno atravesando la ruta $A \rightarrow C$ y otro cubriendo la ruta $B \rightarrow C$.

4 CONCLUSIONES

En este trabajo hemos presentado una extensión del concepto de conjunto difuso añadiendo una onda a su definición. Esta aportación, por un lado no afecta al comportamiento de los conjuntos difusos cuando los estudiamos como funciones de pertenencia y, por otro lado, introduce propiedades adicionales a los conjuntos difusos que pueden ser útiles al manipular información imprecisa mediante términos difusos.

Hemos analizado el comportamiento de la suma de dos conjuntos difusos que tienen ondas con la misma frecuencia pero diferente fase. Esto nos ha llevado a una nueva forma de agregar conclusiones difusas que puede ser interesante aplicar en algunos casos reales. Este comportamiento se ha ilustrado con dos ejemplos.

El estudio de nuevas propiedades que aporta el acoplamiento de ondas al concepto de conjunto difuso y su aplicación a casos prácticos se deja como trabajo a desarrollar.

Agradecimientos

Este trabajado ha sido posibles gracias a la financiación del proyecto de investigación TIC2003-04650: "Algoritmos lingüísticos: una herramienta para Soft-Computing".

Referencias

[1] M. Alonso, E.J. Finn. Física. Pearson Education, 2000.

[2] D. Dubois and H. Prade, Fuzzy sets and systems: theory and applications. Academic Press, 1980.

[3] F. Eshragh & E.H. Mamdani. A general approach to linguistic approximation. *Int. J. Man-Mach. Stud*, 11, Pág. 501-519, 1979.

[4] R.P. Feynmann. Física, Vol. III: Mecánica Cuántica. *Addison-Wesley Iberoamericana*, Fondo Educativo Interamericano, 1987.

[5] G.J. Klir and B. Yuan. Fuzzy sets and fuzzy logic: Theory and Applications. *Prentice Hall*, 1995.

[6] E.H. Mamdani & S. Assilian. An Experiment in linguistic synthesis with fuzzy logic controller. *Int. J. Man-Machine Stud.*, vol. 7, Pág. 1-13, 1975.

[7] S. Mir, D. Zinger, M. Elbuluk. Fuzzy Controller for Inverter Fed Induction Machines. *IEEE Transactions on Industry Applications*, 30 (1), Pág. 78-84, 1994.

[8] C. Pappis, E. Mamdani. A Fuzzy Logic Controller for a Traffic Intersecion. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-7, Pág. 707-717, 1977.

[9] E.S. Santos. Fuzzy and probabilistic programs. *Information Sciences,* 10, Pág 331-345, 1976.

[10] Y. Takagi, M. Sugeno. Fuzzy identification of Systems and its Application to modeling and control. *IEEE transactions on Systems, Man and Cybernetics*, 15, Pág. 116-132, 1985.

[11] K. Tanaka and M. Mizumoto. Fuzzy programs and their execution. en: L.A. Zadeh, K.S. Fu, K. Tanaka and M. Shimura (Eds), *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, Academic Press, Pág. pp.41-76, 1975.

[12] R. M. Tong & P. P. Bonissone. A linguistic Approach to Decisionmaking with Fuzzy Sets, *IEEE Transactions on Systems, Man and Cybernetics,* vol. 10, nº 11, 1980.

[13] L.A. Zadeh. Fuzzy algorithms. *Inf. Control*, 12, Pág. 94-102, 1968.

ANEXO I: CONJUNTOS DIFUSOS CON ONDAS REALES ASOCIADAS

En la definición de conjunto difuso con una onda asociada presentada en el apartado 2, se ha usado una onda compleja $e^{j(\omega x+\theta)}$. Esta onda tiene la propiedad de que su módulo es igual a uno, o sea, $|e^{j(\omega x+\theta)}|=1$. Esta propiedad es vital para conseguir que las funciones de pertenencia de un conjunto difuso estándar y otro con una onda asociada sean equivalentes.

Ahora nos debemos preguntar si es equivalente el significado de un conjunto difuso estándar y el de un conjunto difuso que tenga una onda real (no compleja) asociada.

Para responder a esta cuestión supongamos sin perdida de generalidad un conjunto difuso estándar *A* cuyo soporte es igual a:

$$\text{Soporte}\,(A) = [0, L] \text{ donde } L \in \Re^{+},$$

y una onda real de la forma:

$$onda\,(x) = \cos(n\,\tfrac{\Pi}{L}\,x) \quad \text{donde } n \in \grave{u}.$$

De esta manera, la función de pertenencia $\mu_{A\zeta}(x)$ del conjunto difuso con la onda real asociada es igual a:

$$\mu_{A\zeta}(x) = \left|\mu_A(x) \times onda\,(\omega x+\theta)\right| =$$
$$= \mu_A(x) \times \left|onda\ (\omega x+\theta)\right| =$$
$$= \mu_A(x) \times \left|\cos(n\,\tfrac{\Pi}{L}(\omega x+\theta))\right|.$$

Para un valor de n grande (n=10000), L=1, ω=1 y θ = 0, la función $|onda\ (\omega x+\theta)|$ en el intervalo [0.0, 0.0003] se ilustra en la Figura 4.

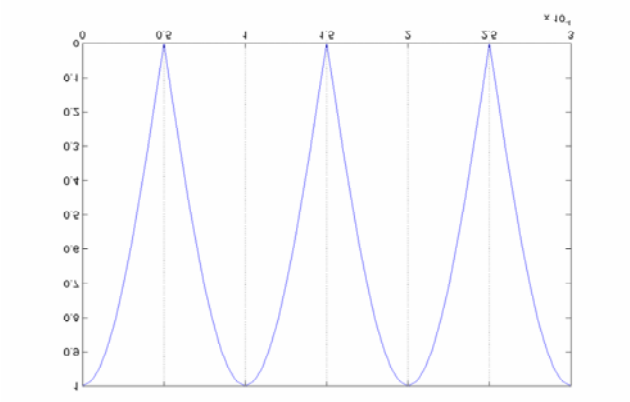


Figura 4: Función "$|onda\ (x)|$" para n=10000 y L=1 en el intervalo [0.0, 0.0003].

Analizando la Figura 4 podemos ver que se cumple:

|onda (ωx+θ)| = 1.0 para x∈ú / nº decimales de x ≤4.

Esto es, hemos diseñado una función onda que consigue que la función de pertenencia $\mu_A(x)$ sea equivalente a la función $\mu_{A\zeta}(x)$ en todos los valores reales con cuatro o menos cifras decimales de precisión. Para incrementar este nivel de precisión sólo deberíamos aumentar el valor de la variable *n* en la función *onda (x)*.

Así podemos advertir que la función de pertenencia de un conjunto difuso estándar y de un conjunto difuso con una onda real asociada son computacionalmente equivalentes en una máquina con una precisión finita, siempre y cuando, la onda asociada tenga una frecuencia suficientemente alta y se hagan coincidir los extremos de las ondas con los valores de precisión finitos considerados.

Función de equivalencia intervalo-valorado Fuzzy. Una aplicación en la segmentación de imágenes.

H. Bustince, E.Barrenechea, M.Pagola y V. Mohedano
Dpt. Automática y Computación
Universidad Pública de Navarra,
Campus Arrosadía s/n, 31106 Pamplona. España,
bustince@unavarra.es

Resumen

En este trabajo presentamos la definición y algunas construcciones de funciones de equivalencia intervalo-valoradas fuzzy. Estudiamos las expresiones obtenidas a partir de funciones de equivalencia fuzzy y de operadores de implicación fuzzy. Además aplicamos las funciones de equivalencia intervalo-valoradas fuzzy a la segmentación de texturas en imágenes.
Palabras Clave: Operador de implicación fuzzy, Función de equivalencia fuzzy, Función de equivalencia intervalo-valorado fuzzy, Segmentación de imágenes.

1 Introducción

Nos planteamos encontrar una expresión de las funciones de equivalencia fuzzy cuando actuan sobre intervalos. Dichas expresiones las obtendremos a partir de funciones de equivalencia fuzzy u operadores de implicación fuzzy actuando sobre agregaciones de los extremos de los intervalos considerados.

En esta introducción presentamos cuatro subsecciones. En la primera, recordamos los conceptos de negación, t-norma y t-conorma (cuando trabajamos con intervalos), y el concepto de operador de agregación. En la segunda recordamos la definición de operador de implicación fuzzy y en la tercera subsección la definición de equivalencia fuzzy, ambas dadas por J.Fodor y M.Roubens ([9]). En la cuarta subsección exponemos la definición de conjunto intervalo-valorado fuzzy y sus primeras propiedades.

En la sección dos definimos la función de equivalencia cuando trabajamos con conjuntos intervalo-valorados fuzzy y especificamos las propiedades que vamos a exigir a tales funciones. A continuación damos una construcción de funciones de equivalencia intervalo-valoradas fuzzy a partir de funciones de equivalencia fuzzy agregando los extremos de los intervalos. También construimos dichas equivalencias a partir de operadores de implicación fuzzy. Establecemos la relación existente entre dichas construcciones. Terminamos esta sección generando equivalencias intervalo-valoradas fuzzy a partir de automorfismos.

En la tercera y última sección, presentamos una aplicación para las funciones de equivalencia intervalo-valoradas fuzzy. Basándonos en el algoritmo FCM ([1]-[2]), proponemos un algoritmo para la segmentación de texturas en imágenes.

1.1 Notaciones especiales

Denotaremos con la letra D al conjunto de todos los subintervalos cerrados de $[0,1]$; es decir:
$D = \{[a,b] | a,b \in [0,1]\}$ donde $a \leq b$. Denotaremos con W a la amplitud del intervalo, $W([a,b]) = b - a$. Diferentes relaciones de orden están estudiadas en ([4],[5]). En particular aquí utilizaremos la siguiente relación:

$$[a,b] \leq [a',b'] \text{ si y sólo si } a \leq a' \text{ y } b \leq b'.$$

Sea n una negación fuerte ([14]-[17]). Llamaremos negación fuerte sobre D y la denotaremos con $\mathbf{n}$ a una función

$$\mathbf{n} : D \to D, \text{ dada por}$$
$$\mathbf{n}([a,b]) = [n(b), n(a)].$$

Es fácil probar que $\mathbf{n}$ satisface las siguientes propiedades:
1) $\mathbf{n}(\mathbf{n}([a,b])) = [a,b]$, es decir, es involutiva.
2) es no creciente, es decir, si $[a,b] \leq [c,d]$ entonces $\mathbf{n}([a,b]) \geq \mathbf{n}([c,d])$.

Sean T y S una t-norma y una t-conorma duales respecto a la negación N (negación estandar N(x)= 1 - x para todo $x \in [0,1]$, en todo el trabajo utilizaremos la negación estandar y la denotaremos con n).
Las expresiones:

$$\mathbf{T} : D^2 \to D \ ,$$
$$\mathbf{T}([a,b],[c,d]) = [T(a,c), T(b,d)]$$

$$\mathbf{S}\text{: } D^2 \to D,$$
$$\mathbf{S}([a,b],[c,d]) = [S(a,c), S(b,d)].$$

satisfacen las siguientes propiedades:
(i) conmutativa,
(ii) asociativa,
(iii) no decreciente en ambos argumentos, es decir,
$\mathbf{T}([a,b],[c,d]) \leq \mathbf{T}([a,b],[c',d'])$ si $[c,d] \leq [c',d']$,
$\mathbf{T}([a,b],[c,d]) \leq \mathbf{T}([a',b'],[c,d])$ si $[a,b] \leq [a',b']$,
$\mathbf{S}([a,b],[c,d]) \leq \mathbf{S}([a,b],[c',d'])$ si $[c,d] \leq [c',d']$,
$\mathbf{S}([a,b],[c,d]) \leq \mathbf{S}([a',b'],[c,d])$ si $[a,b] \leq [a',b']$.
(iv) y las condiciones límite se mantienen, es decir,
$\mathbf{T}([a,b],[1,1]) = [a,b]$,
$\mathbf{T}([a,b],[0,0]) = [0,0]$,
$\mathbf{S}([a,b],[1,1]) = [1,1]$,
$\mathbf{S}([a,b],[0,0]) = [a,b]$.
Es fácil ver que **T** y **S** son extensiones de la t-norma y t-conorma clásicas respectivamente .
Por cuestiones de notación, en este trabajo no vamos a distinguir entre **T** y T ni tampoco entre **S** y S. Debemos tener en cuenta que cuando T actua sobre elementos de D, en realidad nos estamos refiriendo a **T** (construida como se ha expuesto anteriormente), y lo mismo para S y **S**.
Nótese que si $S = \bigvee$ y $T = \bigwedge$, entonces

$$\bigwedge([a,b],[c,d]) = [\bigwedge(a,c), \bigwedge(b,d)]$$
$$\bigvee([a,b],[c,d]) = [\bigvee(a,c), \bigvee(b,d)].$$

En ([4],[5]) está probado que $\{D, \bigvee, \bigwedge\}$ es un retículo distributivo, acotado, no complementado que satisface las Leyes de Morgan.
Una función $M : [0,1]^2 \to [0,1]$ se llama *operador de agregación* ([9],[12]) si satisface las siguientes condiciones:

(A1). $M(0,0) = 0$;
(A2). $M(1,1) = 1$;
(A3). M es no decreciente en ambos argumentos;
(A4). $M(x,y) = M(y,x)$ para todo $x, y \in [0,1]$.

Un operador de agregación es idempotente si $M(x,x) = x$, para todo $x \in [0,1]$.

1.2 Operadores de implicación fuzzy.

En este trabajo nosotros vamos a utilizar la definición de implicación fuzzy presentada por J.Fodor y M.Roubens ([9]). Es decir, un operador de implicación fuzzy es una función

$$I : [0,1]^2 \to [0,1]$$

tal que satisface las siguientes condiciones:
I_1. Si $x \leq z$ entonces $I(x,y) \geq I(z,y)$ para todo $y \in [0,1]$,
I_2. Si $y \leq t$ entonces $I(x,y) \leq I(x,t)$ para todo $x \in [0,1]$,
I_3. $I(0,x) = 1$ para todo $x \in [0,1]$,
I_4. $I(x,1) = 1$ para todo $x \in [0,1]$,
I_5. $I(1,0) = 0$.
Evidentemente estos operadores cumplen:
1)$I(0,0) = I(0,1) = I(1,1) = 1$,
2)$I(1,0) = 0$.
Otras propiedades que normalmente se requieren a la función I, según la aplicación considerada, son las siguientes:
I_6. $I(1,x) = x$;
I_7. $I(x, I(y,z)) = I(y, I(x,z))$;
I_8. $x \leq y$ si y sólo si $I(x,y) = 1$;
I_9. $I(x,0) = n(x)$ es una negación fuerte;
I_{10}. $I(x,y) \geq y$;
I_{11}. $I(x,x) = 1$;
I_{12}. $I(x,y) = I(n(y), n(x))$ con una negación fuerte n;
I_{13}. I es una función continua.

1.3 Equivalencia Fuzzy.

J.Fodor y M.Roubens definen en ([9]) *equivalencia fuzzy* como una operación binaria en el intervalo unidad de la siguiente manera:
Una función $FE : [0,1]^2 \to [0,1]$ se dice *equivalencia fuzzy* si satisface las siguientes condiciones:
1. $FE(x,y) = FE(y,x)$ para todo $x, y \in [0,1]$,
2. $FE(1,0) = FE(0,1) = 0$,
3. $FE(x,x) = 1$ para todo $x \in [0,1]$,
4. Si $x \leq x' \leq y' \leq y$, entonces $FE(x,y) \leq FE(x',y')$.

1.4 Conjuntos intervalo-valorados fuzzy.

Sea X un conjunto ordinario, finito y distinto del conjunto vacío. Un conjunto intervalo-valorado fuzzy ([3]-[8],[10, 11]) en X es una expresión A dada por:

$$A = \{< x_i, M_A(x_i) > | x_i \in X\}$$

donde la función

$$M_A : X \to D$$
$$x_i \to M_A(x_i) = [M_{AL}(x_i), M_{AU}(x_i)]$$

define el grado de pertenencia del elemento x_i en A.

Representaremos con IVFS(X) al conjunto de todos los conjuntos intervalo-valorados fuzzy en el referencial X. Las siguientes definiciones pueden encontrarse en ([3]-[8]). Para todo $A, B \in IVFS(X)$:
1. $A \leq B \Leftrightarrow M_{AL}(x_i) \leq M_{BL}(x_i)$ y $M_{AU}(x_i) \leq M_{BU}(x_i)$ para todo $x_i \in X$
2. $A \preceq B \Leftrightarrow M_{AL}(x_i) \leq M_{BL}(x_i)$ y $M_{AU}(x_i) \geq M_{BU}(x_i)$ para todo $x_i \in X$
3. $A \sqsubseteq B \Leftrightarrow M_{AU}(x_i) \leq M_{BL}(x_i)$ para todo $x_i \in X$
4. $A = B \Leftrightarrow M_{AL}(x_i) = M_{BL}(x_i)$ y $M_{AU}(x_i) = M_{BU}(x_i)$ para todo $x_i \in X$
5. $A' = \{< x_i, \mathbf{n}(M_A(x_i)) > |x_i \in X\} = \{< x_i, [n(M_{AU}(x_i), n(M_{AL}(x_i))] > |x_i \in X\}$

Hacer notar, que para la aplicación que estudiamos en las secciones posteriores nos interesa considerar al conjunto X finito.

2 Equivalencia Intervalo-valorada fuzzy.

Definición1. Una equivalencia intervalo-valorada fuzzy es una función $\chi : D^2 \to D$ tal que:
0. Si $W([a,b]) = W([c,d]) = 0$ entonces $W_{\chi([a,b],[c,d])} = 0$ para todo $[a,b],[c,d] \in D$
1. $\chi([a,b],[c,d]) = \chi([c,d],[a,b])$ para todo $[a,b],[c,d] \in D$,
2. $\chi([a,b],[a,b]) = [1,1]$ para todo $[a,b] \in D$,
3. $\chi([0,0],[1,1]) = \chi([1,1],[0,0]) = [0,0]$,
4. Si $[a,b] \leq [a',b'] \leq [c',d'] \leq [c,d]$ entonces $\chi([a',b'],[c',d']) \geq \chi([a,b],[c,d])$ para todo $[a,b],[a',b'],[c',d'],[c,d] \in D$.

Nota 1. La idea inicial es definir la función de equivalencia intervalo-valorada fuzzy como la extensión natural a intervalos de la función de equivalencia fuzzy, es decir, $\chi([a,b],[c,d]) = [p,q]$ donde $p = inf\{FE(x,y)$ tal que $x \in [a,b], y \in [c,d]\}$ y $q = sup\{FE(x,y)$ tal que $x \in [a,b], y \in [c,d]\}$.
Sin embargo esta construcción no cumple la propiedad 2 de la función de equivalencia intervalo-valorada fuzzy. Como contra ejemplo basta tomar la siguiente función de equivalencia fuzzy $FE(x,y) = \wedge(I(x,y), I(y,x)) = \wedge(\wedge(1, 1-x+y), \wedge(1, 1-y+x))$. En el caso de $\chi([a,b],[a,b]) = [inf\{FE(x,y)\}, sup\{FE(x,y)\}] = [inf\{\wedge(\wedge(1,1-a+b), \wedge(1,1-b+a))\}, sup\{\wedge(\wedge(1,1-a+b), \wedge(1,1-b+a))\}]$, si $a \neq b$ entonces el $inf\{\wedge(\wedge(1,1-a+b), \wedge(1,1-b+a))\}$ es menor que 1, por tanto el intervalo obtenido es distinto del intervalo [1,1].

Ejemplo 1.
$\chi([a,b],[c,d]) = [1-|M(a,b)-M(c,d)|, \bigwedge(1-|M(1-a,1-b)-M(1-c,1-d)|, 1-|M(a,b)-M(c,d)|)]$.

Ejemplo 2. A partir de la Nota 1,
$\chi([a,b],[c,d]) =$
$\begin{cases} [inf\{FE(x,x)\}, sup\{FE(x,x)\}] & si\ [a,b] = [c,d] \\ [inf\{FE(x,y)\}, sup\{FE(x,y)\}] & en\ otro\ caso. \end{cases}$

Nota 2. Queda en trabajos futuros el estudiar la equivalencia intervalo-valorada fuzzy en términos de t-normas intervalo-valoradas y operadores de implicación intervalo-valorados fuzzy ([8]-[9]).

Teorema 1. *Sea una función $FE : [0,1]^2 \to [0,1]$ y sea $M : [0,1]^2 \to [0,1]$ un operador de agregación idempotente que cumple A1, A2, A3, A4. Sea una función $\chi : D^2 \to D$ dada por:*

$$\chi([a,b],[c,d]) = [FE(M(a,b),M(c,d)), \bigvee(FE(M(1-a,1-b),M(1-c,1-d)), FE(M(a,b),M(c,d)))]$$

En estas condiciones,
χ es una función de equivalencia intervalo-valorada fuzzy
si y sólo si
FE es una equivalencia fuzzy tal que $FE(x,y) \geq FE(1-x,1-y)$.

Nota 3. J. Fodor en ([9]) demostró que $I(x,y) = \wedge(I(x,y), I(y,x))$ si I cumple la I_{11}.

Corolario 1. *Sea una función $I : [0,1]^2 \to [0,1]$ que cumple $I_1 - I_5$, I_{11} e I_{12}. Sea $M : [0,1]^2 \to [0,1]$ un operador de agregación idempotente que cumple A1, A2, A3, A4 y sea una función $\chi : D^2 \to D$ dada por:*

$$\chi([a,b],[c,d]) = [\bigwedge(I(M(a,b),M(c,d)), I(M(c,d),M(a,b))), \bigvee(\bigwedge(I(M(1-a,1-b),M(1-c,1-d)), I(M(1-c,1-d),M(1-a,1-b))), \bigwedge(I(M(a,b),M(c,d)), I(M(c,d),M(a,b))))].$$

En estas condiciones χ es una función de equivalencia intervalo-valorada fuzzy.

Corolario 2. *Sea una función $FE : [0,1]^2 \to [0,1]$. Sea $M : [0,1]^2 \to [0,1]$ un operador de agregación idempotente que cumple A1, A2, A3, A4 y sea $\chi : D^2 \to D$ dada por:*

$$\chi([a,b],[c,d]) = [FE(M(a,b),M(c,d)), \bigvee(FE(M(1-a,1-b),M(1-c,1-d)),$$

$$FE(M(a,b),M(c,d)))].$$

En estas condiciones los siguientes items son equivalentes:
i) χ es una función de equivalencia intervalo-valorada.
ii) FE es una equivalencia fuzzy tal que $FE(x,y) \geq FE(1-x,1-y)$.
iii) Existe un operador de implicación I que cumple I_{11} e $I(x,y) \geq I(1-y,1-x)$ tal que

$$\chi([a,b],[c,d]) =$$
$$[\bigwedge(I(M(a,b),M(c,d)),I(M(c,d),M(a,b))),$$
$$\bigvee(\bigwedge(I(M(1-a,1-b),M(1-c,1-d)),$$
$$I(M(1-c,1-d),M(1-a,1-b))),$$
$$\bigwedge(I(M(a,b),M(c,d)),I(M(c,d),M(a,b))))].$$

Corolario 3. *Si $\varphi : [0,1] \to [0,1]$ es un automorfismo y $n(x) = \varphi^{-1}(1-\varphi(x))$ la negacin fuerte generada por él, entonces*
$\chi([a,b],[c,d]) = [\bigwedge(\varphi^{-1}(\bigwedge(1,1-\varphi(M(a,b))+ \varphi(M(c,d)),\varphi^{-1}(\bigwedge(1,1+\varphi(M(a,b))-\varphi(M(c,d))),$
$\bigvee(\bigwedge(\varphi^{-1}(\bigwedge(1,1-\varphi(M(n(a),n(b)))+ \varphi(M(n(c),n(d))),\varphi^{-1}(\bigwedge(1,1+\varphi(M(n(a),n(b)))- \varphi(M(n(c),n(d))),\bigwedge(\varphi^{-1}(\bigwedge(1,1-\varphi(M(a,b))+ \varphi(M(c,d)),\varphi^{-1}(\bigwedge(1,1+\varphi(M(a,b))-\varphi(M(c,d))))]$
es una equivalencia intervalo-valorada fuzzy.

Ejemplo 2.
Sea un automorfismo $\varphi(x) = \sqrt{x}$, y $n(x) = (1-\sqrt{x})^2$.
$\chi([a,b],[c,d]) =$
$[\bigwedge((\bigwedge(1,1-\sqrt{\frac{a+b}{2}}+\sqrt{\frac{c+d}{2}}))^2,(\bigwedge(1,1+ \sqrt{\frac{a+b}{2}}-\sqrt{\frac{c+d}{2}}))^2),\bigvee(\bigwedge((\bigwedge(1,1-\sqrt{\frac{(1-\sqrt{a})^2+(1-\sqrt{b})^2}{2}}+ \sqrt{\frac{(1-\sqrt{c})^2+(1-\sqrt{d})^2}{2}}))^2,((\bigwedge(1,1-\sqrt{\frac{(1-\sqrt{c})^2+(1-\sqrt{d})^2}{2}}+ \sqrt{\frac{(1-\sqrt{a})^2+(1-\sqrt{b})^2}{2}}))^2),\bigwedge((\bigwedge(1,1-\sqrt{\frac{a+b}{2}}+ \sqrt{\frac{c+d}{2}}))^2,(\bigwedge(1,1+\sqrt{\frac{a+b}{2}}-\sqrt{\frac{c+d}{2}}))^2))].$

3 Aplicación de la equivalencia intervalo-valorada fuzzy a texturas.

La obtención de diferentes regiones de la imagen es un tópico importante en los algoritmos de visión por computador. Su objetivo es dividir la imagen en diferentes regiones, cada una de las cuales contenga una textura diferente a la de sus vecinos. Además se considera una textura como uno o más patrones locales que se repiten de una manera periódica.

Existen muchos algoritmos de segmentación de texturas supervisados y no supervisados en la literatura ([2]). La diferencia radica en que la segmentación supervisada requiere de un conocimiento previo del tipo de texturas presentes en la imagen. En general, los algoritmos intentan extraer las características más importantes de las texturas como la tonalidad, direccionalidad, frecuencia y aleatoriedad.
En este trabajo presentamos una nueva visión en el procesamiento de imágenes, tratando la imagen como un conjunto intervalo-valorado fuzzy. Cada pixel de la imagen, tiene una intensidad en la escala de grises, desde el 0 hasta el valor 255. Son 256 valores donde el 0 representa el color negro y el 255 el color blanco, los valores intermedios son toda la escala de grises.
Una imagen A está representada por un conjunto:

$$A = \{(x_{ij}, I(x_{ij}))\} \text{ tal que } I(x_{ij}) \in \{0,1,2,..,255\} \text{ para todo } i = 1,..,m \; j = 1,..,n$$

Es decir, cada x_{ij} es un pixel de la imagen e $I(x_{ij})$ es la intensidad de gris de gris de dicho pixel.
Nosotros utilizamos un método muy sencillo que, dada una imagen creamos su conjunto intervalo-valorado fuzzy asociado.
Recorremos la imagen y a cada pixel le asignamos un intervalo, es decir, al pixel x_{ij} le asociamos el intervalo $[M_{AL}(x_{ij}), M_{AU}(x_{ij})] \in \{0,1,2,..,255\}^2$ para todo $i = 1,..,m \; j = 1,..,n$, tal que $M_{AL}(x_{ij}) \leq M_{AU}(x_{ij})$.
El intervalo asignado a un pixel depende de las intensidades de los pixels de la malla lxl que está centrada en él. La figura 1 representa un ejemplo de tamaño $l = 5$.

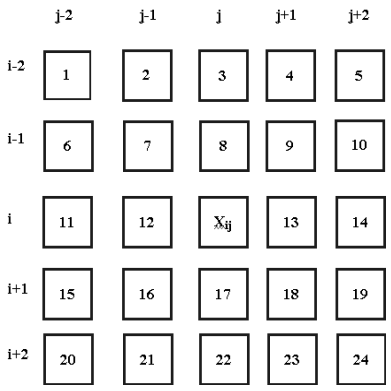


Figura 1: Malla de tamaño 5x5

Por tanto,
$[M_{AL}(x_{ij}), M_{AU}(x_{ij})] = [min(I(x_{rs})), max(I(x_{rs}))]$ para todo $r = i-2,..,i+2$, y para todo $s = j-2,..,j+2$.
El algoritmo propuesto es una modificación al algoritmo FCM (Fuzzy Cluster Means)([1]). El FCM, particiona un conjunto de datos en k-clusters tal que mini-

miza la distancia dentro de cada cluster. Nuestro algoritmo particiona la imagen en k-texturas representadas por k-intervalos, maximizando la equivalencia dentro de las texturas; usando las funciones de equivalencia intervalo-valoradas fuzzy.

3.1 Estructura del algoritmo.

1. Dada una imagen de n pixels crear el conjunto intervalo-valorado
   $A = \{< x_i, M_A(x_i) = [M_{AL}(x_i), M_{AU}(x_i)] > |x_i \in X\}$ con $i = 1..n$.

2. Inicializar
   2.1 K=número de texturas a segmentar.
   2.2 T=número máximo de iteraciones.
   2.3 ϕ=grado de fuzzicación, $1 \leq \phi < \infty$
   2.4 ε=Umbral de finalización, $\varepsilon > 0$
   2.5 W=matriz de pesos, $w_{ij} > 0$ con $i = 1..n$ y $j = 1..K$

3. $t \leftarrow 0$

4. REPETIR
   4.1 $t \leftarrow t + 1$
   4.2 Calcular centros:

$$C_j = \frac{\sum_{i=1}^{n} w_{ij}^{\phi} M_A(x_i)}{\sum_{i=1}^{n} w_{ij}^{\phi}}$$

   4.3 Recalcular matriz de pesos:

$$w_{ij} = \left(\sum_{k=1}^{K} \left(\frac{1 - M(\chi(M_A(x_i), C_j)}{1 - M(\chi(M_A(x_i), C_k)} \right) \right)^{\frac{2}{\phi - 1}}$$

   donde $M(\chi(M_A(x_i), C_k)$ es la agregación con la media aritmética de de los extremos del intervalo.
   4.4 $E_t = ||M^{it} - M^{it-1}||$
   4.5 HASTA ($t = T$ o $E_t \leq \varepsilon$)

5. Etiquetado: $x_i \leftarrow Textura_j$ si y sólo si $w_{ij} > w_{ik}$, para todo $x_i \in A, \quad j, k = 1..K, \quad k \neq j$

Ejemplo 3.
Comprobamos la eficacia de la tranformación de una imagen en un conjunto intervalo-valorado fuzzy, frente a un tratamiento clásico de la imagen, donde cada pixel está representado por su intensidad de gris. En primer lugar, aplicamos el FCM clásico usando solamente el nivel de gris y en segundo lugar un FCM modificado usando el conjunto intervalo-valorado asociado a la imagen. Para ello utilizamos la figura 2.

El número de texturas es $K = 3$, y siendo $\phi = 1.5$,

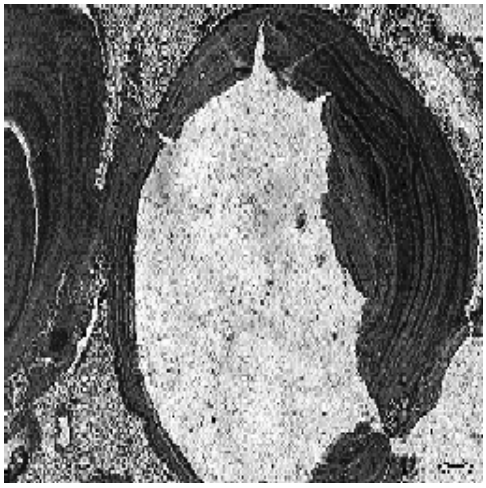


Figura 2: Texturas minerales

$T = 20$ y $\varepsilon = 0.05$. El resultado obtenido de FCM clásico queda reflejado en la figura 3.

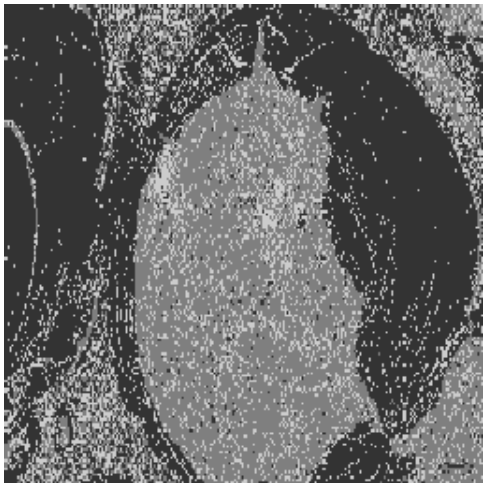


Figura 3: Segmentación usando nivel de gris

En segundo lugar construimos el conjunto intervalo-valorado fuzzy con una malla de tamaño 5x5 y aplicamos el FCM modificado con la función de equivalencia intervalo-valorada χ del ejemplo 1. El resultado obtenido se presenta en la figura 4.

3.2 Conclusiones.

La principal conclusión del trabajo realizado es que la transformación en un conjunto intervalo-valorado fuzzy de una imagen, puede ayudar a conseguir resultados más eficaces en el procesamiento de imágenes. Como hemos visto en el ejemplo anterior usando sólo una característica de cada pixel, la segmentación es mejor en el caso intervalo-valorado. Como futuras

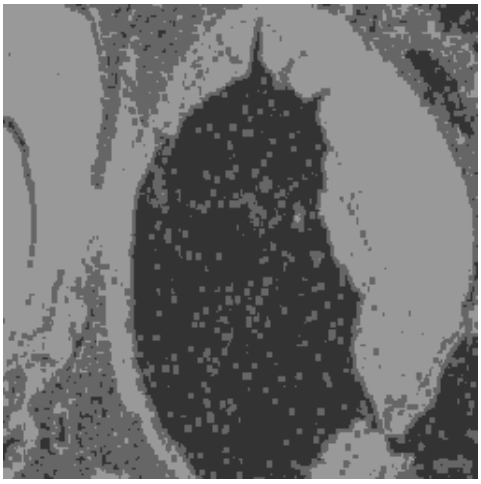


Figura 4: Segmentación usando intervalos

líneas de trabajo, nos proponemos estudiar distintos métodos de procesamiento de imágenes tratadas como conjuntos intervalo-valorados fuzzy.

Referencias

[1] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.*, Plenum, NY (1981).

[2] J.C. Bezdek, J. Keller, R. Krisnapuram, N.R. Pal *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing.*, (Kluwer Academic Publishers) (1999).

[3] P.Burillo and H.Bustince, *Intuitionistic Fuzzy Relations (Part I)*, Mathware and Soft Computing, 2 (1995) 5-38.

[4] P.Burillo and H.Bustince, *Orderings in the referential set induced by an intuitionistic fuzzy relations*, Notes on IFS, 1 (1995) 93-103.

[5] H.Bustince, *Conjuntos Intuicionistas e Intervalo valorados Difusos: Propiedades y Construcción. Relaciones Intuicionistas Fuzzy*, Ph. D. Thesis, Universidad Pública de Navarra, 1994.

[6] H.Bustince and P.Burillo, *Pertubation of intuitionistic fuzzy relations*, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 9 (2001) 81-1003.

[7] H.Bustince and P.Burillo, *Structures on Intuitionistic Fuzzy Relations*, Fuzzy Sets and Systems, 78 (1996) 293-303.

[8] C.Cornelis and G.Deschrijver, *The Compositional Rule of Inference in an Intuitionistic Fuzzy Logic Setting*, Proceedings of the Sixth ESSLLI Students Session, Kristina Striegnitz (editor). 2001.

[9] J.Fodor and M.Roubens, *Fuzzy Preference Modelling and Multicretireis Decision Support*, in: Theory and Decision Library (Kluwer Academic Publishers) (1994).

[10] M.B.Gorzalczany, *A method of inference in approximate reasoning based on interval-valued*, Fuzzy Sets Fuzzy Sets Syst., 21 (1987) 1-17.

[11] S.Jenei, *A more efficient method for defining fuzzy connectives*, Fuzzy Sets and Systems, 90 (1997), 25-35.

[12] G.Klir and B.Yuan, *Fuzzy Sets and Fuzzy Logic: theory and applications*, Prentice Hall, Upper Saddle River, NJ, 1995.

[13] D.Ruan and E.E.Kerre, *Fuzzy Implication Operators and Generalized Fuzzy Method of Cases*, Fuzzy Sets and Systems, 54 (1993), 23-37.

[14] E.Trillas, *Sobre funciones de negación en la teoría de conjuntos difusos*, Stochastica, III-1, (1979) (47-59).

[15] E.Trillas, C.Alsina, J.M.Terricabras *Introducción a la lógica borrosa*, Ariel Matemática, (1995).

[16] E.Trillas and L.Valverde, *On Implication and Indistinguishability in the Setting of Fuzzy Logic*, Readings in Fuzzy Sets for Intelligent Systems, Edited by D.Dubois, H.Prade and R.R.Yager (1993), (97-104).

[17] E.Trillas and L.Valverde, *On mode and implication in approximate reasoning in: M.M.Gupta, A.Kandel, W.Bandler, J.B.Kiszka, Eds.*, Approximate reasoning in Expert Systems (North-Holland, Amsterdam)(1985), (157-166).

OWA–like Aggregation Operators

T. Calvo[1,2]
[1]Dept. of Computer Science,
University of Alcalá, Spain
[2]LOBFI Research Group,
University of The Balearic Islands, Spain
[1,2]tomasa.calvo@{uah.es, uib.es}

J. Lázaro
Dept. of Computer Science,
University of Alcalá, Spain
jesus.lazaro@uah.es

Abstract

We present here different construction methods for aggregation operators obtained by means of Ordered Weighted Averaging Operators (OWA operators). First we consider the Choquet integral based aggregation operators and extend them introducing an additional OWA operator (or its corresponding symmetric fuzzy measure) in its expression. After that, we show an alternative construction method for OWA operator based bipolar aggregation operators. Finally, we present the double OWA operator family as a symmetric comonotone additive double aggregation operator.

Keywords: aggregation operator, bipolar aggregation operator, double aggregation operator, Choquet integral, OWA operator.

1 Introduction

Aggregation operators transform a finite number of inputs into a single output. They are applied in many theoretical and practical domains, such as many–valued logics, fuzzy control, multi–criteria decision making, etc. They are also an important tool of many intelligent techniques, as was observed already by Zimmermann and Zysno [22]. Overview of several aggregation operators and techniques can be found in, e.g., [8, 3]. For an overview of recent theoretical and applicational results in the domain we recommend [4]. Among recent applicational papers recall, e.g., [6, 17]. Determining of appropriate aggregation operators is discussed, e.g., in [1, 16].

Ordered weighted averaging operators were introduced by Yager [19, 11]. Enormous interest in OWA operators has provided many papers and even an edited volume [20]. In particular, max, min and arithmetic mean belong to the OWA family that is useful in generalizing decision making under uncertainty [21].

The aim of this paper is to show new ways to introduce aggregation operators related to the OWA family and to present an extension to other well–known methods, and it is organized as follows: first, in Section 2 we recall the necessary material for the paper. Section 3 is devoted to OWA operators obtained from certain families of aggregation operators, namely Šipoš integral based operators and bipolar aggregation operators [14]. Next, in Section 4 we define the family of double OWA operators related to double aggregation operators [5] and give a characterization for the family. Finally, some conclusions will be given.

In this paper we will deal with normalized fuzzy measures and with aggregation operators as introduced in [12, 4], i.e., with inputs and outputs from the unit interval $[0, 1]$ only, unless explicitly mentioned; some general considerations for the case $[a, b]$ will be given to provide a link between different families of operators.

2 Preliminaries

- A *fuzzy measure* μ on $N = \{1, \ldots, n\}$ is a monotonic set function $f : \mathcal{P}(N) \to \mathbb{R}^+$ such that $\mu(\emptyset) = 0$. μ is *normalized* if $\mu(N) = 1$. The *conjugate* of fuzzy measure μ is a fuzzy measure defined as $\overline{\mu}(X) = 1 - \mu(X^c)$.

- For a function $f : N \to \mathbb{R}^+$, the *Choquet integral* of f with respect to μ is defined by

$$C_\mu(f) = \sum_{i=1}^{n} [f(\sigma(i)) - f(\sigma(i-1))] \cdot \mu(I_{\sigma(i)})$$

  where $f(\sigma(1)) \leq f(\sigma(2)) \leq \ldots \leq f(\sigma(n))$ and $I_{\sigma(i)} = \{\sigma(i), \ldots, \sigma(n)\}$. The Choquet integral is also defined for real-valued functions in the following 2 ways:

$$C_\mu(f) = C_\mu(f^+) - C_{\overline{\mu}}(f^-)$$
$$\check{C}_\mu(f) = C_\mu(f^+) - C_\mu(f^-)$$

which are called, respectively, *asymmetric Choquet integral* and *symmetric Choquet integral*, and where $f^+ = f \vee 0$, $f^- = (-f)^+$.

- An *aggregation operator* is a mapping $A : \bigcup_{n\in\mathbb{N}} [0,1]^n \to [0,1]$ such that:
  (i) $A(x_1,\ldots,x_n) \leq A(y_1,\ldots,y_n)$ whenever $x_i \leq y_i$ for all $i \in \{1,\ldots,n\}$.
  (ii) $A(x) = x$ for all $x \in [0,1]$.
  (iii) $A(\overbrace{0,\ldots,0}^{n-times}) = 0$ and $A(\overbrace{1,\ldots,1}^{n-times}) = 1$.

- A $W : \bigcup_{n\in\mathbb{N}} [0,1]^n \to [0,1]$ mapping is an Ordered Weighted Averaging operator (OWA), if it is given by

$$W(x_1,\ldots,x_n) = \textstyle\sum_{i=1}^n w_{in} \cdot x_{\sigma(i)},$$

where $(x_{\sigma(1)},\ldots,x_{\sigma(n)})$ is a non-decreasing permutation of the input n–tuple $(x_1,\ldots,x_n)$ with the weighting triangle $\triangle = (w_{in})$. The *reverse OWA* denoted by W^r is an OWA such that its weighting triangle $\triangle_{W^r}$ is symmetric respect to $\triangle_W$, i.e., $w_i^r = w_{n+1-i}$.

Evidently, each OWA operator is a symmetric continuous idempotent aggregation operator which is linear and comonotone additive. There is a close relationship between OWA operators and the Choquet integral-based operators; this has been considered, for example, in [11, 10, 13]. Indeed, each OWA operator can be represented by means of the Choquet integral [2] with respect to an appropriate system $(\mu_n)_{n\in\mathbb{N}}$ of symmetric fuzzy measures on $X_n = \{1,\ldots,n\}$. Following [11], this means that each symmetric comonotone additive aggregation operator is just an OWA operator. This operator is a specific type of the Choquet integral [19].

3 OWA operators related to bipolar aggregation operators or Šipoš integral

3.1 Šipoš integral

As it has been referenced, the Choquet integral of the function $f : N \to \mathbb{R}^+$ with respect to a normalized symmetric measure μ is equivalent to the OWA with weights $w_i = \mu(I_{\sigma(i)}) - \mu(I_{\sigma(i+1)})$. Now, given a certain OWA operator $B : \bigcup_{n\in\mathbb{N}} [0,1]^n \to [0,1]$ and $\alpha \in [0,1]$, we can define the following operator $A : \bigcup_{n\in\mathbb{N}} [0,1]^n \to [0,1]$,

$$A(\mathbf{x}) = B(\mathbf{x}^+) - B(\mathbf{x}^-) + \alpha$$

where $\mathbf{x}^+ = \{\mathbf{x} - \alpha \vee 0\}$ and $\mathbf{x}^- = \{\alpha - \mathbf{x} \vee 0\}$. It is clear that both $\mathbf{x}^+$, $\mathbf{x}^-$ are in $[0,1]^n$ and that $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^- + \alpha$.

This operator A is a symmetric aggregation operator:

- A is idempotent as can be easily proven, and therefore $A(\mathbf{0}) = 0$, $A(\mathbf{1}) = 1$.
- A is non–decreasing; to show this, consider $\mathbf{y} \geq \mathbf{x}$. It is clear that $\mathbf{y}^+ \geq \mathbf{x}^+$ and $\mathbf{y}^- \leq \mathbf{x}^-$, and then $A(\mathbf{y}) = B(\mathbf{y}^+) - B(\mathbf{y}^-) + \alpha \geq B(\mathbf{x}^+) - B(\mathbf{x}^-) + \alpha = A(\mathbf{x})$.
- A is symmetric due the symmetry of B.

Observe the similarity between the operator A and the (symmetric) Choquet integral of the function $f : N \to \mathbb{R}$ with respect to a normalized symmetric measure μ:

$$\check{C}_\mu(f) = C_\mu(f^+) - C_\mu(f^-)$$

Note that, in our case, it is not true that $\mathbf{x}^- = (-\mathbf{x})^+$ and that we must add the final α (in the symmetric Choquet integral α equals zero); i.e., the aggregation operator A can be seen as a generalization of the symmetric Choquet integral. But, in general, operator A needs not to be an OWA, as it can be seen in the next example:

Example 1 *Taking $B = \max$ and $\alpha = 0.5$ the following operator A is obtained:*

$A_{\max}(\mathbf{x}) =$

$$\begin{cases} \max\{x_i\}, & \text{if } \mathbf{x} \geq 0.5 \\ \min\{x_i\}, & \text{if } \mathbf{x} \leq 0.5 \\ \min\{x_i\} + \max\{x_i\} - 0.5, & \text{otherwise.} \end{cases}$$

which clearly is not an OWA operator.

Moreover, we can fix two OWA operators $B_1, B_2 : \bigcup_{n\in\mathbb{N}} [0,1]^n \to [0,1]$ to define the more general aggregation operator A as follows:

$$A(\mathbf{x}) = B_1(\mathbf{x}^+) - B_2(\mathbf{x}^-) + \alpha$$

that can be seen as

$$C_{\mu_1,\mu_2}(f) = C_{\mu_1}(\mathbf{x}^+) - C_{\mu_2}(\mathbf{x}^-) + \alpha$$

where μ_1, μ_2 are the symmetric measures generating B_1, B_2 and f is a function $f : N \to \mathbb{R}^+$ such that $f(i) = x_{\sigma(i)}$. This is an extension of the *Cumulative Prospect Theory model* with respect to μ_1, μ_2, see [18], defined as

$$CPT(f) = C_{\mu_1}(f^+) - C_{\mu_2}(f^-).$$

where f^+, f^- are defined as usual.

The aggregation operator A will be an OWA if and only if it can be expressed as the Choquet integral of a symmetric fuzzy measure μ_A, i.e., if there is a symmetric fuzzy measure μ_A such that $C_{\mu_A} = C_{\mu_1,\mu_2}$. As seen in Example 1, $\mu_1 = \mu_2$ is not a sufficient condition for this.

Proposition 1 *Given α, B_1, B_2 as before, the aggregation operator $A(\mathbf{x}) = B_1(\mathbf{x}^+) - B_2(\mathbf{x}^-) + \alpha$ is an OWA operator if and only if $B_2 = B_1^r$, and in this situation $A = B_1$.*

We only present here the main idea to prove this result. If A is an OWA, just taking into account its weighting triangle and the fact that both B_1, B_2 are OWA in the same domain than A its enough to obtain the relationship between them. In the other hand, given $B_2 = B_1^r$ it is easy to see that the permutations σ_{B_1} and σ_{B_2} of the inputs are also one reversal of the other, which allows to simplify the weights into one weighting triangle and to prove that $A = B_1$.

Corollary 1 *If $B_1 = B_2$, A is an OWA operator if and only if the weighting triangle $\triangle_{B_1} = (w_{in})$ is symmetric, i.e., $w_{i,n} = w_{n+1-i,n}$ for all i, n.*

All these results can be expressed via symmetric fuzzy measures as follows

Proposition 2 *Given $\mu_1, \mu_2 : \mathcal{P}(N) \to \mathbb{R}^+$ symmetric fuzzy measures, $\alpha \in [0,1]$ and $\mathbf{x}^+, \mathbf{x}^-$ as previously defined, the aggregation operator*

$$A(\mathbf{x}) = C_{\mu_1}(\mathbf{x}^+) - C_{\mu_2}(\mathbf{x}^-) + \alpha$$

is an OWA operator if and only if $\mu_2 = \overline{\mu_1}$ and, in this case, $\mu_A = \mu_1$.

Remark 1 Observe the resemblance between the above operator A and the asymmetric Choquet integral $C_u(\mathbf{x}) = C_u(\mathbf{x}^+) - C_{\overline{u}}(\mathbf{x}^-)$ when $\mathbf{x}^+ = \{\mathbf{x} \vee 0\}$ and $\mathbf{x}^- = (-\mathbf{x})^+$.

Remark 2 Given α, B_1, B_2, if $B_2 \neq B_1^r$, let $\mathbf{v_i}$ be defined as

$$\mathbf{v_i}_j = \begin{cases} 1 & \text{if } j > n-i \\ 0 & \text{if } j \leq n-i \end{cases}$$

i.e., $\mathbf{v_i} = (\overbrace{0,\dots,0}^{n-i}, \overbrace{1,\dots,1}^{i})$. We have

$$\mathbf{v_i}^+ = (\overbrace{0,\dots,0}^{n-i}, \overbrace{1-\alpha,\dots,1-\alpha}^{i})$$

$$\mathbf{v_i}^- = (\overbrace{\alpha,\dots,\alpha}^{n-i}, \overbrace{0,\dots,0}^{i})$$

and then $A(\mathbf{v_i}) = \sum_{j=n+1-i}^{n} w_j^1(1-\alpha) - \sum_{j=i+1}^{n} w_j^2\alpha + \alpha$ which can be simplified to

$$A(\mathbf{v_i}) = \sum_{j=n+1-i}^{n} w_j^1(1-\alpha) + \sum_{j=1}^{i} w_j^2\alpha$$

i.e., $A(\mathbf{v_i})$ is a weighted mean of the last i weights of B_1 with the first i weights of B_2. Also, as a result of the symmetry of A, the application of A to any vertex of $[0,1]^n$ is a weighted mean of i weights of B_1 and B_2, where i is the ammount of 1–inputs in the vertex.

We can generalize the ideas in this subsection to any interval $[a,b]$. Now, given $\mathbf{x} \in [a,b]^n$, $\alpha \in [a,b]$, we define $\mathbf{x}^+ = \mathbf{a} + (\mathbf{x} - \alpha \vee 0)$ and $\mathbf{x}^- = \mathbf{a} + (\alpha - \mathbf{x} \vee 0)$. Both $\mathbf{x}^+, \mathbf{x}^-$ are in $[a,b]^n$ and we have that $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^- + \alpha$. The OWA operators B_1, B_2 will be defined in $\bigcup_{n\in\mathbb{N}} [a,b]^n \to [a,b]$ and $A : \bigcup_{n\in\mathbb{N}} [a,b]^n \to [a,b]$, $A(\mathbf{x}) = B_1(\mathbf{x}^+) - B_2(\mathbf{x}^-) + \alpha$ is an aggregation operator.

Properties 1, 2 have an analogous result in $[a,b]$. To generalize Remark 2, the natural idea is define $\mathbf{v_i}$ as

$$\mathbf{v_i}_j = \begin{cases} b & \text{if } j > n-i \\ a & \text{if } j \leq n-i \end{cases}$$

but in this case the following value $A(\mathbf{v_i})$ is obtained

$$A(\mathbf{v_i}) = \sum_{j=n+1-i}^{n} w_j^1(b-\alpha) + \sum_{j=1}^{i} w_j^2(\alpha - a) + a \quad (1)$$

where a weighted mean was expected. But this value can be seen as the transformation of the convex sum in $[0,1]$ via the usual bijection $\varphi : [0,1] \to [a,b]$ defined as $\varphi(x) = (b-a)x + a$. To show this idea, consider $X = \sum_{j=n+1-i}^{n} w_j^1$, $Y = \sum_{j=1}^{i} w_j^2$, $\alpha \in [a,b]$. Equation 1 results to be $\varphi(X(1 - \varphi^{-1}(\alpha)) + Y\varphi^{-1}(\alpha))$, i.e., it is the transformation of the convex combination of X and Y using the inverse value of α.

Example 2 *Let $B_1 = (\frac{1}{2}, 0, \frac{1}{2}) = \frac{\max+\min}{2}$, $B_2 = (0,1,0) = \text{med}_3$, and $\alpha = \frac{a+b}{2}$. As $B_1 \neq B_2^r$, A is not an OWA operator, but the following values are immediate:*

- $A(a,a,b) = \frac{1}{2}(b - \frac{a+b}{2}) + 0(\frac{a+b}{2} - a) + a = \frac{3a+b}{4}$
- $A(a,b,b) = \frac{1}{2}(b - \frac{a+b}{2}) + 1(\frac{a+b}{2} - a) + a = \frac{a+3b}{4}$

Taking into account that $\varphi^{-1}(\alpha) = 0.5$, these values can be also be obtained as the transformation of the mean of the sum of the weights:

- $A(a,a,b) = \varphi((\frac{1}{2}) \cdot 0.5 + (0) \cdot 0.5) = \frac{3a+b}{4}$
- $A(a,b,b) = \varphi((\frac{1}{2} + 0) \cdot 0.5 + (0 + 1) \cdot 0.5) = \frac{a+3b}{4}$

3.2 Bipolar aggregation operators

The bipolar scale $[-1,1]$ allows to deal with positive (supporting) information as well as negative (excluding) one. Several aggregation operators act on bipolar scales, as the previously seen Šipoš integral. A definition of bipolar aggregation operators, as well as some of their properties, and the relationship between bipolar aggregation operators and bi–capacities, can be found in [14].

Definition 1 A mapping $B : \bigcup_{n\in\mathbb{N}} [-1,1]^n \to [-1,1]$ will be called a bipolar aggregation operator if

(i) B is non-decreasing, i.e., for any $n \in \mathbb{N}$, $\mathbf{x}, \mathbf{y} \in [-1,1]^n, \mathbf{x} \leq \mathbf{y}$ it holds $B(\mathbf{x}) \leq B(\mathbf{y})$,

(ii) $-1, 0, 1$ are idempotent elements of B, i.e., for any $n \in \mathbb{N}$, $B(c,\ldots,c) = c$ for $c \in \{-1,0,1\}$,

(iii) for $n = 1$, $B(x) = x$ for all $x \in [-1,1]$.

Clearly, any bipolar OWA operator B is a symmetric comonotone additive bipolar aggregation operator. To prove the reverse it is enough to find a weighting triangle that defines the bipolar aggregation operator; if we denote $q_i = B(\mathbf{v_i})$, $i \in \{0,\ldots,n\}$, we will use the values $w_{n+1-i} = q_i - q_{i-1}$ as weights for B. Since the comonotone additive property of B implies its continuity, we have that $B(0,\ldots,0,\overbrace{x,\ldots,x}^{i}) = xq_i$. Now, as we can decompose any input $\mathbf{x}$ in comonotone vectors in the former way, the output $B(\mathbf{x}) = \sum_{i=1}^{n} w_i x_{\sigma(i)}$ is obtained.

Example 3 *Given $\mathbf{x} = (0.9, -0.4, 0, 0.3)$ we have that $\mathbf{x} = \mathbf{y_1} + \mathbf{y_2} + \mathbf{y_3} + \mathbf{y_4}$ where*

- $\mathbf{y_1} = (-0.4, -0.4, -0.4, -0.4)$
- $\mathbf{y_2} = (0.4, 0.4, 0, 0.4)$
- $\mathbf{y_3} = (0.3, 0, 0, 0.3)$
- $\mathbf{y_4} = (0.6, 0, 0, 0)$

and $B(\mathbf{x}) = 0.6q_1 + 0.3q_2 + 0.4q_3 - 0.4q_4 = -0.4w_1 + 0.3w_3 + 0.9w_4$. Observe that although this auxiliary vectors are not comonotone additive, they can be obtained from comonotone vectors; for example, $\mathbf{y_3} + \mathbf{y_4}$ can be represented as $\mathbf{y_3} + (0.3, 0, 0, 0) + (0.3, 0, 0, 0)$ which are effectively comonotones.

Some construction methods for bipolar aggregation operators were also presented in [14]; for example, given an aggregation operator $B : \bigcup_{n\in\mathbb{N}} [0,1]^n \to [0,1]$, a bipolar aggregation operator $A : \bigcup_{n\in\mathbb{N}} [-1,1]^n \to [-1,1]$ could be defined as $A(\mathbf{x}) = B(\mathbf{x}^+) - B(\mathbf{x}^-)$ where $\mathbf{x}^+ = (\mathbf{x} \vee 0)$ and $\mathbf{x}^- = (-\mathbf{x}) \vee \mathbf{0}$.

Now, we can use the ideas presented in subsection 3.1 to obtain a bipolar aggregation operator from two (bipolar) OWA operators. Observe that if we use only one OWA operator B as generator, both construction method result in the same bipolar aggregation operator A; i.e., our method is a more general construction method of bipolar aggregation operators obtained via OWA operators.

Finally, and following the ideas seen in previous subsection, a bipolar aggregation operator obtained from two OWA operators will be an OWA operator itself if and only if original OWA operators are reverse one with respect to the other.

Proposition 3 *Given B_1, B_2 bipolar OWA operators and $\alpha \in [-1,1]$, the aggregation operator $A(\mathbf{x}) = B_1(\mathbf{x}^+) - B_2(\mathbf{x}^-) + \alpha$ is a bipolar OWA operator if and only if $B_2 = B_1^r$, and in this situation $A = B_1$.*

4 OWA operators related to double aggregation operators

A new aggregation technique has recently been introduced in [5]. The corresponding aggregation operators deal with input vectors distinguishing between the sources of data to be aggregated. This approach can be appropriate when it is necessary to obtain a global result. Take, for example, a global evaluation of a product which is evaluated from several technical aspects and from several aesthetic aspects, where the parameters of each type of evaluating aspects are independent. Even in the case of applying simple arithmetic means on each step of aggregation, the final global evaluation differs from global arithmetic mean. To model aggregation problems like the previous one, a special class of aggregation operators, called *double aggregation operators*, was defined.

Definition 2 Let $F, G : \bigcup_{n\in\mathbb{N}} [0,1]^n \to [0,1]$ be two aggregation operators and $H : [0,1]^2 \to [0,1]$ a binary aggregation operator. A *double aggregation* operator is a function

$$A = A_{F,G,H} : \bigcup_{(n,m)\in\mathbb{N}^{[2]}} [0,1]^n \times [0,1]^m \to [0,1]$$

with $\mathbb{N}^{[2]} = \mathbb{N}^2 \cup (\{0\} \times \mathbb{N}) \cup (\mathbb{N} \times \{0\})$, defined by:

$A_{F,G,H}(\mathbf{x}_{(n)}, \mathbf{y}_{(m)}) =$

$$= \begin{cases} H(F(\mathbf{x}_{(n)}), G(\mathbf{y}_{(m)})), & \text{if } n, m \in \mathbb{N} \\ F(\mathbf{x}_{(n)}), & \text{if } m = 0 \\ G(\mathbf{y}_{(m)}), & \text{if } n = 0 \end{cases}$$

for each $\mathbf{x}_{(n)} = (x_1, \ldots, x_n) \in [0,1]^n$, $\mathbf{y}_{(m)} = (y_1, \ldots, y_m) \in [0,1]^m$.

Observe that, for any double aggregation operator A, the boundary values 0 and 1 are idempotent elements.

Now, recall that the following relation $\leq_{2\pi}$ on the set $\mathcal{D} = \bigcup_{(n,m)\in\mathbb{N}^{[2]}} [0,1]^n \times [0,1]^m$ is a *partial ordering.*

Definition 3 Let $\mathbf{z} = (\mathbf{x}_{(n)}, \mathbf{y}_{(m)})$, $\mathbf{z}' = (\mathbf{x}'_{(r)}, \mathbf{y}'_{(s)})$ be elements of $\mathcal{D}$. The relation $\leq_{2\pi}$ is defined as follows: $\mathbf{z} \leq_{2\pi} \mathbf{z}'$ if and only if

- $(n = r = 0, \mathbf{y}_{(m)} \leq_\pi \mathbf{y}'_{(s)})$, or
- $(m = s = 0, \mathbf{x}_{(n)} \leq_\pi \mathbf{x}'_{(r)})$, or
- $(n, m, r, s \in \mathbb{N}, \mathbf{x}_{(n)} \leq_\pi \mathbf{x}'_{(r)}, \mathbf{y}_{(m)} \leq_\pi \mathbf{y}'_{(s)})$.

This allows us to consider the *monotonicity* for a double aggregation operator, as follows: A will be said to be *monotone* if for any $\mathbf{z}, \mathbf{z}' \in \mathcal{D}$, $\mathbf{z} \leq_{2\pi} \mathbf{z}' \Rightarrow A(\mathbf{z}) \leq A(\mathbf{z}')$. Furthermore, it is clear that any double aggregation operator will be monotone according to the former definition.

As we have mentioned, OWA operators [19] are just symmetric Choquet integral based aggregation operators [11, 9], i.e., symmetric comonotone additive aggregation operators. Now, we want to study when a double aggregation operator has both symmetry and comonotone additivity properties.

Theorem 1 *Let $A = A_{F,G,H}$ be a double aggregation operator. Then A is a symmetric comonotone additive double aggregation operator if and only if H is a weighted mean, $H(u,v) = ku + (1-k)v$ for some $k \in [0,1]$, and F and G are Choquet integral based aggregation operators related to some fuzzy measures systems $(\mu_n^F)_{n\in\mathbb{N}}$ and $(\mu_n^G)_{n\in\mathbb{N}}$ respectively, where μ_n^F and μ_n^G are fuzzy measures defined on $X_n = \{1, \ldots, n\}$.*

To prove this theorem it is enough to consider the cases $m = 0$, $n = 0$ and $n = m = 1$ to obtain, respectively, the condition of operators F, G and H.

Following this characterization we can define a *double OWA operator* as a symmetric comonotone additive double aggregation operator.

Due to Theorem 1, each double OWA operator A can be represented in the form $(n, m \in \mathbb{N})$, $A((x_1, \ldots, x_n), (y_1, \ldots, y_m)) = kF(x_1, \ldots, x_n) + (1-k)G(y_1, \ldots, y_m)$, where both F and G are OWA operators.

Example 4 *If we take $k = \frac{1}{2}$, $F = \min$, $G = \max$, then*

$$A(\mathbf{x}_{(n)}, \mathbf{y}_{(m)}) = \begin{cases} \min x_i, & \text{if } m = 0 \\ \max y_j, & \text{if } n = 0 \\ \frac{\min x_i + \max y_j}{2}, & \text{otherwise.} \end{cases}$$

Observe that a double OWA operator coincides with the OWA defined by means of a Choquet integral to 2-symmetric measures (as seen in [15]) and to input vector with 0,1-coordinates.

5 Conclusions

We have presented and discussed new construction methods to obtain OWA based aggregation operators; some of them give a new point of view in well–known methods as Šipoš integral while other present an extension to recently described methods as bipolar aggregation operators. Also, a new class of OWA operators are defined to deal with inputs related with different (or independent) sources; double OWA operators provide a useful tool for this kind of problems without any loss of their well–known classic properties.

Another interesting problem is related to the definition of $\mathbf{x}^+$ and $\mathbf{x}^-$ as multiple choices may lead to the same operators. Special cases arise when $\alpha = \frac{a+b}{2}$ and will be studied in our future work.

6 Acknowledgements

This work is supported by the projects BFN2003–05308, COST 274 Tarski and TIC2002-11942-E.

References

[1] G. Beliakov. Approximation of aggregation operators using splines. *Proc. IPMU'2000*, Madrid, 2000, pp. 680–685.

[2] P. Benvenuti and R. Mesiar. "Integrals with respect to a general fuzzy measure". In M. Grabisch, T. Murofushi and M. Sugeno, eds. *Fuzzy*

Measures and Integrals. Theory and Applications. Physica- Verlag, Heidelberg, 2000, pp. 205–232.

[3] B. Bouchon–Meunier, editor. *Aggregation and Fusion of Imperfect Information*, Physica–Verlag, Heidelberg, 1998.

[4] T. Calvo, A. Kolesárová, M. Komorníková and R. Mesiar. "Aggregation Operators: Properties, Classes and Construction Methods". In T. Calvo, G. Mayor and R. Mesiar, eds. *Aggregation Operators*. Physica–Verlag, Heidelberg, pp. 3–107, 2002.

[5] T. Calvo and A. Pradera. Double aggregation operators, *Fuzzy Sets and Systems* **142** (2004) 15–33.

[6] F. Chiclana, F. Herrera and F. Herrera–Viedma. The ordered weighted geometric operator: properties and applications. *Proc. IPMU'2000*, Madrid, 2000, Vol. **II**, pp. 985–981.

[7] G. Choquet. Theory of capacities. *Ann. Inst. Fourier* **5** (1953-54) 131–295.

[8] D. Dubois and H. Prade. A review of fuzzy set aggregation connectives. *Inform. Sci.* **36** (1985), 85–121.

[9] J. Fodor and M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers, Dordrecht, 1994.

[10] J.C. Fodor, J.–L. Marichal and M. Roubens: Characterization of the ordered weighted averaging operators. *IEEE Transactions on Fuzzy Systems* **3** (1995) 236–240.

[11] M. Grabisch. Fuzzy integral in multicriteria decision making. *Fuzzy Sets and Systems* **69** (1995) 279–298.

[12] G.J. Klir and T.A. Folger. *Fuzzy Sets, Uncertainty, and Information*. Prentice Hall, Englewood Cliffs, 1988.

[13] J.–L. Marichal. An axiomatic approach of the discrete Choquet integral as a tool to aggregate interacting criteria. *IEEE Transactions on Fuzzy Systems* **8** (2000) 800–807.

[14] A. Mesiarová, J. Lázaro and T. Calvo. Bipolar Aggregation Operators. *Proc. Agop'2003*, Alcalá-Spain, 2003, pp. 119–122.

[15] P. Miranda and M. Grabisch. p-symmetric bicapacities. *Proc. Agop'2003*, Alcalá-Spain, 2003, pp. 123–128.

[16] C. Moraga. "Neuro–fuzzy modeling of compensating systems". In P. Sincak and J. Vascak, eds., *Quo Vadis Computational Intelligence*. Physica-Verlag, Heidelberg, 2000, 385–398.

[17] E. Tsiporkova and H.-J. Zimmermann. Aggregation of compatibility and equality: a new class of similarity measures for fuzzy sets, *Proc. IPMU'98*, Paris, 1998, Vol. **II**, pp. 1769–1776.

[18] A. Tversky and D. Kahneman. Advances in Prospect Theory: Cumulative Representation of Uncertainty, *Journal of Risk and Uncertainty* **5** (1992) 297-323.

[19] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. Syst., Man and Cybern.*. **18** (1988) 183–190.

[20] R.R. Yager and J. Kacprzyk. *The Ordered Weighted Averaging Operators, Theory and applications*. Kluwer Academic Publishers, Boston, Dordrecht, London, 1997.

[21] R.R. Yager and D.P. Filev. *Essentials of Fuzzy Modelling and Control*. J. Wiley & Sons, New York, 1994.

[22] H.-J. Zimmermann and P. Zysno. Latent connectives in human decision making, *Fuzzy Sets and Systems* **4** (1980) 37–51.

VERIFICACIÓN DE ALGORITMOS DE CLASIFICACIÓN CON PRECONDICIÓN DIFUSA

Victoria López
Dept. Estadística e I.O. I
Universidad Complutense
vlopez@mat.ucm.es

Daniel Gómez
Dept. Estadística e I.O. III
Universidad Complutense
dagomez@estad.ucm.es

Javier Montero
Dept. Estadística e I.O. I
Universidad Complutense
javier_montero@mat.ucm.es

Resumen

En este artículo abordaremos el problema de la verificación formal [7, 13] de un algoritmo con precondiciones y entradas difusas. Se tratará la verificación mediante la semántica axiomática realizando transformaciones de predicados difusos. Para ello comenzaremos presentando la especificación formal de algoritmos nítidos que ampliaremos al contexto difuso para dar paso a la metodología de verificación formal nítida y difusa. Presentaremos un problema de decisión-clasificación para ejemplificar el objetivo de este artículo.

Palabras Clave: Verificación, Especificación, Algoritmos Difusos.

1 INTRODUCCIÓN

En este trabajo se presenta la verficación de algoritmos de clasificación difusa de imágenes digitales obtenidas vía satélite (teledetección).

Actualmente la clasificación multiespectral puede realizarse utilizando una gran variedad de algoritmos, que se pueden clasificar en dos grandes grupos: clasificación nítida y clasificación utilizando conjuntos difusos.

En la **clasificación supervisada**, la identidad, localización y características de algunos tipos de terreno son conocidos a priori, mediante análisis de fotografías aéreas, mapas y experiencia personal. El analista intenta encontrar lugares específicos en los datos que representen ejemplos de estos tipos de terreno conocidos. Estas áreas se conocen como *training sites*, ya que las características de estos tipos de area se suelen determinar mediante algoritmos de aprendizaje. Cada pixel es evaluado y será asignado a la clase a la cual tiene mayor probabilidad de pertenecer (en el contexto difuso se obtendrán funciones de pertenencia para cada una de las clases).

En la **clasificación no supervisada**, las clases, así como otras características, no son conocidas a priori. En estos casos se ha de agrupar los pixels con características similares dentro de un cluster de acuerdo con algún criterio determinado (ver Amo et al [1, 2]). El algoritmo que se evalúa en este trabajo es proceso de una larga investigación en clasificación no supervisada para la determinación de regiones homogéneas y posterior clasificación difusa (ver [1, 2, 3, 4]).

Uno de los principales problemas que surge de forma natural en cualquier algoritmo de clasificación no supervisado (tanto en el caso nítido como en el difuso) es el de la validación del proceso de clasificación. En la mayoría de las ocasiones este proceso de evaluación es hecho de forma empírica por expertos que deciden si la mayoría de las clasificaciones que se realizan son adecuadas o no lo son.

Sin embargo este método de evaluación tiene ciertos problemas:

- En algunos campos (problemas de optimización de rutas) existen problemas test para determinar las bondades de un algoritmo, esto puede hacerse porque tienen una función objetivo claramente definida. En problemas de clasificación de imágenes vía satélite no existe esta colección de problemas test y por tanto hace más difícil una valoración desde un punto de vista empírico.
- Frente a los mismos resultados pueden existir interpretaciones y validaciones distintas (ver [8]) hechas por diferentes expertos (lo que dificulta la valoración del algoritmo por esta vía).

Por este motivo existe una necesidad de realizar una

validación del algoritmo de clasificación que tenga en cuenta el proceso y no únicamente el resultado final. En el caso difuso la verficación es más flexible y podría entenderse como un análisis del proceso desarrollado.

Este trabajo está dividido de la siguiente forma: en la sección 2 se introduce la especificación formal de algoritmos en el caso nítido y difuso. En la sección 3 se ve la verificación formal para el caso iterativo y se analiza el proceso de verificación de algoritmos difusos. En la sección 4 se aborda el problema de la especificación y diseño de un algoritmo de clasificación-decisión propuesto por los autores en trabajos anteriores [1, 2, 3, 4, 5] y se verifica en la sección 5. Finalmente en la sección 6 se presentan las conclusiones finales.

2 ESPECIFICACIÓN FORMAL DE ALGORITMOS

2.1 ESPECIFICACIÓN NÍTIDA

La especificación formal es la parte de la algorítmica que se ocupa de determinar, mediante aserciones lógicas, las condiciones de funcionamiento y el resultado que va a ser calculado por el algoritmo [11]. Consta de tres partes fundamentales:
Cabecera donde se especifica el nombre del subprograma, los datos parámetros con modo y tipo.
Precondición, predicado lógico que representa los requisitos que deben cumplir los datos de entrada para garantizar una ejecución correcta del algoritmo.
Postcondición, predicado lógico que se hace cierto tras una ejecución correcta del algoritmo, nos informa del valor o valores que tomarán los parámetros de salida en función de los datos de entrada.

2.2 ESPECIFICACIÓN DIFUSA

Los asertos en la especificación pueden no ser estrictamente ciertos o falsos, sino un grado de adecuación a las restricciones del algoritmo.

Ejemplo 2.1 *Se dice que una matriz de $n \times m$ números enteros es melchoriforme si posee algún elemento 'rubio'. Un elemento es 'rubio' si su valor coincide con la suma de los restantes elementos de la matriz.*
En su versión nítida, obtendríamos la suma de los elementos de la matriz y realizaríamos la búsqueda del elemento descrito. Sin embargo, puede interesarnos conocer si existe algún elemento que sin ser 'rubio' esté cerca de serlo. De esta forma proponemos la versión difusa del problema.

$$suma(m) = \sum_{i=1}^{n}\sum_{j=1}^{m} m[i,j]$$

$$rubio(i,j,m) \Leftrightarrow (suma(m) = 2m[i,j])$$

el predicado 'rdif' representa la versión difusa del predicado 'rubio':

$$rdif(i,j,m) \Leftrightarrow (suma(m) \cong 2m[i,j])$$

La especificación del algoritmo melchoriforme difuso resulta:

fun *Melchorif (m:* **array***[1..n,1..n])* **of** **int** *):*
ret *b:***bool***;*
$\{PRE_{dif}: Cierto\}$
$\{POST_{dif} : b \Leftrightarrow \exists(i,j) \in \{1..n\} \times \{1..m\} : rdif(i,j,m)\}$

donde la precondición Cierto indica que los datos de entrada no tienen que cumplir ninguna restricción. Por otra parte, la postcondición será cierta en cierto grado, es decir, el predicado $POST_{dif}$ será 1 si en la matriz existe un elemento (i,j) que cumpla el predicado nítido $rubio(i,j,m)$; $POST_{dif}$ será 0 todos los elementos de la matriz cumplen $rdif(i,j,m) = 0$; en otro caso $POST_{dif}$ tomará un valor en (0,1) que representará el grado de certeza en que la matriz dada cumple la propiedad melchoriforme, que debe corresponder con:

$$max_{1\leq i\leq n, 1\leq j\leq m}(suma(m) \cong 2m[i,j])$$

3 VERIFICACIÓN FORMAL

La verificación de un algoritmo consiste en demostrar su funcionamiento es válido y es el esperado en cualquier ejecución posible, es decir, el resultado (output) guarda con los datos de entrada (input) la relación establecida por la especificación del problema [6, 9, 12, 13].

3.1 ALGORITMOS ITERATIVOS. PARADIGMA ITERATIVO

Aplicaremos este proceso, sin pérdida de generalidad, únicamente a los algoritmos iterativos cuyo esqueleto se corresponda con el esquema de la figura 1. En esta figura denotamos:
'Inic;' instrucciones de inicialización.
'Cond' determina la permanencia en el proceso.
'SecInst' es la secuencia de instrucciones de una etapa en el proceso.

En el proceso de verificación [13] son precisos los siguientes elementos:

Figura 1: Esquema Iterativo

{PRE}
Inic;
{INV}
while Cond **do**
SecInst;
endwhile
{POST}

Figura 2: Esquema de verificacion en iterativos

Base de la inducción:
$\{PRE\}Inic\{INV\}$
Paso inductivo:
$\{INV \wedge Cond\}SecInst\{INV\}$
Consecución del objetivo:
$\{INV \wedge \neg Cond\} \rightarrow \{POST\}$
Definición de una función tamaño: $f \geq 0$
Decrecimiento estricto de f:
$f_k > f_{k+nv} : nv = nv(k) \in nat \forall k \in nat$

- Derivación de la condición invariante a partir de la postcondición.
- Prueba inductiva: A partir de la precondición, la ejecución del algoritmo, conduce a la postcondición.
- Tamaño del problema: Todos los bucles del algoritmo se ejecutan un número finito de veces.

La figura 2 es una descripción de las pruebas a realizar para la consecución de la verificación de un bucle.

4 DISEÑO Y ESPECIFICACIÓN DE UN ALGORITMO DE CLASIFICACIÓN-DECISIÓN

El problema es el de decidir cuando la clasificación difusa de una imagen de píxeles es correcta, al menos en grado suficiente, de acuerdo con la opinión de un experto. Una propuesta a la especificación del algoritmo asociado es:

proc DecideyClasifica (**in** *im*: Timagen;
out *imc*:Tlista **of** Timagen; **out** *k*: **int**);
{PRE: *im* cumple las características de imagen }
{POST: *imc* es una lista de matrices resultado de la clasificación suficiente de la imagen *im*. La variable *k* devuelve la longitud de la lista *imc*}

4.1 DISEÑO DE UN ALGORITMO DE CLASIFICACIÓN-DECISIÓN

Definimos un algoritmo difuso como aquél cuya especificación contiene algún predicado vago. Habitualmente encontraremos este tipo de predicados en la parte de precondicionamiento o postcondicionamiento del problema.

El siguiente algoritmo utiliza la opinión de dos expertos independientes para realizar la valoración del proceso. El algoritmo recibe una imagen y los niveles de confianza exigibles frente a la valoración de los expertos. El algoritmo devuelve una lista de matrices cuyos elementos son valores en el intervalo [0,1] y que representa la clasificación de la imagen así como un valor k donde se almacena la longitud de dicha lista, o lo que es lo mismo, el número de clases resultante.

El esquema de nuestro primer algoritmo representante de la primera parte del proceso es el siguiente: En primer lugar realizamos una llamada al proceso $ClasNit$ que produce una primera clasificación nítida de la imagen y le preguntaremos a un experto –el experto interno al proceso– su opinión sobre dicha clasificación.

La consulta al experto se realiza mediante el algoritmo *Decide* que devuelve una variable booleana *vale* a ***Cierto*** si el experto opina que la clasificación recibida es válida, lo que el experto decide tras comparar su propia valoración del algoritmo (α_{res}) con el valor propuesto (α_{int}) por el cliente. En ese caso, se entiende que el experto está de acuerdo con la clasificación realizada, por lo que acepta que el número de clases obtenido en la clasificación anterior sea correcto. En caso contrario, el experto devuelve *vale* a ***Falso*** y propone un número de clases para provocar una nueva clasificación con este parámetro prefijado. Por ello decidimos aumentar el número de parámetros de entrada del proceso añadiendo α_{int}.

Mientras el proceso no alcance la satisfacción del experto interno (*vale* sea ***Falso***)9se realizarán clasificaciones difusas que el experto valorará.

proc $DecideyClasifica$ (**in** *im*: Timagen;
in α_{int} : $[0, 1]$;
out *imc*:Tlista **of** Timagen;
out k, k_0: **int**;
out α_{res}: [0,1]);
{PRE $_{DecideyClasifica}$}
ClasNit(im,imc,k);
$k_0 := k$;
Decide(im, imc, k, nk, vale,α_{res}*)*;
while$\neg vale$ **do**
$ClasDif(im, nk, imc, k)$;
$Decide(im, imc, k, \alpha_{int}, nk, vale, \alpha_{res})$;
endwhile

{POST $_{DecideyClasifica}$}
endproc;

El algoritmo padre es:

proc *ProcesoCompleto* (**in** *im*: Timagen;
in $\alpha_{ext}, \alpha_{int}$: [0, 1];
out *imc*:Tlista **of** Timagen;
out $\alpha_{res}, \alpha'_{res}$: [0,1];
out k_0, k: **int**);
{PRE $_{ProcesoCompleto}$}
DecideyClasifica(im, α_{int}*,imc,k,*α_{res}*)*;
Decide2(im, imc, k,α_{res}*, nk, vale,*α'_{res}*)*;
{POST$_{ProcesoCompleto}$}
endproc;

Los valores α_{res} y α'_{res} son booleanos en el contexto difuso, por lo que su rango de posibilidades no sólo serán ***Cierto*** o ***Falso*** sino que será un rango del intervalo real [0,1].

A este tipo de rango deberán pertenecer también las variables α_{int} y α'_{ext}. Para generalizar todos los casos supondremos que estas variables toman valores en el intervalo [0,1] donde establecemos la imagen de las siguientes funciones de valoración que definimos a continuación para distinguir la actuación de los expertos:

$\diamond$ $Valor_{int}(imc, im)$ = valor que el experto interno asigna a la clasificación *imc* de la imagen im $(= \alpha_{res})$, número de clases propuesto para la siguiente clasificación

$\diamond$ $Valor_{ext}(imc, im)$ = valor que el auditor, –experto externo–, asigna a la clasificación *imc* de la imagen $im(= \alpha'_{res})$

$\diamond$ $Valor_{final}(imc, im, k_0, \alpha_{res}, \alpha'_{res})$ = función estimadora que produce una valoración de la clasificación teniendo en cuenta además el valor producido por la clasificación nítida: k_0, la última valoración realizada por el experto interno: α_{res} y la valoración realizada por el experto externo: α'_{res}

4.2 ESPECIFICACIÓN FORMAL DE *DecideyClasifica*

La precondición de este procedimiento deberá exigir que la imagen de entrada im cumpla las características de una imagen de píxeles. Para simplificar usaremos el predicado lógico $EsImagen(im)$.

La postcondición de *DecideyClasifica* debe indicar que el parámetro de salida *imc* contiene una clasificación difusa con valoración suficiente. De esta forma la postcondición de *DecideyClasifica* resulta:

$$\{POST : Valor_{int}(imc, im) \succeq \alpha_{int} \wedge k = long(imc)\}$$

donde gracias a la condición de generalidad de los predicados difusos sobre los nítidos podemos observar este predicado como postcondición válida cuando no se ejecuta el bucle –clasificación nítida– como cuando se ejecuta el bucle –clasificación difusa– .

4.3 ESPECIFICACIÓN FORMAL DE *ProcesoCompleto*

La precondición del proceso completo coincide básicamente con la precondición de *DecideyClasifica*, basta con añadir una cláusula más para el parámetro α_{ext} obteniéndose:

$$\{PRE : EsImagen(im) \wedge (0 \leq \alpha_{int}, \alpha_{ext} \leq 1)\}$$

En cuanto a la postcondición de *ProcesoCompleto* podemos establecer la siguiente:

$$\{POST : (\alpha_{res} \succeq \alpha_{int}) \wedge_d (\alpha'_{res} \succeq \alpha_{ext})$$

$$\wedge_d (f(im, imc, k_0, k, \alpha_{res}, \alpha'_{res}) \succeq g(\alpha_{int}, \alpha_{ext})\}$$

donde la primera cláusula indica que el experto interno admite como suficiente la clasificación de la imagen respecto al valor α_{int} prefijado por el cliente. De forma análoga la segunda cláusula indica que el experto externo admite como suficiente la clasificación de la imagen respecto al valor α_{ext} prefijado de antemano también por el cliente. La tercera cláusula comprueba una preferencia entre la valoración del proceso completo calculada mediante una función f que tendrá en cuenta los datos más relevantes indicados en sus argumentos y una función g de los niveles de confianza α_{int} y α_{ext}. Estas funciónes f y g pueden definirse como se desee en cada caso concreto y su comparación determinará junto con la primera cláusula un predicado difuso que nos dará información sobre la bondad del proceso completo de clasificación-decisión llevado a cabo.
La conjunción difusa de estas tres cláusulas conformarán el grado de adecuación del resultado del proceso a nuestro objetivo.

4.3.1 Especificación de los subprogramas

♣ Proceso de clasificación nítida
El procedimiento *ClasNit* tiene la siguiente especificación formal asociada:

proc *ClasNit* (**in** *im*: Timagen;
out *imc*:Tlista **of** Timagen; **out** k: **int**);
$\{PRE_{ClasNit} : EsImagen(im)\}$
$\{POST_{ClasNit} : EsCN(imc, im) \wedge (long(imc) = k) \wedge k_0 = k\}$

donde los predicados lógicos *EsImagen* y *EsCN* se pueden definir en lenguaje natural como:
$EsImagen(im) \equiv im$ cumple las condiciones de imagen digital

$EsCN(imc, im) \equiv imc$ es una clasificación nítida de los píxeles de la imagen im.
Este predicado se define formalmente por:

$$EsCN(imc, im) \equiv$$
$$\forall Mat \in imc[\forall i \in \{1..N\} \forall j \in \{1..M\}$$
$$(Mat[i,j] = 0 \vee Mat[i,j] = 1)] \wedge$$
$$\forall i \in \{1..N\} \forall j \in \{1..M\} (\sum_{Mat \in imc} Mat[i,j] = 1)$$

que corresponde con la definición clásica de partición.

♣ Proceso de clasificación difusa
De forma muy parecida podemos especificar el procedimiento de clasificación difusa:

proc $ClasDif$ (**in** im: Timagen;
in k: **int**;
in/out imc:Tlista **of** Timagen;
out nk: **int**);
$\{PRE_{ClasDif} : EsImagen(im) \wedge (EsCN(imc, im) \vee EsCD(imc, im)) \wedge k = long(imc) \wedge imc = IMC\}$
$\{POST_{ClasDif} : EsCD(imc, im) \wedge (long(imc) = nk)\}$

donde se aparece un nuevo predicado $EsCD$ definido informalmente por

$EsCD(imc, im) \equiv imc$ es una clasificación difusa de los píxeles de la imagen im.
Y formalmente:

$$EsCD(imc, im) \equiv \forall p \in im, \sum_{C \in imc} \mu_C(p) = 1$$

donde $p = (i, j)$ representa un píxel de la imagen im y $\mu_C(p) = C[i, j]$ es el valor de pertenencia del píxel p a la clase C que será una matriz de la clasificación imc

♣ Consultas a los expertos

En cuanto a los procedimientos de consulta a los expertos $Decide$ y $Decide2$ su especificación formal es la siguiente:

proc $Decide$ (**in** im: Timagen;
in imc:Tlista **of** Timagen;
in k: **int**;
in α_{int}: [0,1];
out nk: **int**;
out $vale$: **bool**);
out α_{res}:[0,1];
$\{PRE_{Decide} : EsImagen(im) \wedge (EsCN(imc, im) \vee EsCD(imc, im)) \wedge k = long(imc)\}$
$\{POST_{Decide} : \alpha_{res} = Valor_{int}(imc, im) \wedge$
$vale \Leftrightarrow (nk = k \wedge \alpha_{res} \succeq \alpha_{int})\}$

proc $Decide2$ (**in** im: Timagen;
in imc:Tlista **of** Timagen;
in k: **int**;
in α_{ext}: [0,1];
out $vale$: **bool**;
out α'_{res}:[0,1];);
$\{PRE_{Decide2} : EsImagen(im) \wedge (EsCN(imc, im) \vee EsCD(imc, im)) \wedge k = long(imc)\}$
$\{POST_{Decide2} : \alpha'_{res} = Valor_{ext}(imc, im) \wedge$
$vale \Leftrightarrow (nk = k \wedge \alpha'_{res} \succeq \alpha_{ext})\}$

La verificación formal introducirá el cálculo de predicados difuso para evaluar el factor de certeza de cada aserto.

5 Verificación del algoritmo de decisión-clasificación

Uniendo la clasificación difusa con la participación del experto externo se obtiene el proceso completo:
proc $ProcesoCompleto$ (**in** im: Timagen;
in $\alpha_{int}, \alpha_{ext}$:[0,1];
out imc:Tlista **of** Timagen;
out $\alpha_{res}, \alpha'_{res}$: **[0,1]**;
out k_0, k: **int**);
$\{\text{PRE}_{ProcesoCompleto}\} \Rightarrow \{\text{PRE}_{ClasNit}\}$
ClasNit(im,imc,k);
$k_0 := k$;
$\{POST_{ClasNit}\} \Rightarrow \{\text{PRE}_{Decide}\}$
Decide(im, imc, k, α_{int}*, nk, vale,* α_{res}*)*;
$\{\text{POST}_{Decide}\} \Rightarrow \{\text{INV: ... }\}$
while $\neg vale$ **do**
$\{\text{INV} \wedge \neg vale\} \Rightarrow \{\text{PRE}_{ClasDif}\}$
$ClasDif(im, nk, imc, k)$
$\{\text{POST }_{ClasDif}\} \Rightarrow \{\text{wp(INV)}\} \Rightarrow \{\text{PRE}_{Decide}\}$
$Decide(im, imc, nk, \alpha_{int}, k, vale, \alpha_{res})$
$\{\text{POST}_{Decide}\} \Rightarrow \{\text{INV: ...}\}$
endwhile
$\{\text{POST}_{DecideyClasifica}\} \Rightarrow \{\text{PRE}_{Decide2}\}$
Decide2(im, imc, k, $\alpha_{ext}, vale, \alpha_{res}$*)*;
$\{\text{POST}_{Decide2}\} \Rightarrow \{\text{POST }_{ProcesoCompleto}\}$
endproc;

Un invariante adecuado es:

$\{\text{INV: } (\alpha_{res}, nk) = Valor_{int}(imc, im) \wedge_d vale \Leftrightarrow (\alpha_{res} \succeq \alpha_{int}) \wedge_d vale \Leftrightarrow (k = nk)\}$

Se cumple trivialmente tras las instrucciones de inicialización del algoritmo. Para comprobarlo basta con comparar la postcondición del proceso $Decide$ tras su primera ejecución:
$\{\text{POST}_{Decide}\} \Rightarrow \{\text{INV: ... }\}$
lo que constituye la base de la inducción que fundamenta la corrección.

En cuanto al paso inductivo, supondremos como hipótesis

$$\{INV \wedge \neg vale\}$$

por lo que se produce una entrada en el bucle y como consecuencia la ejecución de una nueva clasificación y una nueva consulta al experto interno. Tras lo cual puede haberse perdido la invarianza pero la llamada al proceso *Decide* de consulta al experto restablecerá el valor de las variables.

En cuanto a la consecución del objetivo del bucle, es trivial observar que:

$$\{INV \wedge vale\} \Rightarrow POST_{DecideyClasifica}$$

con lo que se obtiene la corrección completa de la primera parte del proceso, quedando únicamente pendiente la corrección de la secuencia de código relativa a la consulta al experto externo, esto es:

$\{\text{POST}_{DecideyClasifica}\} \Rightarrow \{\text{PRE}_{Decide2}\}$
Decide2(im, imc, k, $\alpha_{ext}, \alpha_{res}$*, vale)*;
$\{\text{POST}_{Decide2}\Rightarrow$ $\{\text{POST}_{\ ProcesoCompleto}\}$

La postcondición del proceso completo, está constituida por tres cláusulas difusas que a su vez se conectan mediante el operador binario difuso $f_{\wedge} \equiv \wedge_d$ generalizado expuesto en la sección 3.2. de este artículo. De esta forma la postcondición del proceso completo se cumplirá en cierto grado y dicho grado a su vez puede servir como valoración final del proceso de clasificación-decisión desarrollado a lo largo de nuestro artículo:

$$Valor_{final}(imc, im, k_0, \alpha_{res}, \alpha'_{res}) =$$

$$(\alpha_{res} \succeq \alpha_{int}) \wedge_d (\alpha'_{res} \succeq \alpha_{ext})$$

$$\wedge_d (f(im, imc, k_0, k, \alpha_{res}, \alpha'_{res}) \succeq g(\alpha_{int}, \alpha_{ext})\}$$

6 Conclusiones

Los algoritmos de clasificación-decisión difusos pueden ser debidamente diseñados, justificados y valorados mediante un método formal que detecte problemas y su posible solución generando diseños óptimos para procesos concretos. De esta forma podremos medir la bondad del algoritmo y establecer así un orden en el conjunto de algoritmos difusos asociados a un problema.

Referencias

[1] A. Amo, J. Montero y G. Biging: *Relevancia y redundancia en sistemas de clasificación difusa.* En: Proceedings Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF 2000 (A. Ollero, S. Sánchez, B. Arrue and I. Baturone, eds., Universidad de Sevilla, Sevilla); pp. 521-525 (2000).

[2] A. Amo, D. Gómez, J. Montero and G. Biging: Relevance and redundancy in fuzzy classification systems. Mathware and Soft Computing 8, 203-216 (2001)

[3] A.Amo, D. Gómez, J. Montero: Spectral fuzzy classification: a supervised approach. Mathware and Soft Computing 10:141-154 (2003)

[4] A.Amo, J. Montero, G. Biging y V. Cutello: Fuzzy classification systems. EUR. JOUR. OPER. RES. (to appear)

[5] A.Amo, J. Montero y V. Cutello: *On the principles of fuzzy classification.* En: Proceedings North American Fuzzy Information Processing Society Conference; 675–679 (1999).

[6] H.K. Berg y otros. Formal Methods of Program Verification and Specification. *Ed. Prentice Hall*, New Jersey, 1982.

[7] M. Broy y B. Krieg. Derivation Of Invariant Assertions During Program Development by Transformation. *ACM Transactions on Programming Languages and Systems*, Vol. 2, N 3, 321–327 1980.

[8] D. Gómez. *Algunas aportaciones sobre relaciones de preferencia.* Tesis doctoral, Universidad Complutense de Madrid, 2003.

[9] D. Gries. An Introduction to Current Ideas on the Derivation of Correctness Proofs and Correct Programs. *IEEE Software Engineering*, Vol. 2, N 4, 1976.

[10] J. R. Jensen: *Introductory digital image processing.* 1986 Prentice-Hall.

[11] A. Kaldewaij *Programming: The Derivation of Algorithms*, Ed.:Prentice Hall International, 1990.

[12] Z. Manna. The Correctness of Programs. *Journal of Computer and System Sciences*, Vol. 3, N 2, 1969.

[13] R. Peña Marí, *Diseño de Programas, Formalismo y Abstracción*, Ed.: Prentice Hall, 1999

An IOWA Operator Based on Additive Consistency for Group Decision Making

S. Alonso
Dept. of Computer Science and A.I, Granada Univ., salonso@decsai.ugr.es

F. Chiclana
CCI, De Montfort Univ. Leicester, UK. chiclana@dmu.ac.uk

F. Herrera
Dept. of Computer Science and A.I, Granada Univ., herrera@decsai.ugr.es

E. Herrera-Viedma
Dept. of Computer Science and A.I, Granada Univ., viedma@decsai.ugr.es

Abstract

In group decision-making, experts' preferences are usually expressed by means of fuzzy preference relations. Because not all experts may be capable of maintaining consistency between all the possible pair of feasible options of the problem, it is worthwhile evaluating the degree of inconsistency of a fuzzy preference relation so that more importance can be given to the more consistent experts. To do this, a new consistency measure of a fuzzy preference relation, based on the additive consistency property, is proposed. This consistency measure is also used to propose an IOWA operator to aggregate experts' fuzzy preference relations into a collective one in such a way that high weighting values are associated to those experts with high consistency degrees.

Keywords: Additive consistency, IOWA operator, Group decision-making, majority.

1 Introduction

Decision-making is a process of selecting the best alternative from a feasible for the purpose of attaining a goal or goals. Most major decisions are made by group that may include people from different departments or from different organizations [1, 3]. Thus, many different representation formats can be used to express preferences: i) preference ordering of the alternatives, ii) utility functions, iii) fuzzy preference relations [1]. The latter has been widely used due to their effectiveness as a tool for modelling decision processes and, above all, its utility and easiness of use when we want to aggregate experts' preferences into group preferences [3, 4].

Due to the complexity of most decision-making problems, experts's preferences may not satisfy formal properties that fuzzy preference relations are assumed to verify. One of these properties, consistency, is associated with the *transitivity property*. It is obvious that consistent information, that is, information which does not imply any kind of contradiction, is more relevant than information that contains some contradictions.

A classical choice scheme for a GDM problem follows two steps before it achieves a final decision [1]: *aggregation* and *exploitation*. The aggregation step of a GDM problem consists of combining the experts' individual preferences into a group collective one in such a way that it summarizes or reflects the properties contained in all the individual preferences. The exploitation phase transforms the global information about the alternatives into a global ranking of them. This can be done in different ways, the most common one being the use of a ranking method to obtain a score function.

In this paper, we will focus on the aggregation phase of the resolution scheme for a GDM problem, where the information provided is expressed in terms of fuzzy preference relations which are not supposed to be fully consistent. In these cases, more importance should be given to the experts that provide the more *consistent* information. To do this, we will develop a new Induced Ordered Weighted Averaging (IOWA) operator to aggregate the individual preferences relations that take into account their associated consistency degrees. We call this the Additive Consistency IOWA (AC-IOWA) operator.

The rest of paper is set out as follows: section 2 briefly presents the concept of fuzzy preference relation. Section 3 deals with the issue on consistency of fuzzy preference relations, and in particular focuses on the the additive consistency, and the introduction of a measure of consistency based on it. In section 4 the aggregation of preferences using OWA and IOWA operators is presented, an the AC-IOWA operator is defined. Sec-

tion 5 gives an example, and in section 6 we point out our concluding remarks and future research.

2 Fuzzy Preference Relations

In GDM problems the best alternative(s) among a finite set of feasible options, $X = \{x_1, ..., x_n\}$, $(n \geq 2)$, according to the preferences provide by a group of experts $E = \{e_1, \ldots, e_m\}$, has to be chosen. In such a decision situation, alternatives are usually pairwise compared, which makes the fuzzy preference relation the best representation format.

Definition 1. *[4] A fuzzy preference relation P on a set of alternatives X is a fuzzy set on the product set $X \times X$, that is characterized by a membership function*

$$\mu_P \colon X \times X \to [0, 1]$$

When cardinality of X is small, the preference relation may be conveniently represented by the $n \times n$ matrix $P = (p_{ij})$ being $p_{ij} = \mu_P(x_i, x_j) \quad \forall i, j \in \{1, \ldots, n\}$ interpreted as the preference degree or intensity of the alternative x_i over x_j: $p_{ij} = 1/2$ indicates indifference between x_i and x_j ($x_i \sim x_j$), $p_{ij} = 1$ indicates that x_i is absolutely preferred to x_j, and $p_{ij} > 1/2$ indicates that x_i is preferred to x_j ($x_i \succ x_j$). Based on this interpretation we have that $p_{ii} = 1/2 \quad \forall i \in \{1, \ldots, n\}$ ($x_i \sim x_i$).

As can be seen from the previous definition, an expert only needs to provide every p_{ij} value to efficiently express its criteria over the set of alternatives. However, as it will be shown in the next section, this information does not guarantee that certain "basic" properties that are usually assumed for fuzzy preference relations are satisfied.

3 On Consistency of Fuzzy Preference Relations

In terms of fuzzy preference relations, full consistency implies no contradiction between any preference values. When an expert has to express his preferences over a set of n alternatives, he has to provide n^2 preference values, and to maintain full consistency between all of them can be a difficult task.

3.1 Additive Consistency

Definition 1 does not imply any kind of consistency of a fuzzy preference relation. In fact, preferences expressed by an expert in a fuzzy preference relation can be contradictory. As studied in [2], for making a rational choice, a set of properties to be satisfied by such fuzzy preference relations have been suggested. One of these properties is *additive transitivity* [5]:

$$(p_{ij} - 0.5) + (p_{jk} - 0.5) = (p_{ik} - 0.5) \quad \forall i, j, k \in \{1, \ldots, n\} \tag{1}$$

or equivalently:

$$p_{ij} + p_{jk} + p_{ki} = 1.5 \quad \forall i, j, k \in \{1, \ldots, n\}. \tag{2}$$

This type of transitivity has the following interpretation: suppose we do want to establish a ranking between three alternatives x_i, x_j and x_k. If we do not have any information about these alternatives it is natural to start assuming that we are in an indifference situation, that is, $x_i \sim x_j \sim x_k$, and therefore when giving preferences this situation is represented by $p_{ij} = p_{jk} = p_{ik} = 0.5$. Suppose now that we have a piece of information that says alternative $x_i \prec x_j$, that is $p_{ij} < 0.5$. It is clear that p_{jk} or p_{ki} have to change, otherwise there would be a contradiction, because we would have $x_i \prec x_j \sim x_k \sim x_i$. If we suppose that $p_{jk} = 0.5$ then we have the situation: x_j is preferred to x_i and there is no difference in preferring x_j to x_k. We must conclude then that x_k has to be preferred to x_i. Furthermore, as $x_j \sim x_k$ then $p_{ji} = p_{ki}$, and so $p_{ij}+p_{jk}+p_{ki} = p_{ij}+p_{jk}+p_{ji} = 1+0.5 = 1.5$. We have the same conclusion if $p_{ki} = 0.5$. In the case of being $p_{jk} < 0.5$, then we have that x_k is preferred to x_j and this to x_i, so x_k should be preferred to x_i. On the other hand, the value p_{ki} has to be equal or greater than p_{ji}, being equal only in the case of $p_{jk} = 0.5$ as we have seen. Interpreting the value $p_{ji} - 0.5$ as the intensity of preference of alternative x_j over x_i, then it seems reasonable to suppose that the intensity of preference of x_k over x_i should be equal to the sum of the intensities of preferences when using and intermediate alternative x_j, that is, $p_{ki} - 0.5 = (p_{kj} - 0.5) + (p_{ji} - 0.5)$. The same reasoning can be applied in the case of $p_{jk} > 0.5$.

From the previous equations we obtain the following expression:

$$p_{ij} + p_{jk} - 0.5 = p_{ik} \quad \forall i, j, k \in \{1, \ldots, n\}. \tag{3}$$

A fuzzy preference relation is additive consistent when for every three options in the problem $x_i, x_j, x_k \in X$ their associated preference degrees p_{ij}, p_{jk}, p_{ik} satisfy (3). An additive consistent fuzzy preference relation will be referred as consistent throughout the paper, as this is the only transitivity property we are considering.

3.2 Consistency Measure

Additive consistency can be used to calculate the value of a preference degree p_{ik} using other preference degrees in a fuzzy preference relation. Indeed,

$$cp_{ik}^{j} = p_{ij} + p_{jk} - 0.5 \tag{4}$$

where cp_{ik}^{j} means the calculated value of p_{ik} via j, that is, using p_{ij} and p_{jk}. Obviously, if the information provided in a fuzzy preference relation is completely consistent then cp_{ik}^{j} $(\forall j)$ and p_{ik} coincide. However, the information given by an expert usually does not satisfy (3), because the information provided by an expert usually suffers from a certain degree of inconsistency. In these cases, the value

$$\varepsilon p_{ik} = \frac{\sum_{\substack{j=1 \\ j\neq i,k}}^{n} \left| cp_{ik}^{j} - p_{ik} \right|}{n-2} \tag{5}$$

can be used to measure the error expressed in a preference degree between two options. This error can be interpreted as the consistency level between the preference degree p_{ik} and the rest of the preference values of the fuzzy preference relation. Clearly, when $\varepsilon p_{ik} = 0$ then there is no inconsistency at all between p_{ij} and the other preference values, and the higher the value of εp_{ik} the more inconsistent is p_{ik} with respect to the rest of information.

The *consistency level* for the whole fuzzy preference relation P is defined as follows:

$$CL_P = \frac{\sum_{\substack{i,k=1 \\ i\neq k}}^{n} \varepsilon p_{ik}}{n^2 - n}. \tag{6}$$

When $CL_P = 0$, then the preference relation P is fully additive consistent, otherwise, the higher CL_P the more inconsistent is P.

4 A new IOWA Operator Based on Additive Consistency

The collective fuzzy preference relation, which indicates the global preference between all pairs of alternatives, is obtained by aggregating all the individual preferences. To do this, several different aggregation operators have been proposed, including the OWA [6] and IOWA [8, 9] operators.

4.1 OWA and IOWA operators

A fundamental aspect of the OWA operators is the reordering of the arguments to aggregate, based upon the magnitude of their respective values:

Definition 2. *[6] An OWA operator of dimension n is a function $\phi\colon \mathbb{R}^n \to \mathbb{R}$, that has associated to it a set of weights or weighting vector $W = (w_1, \ldots, w_n)$ such that, $w_i \in [0,1]$ and $\sum_{i=1}^{n} w_i = 1$, and is defined to aggregate a list of values $\{p_1, \ldots, p_n\}$ according to the following expression,*

$$\phi(p_1, \ldots, p_n) = \sum_{i=1}^{n} w_i \cdot p_{\sigma(i)}$$

being σ a permutation of $\{1, \ldots, n\}$ such that $p_{\sigma(i)}$ is the i-th highest value in the set $\{p_1, \ldots, p_n\}$.

In the process of quantifier guided aggregation, given a collection of n criteria represented as fuzzy subsets of the alternatives X, the OWA operator has been used to implement the concept of fuzzy majority in the aggregation phase by means of a *fuzzy linguistic quantifier* which indicates the proportion of criteria 'necessary for a good solution' [7]. This implementation is done by using the quantifier to calculate the OWA weights. When a fuzzy quantifier Q is used to compute the weights of the OWA operator ϕ, then it is symbolized by ϕ_Q.

In [7], Yager proposed a procedure to evaluate the overall satisfaction of Q important (u_k) criteria (experts) (e_k) by the alternative x. In this procedure, once the satisfaction values to be aggregated have been ordered, the weighting vector associated to an OWA operator using a linguistic quantifier Q are calculated following the expression

$$w_i = Q\left(\frac{\sum_{k=1}^{i} u_{\sigma(k)}}{T}\right) - Q\left(\frac{\sum_{k=1}^{i-1} u_{\sigma(k)}}{T}\right) \tag{7}$$

being $T = \sum_{k=1}^{n} u_k$ the total sum of importance, and σ the permutation used to produce the ordering of the values to be aggregated. This approach for the inclusion of importance degrees associates a zero weight to those expert with zero importance degree.

In the OWA aggregation the weights are not associated with the arguments bur with the order position of the arguments. However, as it happens with fuzzy preference relations, the values to be aggregated cannot be directly compared, or sometimes, a different order for the arguments is preferred to be used. To allow different orders in an OWA operator, Yager and Filev defined the Induced OWA operator as follows:

Definition 3. *[8, 9] An IOWA operator of dimension n is a function $\Phi_W\colon (\mathbb{R}\times\mathbb{R})^n \to \mathbb{R}$, to which a set of weights or weighting vector is associated, $W = (w_1, \ldots, w_n)$, such that $w_i \in [0,1]$ and $\Sigma_i w_i = 1$, and it is defined to aggregate the set of second arguments of a list of n 2-tuples $\{\langle u_1, p_1\rangle, \ldots, \langle u_n, p_n\rangle\}$ according to the following expression,*

$$\Phi_W\left(\langle u_1, p_1\rangle, \ldots, \langle u_n, p_n\rangle\right) = \sum_{i=1}^{n} w_i \cdot p_{\sigma(i)}$$

being σ a permutation of $\{1, \ldots, n\}$ such that $\langle u_{\sigma(i)}, p_{\sigma(i)}\rangle$ is the 2-tuple with $u_{\sigma(i)}$ the i-th highest value in the set $\{u_1, \ldots, u_n\}$.

In the above definition, the reordering of the set of values to be aggregated, $\{p_1, ..., p_n\}$, is induced by the reordering of the set of values $\{u_1, ..., u_n\}$ associated to them, which is based upon their magnitude. Due to this use of the set of values $\{u_1, ..., u_n\}$, Yager and Filev called them the values of an *order inducing variable* and $\{p_1, ..., p_n\}$ the values of the *argument variable* [9]. As we have mentioned, the main difference between the OWA and the IOWA operators resides on the reordering step of the argument variable. An immediate consequence of this definition is that if the order inducing variable is the argument variable then the IOWA operator is reduced to the OWA operator.

4.2 Additive Consistency IOWA operator

As said in section 3, consistent information should be given more importance in the aggregation process of a GDM problem. The general procedure for the inclusion of importance degrees in the aggregation process involves the transformation of the preference values, p_{ij}^k, under the importance degree u_k to generate a new value, $\bar{p}_{ij}^k$. This activity is carried out by means of a transformation function (t-norm operator) g, $\bar{p}_{ij}^k = g(p_{ij}^k, u_k)$.

In our case, we may as well implement the consistency degrees by an alternative approach, which consists of using them as the order inducing values of the IOWA operator. Indeed, the closer CL_P is to 0 the more consistent the information represented by P, and thus more importance should be placed on that information. In other words, we could use these values to define the ordering of the preferences to be aggregated, in which case we would be implementing the concept of consistency in the aggregation process of our decision-making. This kind of aggregation process defines an IOWA operator that we call the Additive Consistency IOWA (AC-IOWA) operator and denote it as Φ_W^{AC}.

Definition 4. *If a set of experts, $E = \{e_1, \ldots, e_m\}$, provides preferences about a set of alternatives, $X = \{x_1, \ldots, x_n\}$, by means of the fuzzy preference relations, $\{P^1, \ldots, P^m\}$, then the AC-IOWA operator of dimension n, Φ_W^{AC}, is an IOWA operator whose set of order inducing values is $\{1 - CL_{P^1}, ..., 1 - CL_{P^m}\}$.*

The application of the AC-IOWA operator, with an appropriate RIM quantifier Q function to obtain the weighting vector, will associate more important weight in the aggregation to the most consistent experts. This is illustrated in the next section.

5 Example

Suppose that three experts e_1, e_2 and e_3 provide the following fuzzy preference relations over a set of five alternatives $\{x_1, ..., x_5\}$:

$$P^1 = \begin{pmatrix} - & 0.3 & 0.0 & 0.2 & 0.2 \\ 0.7 & - & 0.2 & 0.4 & 0.3 \\ 0.9 & 0.8 & - & 0.7 & 0.8 \\ 0.8 & 0.6 & 0.3 & - & 0.6 \\ 0.8 & 0.6 & 0.2 & 0.5 & - \end{pmatrix}$$

$$P^2 = \begin{pmatrix} - & 0.3 & 0.8 & 0.4 & 0.2 \\ 0.7 & - & 0.5 & 0.6 & 0.8 \\ 0.2 & 0.6 & - & 0.4 & 0.6 \\ 0.5 & 0.3 & 0.7 & - & 0.5 \\ 0.6 & 0.6 & 0.4 & 0.5 & - \end{pmatrix}$$

$$P^3 = \begin{pmatrix} - & 0.6 & 0.6 & 0.3 & 0.4 \\ 0.4 & - & 0.6 & 0.5 & 0.3 \\ 0.4 & 0.4 & - & 0.3 & 0.3 \\ 0.7 & 0.5 & 0.7 & - & 0.7 \\ 0.6 & 0.7 & 0.7 & 0.3 & - \end{pmatrix}$$

To apply the AC-IOWA operator we need to calculate the CL_P value associated with every fuzzy preference relation. To do so we must first calculate every $\epsilon p_{ij}, \forall i, j \in \{1, ..., 5\}, i \neq j$ for every preference relation. Lets take as an example the calculation of ϵp_{34} on P^1:

$$\epsilon p_{34} = \frac{|cp_{34}^1 - p_{34}| + |cp_{34}^2 - p_{34}| + |cp_{34}^5 - p_{34}|}{3}$$

where

$$cp_{34}^1 = p_{31} + p_{14} - 0.5 = 0.9 + 0.2 - 0.5 = 0.6$$
$$cp_{34}^2 = p_{32} + p_{24} - 0.5 = 0.8 + 0.4 - 0.5 = 0.7$$
$$cp_{34}^5 = p_{35} + p_{54} - 0.5 = 0.8 + 0.5 - 0.5 = 0.8$$

and thus

$$\epsilon p_{34} = \frac{|0.6 - 0.7| + |0.7 - 0.7| + |0.8 - 0.7|}{3} = 0.07$$

Calculating every ϵp_{ij} for P^1 we have:

$$\epsilon P^1 = \begin{pmatrix} - & 0.0 & 0.03 & 0.0 & 0.1 \\ 0.07 & - & 0.07 & 0.03 & 0.17 \\ 0.13 & 0.07 & - & 0.07 & 0.13 \\ 0.07 & 0.03 & 0.0 & - & 0.1 \\ 0.07 & 0.03 & 0.1 & 0.03 & - \end{pmatrix}$$

Finally, the CL_P value for P^1 is obtained as the average of every ϵp_{ij}:

$$CL_{P^1} = (3 \times 0.0 + 5 \times 0.03 + 3 \times 0.1 + 6 \times 0.07 + 1 \times 0.17 + 2 \times 0.13) \ / \ 20 = 0.07$$

Following the same steps we can calculate each CL_P for every fuzzy preference relation:

$$CL_{P^2} = 0.28 \quad CL_{P^3} = 0.14$$

From the results obtained we can conclude that the expert e_1 is the most consistent while e_2 is the most inconsistent one.

Once every consistency level has been obtained we can aggregate experts' preferences into a global fuzzy preference relation P^{Global} using the AC-IOWA operator. In this example we will guide the IOWA operator by means of the fuzzy linguistic quantifier"most of". Yager in [7] considers the parameterized family of RIM quantifiers $Q(r) = r^{\alpha},\ \alpha \geq 0$, and the particular function with $\alpha = 2$ to represent the linguistic quantifier "most of". This function is strictly increasing but when used with the IOWA or OWA operators, associates high weighting values to those experts with a low consistency value. In order to overcome this drawback, two approaches could be adopted: i) the experts are ordered using the opposite criteria, i.e. the first one being the one with lowest consistency degree, or ii) a RIM quantifier with $\alpha < 1$ is used. We choose the second one, and in particular we use the RIM function $Q(r) = r^{1/2}$ to represent the linguistic quantifier "most of". Using expression (7), the corresponding weighting vector is $(0.61, 0.24, 0.15)$:

$$\begin{aligned} P^{Global} &= \Phi^{AC}_{most}(\langle 0.07, P^1\rangle, \langle 0.28, P^2\rangle, \langle 0.14, P^3\rangle) \\ &= 0.61 \cdot P^1 + 0.24 \cdot P^3 + 0.15 \cdot P^2 \\ &= \begin{pmatrix} - & 0.37 & 0.26 & 0.25 & 0.25 \\ 0.63 & - & 0.34 & 0.45 & 0.38 \\ 0.68 & 0.67 & - & 0.56 & 0.65 \\ 0.73 & 0.53 & 0.46 & - & 0.61 \\ 0.72 & 0.65 & 0.35 & 0.45 & - \end{pmatrix} \end{aligned}$$

In the exploitation step of the resolution process, this collective fuzzy preference relation is transformed into a global ranking of the alternatives from which a final solution for the GDMP problem is obtained.

6 Concluding Remarks and Future Works

In this work we have presented an IOWA operator that gives more importance to those experts providing fuzzy preference relations with a high level of consistency. To measure the consistency level of a fuzzy preference relation, a new consistency measure based on the concept of additive transitivity has been introduced. This consistency measure has been used to define the order inducing variable for the IOWA operator. Therefore, the higher the consistency level of a fuzzy preference relation the higher it contributes to the collective fuzzy preference relation. This is a natural and rational assumption to achieve good quality solutions to GDM problems where the information provided does not comply common properties (as consistency) usually required to solve them.

In future research, a complete decision model for GDM problems with inconsistent fuzzy preference relations will be developed, in which the consistency measure will be used to estimate or reconstruct possible missing values in fuzzy preference relations.

Acknowledgements

This work has been supported by the National Network in Decision Making, Preference Modelling and Aggregation TIC-2002-11942-E (Ministerio de Ciencia y Tecnología, España).

References

[1] Chiclana, F., Herrera, F., Herrera-Viedma, E.: Integrating three representation models in fuzzy multipurpose decision making based on fuzzy preference relations. Fuzzy Sets and Systems **97** (1998) 33–48

[2] Herrera-Viedma, E., Herrera, F., Chiclana, F., Luque, M.: Some issues on consistency of fuzzy preference relations. European Journal of Operational Research **154** (2004) 98–109

[3] Kacprzyk, J.: Group decision making with a fuzzy linguistic majority. Fuzzy Sets and Systems **18** (1986) 105–118

[4] Orlovski, S. A.: Decision-making with fuzzy preference relations, Fuzzy Sets and Systems **1** (1978) 155–167

[5] Tanino, T.: Fuzzy preference orderings in group decision making. Fuzzy Sets and Systems **12** (1984) 117–131

[6] Yager, R. R.: On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Transaction on Systems, Man and Cybernetics **18** (1988), 183–190.

[7] Yager, R.R.: Quantifier guided aggregation using OWA operators, *International Journal of Intelligent Systems*, Vol 11 (1996) 49–73.

[8] Yager, R. R., Filev, D. P.: Operations for granular computing: mixing words and numbers. Proceedings of the FUZZ-IEEE World Congress on Computational Intelligence, Anchorage (1998) 123–128.

[9] Yager, R. R., Filev, D. P.: Induced ordered weighted averaging operators. IEEE Transaction on Systems, Man and Cybernetics **29** (1999), 141–150.

OPINIÓN DE MAYORÍA EN TOMA DE DECISIÓN EN GRUPO MEDIANTE EL OPERADOR QMA-OWA

Peláez J.I. Doña J.M. Mesas A.
Dpto. Lenguajes y Ciencias de la Computación
E.T.S.I. Informática. Campus de Teatinos.
Universidad de Málaga
Málaga 29071. Spain
E-mail: jignacio@lcc.uma.es

D. L. La Red
Dpto. de Informática
Universidad Nacional del Nordeste Corrientes.
Argentina
E-mail: lrmdavid@exa.unne.edu.ar

Resumen

Uno de los principales problemas en la decisión en grupo es la definición de una estrategia de decisión que considere la opinión individual de los decisores. En esta estrategia el concepto de mayoría tiene especial importancia ya que generalmente se necesita obtener una opinión global que sintetice las opiniones de la mayoría de los decisores. La reducción de las opiniones individuales en un valor representativo (opinión de la mayoría) se realiza normalmente mediante un proceso de agregación. Con la teoría de conjunto borrosos el concepto de mayoría puede expresarse con cuantificador lingüístico. En este trabajo se proponen dos enfoques para obtener la opinión de mayoría. En primer lugar se considera la aplicación del cuantificador de forma general sobre el total de los elementos del conjunto de referencia. Mientras que en el segundo enfoque se aplica el cuantificador sobre grupos de mayoría. Para modelar la semántica del cuantificador lingüístico se usa el operador de mayoría MA-OWA, presentando una nueva definición de sus pesos y un nuevo proceso de normalización de mayoría.

Palabras clave: Operadores de Mayoría, cuantificadores lingüísticos, toma de decisión en grupo.

1 INTRODUCCIÓN

Los problemas de decisión en grupo son situaciones habituales dentro de las actividades humanas en las que un conjunto de expertos trabajan en un proceso de decisión que consiste en la evaluación de un conjunto de alternativas. El primer paso de este proceso consiste en la valoración individual de cada experto, es decir, cada decisor considera cada alternativa en base a un esquema de evaluación [3, 5, 13]. Al final de este paso cada alternativa tiene asociado un valor de juicio dentro de una escala (numérica o lingüística). El siguiente paso consiste en determinar para cada alternativa un valor de consenso que sintetice las valoraciones individuales de cada decisor. Este valor de consenso debe ser representativo de una estimación colectiva y se obtiene comúnmente a través de de la agregación de las opiniones de los expertos. De forma general se considera durante el proceso un grado de consenso para cada alternativa. Finalmente, se concluye con la selección de la mejor alternativa/s como valor de solución más representativo del problema [2, 3, 5, 6].

Para obtener un valor final de consenso se han propuesto diversas técnicas borrosas en las que se realiza una agregación guiada a través del concepto de mayoría, entendiéndose como mayoría aquella evaluación colectiva en la que intervienen la mayoría de los expertos involucrados en el problema de decisión. En estas técnicas el resultado que se produce no tiene por qué ser necesariamente de unanimidad aunque sí se obtiene un resultado en el que están de acuerdo una mayoría difusa de los decisores [5, 6, 8, 10].

En los métodos de toma de decisión borrosos, el concepto de mayoría se modela generalmente a través del uso de cuantificadores lingüísticos, tales como *al menos el 80%* y *muchos*. Un cuantificador lingüístico se define formalmente como un subconjunto borroso dentro de un dominio numérico. La semántica de un subconjunto borroso se describe a través del uso de una función de pertenencia la cual describe la compatibilidad de un valor absoluto o porcentual con respecto al concepto expresado por el cuantificador lingüístico [1, 7]. De esta forma, el cuantificador lingüístico se puede ver como un concepto borroso que se refiere a la cantidad de elementos a considerar en el conjunto de referencia.

$$\mu_{muchos}(x) = \begin{cases} 1 & x \geq 0.9 \\ 2x-0.8 & 0.4<x<0.9 \\ 0 & x \leq 0.4 \end{cases}$$

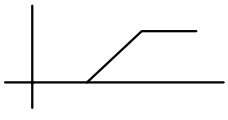


Fig. 1: Def. del cuantificador lingüístico *muchos*.

En la decisión en grupo, los cuantificadores lingüísticos se usan para indicar una estrategia de fusión que guía el proceso de agregación de las opiniones de los expertos. El resultado de esta agregación deberá representar la semántica del cuantificador lingüístico.

En este trabajo se considera el uso de los cuantificadores lingüísticos para especificar el proceso de agregación analizándose la semántica en el desarrollo de la agregación. En particular se muestra cómo la definición usual de los operadores OWA basada en los cuantificadores lingüísticos no refleja la semántica de consenso de la mayoría usada en los problemas de toma de decisión en grupo. Por esta razón se proponen dos nuevos operadores de mayoría basados en la agregación de los operadores de mayoría MA-OWA

El artículo se estructura como sigue: en la sección 2 se muestra la agregación guiada por cuantificadores a trasvés del operador OWA y los problemas que presenta al modelar el concepto de mayoría en la toma de decisión en grupo; en la sección 3 se definen los operadores de mayoría y se presentan dos nuevas técnicas de mayoría cuantificada para la obtención de soluciones de consenso para el problema de toma de decisión en grupo. Finalmente se exponen las conclusiones.

2 OPERADORES DE AGREGACIÓN EN TOMA DE DECISIÓN EN GRUPO

El concepto de agregación guiada por un cuantificador ha sido formalmente definido a través de los operadores OWA (Ordered Weighted Averaging) [14, 15] y de las integrales difusas [4].

En esta sección se estudia la aplicación de los operadores OWA a los problemas de toma de decisión, analizándose el proceso de agregación guiada y los problemas que surgen en la toma de decisión al representar la semántica de mayoría.

2.1. OPERADORES OWA (ORDERED WEIGHTED AVERAGING)

Un operador OWA es una función del tipo $F: R^n \to R$ a la que se le asocia un vector W de longitud n $W = [w_1, w_2, \ldots, w_n]^T$ donde $w_i \in [0,1]$ y $\sum_{i=1}^{n} w_i = 1$.

$OWA(a_1, a_2, \ldots, a_n) = \sum_{j=1}^{n} w_j \cdot b_j$ donde b_j es el *j*-ésimo valor mayor de las a_n.

La obtención de los pesos en los operadores OWA se suele realizar a través de los cuantificadores lingüísticos. Un cuantificador se define como una función $Q: [0,1] \to [0,1]$ donde $Q(0)=0$, $Q(1)=1$ y $Q(x) \geq Q(y)$ para $x>y$. Para un valor dado $x \in [0,1]$, $Q(x)$ es el grado en el que x satisface el concepto borroso representado por el cuantificador. Basándose en esta función Q, el vector de pesos OWA se determina a través de Q como sigue:

$$w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right) \qquad (1)$$

Estos pesos tienen la función de incrementar o decrementar la importancia de los diferentes componentes de la agregación según la semántica asociada al operador a través de Q,, es decir, el cuantificador determinará la estrategia de construcción del vector de pesos.

2.2. SEMÁNTICA DE AGREGACIÓN DE LOS OPERADORES OWA. CONCEPTO DE MAYORÍA

La semántica que se usa comúnmente en los operadores OWA es una generalización de los cuantificadores lógicos *Existe* y *Todos*. En este entorno los argumentos se valoran como valores de verdad o grados de satisfacción. Por ejemplo, si Q es el cuantificador entonces la agregación que realiza el operador OWA genera un valor que se corresponde con el valor de verdad de la expresión "Q de los elementos que se agregan se satisfacen". Esta semántica es la más usada en los problemas de toma de decisión en grupo. No obstante, debido a la formulación del cálculo de los pesos en los operadores OWA la agregación que se produce no es representativa del concepto de mayoría usado en la toma de decisión en grupo.

Por ejemplo, sea el cuantificador lingüístico *al menos el 80%*. Entonces el vector de pesos para 5 elementos cualesquiera es: $W_{al\ menos\ 80\%}$ = [0 0 0 1 0].

Como se puede ver, el cuantificador genera una agregación donde la semántica se corresponde con la expresión "es verdad que al menos el 80% de los criterios son totalmente satisfechos", lo que se corresponde con el cuarto elemento independientemente del grado de satisfacción de los valores previos, es decir, producen el mismo resultado (0.1) la agregación de los vectores [1 1 1 0.1 0] y [0.1 0.1 0.1 0.1 0].

Esta semántica junto con la definición del cálculo de pesos del operador OWA no produce una solución que sea representativa a la cantidad especificada por el cuantificador. Esto es debido a que el vector de pesos se interpreta para cada peso i como el incremento en la satisfacción del criterio i+1 con respecto a la satisfacción del criterio i. Esto produce una agregación que puede no ser válida para los problemas de toma de decisión en grupo, ya que la semántica del cuantificador que guía el proceso de agregación no modela adecuadamente el concepto de mayoría tal y como este se usa en las

aplicaciones de decisión donde se debe agregar considerando Q de los valores más relevantes en el conjunto de referencia en lugar de Q de los valores son totalmente satisfechos. Para ello, un operador con una semántica válida para calcular una opinión de la mayoría debe producir un valor que sea representativo de la cantidad expresada por el cuantificador lingüístico.

3 OPERADORES DE MAYORÍA CUANTIFICADOS QMA-OWA

Los operadores de mayoría MA-OWA [9, 10, 11] surgen por la necesidad de obtener valores de agregación que sean representativos de la mayoría de los elementos a agregar sin omisión de las minorías, ya que los operadores de agregación comunes [2, 15] tienden a enfatizar la opinión de las minorías a expensas de las mayorías generando valores de agregación que no son válidos para los problemas de toma de decisión en grupo. Los operadores de mayoría se definen como operadores neat OWA [9] sin hacer uso de los cuantificadores lingüísticos.

En esta sección se presentan dos nuevos operadores de mayoría en los que se introducen el uso de cuantificadores para guiar el proceso de agregación. En primer lugar se considera la aplicación del cuantificador de forma general sobre el total de los elementos del conjunto de referencia. El segundo método se basa en el concepto de grupos de mayoría. Así mismo se introduce un nuevo proceso de normalización en el cálculo de los pesos para obtener un vector más característico para el concepto representado por la expresión del cuantificador lingüístico.

3.1. OPERADOR DE MAYORÍA MA-OWA

Como se muestra en [9] el operador *MA-OWA* se define como:

$$F_{MA}(a_1,a_2,\ldots,a_n)=\sum_{i=1}^{n} w_i\cdot b_i=\sum_{i=1}^{n} f_i\cdot(b_1,b_2,\ldots,b_n)\cdot b_i \qquad (2)$$

donde $w_i\in[0,1]$ con $\sum_{i=1}^{n} w_i=1$ y b_i es el i-ésimo elemento de $a_1,\ldots, a_n$ ordenado en orden ascendente según las cardinalidades.

Los pesos del operador *MA-OWA* se calculan como sigue.

Sea δ_i la cardinalidad del elemento i con $\delta_i > 0$, entonces:

$$w_i=f_i(b_1,\ldots,b_n)=\frac{\gamma_i^{\delta_{\min}}}{\theta_{\delta_{\max}}\cdot\theta_{\delta_{\max-1}}\cdot\ldots\cdot\theta_{\delta_{\min+1}}\cdot\theta_{\delta_{\min}}}+$$

$$+\frac{\gamma_i^{\delta_{\min+1}}}{\theta_{\delta_{\max}}\cdot\theta_{\delta_{\max-1}}\cdot\ldots\cdot\theta_{\delta_{\min+1}}}+\ldots+\frac{\gamma_i^{\delta_{\max}}}{\theta_{\delta_{\max}}} \qquad (3)$$

donde

$$\gamma_i^k=\begin{cases}1 & si\ \delta_i\geq k\\ 0 & otro\,caso\end{cases} \quad \text{y}$$

$$\theta_i=\begin{cases}(número\ de\ elementos\ con\ \text{cardinalidad}\geq i)+1 & si\ i\neq\delta_{\min}\\ número\ de\ elementos\ con\ \text{cardinlaidad}\geq i & otro\,caso\end{cases}$$

Los operadores de mayoría realizan la agregación en función del elemento δ_i que mide, de forma general, la importancia del elemento i a través de su cardinalidad. En los procesos de mayoría se considera la formación de grupos de discusión o de mayoría en función de sus similaridades o distancias entre las opiniones de los expertos, considerándose dentro del mismo grupo todos aquellos valores con un mínimo de separación. Como se observa, la forma de cálculo del valor δ_i es independiente de la definición de los operadores de mayoría.

En este trabajo, la cardinalidad se calcula usando la siguiente función de distancia:

$$dist(a_i,a_j)=\begin{cases}1 & if\ \ |a_i-a_j|\leq x\\ 0 & otro\,caso\end{cases} \qquad (4)$$

Por lo que la cardinalidad del elemento a_i es

$$\delta_i=\sum_{j=1}^{n} dist(a_i,a_j) \qquad (5)$$

Donde el valor x modeliza el tamaño de cada grupo. Socialmente este grado se corresponde con la flexibilidad de los decisores para agruparse y reforzar sus opiniones.

3.2. NORMALIZACIÓN DE MAYORÍA

En los operadores de mayoría cuantificada la cardinalidad total de los elementos y de cada uno de los grupos de mayorías se ve modificada por el cuantificador, por ello, los nuevos pesos cuantificados deberán modificarse para ser normalizados ya que se debe seguir cumpliendo la propiedad $\sum_{i=1}^{n} w_i=1$. Esta modificación se denomina normalización de mayoría y usa la suma de los valores del cuantificador como nueva cardinalidad, incrementando los pesos en la proporción expresada por dichos valores, de esta forma se obtiene una normalización cuantificada. La definición formal de la normalización de mayoría es la siguiente:

Sea $(w_1, w_2,\ldots, w_n)$ y $(q_1, q_2,\ldots,q_n)$ donde $0\leq q_i\leq 1$ y $\sum_{i=1}^{n} w_i=1$, además se establece $\rho=\sum_{i=1}^{n} w_i\cdot q_i\leq 1$.

Si $\varepsilon = 1-\rho$, entonces $\varepsilon / \sum_{i=1}^{n} q_i$ es la cantidad a ser añadida a cada elemento en la cantidad expresada por q_i.

Luego se demuestra que la normalización de mayoría es:

$$\sum_{i=1}^{n}(w_i \cdot q_i + \frac{\varepsilon}{\sum_{i=1}^{n} q_i} \cdot q_i) = \sum_{i=1}^{n} w_i \cdot q_i + \frac{\varepsilon}{\sum_{i=1}^{n} q_i} \cdot \sum_{i=1}^{n} q_i = \rho + 1 - \rho = 1$$

Como se observa en su definición, esta nueva normalización de mayoría modifica cada peso incrementándolo en función del cuantificador usado en cada caso con respecto a los demás elementos cuantificados, añadiendo, en función de este valor, a cada peso la cantidad ε que corresponda. De esta forma se puede obtener una normalización que se integra en el proceso de agregación. La normalización clásica, por ejemplo, realiza el reajuste de los pesos considerando únicamente el tamaño del peso sin normalizar con respecto al tamaño total del conjunto que se desea normalizar, de esta forma se favorecen a aquellos elementos con mayor valor inicial en lugar de favorecer a aquellos que se han cuantificado más favorablemente (independientemente de su valor inicial).

3.3. CUANTIFICACIÓN INDIVIDUAL

Usualmente los cuantificadores se aplican como una función de corte sobre el conjunto de valores de referencia indicando una cuantificación sobre los elementos a agregar. La cuantificación individual consiste en indicar el grado con que cada miembro del conjunto de valores representa la mayoría. Para ello, es necesario considerar cada elemento de forma individual, por ejemplo, si tenemos el valor 2 con cardinalidad 3, se tomará como tres 2 con cardinalidad 1 en lugar del modelo anterior, luego, los pesos calculados previamente se divide por la cardinalidad de cada elemento para obtener de este modo los pesos individuales.

Para obtener el nuevo vector de pesos cuantificados se usa la siguiente expresión:

$$w_i^Q = w_i^N \cdot Q\left(\frac{i}{n}\right) + \left[Q\left(\frac{i}{n}\right) \cdot \frac{1 - \sum_{i=1}^{n}\left(w_i^N \cdot Q\left(\frac{i}{n}\right)\right)}{\sum_{i=1}^{n} Q\left(\frac{i}{n}\right)} \right] \quad (6)$$

donde Q es el cuantificador.

Entonces el operador de mayoría cuantificado QMA-OWA es $\sum_{i=1}^{n} w_i^Q \cdot b_i$.

Ejemplo: Se desean agregar 7 elementos que representan la opinión de un conjunto de expertos [0 0 0.7 0.8 1 1 1] para obtener un valor que los sintetice en uno solo.

- Operador OWA con el cuantificador *muchos*:

Conjunto de valores ordenados:

B = [1 1 1 0.8 0.7 0 0]

Vector de pesos:

W = [0 0 0.057 0.285 0.285 0.285 0.088]

$$\text{OWA} = \sum_{i=1}^{n} w_i \cdot b_i = 0.485$$

- Operador MA-OWA (sin cuantificador):

Se consideran 4 elementos B = [0.7 0.8 0 1] con cardinalidad [1 1 2 3] respectivamente.

El vector de pesos ordenado en orden ascendente según las cardinalidades es:

W = [0.042 0.042 0.208 0.708]

$$\text{MA-OWA} = \sum_{i=1}^{n} w_i \cdot b_i = 0.771$$

- Operador cuantificado QMA-OWA con *muchos*:

Sean los nuevos conjuntos de elementos y pesos individuales B_N = [0.7 0.8 0 0 1 1 1] y W_N = [0.041 0.041 0.208/2 0.208/2 0.708/3 0.708/3 0.708/3]. Con el vector Q = [0 0 0.057 0.343 0.629 0.914 1]

Entonces

$$w_1^Q = 0.042 \cdot 0 + 0 \cdot \frac{1 - 0.642}{2.943} = 0$$

$$w_2^Q = 0.042 \cdot 0 + 0 \cdot \frac{1 - 0.642}{2.943} = 0$$

....

$$w_6^Q = 0.236 \cdot 0.914 + 0.914 \cdot \frac{1 - 0.642}{2.943} = 0.327$$

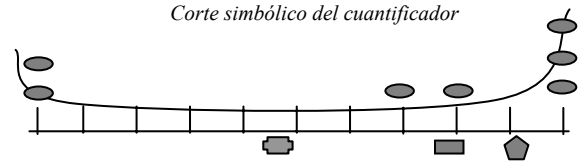
$$w_7^Q = 0.236 \cdot 1 + 1 \cdot \frac{1 - 0.642}{2.943} = 0.358$$

Luego el vector de pesos es

W^Q = [0 0 0.013 0.077 0.225 0.327 0.358]

Resultando QMA-OWA = $\sum_{i=1}^{n} w_i^Q \cdot b_i$ =0.910

Gráficamente:



Donde

Valores a agregar

Resultados de las agregaciones de Mayoría, OWA y QMA-OWA respectivamente.

Fig. 2: Representación gráfica usando diferentes operadores de agregación donde se muestra el corte del cuantificador.

Si analizamos el vector *Q* es fácil observar cómo el cuantificador corta los elementos 0.7 y 0.8, y prácticamente la totalidad de 0. Se puede ver que el operador QMA-OWA produce un resultado representativo de la mayoría expresada por el cuantificador (*muchos*), mientras que el operador OWA (0.485) trabajando con el mismo cuantificador produce un resultado que no es representativo del concepto de mayoría del cuantificador (más del 70% de los elementos es mayor que 0.7). Como hemos visto, esto se debe a que en la semántica de un operador OWA tradicional el valor de la agregación se corresponde con un grado de la satisfacción de una expresión del tipo "*Q* de los valores son totalmente satisfechos", mientras que un operador con una semántica de mayoría debe producir un valor que sea representativo de *Q* de los valores más relevantes en el conjunto de referencia.

Es importante señalar que la semántica del cuantificador se ve afectada de forma significativa por el componente no lineal del operador de agregación, es decir, por la forma en que los argumentos se reordenan. En los operadores OWA tradicionales, después del paso de reordenación, si los valores representativos se encuentran a la izquierda del vector se consideran de forma nula o de forma mucho menor que si se encontraran el la parte derecha, incluso cambiando la forma de ordenación el problema en la distribución de los pesos continuaría, ya que al medir un valor de incremento los pesos pueden decrecer en las últimas posiciones.

Para poder representar el concepto de mayoría, los operadores QMA-OWA realizan la agregación en función de la semántica del cuantificador y de la importancia δ_i de los elementos dentro del vector de valores a agregar.

3.4. CUANTIFICACIÓN EN GRUPOS

Desde un punto de vista social estricto la cuantificación general puede producir problemas de distribución, ya que el corte del cuantificador excluye un conjunto de elementos del proceso de agregación. En los problemas de toma de decisión esta exclusión puede generar situaciones donde los expertos no queden satisfechos con el resultado final [12]. Para resolver estos tipos de problemas produciendo un valor de consenso que use la semántica del cuantificador se desarrolla un nuevo operador que usa el concepto de grupos de mayoría, de esta forma, el cuantificador se aplicará sobre cada grupo de mayoría produciendo un nuevo grupo con la cardinalidad expresada por *Q*.

Los pesos se calculan para cada grupo, dividiendo el peso original de cada elemento por su cardinalidad para después aplicar el cuantificador sobre el grupo.

Para obtener el nuevo vector de pesos cuantificados se usa la siguiente expresión:

$$w_i^Q = \frac{w_i}{\delta_i} \cdot \sum_{j=1}^{\delta_i} Q\left(\frac{\mathrm{j}}{\delta_i}\right) + \left[\sum_{j=1}^{\delta_i} Q\left(\frac{\mathrm{j}}{\delta_i}\right) \cdot \frac{1 - \sum_{i=1}^{n} \frac{w_i}{\delta_i} \cdot \sum_{j=1}^{\delta_i} Q\left(\frac{\mathrm{j}}{\delta_i}\right)}{\sum_{i=1}^{n} \sum_{j=1}^{\delta_i} Q\left(\frac{\mathrm{j}}{\delta_i}\right)} \right] \quad (7)$$

Donde *Q* es el cuantificador, *n* es el número de grupos de mayoría y δ_i es la cardinalidad del grupo *i*.

El operador cuantificado QMA-OWA con grupos de mayoría es: $\sum_{i=1}^{n} w_i^Q \cdot b_i$.

Ejemplo: Sea B = [0.7 0.8 0 1] con cardinalidad [1 1 2 3] respectivamente, entonces el vector de pesos de mayoría es W = [0.042 0.042 0.208 0.708], y sea el cuantificador definido en la figura 1, entonces:

Los vectores Q para grupos con cardinalidad 1, 2 y 3 respectivamente son: [1]; [0.2 1]; [0 0.533 1]

Luego

$$w_1^Q = \frac{0.042}{1} \cdot 1 + 1 \cdot \frac{1 - 0.570}{4.733} = 0.132;$$

$$w_2^Q = \frac{0.042}{1} \cdot 1 + 1 \cdot \frac{1 - 0.570}{4.733} = 0.132;$$

$$w_3^Q = \frac{0.208}{2} \cdot 1.2 + 1.2 \cdot \frac{1 - 0.570}{4.733} = 0.234;$$

$$w_4^Q = \frac{0.708}{3} \cdot 1.533 + 1.533 \cdot \frac{1 - 0.570}{4.733} = 0.501$$

Obteniéndose W^Q = [0.132 0.132 0.234 0.501]

Finalmente QMA-OWA = $\sum_{i=1}^{n} w_i^Q \cdot b_i$ =0.7

Gráficamente:

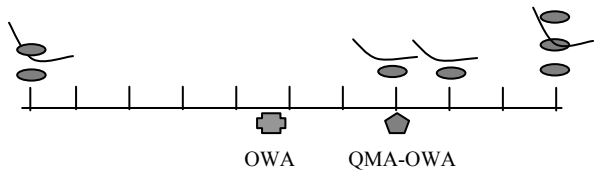


Fig. 3: Representación gráfica usando diferentes agregaciones donde se muestra el corte del cuantificador para cada grupo de mayoría.

Usando los grupos de mayoría en el proceso de cuantificación no se producen discriminaciones en los valores de opinión de los expertos. De esta forma se produce un resultado de fusión que es representativo de la semántica expresada por Q para todos los grupos y se eliminan los problemas de distribución.

4 CONCLUSIONES

En este trabajo se ha estudiado la importancia obtener una opinión global que sintetice las opiniones de la mayoría de los decisores en los problemas de toma de decisión en grupo. Para ello se ha considerado el uso de los cuantificadores lingüísticos para guiar la agregación y se ha analizando su semántica. También se ha mostrado cómo la definición usual de los operadores OWA basada en los cuantificadores lingüísticos no refleja la semántica de consenso de la mayoría usada en los problemas de toma de decisión en grupo. Por esta razón se han propuesto dos nuevos operadores de mayoría cuantificados basados en la agregación de los operadores MA-OWA. La primera propuesta considera la aplicación del cuantificador sobre el total de los elementos a agregar. El segundo método se basa en la consideración del concepto de grupos de mayoría para eliminar los problemas de distribución. Para el desarrollo de ambas técnicas se ha definido una nueva normalización denominada normalización de mayoría.

Agradecimientos

Este trabajo está apoyado por los proyectos TIC2002-04242-C03-02 y TIC2002-11942-E.

Referencias

[1] Barwise J. and Cooper R., Generalized Quantifiers in Natural Language. Linguistic and Philosophy, 4:159-220, 1981

[2] Delgado M. Verdegay J. L. and Vila M. A. 1993. On aggregation operations of linguistic labels, Internat. J. Intelligent Systems 8. 351-370.

[3] Fishburn P.C., A comparative analysis of Group Decision Methods, Behavioral Science, Vol. 16(6), 538-544, 1971.

[4] Grabisch M., Fuzzy integrals in multicriteria decision making, Fuzzy Sets and Systems, 69, 279-298, 1995.

[5] Herrera F., Herrera-Viedma E. (2000). Linguistic decision analysis: steps for solving decision problems under linguistic information. Fuzzy Sets and Systems 115, 67-82.

[6] Herrera F., Herrera-Viedma E., Verdegay J.L., Direct Approach Processes in Group Decisión Making Using Linguistic OWA Operators, Fuzzy Sets and Systems, 79, 1996.

[7] Keenan E.L. and Westerstal D., Generalized quantifiers in Linguistic and Logics, in van Benthem J., ter Meulen A. (eds) Handbook of logic and language, Amsterdam: North-Holland, 837-893, 1997.

[8] Nurmi H. and Kacprzyk J. 1991. On Fuzzy Tournaments and Their Solution Concepts in Group Decision Making. European Journal of Operational Research.

[9] Pelaez J.I., Doña J.M., Majority Additive-Ordered Weighting Averaging: A New Neat Ordered Weighting Averaging Operators Based on the Majority Process, International Journal of Intelligent Systems, 18, 4 (2003) 469-481.

[10] Pelaez J.I., Doña J.M., LAMA: A Linguistic Aggregation of Majority Additive Operator, International Journal of Intelligent Systems, (2003).

[11] Pelaez J.I., Doña J.M., La Red D., Analysis of the Majority Process in Group Decision Making Process, 9th International Conference on Fuzzy Theory and Technology, North Carolina. USA. (2003).

[12] Robertson G. Webb G. 1998. Cake-Cutting Algorithms: Be Fair If You Can. A K Peters Ltd. Portland, OR.

[13] Saaty T, L. 1980. The Analytic Hierarchy Process. Macgraw Hill.

[14] Yager R. 1988. On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans. On Systems, Man and Cybernetics 18. 183-190.

[15] Yager R. 1993. Families of OWA operators. Fuzzy Sets and Systems. 59. 125-148.

SELECCIÓN ENDÓGENA DE EXPERTOS

de Andrés Calle, Rocío
Departamento de Economía Aplicada
Facultad de Ciencias Económicas y Empresariales
Universidad de Valladolid
e-mail: rocioac@eco.uva.es

García Lapresta, José Luis
Departamento de Economía Aplicada
Facultad de Ciencias Económicas y Empresariales
Universidad de Valladolid
e-mail: lapresta@eco.uva.es

Resumen

Uno de los problemas recientemente abordados en la literatura sobre toma de decisiones colectivas consiste en identificar y seleccionar los individuos de un grupo respecto de un determinado atributo. Para ello se tienen en cuenta las opiniones que cada uno de los agentes tiene sobre sí mismo, así como del resto de los miembros que componen el colectivo. En la literatura existente sólo se han tenido en cuenta opiniones dicotómicas: cada agente muestra su aprobación o desaprobación respecto de cada individuo sobre su idoneidad respecto del atributo considerado. En este trabajo se aborda este asunto desde la perspectiva de la lógica difusa, de forma que los individuos gradúan sus valoraciones en el intervalo unidad. En primer lugar se lleva a cabo la construcción de la opinión colectiva sobre un individuo mediante reglas graduales independientes, de manera que dicha valoración colectiva dependa únicamente de las opiniones que tienen sobre él cada uno de los agentes, incluido él mismo. En distintos tipos de problemas la opinión que un individuo tiene sobre sí mismo puede contar en mayor o menor medida, según las reglas al uso o el grado de liberalismo existente en el colectivo. Para abordar este aspecto se ha utilizado una clase de medias cuasiaritméticas ponderadas. Una vez establecida la valoración colectiva de cada agente, se consideran reglas de decisión que, a partir de las valoraciones anteriores, permiten seleccionar los agentes, según diferentes niveles de cualificación. En el trabajo se muestran las principales propiedades de las reglas graduales propuestas, así como de las reglas de decisión subyacentes.

Palabras Clave: Toma de decisiones colectivas, Medias cuasiaritméticas ponderadas, Liberalismo.

1 INTRODUCCIÓN

En este trabajo se aborda el problema de cómo decidir qué agentes de un grupo satisfacen un determinado atributo, teniendo en cuenta las opiniones que los propios individuos tienen sobre ellos mismos y sobre los demás miembros del colectivo. Uno de los orígenes de este tipo de problemas se basa en la identificación de los miembros de un colectivo respecto de atributos religiosos y étnicos: ¿Quién es judío? (Kasher [11]). Posteriormente, Kasher – Rubinstein [12], Samet – Schmeidler [14], Billot [4] y Dimitrov – Sung – Xu [7] analizan este mismo problema desde la perspectiva de la Teoría de la Elección Social, proporcionando caracterizaciones axiomáticas de varias reglas de decisión, entre las cuales caben destacar las liberales, las dictatoriales, las oligárquicas y las de consentimiento.

El problema mencionado puede generalizarse a muy diversos atributos, en particular a la selección de expertos dentro de un grupo, a partir de las opiniones de los propios miembros del colectivo. Se ha de señalar que los trabajos mencionados abordan la cuestión desde una perspectiva dicotómica, es decir, los agentes únicamente pueden opinar de forma extrema, señalando qué individuos satisfacen o no un determinado atributo.

En este trabajo trataremos el problema desde el punto de vista de la lógica difusa. Los agentes evalúan el cumplimiento del atributo considerado, para cada uno de los individuos y sobre sí mismo, mediante valores numéricos entre 0 y 1. Dicho de otra forma, cada agente aporta con su opinión un subconjunto difuso del conjunto de agentes. A partir de esta información se trata de obtener un nuevo subconjunto difuso agregado, de manera que el proceso de agregación respete algunos principios básicos, que permita a tal subconjunto difuso representar adecuadamente las opiniones individuales en una única opinión colectiva.

A la vista de esta nueva información se abre un nuevo proceso, ahora de *desfuzzyficación*, con objeto de tomar una decisión final sobre qué agentes son seleccionados respecto del atributo considerado: en el ejemplo apuntado, los expertos que cuentan con el beneplácito interno del colectivo de agentes. Para ello hemos considerado α-cortes del subconjunto difuso agregado, que pueden entenderse como niveles de cualificación a exigir para llevar a cabo la tarea a desempeñar.

Para llevar a cabo la primera tarea de agregación hemos considerado reglas graduales independientes, las cuales permiten obtener una opinión colectiva sobre cada agente teniendo en cuenta las opiniones de todos los individuos sobre aquél que se evalúa. Este proceso equivale a disponer de tantos operadores de agregación como agentes haya. Una vez establecido el marco general, hemos utilizado como operadores de agregación una clase de medias cuasiaritméticas ponderadas, las cuales permitirán graduar diferentes tipos de liberalismo con la ayuda de un parámetro. Dicho liberalismo permitirá reflejar la importancia que se quiera dar a la opinión que cada individuo tenga sobre sí mismo, según el tipo de problema decisional abordado. Algunos estudios sobre medias cuasiaritméticas ponderadas y operadores de agregación pueden ser encontrados en Aczél [1, 2], Aczél – Alsina [3], Bullen – Mitrinovic – Vasic [5], Ovchinnikov [13], Fodor – Roubens [8] y Calvo – Kolesárova – Komorníková – Mesiar [6], entre otros.

Tanto en la primera fase del proceso de decisión, recién expuesto, como en la segunda, basada en reglas de decisión asociadas a las reglas graduales independientes, se lleva a cabo un análisis de las propiedades (las demostraciones no se incluyen por motivos de espacio).

El trabajo esta organizado de la siguiente forma. La Sección 2 se dedica a las reglas graduales independientes, operadores de agregación y medias cuasiaritméticas ponderadas. En la Sección 3 consideramos la clase de medias cuasiaritméticas ponderadas que permite abordar los diferentes grados de liberalismo. La Sección 4 está dedicada a las reglas de decisión asociadas a las reglas graduales independientes.

2 REGLAS GRADUALES INDEPENDIENTES

Supongamos que un determinado grupo de agentes deben decidir qué miembros son adecuados para llevar a cabo una tarea determinada que exige determinadas cualidades. Dado que tal atributo es vago, cada uno de los agentes podrá opinar de forma gradual sobre sí mismo y sobre el resto de los individuos, mediante valores numéricos en el intervalo unidad. La agregación de estas valoraciones individuales se llevará a cabo mediante reglas graduales, las cuales asignarán una valoración colectiva entre 0 y 1 sobre cada agente, teniendo en cuenta las opiniones individuales. Se supondrá que estas reglas graduales son independientes, en el sentido de Samet – Schmeidler [14], de tal manera que sólo se consideran las opiniones sobre el individuo a evaluar y no otras, por ser irrelevantes.

Notación. Sea $N=\{1,\ldots,n\}$ el conjunto de individuos pertenecientes a un grupo y 2^N el conjunto potencia de *N*, es decir, el conjunto de todos los subconjuntos de *N*. Se utilizará negrita para denotar los vectores en $[0,1]^n$: $\mathbf{x}=(x_1,\ldots,x_n)$, $\mathbf{1}=(1,\ldots,1)$, $\mathbf{0}=(0,\ldots,0)$. Dados dos vectores $\mathbf{x}, \mathbf{y}\in[0,1]^n$, con $\mathbf{x}\geq\mathbf{y}$ denotamos $x_i\geq y_i$ para todo $i\in\{1,\ldots,n\}$. Dada una biyección $\pi: N\to N$, con $\pi(\mathbf{x})$ denotamos $(x_{\pi(1)},\ldots,x_{\pi(n)})$. Un *perfil* P es una matriz $n\times n$ con coeficientes en el intervalo $[0,1]$:

$$P=\begin{pmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{pmatrix},$$

donde P_{ij} es la valoración que el individuo *i* da sobre el individuo *j* en cuanto a su grado de cumplimiento del atributo fijado. De esta forma, cada fila i en dicho perfil *P* describe las valoraciones del individuo *i* sobre el resto de los agentes, incluido él mismo. Cada columna *j* de dicho perfil muestra las valoraciones acerca del individuo *j*, incluida la que tiene sobre sí mismo. El conjunto de perfiles, $M_{n\times n}([0,1])$, está formado por todas las posibles matrices de orden $n\times n$ con coeficientes en el intervalo $[0,1]$.

Definición 1. Una *regla gradual* es una función $f: M_{n\times n}([0,1])\to[0,1]^n$ que asocia a cada perfil $P\in M_{n\times n}([0,1])$ un vector

$$f(P)=(f_1(P),\ldots,f_n(P))\in[0,1]^n,$$

donde $f_j(P)$ es la valoración colectiva obtenida por el individuo j de acuerdo con f en el perfil *P*.

En García Lapresta – Llamazares [9, Theorem 1] se proporciona una condición necesaria y suficiente para la neutralidad en las reglas de agregación de preferencias difusas. Tomando esa misma idea podemos introducir un axioma de independencia, coincidente con el introducido en Samet – Schmeidler [14], ahora para reglas graduales.

Definición 2. Una regla gradual $f: M_{n\times n}([0,1])\to[0,1]^n$ es *independiente* si para cada individuo $j\in N$ existe una

función $F_j:[0,1]^n \to [0,1]$, denominada *operador de agregación*, tal que

$$f_j(P) = F_j(P_{1j},\ldots,P_{jj},\ldots,P_{nj}),$$

para cada perfil $P \in M_{n\times n}([0,1])$.

La independencia implica que la valoración colectiva de cada individuo es obtenida teniendo en cuenta únicamente las opiniones sobre el individuo evaluado. Entonces:

$$f(P) = (F_1(P_{11},\ldots,P_{n1}),\ldots,F_n(P_{1n},\ldots,P_{nn})).$$

Así, considerar una regla gradual independiente es equivalente a tomar n operadores de agregación $F_1,\ldots,F_n$.

A continuación presentamos algunos conceptos acerca de los operadores de agregación incluidos en la definición de las reglas graduales independientes.

Definición 3. Sea $F:[0,1]^n \to [0,1]$ un operador de agregación.

1. F es *unánime* si para todo $t \in [0,1]$: $F(t\cdot\mathbf{1}) = t$.
2. F *es anónimo* si para toda biyección $\pi: N \to N$ y todo $\mathbf{x} \in [0,1]^n$: $F(\pi(\mathbf{x})) = F(\mathbf{x})$.
3. F es *monótono* si para cualesquiera $\mathbf{x},\mathbf{y} \in [0,1]^n$:
$$\mathbf{x} \geq \mathbf{y} \Rightarrow F(\mathbf{x}) \geq F(\mathbf{y}).$$
4. F es *compensativo* si para todo $\mathbf{x} \in [0,1]^n$:
$$\min\{x_1,\ldots,x_n\} \leq F(\mathbf{x}) \leq \max\{x_1,\ldots,x_n\}.$$
5. F es *auto-dual* si para todo $\mathbf{x} \in [0,1]^n$:
$$F(\mathbf{1}-\mathbf{x}) = 1 - F(\mathbf{x}).$$

Observación 1. Sea $F:[0,1]^n \to [0,1]$ un operador de agregación.

1. Si F es compensativo, entonces F es unánime.
2. Si F es monótono, entonces F es unánime si y sólo si F es compensativo.
3. Si F es auto-dual, entonces $F(\mathbf{1}) = 1$ equivale a $F(\mathbf{0}) = 0$.

Definición 4. Dada una regla gradual independiente *f* con operadores de agregación asociados $F_1,\ldots,F_n$, diremos que f es *unánime*, *anónima*, *monótona, compensativa*, *auto-dual* o *continua* si $F_1,\ldots,F_n$ satisfacen las propiedades con el mismo nombre.

El anonimato es una propiedad muy restrictiva para abordar el liberalismo, ya que la valoración que cada individuo tiene sobre sí mismo puede tener un tratamiento diferenciado a la que el agente tiene sobre el resto de los agentes del grupo. Por esta razón hemos introducido una versión más débil del anonimato, en donde la simetría de las reglas graduales independientes ahora es sólo aplicable a las opiniones de aquellos individuos que no son evaluados en ese momento.

Definición 5. Dada una regla gradual independiente *f* con operadores de agregación asociados $F_1,\ldots,F_n$, diremos que *f* es *débilmente anónima* si para todo $j \in N$ y toda biyección $\pi: N-\{j\} \to N-\{j\}$:

$$F_j(P_{\pi(1)j},\ldots,P_{\pi(j-1)j},P_{jj},P_{\pi(j+1)j},\ldots,P_{\pi(n)j}) =$$
$$= F_j(P_{1j},\ldots,P_{(j-1)j},P_{jj},P_{(j+1)j},\ldots,P_{nj}),$$

para todo perfil $P \in M_{n\times n}([0,1])$.

Obviamente, el anonimato implica el anonimato débil.

Una clase de operadores de agregación interesante para abordar el problema del liberalismo es la correspondiente a las medias cuasiaritméticas ponderadas. En primer lugar introducimos las medias cuasiaritméticas.

Definición 6. Una función $\varphi:[0,1]\to[0,1]$ es un *automorfismo de orden* si φ es biyectiva y creciente.

De acuerdo con García Lapresta – Llamazares [9, pp. 684-685], si φ es un automorfismo de orden, entonces φ es continuo, estrictamente creciente y verifica $\varphi(0) = 0$ y $\varphi(1) = 1$; además, φ^{-1} también es un automorfismo de orden con las mismas propiedades.

Definición 7. Dado un automorfismo de orden $\varphi:[0,1]\to[0,1]$, la *media cuasiaritmética asociada a* φ es el operador de agregación $F_\varphi:[0,1]^n \to [0,1]$ definido por:

$$F_\varphi(x_1,\ldots,x_n) = \varphi^{-1}\left(\frac{\varphi(x_1)+\cdots+\varphi(x_n)}{n}\right).$$

Es fácil comprobar que F_φ es siempre unánime, anónimo, monótono, compensativo y continuo. En García Lapresta – Llamazares [10, pp. 471-473] se demuestra que para todo automorfismo de orden $\varphi:[0,1]\to[0,1]$, F_φ es auto-dual si y sólo si se verifica $\varphi(1-x) = 1-\varphi(x)$ para todo $x \in [0,1]$. En este sentido, podemos decir que el automorfismo de orden φ es *auto-dual*. Resulta

inmediato comprobar que si φ es auto-dual, entonces $\varphi\left(\frac{1}{2}\right)=\frac{1}{2}$ y φ^{-1} es también auto-dual.

A pesar de que las medias cuasiaritméticas tienen buenas propiedades, el anonimato puede ser contraproducente en la modelización del liberalismo, dentro del marco de las reglas graduales independientes. Por esta razón, consideraremos medias cuasiaritméticas ponderadas que, si bien no son anónimas, cumplen algunas de las propiedades más interesantes de las medias cuasiaritméticas.

Definición 8. Sean $\varphi:[0,1]\to[0,1]$ un automorfismo de orden y $\mathbf{w}=(w_1,\ldots,w_n)\in[0,1]^n$ un vector de pesos tal que $w_1+\cdots+w_n=1$. La *media cuasiaritmética ponderada asociada a* φ *y* $\mathbf{w}$ es el operador de agregación $F_\varphi^{\mathbf{w}}:[0,1]^n\to[0,1]$ definido por:

$$F_\varphi^{\mathbf{w}}(x_1,\ldots,x_n)=\varphi^{-1}\left(w_1\varphi(x_1)+\cdots+w_n\varphi(x_n)\right).$$

Resulta inmediato comprobar que $F_\varphi^{\mathbf{w}}$ es siempre unánime, monótono, compensativo y continuo. Además, si φ es auto-dual, entonces $F_\varphi^{\mathbf{w}}$ es también auto-dual; y $F_\varphi^{\mathbf{w}}$ es anónimo si y sólo si $w_i=\frac{1}{n}$ para todo $i\in\{1,\ldots,n\}$, es decir, $F_\varphi^{\mathbf{w}}$ es la media cuasiaritmética asociada a φ, F_φ.

3 LIBERALISMO

A la hora de establecer qué individuos son adecuados respecto de un determinado atributo o tarea a desempeñar, el liberalismo extremo consiste en que la valoración colectiva de un determinado agente depende únicamente de la opinión que el propio agente tiene sobre sí mismo. En el otro extremo tendríamos el caso en el que no se tienen en consideración las opiniones que los agentes tienen sobre sí mismos.

Con el fin de plantear un modelo donde el liberalismo esté contemplado de forma gradual, a continuación se introduce, a modo de ejemplo, una clase de reglas graduadas independientes asociadas a una familia concreta de medias cuasiariméticas ponderadas.

Sea $\gamma\in[0,1]$ un parámetro. Para cada agente $j\in N$, consideramos el vector de pesos $\mathbf{w}=(w_1,\ldots,w_n)$ definido por:

$$w_i=\begin{cases}\gamma, \text{ if } i=j,\\ \dfrac{1-\gamma}{n-1}, \text{ if } i\neq j.\end{cases}$$

Obsérvese que $(w_1,\ldots,w_n)\in[0,1]^n$ y $w_1+\cdots+w_n=1$.

En primer lugar tendremos en cuenta la media aritmética ponderada correspondiente asociada a $\mathbf{w}$ ($\varphi(x)=x$ para todo $x\in[0,1]$). Así, la valoración colectiva del agente *j* viene dada por:

$$F_j\left(P_{1j},\ldots,P_{jj},\ldots,P_{nj}\right)=\gamma P_{jj}+\frac{1-\gamma}{n-1}\sum_{i\neq j}P_{ij}\ , \quad (1)$$

para todo perfil $P\in M_{n\times n}([0,1])$.

En los casos extremos de γ, hay claras interpretaciones en relación al liberalismo:

- Para $\gamma=0$, la valoración colectiva acerca del agente *j* no tiene en cuenta la propia opinión del agente *j* sobre sí mismo y es obtenida como la media aritmética de las opiniones individuales de los agentes, exceptuando la del propio individuo evaluado:

  $$F_j\left(P_{1j},\ldots,P_{jj},\ldots,P_{nj}\right)=\frac{1}{n-1}\sum_{i\neq j}P_{ij}\ ,$$

  para cada perfil $P\in M_{n\times n}([0,1])$.

- Para $\gamma=\frac{1}{n}$, la valoración colectiva sobre el individuo *j* es justamente la media aritmética de las opiniones de todos los agentes:

  $$F_j\left(P_{1j},\ldots,P_{jj},\ldots,P_{nj}\right)=\frac{1}{n}P_{jj}+\frac{1}{n}\sum_{i\neq j}P_{ij}=\frac{1}{n}\sum_{i=1}^{n}P_{ij}\ ,$$

  para cada perfil $P\in M_{n\times n}([0,1])$.

  Ahora la opinión del individuo evaluado es una más y tiene la misma influencia que la del resto de los agentes sobre su propia valoración colectiva.

- Para $\gamma=1$, la valoración colectiva sobre un agente queda determinada por la opinión que el propio agente tiene sobre sí mismo:

  $$F_j\left(P_{1j},\ldots,P_{jj},\ldots,P_{nj}\right)=P_{jj}\ ,$$

  para cada perfil $P\in M_{n\times n}([0,1])$.

  En este caso carecen de importancia las opiniones que el resto de los agentes tienen sobre el individuo a evaluar y la regla gradual independiente asociada a $F_1,\ldots,F_n$ sería la más liberal de todas las de la clase presentada.

Ahora consideraremos un caso más general, en donde se utilizan medias cuasiaritméticas ponderadas.

Consideraremos de nuevo un parámetro $\gamma \in [0,1]$ y para cada individuo $j \in N$ sea $\mathbf{w} = (w_1, \ldots, w_n)$ el vector de pesos:

$$w_i = \begin{cases} \gamma, \text{ if } i = j, \\ \dfrac{1-\gamma}{n-1}, \text{ if } i \neq j. \end{cases}$$

Así, podemos definir:

$$F_j\left(P_{1j}, \ldots, P_{jj}, \ldots, P_{nj}\right) = \varphi^{-1}\left(\gamma\,\varphi\left(P_{jj}\right) + \frac{1-\gamma}{n-1}\sum_{i \neq j}\varphi\left(P_{ij}\right)\right), \tag{2}$$

para cada perfil $P \in M_{n\times n}\left([0,1]\right)$.

Otra vez el valor del parámetro γ proporciona diferentes tipos de liberalismo, pero ahora las opiniones individuales han sido transformadas por medio del automorfismo de orden considerado. Asociadas a estos operadores de agregación tenemos otra clase de reglas graduales independientes. Obviamente, para valores crecientes de γ obtenemos reglas graduales independientes más liberales.

Proposición 1. Para cada automorfismo de orden $\varphi : [0,1] \to [0,1]$ y $\gamma \in [0,1]$, las reglas graduales independientes asociadas a las medias cuasiaritméticas ponderadas definidas en (2) son unánimes, débilmente anónimas, monótonas, compensativas y continuas. Además, si φ es auto-dual, entonces la regla gradual asociada también es auto-dual.

4 REGLAS DE DECISIÓN

Las reglas graduales proporcionan una importante información acerca de los miembros de un colectivo, teniendo en cuenta las opiniones individuales de los agentes. Sin embargo, nuestro objetivo consiste en seleccionar de manera efectiva qué individuos van a formar parte de un determinado subgrupo, de acuerdo al atributo considerado. Para dar solución a este problema, hemos considerado una clase de reglas de decisión asociadas a cada regla gradual independiente. Esta clase está definida por medio de un parámetro que va a permitir usar diferentes umbrales de cualificación en la decisión colectiva.

Definición 9. Una *regla de decisión* es una aplicación $g : M_{n\times n}\left([0,1]\right) \to 2^N$ que asocia a cada perfil $P \in M_{n\times n}\left([0,1]\right)$ un subconjunto de agentes $g(P) \subseteq N$.

Definición 10. Sea $g : M_{n\times n}\left([0,1]\right) \to 2^N$ una regla de decisión.

1. g es *independiente* si para todo par de perfiles $P, Q \in M_{n\times n}\left([0,1]\right)$ y todo $j \in N$ tales que $P_{ij} = Q_{ij}$ para todo $i \in N$:
$$j \in g(P) \quad \Leftrightarrow \quad j \in g(Q).$$
2. g es *unánime* si para todo perfil $P \in M_{n\times n}\left([0,1]\right)$ y todo $j \in N$:
$$\forall i \in N \ P_{ij} = 1 \quad \Rightarrow \quad j \in g(P).$$
3. g es *monótona* si para todo par de perfiles $P, Q \in M_{n\times n}\left([0,1]\right)$ tales que $P_{ij} \leq Q_{ij}$ para cualesquiera $i, j \in N$: $g(P) \subseteq g(Q)$.
4. g es *auto-dual* si para todo perfil $P \in M_{n\times n}\left([0,1]\right)$ y todo $k \in N$:
$$k \in g(P) \ \Rightarrow \ k \notin g(Q),$$
donde $Q_{ij} = 1 - P_{ij}$ para todo $i, j \in N$.
5. g es *no degenerada* si para todo $j \in N$ existen perfiles $P, Q \in M_{n\times n}\left([0,1]\right)$ tales que $j \in g(P)$ y $j \notin g(Q)$.

Definición 11. Sea $f : M_{n\times n}\left([0,1]\right) \to [0,1]^n$ una regla gradual y $\alpha \in [0,1]$. La *α-regla de decisión asociada a f* es la función $f^\alpha : M_{n\times n}\left([0,1]\right) \to 2^N$ definida por

$$f^\alpha(P) = \left\{ j \in N \mid f_j(P) \geq \alpha \right\},$$

para cada perfil $P \in M_{n\times n}\left([0,1]\right)$.

Podemos observar que para cada perfil $P \in M_{n\times n}\left([0,1]\right)$, se verifica $f^0(P) = N$, y si $0 \leq \alpha \leq \beta \leq 1$, entonces $f^\beta(P) \subseteq f^\alpha(P)$.

Hemos de notar que el concepto de α-regla de decisión generaliza el de *reglas de consentimiento* de Samet – Schmeidler [14] (en su caso para las valoraciones dicotómicas).

Observación 3. Dada una regla gradual independiente f con operadores de agregación asociados $F_1, \ldots, F_n$, y $\alpha \in [0,1]$, la α-regla de decisión asociada a f viene definida por:

$$f^\alpha(P) = \left\{ j \in N \mid F_j\left(P_{1j}, \ldots, P_{nj}\right) \geq \alpha \right\},$$

para todo perfil $P \in M_{n\times n}([0,1])$. Obviamente f^α es una regla de decisión independiente para cada $\alpha \in [0,1]$.

En la siguiente proposición mostramos algunas propiedades que las reglas de decisión heredan de las reglas graduales independientes correspondientes.

Proposición 2. Sea $f : M_{n\times n}([0,1]) \to [0,1]^n$ una regla gradual independiente.

1. Si f es unánime, entonces f^α es unánime para todo $\alpha \in [0,1]$.
2. f es monótona si y sólo si f^α es monótona para todo $\alpha \in [0,1]$.
3. Si f es auto-dual, entonces f^α es auto-dual para todo $\alpha \in (\frac{1}{2},1]$.

En el siguiente resultado se pone de manifiesto que las α-reglas de decisión asociadas a las medias cuasiaritméticas ponderadas definidas en (2) satisfacen algunas interesantes propiedades.

Proposición 3. Para cada automorfismo de orden $\varphi : [0,1] \to [0,1]$ y $\alpha, \gamma \in [0,1]$, las α-reglas de decisión asociadas a las reglas graduales independientes definidas por (2) son unánimes, débilmente anónimas y monótonas. Además, f^α es no degenerada para $\alpha > 0$ y auto-dual cuando $\varphi : [0,1] \to [0,1]$ es auto-dual y $\alpha > 0.5$.

Agradecimientos

Este trabajo ha sido parcialmente financiado por la Junta de Castilla y León (Consejería de Educación y Cultura, Proyecto VA057/02), el Ministerio de Ciencia y Tecnología, Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica (I+D+I) (Proyecto BEC2001-2253) y FEDER. Los autores agradecen a Dinko Dimitrov por haberles introducido en la literatura sobre el problema "Who is a J?", así como a Miguel Ángel Ballester por sus comentarios y sugerencias.

Referencias

[1] Aczél, J. (1966): *Lectures on Functional Equations and their Applications*. Academic Press, Nueva York.

[2] Aczél, J. (1987): *A Short Course on Functional Equations*. D. Reidel Publishing Company, Dordrecht.

[3] Aczél, J. – Alsina, C. (1987): "Synthesizing judgments: a functional equations approach". *Mathematical Modelling*, 9/3-5, pp. 311-320.

[4] Billot, A. (2003): "How the liberalism kills democracy or Sen´s theorem revisited". *Public Choice*, 116, pp. 247-270.

[5] Bullen, P.S. – Mitrinovic, D.S. – Vasic, P.M. (1988): *Means and Their Inequalities*. Reidel, Dordrecht.

[6] Calvo, T. – Kolesárova, A. – Komorníková, M. – Mesiar, R. (2002): "Aggregation operators: Properties, classes and constructions models". En Calvo, T. – Mayor, G. – Mesiar, R. (eds.) *Aggregation Operators: New Trends and Applications*. Physica-Verlag, Heidelberg, pp. 3-104.

[7] Dimitrov, D. – Sung, S.C. – Xu, Y. (2003): "Procedural group identification". Tilburg University, CentER Discussion Paper 2003-10.

[8] Fodor, J. – Roubens, M. (1994): *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers, Dordrecht.

[9] García Lapresta, J.L. – Llamazares, B. (2000): "Aggregation of fuzzy preferences: Some rules of the mean". *Social Choice and Welfare*, 17, pp. 673-690.

[10] García Lapresta, J.L. – Llamazares, B. (2001): "Majority decisions based on difference of votes". *Journal of Mathematical Economics*, 45, pp. 463-481.

[11] Kasher, A. (1993): "Jewish collective identity". En D.T. Goldberg – M. Krausz (eds.) *Jewish Identity*. Temple University Press, Philadelphia, pp. 56-78.

[12] Kasher, A. – Rubinstein, A. (1997): "On the question 'Who is a J?' A social choice approach". *Logique & Analyse*, 160, pp. 385-395.

[13] Ovchinnikov, S.V. (1990): "Means and social welfare functions in fuzzy binary relation spaces". En Kacprzyk, J. – Fedrizzi, M. (eds.) *Multiperson Decision Making Using Fuzzy Sets and Possibility Theory*, Kluwer Academic Publishers, Dordrecht, pp. 143-154.

[14] Samet, D. – Schmeidler, D. (2003): "Between liberalism and democracy". *Journal of Economic Theory*, 110, pp. 213-233.

[15] Sung, S.C – Dimitrov, D. (2003): "On the axiomatic characterization of 'Who is a J?'". Tilburg University, CentER Discussion Paper 2003-89.

MODELANDO EL CONSENSO EN PROBLEMAS DE TOMA DE DECISIONES EN GRUPO CON INFORMACIÓN LINGÜÍSTICA MULTIGRANULAR

F. Mata, L. Martínez[0]
Dept. Informática
Universidad de Jaén
23700- Jaén, Spain
e-mail: fmata,martin@ujaen.es

E. Herrera-Viedma
Dept. CC e I.A
Universidad de Granada
18071- Granada, Spain
e-mail: viedma@decsai.ugr.es

F. Chiclana
Centre for Computational Intelligence
De Montfort University
Leicester LE1 9BH, UK,
e-mail: chiclana@dmu.ac.uk

Resumen

Para resolver un problema de toma de decisiones en grupo se aplican dos procesos, un proceso de consenso y un proceso de selección. En el primero los expertos realizan varias rondas de discusión en un intento de aproximar sus opiniones, mientras en el segundo se realiza la selección de las alternativas solución de acuerdo a las opiniones consensuadas.

En este trabajo presentamos un modelo de consenso para problemas de Toma de Decisiones en Grupo (TDG) que permite automatizar el proceso de consenso, asumiendo que los expertos expresan sus opiniones mediante relaciones de preferencia lingüísticas multigranulares. Este modelo lleva a cabo dos tareas, i) mide el consenso entre los expertos y si éste es aceptable cede el paso al proceso de selección, en caso contrario, ii) se generan recomendaciones sobre los cambios en las opiniones de los expertos para aproximar sus posiciones. Éstas son remitidas a los expertos mediante un mecanismo de realimentación automática.

Palabras Clave: Consenso, Información Lingüística Multigranular, Toma de Decisiones en Grupo, Relaciones de Preferencia.

1 INTRODUCCIÓN

Un problema de TDG se puede definir como una situación de decisión en la que intervienen varios expertos que pueden tener diferente percepción o conocimiento sobre el problema y que intentan encontrar la mejor solución al mismo.

[0]Soportado parcialmente por los Proyectos de Investigación TIC 2002-03348 y TIC 2002-11942-E

En algunos problemas de TDG es posible que los expertos expresen sus opiniones de una manera precisa utilizando valores cuantitativos. Sin embargo suele ocurrir que los expertos trabajen con información vaga o imprecisa debido a que han de calificar aspectos cualitativos difícilmente valorables mediante valores numéricos. En estos casos es más conveniente utilizar términos lingüísticos que se adecúan mejor a este tipo de aspectos [3].

El uso del Enfoque Lingüístico Difuso ha dado buenos resultados en la representación y valoración de información cualitativa, utilizando variables lingüísticas para representar la información cualitativa [3, 7].

En los problemas de TDG en los que intervienen un conjunto amplio y variado de expertos, es muy frecuente que éstos pertenezcan a diferentes áreas de estudio o bien tengan un grado distinto de conocimiento sobre el problema. Esto puede obligarles al uso de conjuntos de términos lingüísticos con diferente semántica y/o granularidad para expresar sus opiniones. En estos casos decimos que el problema TDG está definido en un contexto lingüístico multigranular.

Para resolver un problema de TDG es necesario la aplicación de dos procesos (ver Figura 1):

i) Proceso de Consenso: En éste los expertos discuten y cambian sus opiniones con el propósito de alcanzar un grado de acuerdo o consenso apropiado. Aparece la figura del moderador que se encarga de comprobar el nivel de consenso entre los expertos y en caso necesario sugerir los cambios en las opiniones de los expertos para alcanzar un grado de consenso mayor.

ii) Proceso de Selección: Una vez alcanzado el nivel consenso adecuado, se selecciona la mejor o mejores alternativas solución al problema planteado.

En esta contribución pretendemos abordar el proceso de consenso en problemas de TDG en las que los expertos utilizan diferentes conjuntos de términos

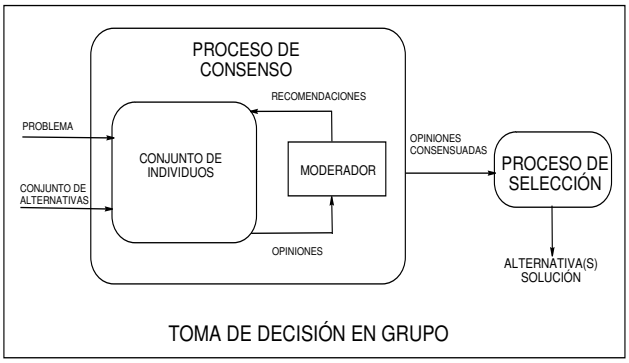


Figura 1: Procesos en los problemas de TDG.

lingüísticos para expresar sus opiniones por medio de relaciones de preferencia lingüísticas multigranulares. Para resolverlo proponemos un modelo de consenso que permite realizar el proceso de consenso y automatizar las funciones de la figura del moderador. Una descripción gráfica de todas las tareas que lleva a cabo el modelo de consenso que proponemos se presenta en la Figura 2. Como se observa, una vez que los expertos han expresado sus opiniones, un proceso de consenso calcula el nivel de consenso entre ellos. Si éste es aceptable, se da por finalizada el proceso de consenso, en caso contrario, se generan una serie de recomendaciones que apoyándose en un mecanismo de realimentación se hacen llegar a los expertos. A continuación se repite de nuevo todo el proceso.

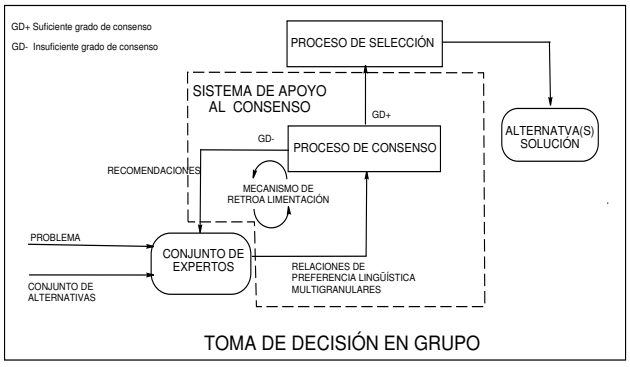


Figura 2: Modelo consenso en TDG con información lingüística multigranular

Para medir el nivel consenso, se usan dos tipos de medidas:

a) Grados de consenso: Miden el grado de consenso entre todos los expertos.
b) Medidas de proximidad: Miden la proximidad o distancia de la opinión de cada experto respecto a la opinión colectiva de todos ellos.

De este modo, el modelo de consenso contempla cuatro fases que se repiten sucesivamente hasta que se alcanza el nivel de consenso deseado o se decide que dicho proceso ha terminado. Estas cuatro fases son las siguientes:

1. Unificación de la información.
2. Cálculo de los grados de consenso.
3. Control del consenso.
4. Generación de recomendaciones o consejos usando las medidas de proximidad.

Esta contribución está estructurada de la siguiente forma: En la Sección 2 se presentan brevemente los problemas de TDG definidos en contextos lingüísticos multigranulares. En la Sección 3 presentamos nuestro modelo de consenso para problemas de TDG en contextos lingüísticos multigranulares. Finalmente, en la Sección 4 presentaremos nuestras conclusiones.

2 PROBLEMAS DE TDG EN CONTEXTO LINGÜÍSTICO MULTIGRANULAR

Los problemas de TDG se caracterizan por la existencia de varios expertos que dan su opinión sobre la mejor solución a un determinado problema. Una forma clásica de expresar su opinión es utilizando relaciones de preferencia [2]. Sea $X = \{x_1, x_2, \ldots, x_n\}$ $(n \geq 2)$ un conjunto de alternativas finito valoradas por un conjunto de expertos $E = \{e_1, e_2, \ldots, e_m\}$ $(m \geq 2)$, donde cada experto e_i proporciona sus preferencias sobre X por medio de relaciones de preferencia, $P_{e_i} \subset X \times X$, donde cada valor $\mu_{P_{e_i}}(x_j, x_k) = p_i^{jk}$ denota el grado de preferencia de la alternativa x_j sobre x_k expresada por el experto e_i.

Tradicionalmente, en los problemas de TDG los expertos han expresado sus opiniones utilizando relaciones de preferencia con valores numéricos [6, 8], $\mu_{P_{e_i}} : X \times X \rightarrow [0, 1]$. Sin embargo, en muchas ocasiones se trabaja con información imprecisa que requiere utilizar valoraciones cualitativas en vez de cuantitativas. En estos casos, el Enfoque Lingüístico Difuso ha sido muy aplicado [3, 7].

En este trabajo asumimos que los expertos utilizan información lingüísticas para expresar sus preferencias, $\mu_{P_{e_i}} : X \times X \rightarrow S$, donde S es un conjunto de términos lingüísticos $S = \{s_0, s_1, \ldots, s_g\}$ caracterizado por su granularidad, $\#(S) = g + 1$ y que satisface que:

1. Es ordenado, $s_i \geq s_j$, si $i \geq$ j.
2. Existe un operador de negación, $\text{Neg}(s_i) = s_j$ tal que $j = g - i$.
3. Existe un operador mximo: $\text{Max}(s_i, s_j) = s_i$, si $s_i \geq s_j$.
4. Existe un operador mínimo: $\text{Min}(s_i, s_j) = s_i$, si $s_i \leq s_j$.

La semántica de los términos se representa mediante números difusos definidos en el intervalo [0,1]. Un ejemplo de asignación de la semática a un conjunto de siete términos mediante números difusos triangulares sería el siguiente:

$P = Perfecto = (.83, 1, 1)$ $\quad VH = Muy_Alto = (.67, .83, 1)$
$H = Alto = (.5, .67, .83)$ $\quad M = Medio = (.33, .5, .67)$
$L = Bajo = (.17, .33, .5)$ $\quad VL = Muy_Bajo = (0, .17, .33)$
$N = Nada = (0, 0, .17)$.

que gráficamente se representa en la Figura 3.

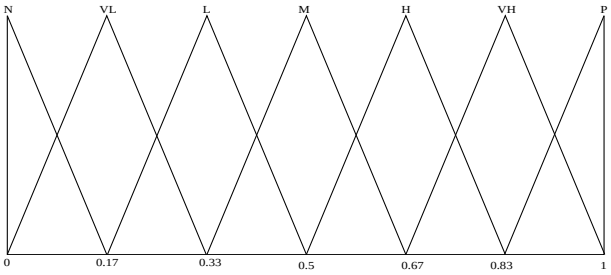


Figura 3: Conjunto de siete etiquetas con su semántica

Destacar que la situación ideal en un problema de TDG sería que todos los expertos e_i utilizasen el mismo conjunto de términos lingüísticos S. Sin embargo puede ocurrir que pertenezcan a diferentes áreas de investigación, y por tanto, tengan distinto grado de conocimiento sobre el problema, y por ello necesidad de usar diferentes conjuntos de etiquetas S_i con diferente granularidad y semántica. En estos casos, necesitaremos herramientas para trabajar y modelar información lingüística multigranular [4].

Entonces, en esta contribución tratamos problemas de TDG en contextos lingüísticos multigranulares, donde cada experto e_i utiliza su propio S_i para expresar sus opiniones. De este modo, asumimos que cada experto e_i proporciona una relación de preferencia P_{e_i} valorada lingüísticamente sobre S_i, donde $p_i^{jk} \in S_i$, $S_i = \{s_0^i, \ldots, s_p^i\}$, siendo $p+1$ la cardinalidad de S_i.

3 UN MODELO DE CONSENSO EN TDG EN CONTEXTO LINGÜÍSTICO MULTIGRANULAR

En esta Sección presentamos un modelo de consenso para modelar los procesos de consenso en los problemas de TDG presentados anteriormente. Este modelo de consenso contempla cuatro fases para modelar los procesos de consenso (ver Figura 4):

1. *Unificación de la información.* Se reciben el conjunto de preferencias lingüísticas multigranulares dadas por los expertos y se encarga de unificar toda la información para expresarla en un único dominio.
2. *Cálculo de los Grados de Consenso.* Se calculan los grados de consenso en los diferentes niveles de representación de las relaciones de preferencia.
3. *Control del Consenso.* Se comprueba si el grado de consenso es suficiente para finalizar el proceso de consenso y comenzar con el proceso de selección o por el contrario continuar con el proceso de consenso.
4. *Generación de Recomendaciones.* Se calculan las medidas de proximidad y utilizando un sistema de recomendaciones orientado se recomiendan los cambios en las opiniones de los expertos.

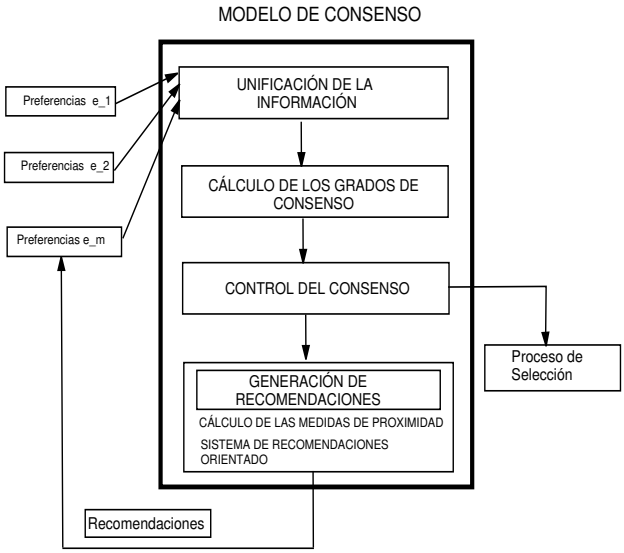


Figura 4: Modelo de consenso en contexto lingüístico multigranular

Cada una de estas fases se describen con mayor detalle en las siguientes subsecciones.

3.1 UNIFICACIÓN DE LA INFORMACIÓN

Dado que en esta contribución estamos trabajando en contextos lingüísticos multigranulares, en primer lugar, se lleva a cabo la unificación de la información, transformando todas las preferencias lingüísticas multigranulares a un solo dominio que denominaremos conjunto de términos lingüísticos básicos S_T [4]. Para realizar las transformaciones utilizamos funciones de transformación multigranulares, $\tau_{S_i S_T}$, que convierten los términos lingüísticos pertenecientes a S_i en conjuntos difusos definidos en S_T.

$$\tau_{S_i S_T} : S_i \longrightarrow F(S_T), \quad \forall\ S_i$$

$$\tau_{S_i S_T}(p_i^{jk}) = \{(c_h, \alpha_{ih}^{jk})\ /\ h = 0, \ldots, g\}.$$

donde al menos $\exists\ \alpha_{ih}^{jk} > 0$

Tras este proceso de unificación y asumiendo que cada conjunto difuso se representa mediante sus grados de pertenencia, $(\alpha_{i0}^{jk}, \ldots, \alpha_{ig}^{jk})$, dispondremos de una relación de preferencia P_{e_i} para cada experto e_i, donde cada preferencia lingüística p_i^{jk} es un conjunto difuso:

$$P_{e_i} = \begin{pmatrix} p_i^{11} = (\alpha_{i0}^{11}, \ldots, \alpha_{ig}^{11}) & \cdots & p_i^{1n} = (\alpha_{i0}^{1n}, \ldots, \alpha_{ig}^{1n}) \\ \vdots & \cdots & \vdots \\ p_i^{n1} = (\alpha_{i0}^{n1}, \ldots, \alpha_{ig}^{n1}) & \cdots & p_i^{nn} = (\alpha_{i0}^{nn}, \ldots, \alpha_{ig}^{nn}) \end{pmatrix}$$

Una vez que las preferencias de todos los expertos han sido unificadas en un mismo dominio, entonces se procede a calcular los grados de consenso entre los expertos.

3.2 GRADOS DE CONSENSO ENTRE EXPERTOS

Los grados de consenso miden el nivel de consenso entre los expertos. Para medirlo, se calcula el grado de consenso en tres niveles diferentes: nivel de pares de alternativas, nivel de alternativas y nivel de relación [5]. Para obtenerlos necesitamos una matriz de consenso que se obtiene agregando la distancia entre las opiniones de todos los expertos, comparándolas dos a dos.

La distancia entre los expertos se calcula utilizando matrices de distancia para cada dos expertos, e_i, e_j. Los valores de estas matrices expresan la distancia entre dos preferencias p_i^{lk}, p_j^{lk}. Estos valores se obtienen utilizando la función de diferencia que se presentó en detalle en [5]. Esta función calcula la distancia utilizando el valor central de cada preferencia (conjunto difuso), cv_i^{lk}, que representa la distribución de la información del experto e_i en el conjunto difuso $p_i^{lk} = (\alpha_{i0}^{lk}, \ldots, \alpha_{ig}^{lk})$. Se calcula como:

$$cv_i^{lk} = \frac{\sum_{j=0}^{g} index(s_j^i) \cdot \alpha_{ij}^{lk}}{\sum_{j=0}^{g} \alpha_{ij}^{lk}}, \qquad (1)$$

donde $index(s_j^i) = j$.

Usando los valores centrales obtenemos la distancia entre dos preferencias p_i^{lk}, p_j^{lk} como:

$$d(p_i^{lk}, p_j^{lk}) = 1 - \left| \frac{cv_i^{lk} - cv_j^{lk}}{g} \right| \qquad (2)$$

Los grados de consenso se obtienen realizando los siguientes cálculos:

1. Cálculo de los valores centrales, cv_i^{lk}: Calculamos los valores centrales de todas las preferencias de los expertos aplicando la expresión (1) para $i = 1, \ldots, m\ ,\ l, k = 1, \ldots, n$ y $l \neq k$.

2. Cálculo de la matrices de distancia, $DM_{i,j}$: Calculamos una matriz de distancia $DM_{i,j}$ para cada par de expertos e_i, e_j, donde cada elemento representa la distancia entre pares de alternativas:

$$dm_{ij}^{lk} = d(p_i^{lk}, p_j^{lk}) = 1 - \left| \frac{cv_i^{lk} - cv_j^{lk}}{g} \right| \qquad (3)$$

3. Cálculo de la matriz de consenso, CM: Agregando todas las $DM_{i,j}$ a nivel de pares de alternativas obtenemos la matriz de consenso CM. Utilizamos como operador de agregación, ϕ, la media aritmética.

$$cm^{lk} \quad = \quad \phi(dm_{ij}^{lk}, i, j = 1, \ldots, m\ ,\ i < j)$$

$\forall\ l, k = 1, \ldots, n.$

4. Cálculo de los grados de consenso: Calculamos el grado de consenso entre pares de alternativas, alternativas y expertos.

   Consenso sobre pares de alternativas cp^{lk}: Evalúa el consenso sobre cada par de alternativas x_l, x_k. En este caso coinciden con los valores de CM.

$$cp^{lk} = cm^{lk}, \quad para \quad l, k = 1, \ldots, n \quad y \quad l \neq k$$

   Consenso sobre las alternativas, ca^l: Evalúa el consenso sobre cada alternativa x_l

$$ca^l = \frac{\sum_{k=1}^{n} cp^{lk}}{n} \qquad (4)$$

   Consenso sobre la relación, ce: Mide el grado de consenso total considerando todos los pares de la relación.

$$ce = \frac{\sum_{l=1}^{n} ca^l}{n} \qquad (5)$$

   Si el valor ce es pequeño, el grado de consenso es bajo y por lo tanto las opiniones de los expertos están bastante alejadas. Este valor es utilizado para finalizar el proceso de consenso cuando se alcance el consenso deseado.

3.3 CONTROL DEL CONSENSO

El valor ce y un parámetro que denominaremos umbral de consenso ($\gamma/\gamma \in [0, 1]$) se usan para controlar el grado de consenso tal y como se describe en la Figura 5.

Este proceso se lleva a cabo según las siguientes reglas:

- Si $ce >= \gamma$, entonces los expertos han alcanzado el grado de consenso deseado y puede comenzar el proceso de selección de las alternativas solución.

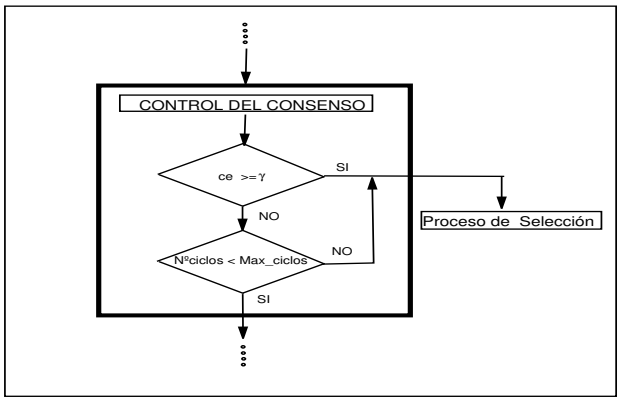


Figura 5: Control del consenso

- Si $ce < \gamma$, la fase de discusión en el que se basa el proceso de consenso debe continuar.

Es necesario indicar que si el valor del umbral de consenso γ es muy alto, es posible que los expertos nunca alcancen un grado de consenso igual o superior a γ y por lo tanto se entre en un bucle infinito. Para evitar esta situación, hemos definido un parámetro denominado Max_ciclos que controla si se han realizado el número máximo de rondas de consenso previstas previamente. Si se produce esta situación, terminará el proceso de consenso dando lugar al comienzo del proceso de selección.

3.4 GENERACIÓN DE RECOMENDACIONES

El resultado de las operaciones que se llevan a cabo en esta fase es un conjunto de recomendaciones donde se sugiere la dirección de los cambios en las opiniones de los expertos más alejados del consenso. Para hacer esto, primero es necesario calcular la medidas de proximidad de cada experto y seguidamente aplicar una serie de reglas de recomendación. Cada una de estas operaciones se describen en las dos siguientes subsecciones.

3.4.1 Cálculo de las medidas de proximidad

Las medidas de proximidad evalúan la distancia entre la opinión de un experto y la opinión colectiva de todos ellos. Igual que con los grados de consenso, las medidas de proximidad también las calculamos respecto a pares de alternativas, alternativas y relación. Para obtenerlas, necesitamos previamente obtener una relación de preferencia colectiva P_{e_c} que represente la opinión colectiva de todos los expertos y que se calcula agregando todas las opiniones individuales siguiendo el mismo modelo presentado en [4],

$$P_{e_c} = \begin{pmatrix} p_c^{11} & \cdots & p_c^{1n} \\ \vdots & \cdots & \vdots \\ p_c^{n1} & \cdots & p_c^{nn} \end{pmatrix}$$

donde cada p_c^{jk} es un conjunto difuso.

Las medidas de proximidad se obtienen realizando los siguientes cálculos:

1. Cálculo de las matrices de proximidad, PM_i: Para cada experto e_i se calcula una matriz de proximidad PM_i que representa la distancia entre la opinión individual de este experto y la opinión colectiva

$$PM_i = \begin{pmatrix} d(p_i^{11}, p_c^{11}) & \cdots & d(p_i^{1n}, p_c^{1n}) \\ \vdots & \cdots & \vdots \\ d(p_i^{n1}, p_c^{n1}) & \cdots & d(p_i^{nn}, p_c^{nn}) \end{pmatrix}$$

2. Cálculo de las medidas de proximidad: Calculamos las medidas de proximidad en tres niveles: pares de alternativas, alternativas y expertos:

   Proximidad del par, pp_i^{jk}: Mide la proximidad entre cada par de alternativas. Coincide con los valores de PM_i

   $$pp_i^{lk} = pm_i^{lk}$$

   para $l, k = 1, \ldots, n$ y $l \neq k$

   Proximidad de la alternativa, ap_i^l: Mide la proximidad en cada alternativa x_l

   $$ap_i^l = \frac{\sum_{k=1}^{n} pp_i^{lk}}{n} \quad (6)$$

   Proximidad en la relación, ep_i: Mide la proximidad en la relación

   $$ep_i = \frac{\sum_{l=1}^{n} ap_i^l}{n} \quad (7)$$

   Si el valor ep_i está próximo a 1, esto significa que su opinión coincide con la del resto del grupo, en caso contrario, si está próximo a 0, la opinión del experto está muy alejada del resto.

3.4.2 Sistema de Recomendaciones Orientado

Nuestro modelo de consenso consta de un módulo que hemos denominado Sistema de Recomendaciones Orientado. Define una serie de reglas que permiten identificar las opiniones de los expertos más alejadas del consenso y sugerir la dirección de los cambios, con el propósito de conseguir gradualmente que acerquen sus opiniones y alcancen el consenso deseado. Para conseguir este propósito lleva a cabo los siguientes pasos:

1. Identificación de los expertos más alejados del consenso. Debemos decidir el número de expertos que deben cambiar de opinión, por ejemplo el 50%. A continuación ordenamos todos los expertos por su proximidad y seleccionamos los más alejados.
2. Identificación de las alternativas a cambiar. Estos expertos han de cambiar solo las preferencias dadas sobre las alternativas cuyo grado de consenso sea menor que el umbral de consenso, $ca^l < \gamma$. De esta forma evitamos hacer cambios sobre alternativas con un aceptable grado de consenso.
3. Identificación de los pares de alternativas a cambiar. Una vez hemos identificado el experto e_i y alternativa x_l a cambiar, todas las preferencias p_i^{lk} $(k = 1, \ldots, n)$ cuya proximidad $pp_i^{lk} < \beta$ deben ser modificadas. El parámetro β actúa como un umbral de proximidad que permite seleccionar las preferencias más alejadas de la opinión colectiva.
4. Aplicación de las reglas de dirección (DR). Para finalizar esta fase, tan solo nos falta averiguar la dirección de los cambios en los pares de alternativas p_i^{lk} que deben cambiar. Este problema se resuelve definiendo dos pares de parámetros de dirección que llamaremos principal (ml) y secundario (sl). Un par para la opinión del experto (eml, esl) y otro par para la opinión colectiva (cml, csl). Cada uno de los parámetros de dirección toman respectivamente el valor y posición de las dos etiquetas mayores del conjunto difuso de la preferencia del experto $(eml_{pos}, eml_{val}, esl_{pos}, esl_{val})$ y preferencia colectiva $(cml_{pos}, cml_{val}, csl_{pos}, csl_{val})$. Los valores de estos parámetros son utilizados por las siguientes reglas de dirección:

DR.1. Si $eml_{pos} > cml_{pos}$ entonces el experto debe disminuir la valoración dada sobre el par de alternativas x_l, x_k.

DR.2. Si $eml_{pos} < cml_{pos}$ entonces el experto debe aumentar la valoración dada sobre el par de alternativas x_l, x_k.

DR.3. Si $eml_{pos} = cml_{pos}$ entonces aplicar las mismas reglas pero comparando eml_{val} y cml_{val}.

DR.4. Si ($eml_{pos} = cml_{pos}$ y $eml_{val} = cml_{val}$), entonces aplicar las mismas reglas para el parámetro de dirección secundario sl.

4 CONCLUSIONES

En este trabajo hemos presentado un modelo de consenso que permite modelar el proceso de consenso en problemas de TDG en contextos lingüísticos multigranulares. Este modelo utiliza medidas de consenso y proximidad para evaluar el nivel de acuerdo entre expertos e identificar los expertos más alejados del consenso.

Como principal caracterstica cuenta con un sistema de recomendaciones orientado que genera un conjunto de recomendaciones en las que se sugiere a expertos concretos que cambien sus opiniones más discrepantes, el cual le permite suplantar la figura del moderador en los procesos de consenso en TDG.

Referencias

[1] P.P. Bonissone y K.S. Decker. Selecting Uncertainty Calculi and Granularity: An Experiment in Trading-off Precision and Complexity, en: L.H. Kanal y J.F. Lemmer, Eds., *Uncertainty in Artificial Intelligence*, Pág. 217-247, North-Holland, 1986.

[2] J. Fodor y M. Roubens. Fuzzy Preference Modelling and Multicriteria Decision Support. *Kluwer Academic Publishers*, Dordrecht, 1994.

[3] F. Herrera, E. Herrera-Viedma y J.L. Verdegay. A Sequential Selection Process in Group Decision Making with Linguistic Assessment, *Information Sciences*, Pág. 223-239, **85**, 1995.

[4] F. Herrera, E. Herrera-Viedma y L. Martínez. A fusion aproach for managing multi-granularity linguistic term sets in decision making. *Fuzzy Sets and Systems*, Pág. 43-58, **114**, 2000.

[5] E. Herrera-Viedma, F. Mata, L. Martínez y F. Chiclana. Consensus Measurements in Multi-Granular Linguistic Group Decision Making. Actas del congreso, *Modelos de Decisión para la Inteligencia Artificial*. Barcelona (España), 2004.

[6] J. Kacprzyk, H. Nurmi, y M. Fedrizzi, Eds. *Consensus under Fuzziness.* Kluwer Academic Publishers, Boston, 1997.

[7] R.R. Yager. An Approach to Ordinal Decision Making. *Int. J. of Approximate Reasoning*, **12** Pág. 237-261, 1995.

[8] H.J. Zimmermann. Fuzzy Set Theory and Its Applications. *Kluwer Academic Publishers*, Boston, 1996.

Applying fuzzy logic to model user preferences on multimedia sessions *

Juan A. Botía, Mercedes Valdés, Pedro M. Ruiz and Antonio F. Gómez-Skarmeta
Departmento de Ingeniería de la Información y las Comunicaciones
Universidad de Murcia

Resumen

This work is enclosed inside a more general task which is the proposal of a new approach based on intelligent agents for the initiation, control and adaptation of a multimedia session. The overall contribution of this work is an intent to fill the gap between what the different users implied consider as their preferred multimedia session characteristics and the actual multimedia session established and maintained. Particularly, we focus here on specifying user preferences on media by using fuzzy logic. This preferences will guide the process of scoring, from the user point of view, any possible multimedia session configuration. In turn, this will allow for a further negotiation on possible session to establish between communicating peers.

1 Introduction

Multi-agent systems are called to play an important role in the new information systems coming into the third millennium. The kind of applications we refer to will have new features like, for example, they will be user-centric, adaptable to changing conditions and context aware [2]. Moreover, they will have to be able to execute in more and more heterogeneous environments both in terms of data transport telecommunication networks and devices connected to that networks and hosting the applications. Examples of such kind of systems are [9, 15]. This examples emphasize two simple but powerful ideas: the first idea is that computers should be present in the typical daily tasks at the work place and the second one is that applications should tightly adapt to the user.

This article deals with the *user-centric* feature. It shows our approach to allow an user to specify preferences on the media used for potential multimedia sessions which could be established with the rest of users of our application. The general task we are involved in is the management of a complete multimedia session between applications built by using intelligent agents, inside of a system we are currently developing in our research group. This is basically a multi-agent system compound by agents which act on behalf of the user (i.e. user agents) and the typical user is either a pupil or a teacher of our Computer Science Faculty. The assistance given by these agents is summarized by the following main features: (1) facilities for the pupil to obtain appointments with professors to make inquiries, (2) facilities for discussion and scientific collaboration between researchers and pupils, (3) facilities for concrete collaborations between professors and pupils and (4) the most important issue we are pursuing: to augment professors availability by providing access to them at any time and any place of the Faculty through the use of a variety of heterogeneous devices such a personal computer, a laptop or a PDA (Personal Digital Assistant).

The rest of the article is organized as follows: in section 2 we introduce the issue of the negotiation of the multimedia session between two peers from a multi-agent perspective. In section 3 our fuzzy modeling [1] approach for allowing an user to express his interests of the different media involved in a multimedia session is explained. Finally, section 4 gathers all conclusions extracted from the work and points some future works.

2 The negotiation process

When a multimedia session is going to be established between two or more users, each user will access the session with a concrete terminal. Each terminal in

*Part of this work has been funded by the Spanish Science and Technology Ministry (MCYT) by means of the 'Ramon y Cajal' workprogramme as well as the *SAM (TIC2002-04531-C04-03)* project.

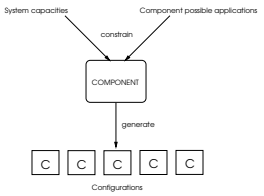


Figura 1: Obtaining configurations from the components and system capacities

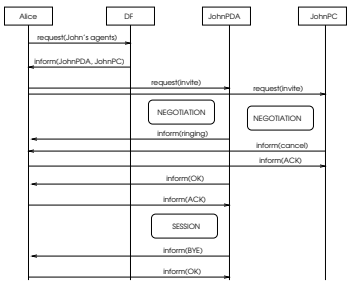


Figura 2: Session establishment and negotiation between an initiator and two possible peers.

turn will be characterized by the hardware and software resources it needs. Resources define the limitations about what, and what not, can be visualized, printed and captured with the terminal (or device). **System capacities** are defined as all the features offered and limited by the system. Moreover, when the session is going to be established, the different media that will be used have to be specified. **Session Components** are the different communication elements that will be used. This paper considers only audio and video components. A component is defined by (a) the functionality it provides and (b) one of more ways to provide the functionality. Each one of the different ways of providing functionality is named a **configuration**. The relation between these three concepts is represented at figure 1. The negotiation process generates two different kinds of configurations: potential and real configurations. A potential configuration is the set of possible configurations which can be adopted by the system. A real configuration is a concrete instantiation of one of the potential ones.

When designing the negotiation mechanism form session establishment, two different problems must be solved in turn: (1) the design of a mechanism by means of which the user can specify preferences on audio and video and (2) the definition of a protocol and negotiation strategy to be used by the agents implied in the determination of the real configuration the session will have meanwhile reaching an acceptable satisfaction level for these agents. This article is devoted to explain our approach to allow an user specifying preferences on audio and video by means of fuzzy logic (i.e., the first problem). However, the negotiation mechanism will be introduced now to make our proposal more understandable.

To better locate the negotiation process [11, 12] we will use a sequence diagram, like the one appearing at figure 2. In that figure, the user agent of Alice, let it be denoted as `Alice`, wants to establish a multimedia session with John through one of his user agents. At the moment of initiating the interaction, John is connected to the system by `JohnPDA` and `JohnPC`, corresponding to an user agent at a PDA and other user agent running at its desktop personal computer, respectively. The first interaction is initiated by `Alice`, requesting the directory facilitator (DF) the addresses of John's agents. After that, `Alice` sends an INVITE command to `JohnPDA` and `JohnPC`. This command, and the rest of commands of the interaction, is one of the possible commands of the SIP [10] Internet protocol, used for session management. Once this is done, the negotiation process starts. This task can be seen as a search process [3], in which the three agents move from point to point of the search space until `Alice` meets with one of the other two at one of these points. Obviously, each point is a possible agreement and it will be compound of the description of the multimedia session the users will maintain for communication and an utility value each agent will obtain on reaching the agreement at the specific point. But, how to score each possible point? This score will depend on user preferences on the media implied in the session. We will use a fuzzy inference system (FIS) to model this issue and compound with this model the ultimate utility function needed by each agent to enable the negotiation. Once `Alice` decides that it wants to establish the session with `JohnPDA`, the session is normally established and, eventually, terminated.

3 Modeling user preferences with fuzzy logic

Consider the user John. In principle, John is able of establishing a session with any other user of the system. Generically, John will have **generic preferences** on using the system. A preference of this kind could be, for example, the window look-and-feel style he likes for the application. He can also be connected to the system by a number of different devices, a PDA, a laptop or a personal computer, running a FIPA platform like JADE or LEAP. For each kind of device he will also have **device preferences**. For example, two different sound codecs would be preferred in a PDA and a Desktop Computer. Finally, communication with the

Parameter	Values	Explanation
AUDCOD	PCM, G711-u, G722, GSM	Audio codec
VIDCOD	MJPEG, H.263	Video codec
FSIZE	CIF, QCIF, 160x128	Size of video frames
FPS	$\{0, \ldots, 12\}$	Frames by second sent
QFVIDEO	[0,100]	Quantify f. of video codec

Tabla 1: Parameters considered as interesting topics for the user

other users will be done by using concrete applications like, for example, videoconferencing, chat, e-mail and so on. He will also have **media preferences** for that. This paper focuses only on media preferences.

3.1 Audio and video user preferences

Once the media are fixed to audio and video, the adaptation level we have worked on has into account the five parameters appearing at table 1. Obviously, we can make a correspondence between each one of this parameters and a topic over which the user could express his interest about. In fact, the user will express his preferences over audio and video by assigning a number in $[0, 1]$ for each topic of table 1. The higher the number, the higher his interest.

The link between specification of preferences by the user and the negotiation phase is the utility function used by each agent in the negotiation. This function will provide a valuation of utility in $[0, 1]$ for each real configuration that could be adopted as a possible multimedia session set up.

Given a real configuration, the agent needs to calculate an acceptance degree for it (**utility**), depending on what the user considers as interesting configurations. For this aim the agent must take into account the interest of the user about each configuration parameter. For example, if the interest of the user about the audio codec is high, then a PCM video codec would be the most suitable option (this is the best codec available), at least taking into account this particular parameter. Besides, the user will have certain preferences about the remaining parameters and they all must contribute to the calculus of the global utility for the real configuration. In order to model how the user interest about each one of the parameters contributes to the global utility of a given real configuration, we have used **fuzzy inference systems** (FIS) [14, 6].

3.2 Related work

But before we get inside the details we will point to some related work we have found in the literature. If we look at the problem from a high abstraction level, what we have to find is a function which evaluates different combinations of values for a number of attributes. Techniques for evaluating different combinations of attribute values can be drawn from the domain of Multi-attribute Utility Theory [5, 8]. One of the most well known application of these kind of techniques is the evaluation of a given product in an electronic shopping environment. For example, in [13] an adaptive recommender system is proposed. It supports the user in finding interesting products in an electronic product catalog. This system gives a measure of how well the catalog entries match the user preferences taking into account weights representing the relative importance between attributes and functions quantifying their respective utility. Finally, the evaluation of each product is obtained combining the partial utilities by means of an aggregation function. Another example can be found in [7], where a multiagent trading environment is proposed to support electronic trading. The buyer expresses his preferences in terms of *don't care, wont accept, must have*, or even a percentage of interest. A purchasing adviser agent analyzes the buyer's requirements and selects the most suitable product from a list designed by a merchant agent ins a similar manner to that of the previous example. However the most interesting issue there is the way in which the merchant agent generates the list of suitable products. It uses fuzzy logic to match the user preferences with every product. If a product exhibits a feature that exactly matches the buyer's requirements for that feature, then the product is given a score equal to the weight the buyer has attached to that feature. If not, fuzzy rules are applied to enable the product to score some fraction of the weight. Another related example can be found in [4] in which user preferences are expressed in terms of linguistic labels as: *like, interest, indifference* and *dislike*.

3.3 Fuzzy inference systems

A FIS is a computational tool to describe systems by means of its input-output relations. Its foundations can be found in fuzzy sets and fuzzy reasoning theory. It structures the knowledge using IF-THEN rules in terms of fuzzy quantities.

Three main elements can be distinguished in a FIS. The first is the **rules base**. For example, a FIS involving two inputs and one output usually has the form:

Rule 1: IF x is A_1 AND y is B_1 THEN z is C_1
ALSO
Rule 2: IF x is A_2 AND y is B_2 THEN z is C_2,
ALSO
...
ALSO
Rule r: IF x is A_r AND y is B_r THEN z is C_r

where $x \in X$ and $y \in Y$ are the input variables, $z \in Z$ is the output one and A_i, B_i and C_i with $i = 1, \ldots, r$, are fuzzy sets. Since these fuzzy amounts can have an associated linguistic term [16], the second element of

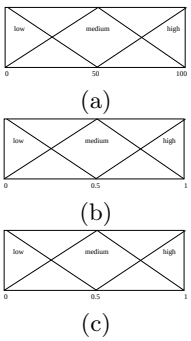


Figura 3: The dictionary: (a) for video quality factor, (b) for user interest; (c) for utility.

a FIS is a **dictionary** that associates each linguistic term with a fuzzy set. For example, each fuzzy set in the previous rules base could be associated to a linguistic label as *high*, *low*, etc.. So the dictionary links each linguistic term with its definition as fuzzy set. In figure 3, the dictionary for video quality factor, user interest, and global utility fuzzy variables can be found.

Finally, the third element is a **fuzzy reasoning method (FRM)** by means of which an output is inferred starting from a given input. Basically, a concrete FRM explains how to use a given input and the rule base to produce an output.

For our problem five FISs, F_i being $i = 1, \ldots, 5$, have been designed. F_i models how the user interest in parameter i (corresponding to the i-th row of table 1) affects to the global utility of a given real configuration. Therefore, the final utility is obtained by aggregating (fuzzy union) the fuzzy outputs of the five FISs. However, since this output is a fuzzy value and we want to obtain a real one, a *defuzzification* step must be done. We use for that the so called *center of area* (COA) defuzzification method.

3.4 A FIS for the Audio Codec

An an example of using a FIS for a discrete topic, the following is the FIS FIS_1 that models the contribution to utility from the audio codec, and user interest on the topic:

Rule 1: IF $interest_{audcod}$ is Low AND $conf_{audcod}$ is GSM
THEN $utility$ is $Medium$
ALSO
Rule 2: IF $interest_{audcod}$ is Low AND $conf_{audcod}$ is $G711$
THEN $utility$ is $Medium$
ALSO
Rule 3: IF $interest_{audcod}$ is Low AND $conf_{audcod}$ is $G712$
THEN $utility$ is $Medium$
ALSO
Rule 4: IF $interest_{audcod}$ is Low AND $conf_{audcod}$ is PCM
THEN $utility$ is $Medium$
ALSO
Rule 5: IF $interest_{audcod}$ is $Medium$ AND $conf_{audcod}$ is GSM
THEN $utility$ is Low
ALSO
Rule 6: IF $interest_{audcod}$ is $Medium$ AND $conf_{audcod}$ is $G711$
THEN $utility$ is $Medium$
ALSO
Rule 7: IF $interest_{audcod}$ is $Medium$ AND $conf_{audcod}$ is $G712$
THEN $utility$ is $Medium$
ALSO
Rule 8: IF $interest_{audcod}$ is $Medium$ AND $conf_{audcod}$ is PCM
THEN $utility$ is $Medium$
ALSO
Rule 9: IF $interest_{audcod}$ is $High$ AND $conf_{audcod}$ is GSM
THEN $utility$ is Low
ALSO
Rule 10: IF $interest_{audcod}$ is $High$ AND $conf_{audcod}$ is $G711$
THEN $utility$ is Low
ALSO
Rule 11: IF $interest_{audcod}$ is $High$ AND $conf_{audcod}$ is $G712$
THEN $utility$ is $Medium$
ALSO
Rule 12: IF $interest_{audcod}$ is $High$ AND $conf_{audcod}$ is PCM
THEN $utility$ is $High$

where $interest_{audcod}$ refers to the real interest the user has on the audio codec, $conf_{audcod}$ refers to the particular audio codec used and $utility$ is the contribution of the topic (i.e. audio codec) to the global utility. The first four rules express that, when the user in not interested in the audio codec (i.e. the user does not mind if the audio sounds good or bad), the utility of the given configuration will be medium in such a way that it does not interfere a possible particular preference of the other side. Rule labeled as $Rule_5$ express that the higher the interest in the audio codec, the lower the utility of a configuration with GSM (as GSM is the worst audio codec that can be used at the system). Finally, rule number 12 express that the most interesting audio codec is PCM.

We have considered three linguistic labels to express the user interest about a configuration parameter: *Low, Medium* and *High*. Their associated fuzzy sets are illustrated in figure 3(b). The partition expressing the different levels of utility is shown in figure 3(c). Finally we must remark that the values of the parameter corresponding to the audio codec are discrete ones but they can be managed as singleton fuzzy sets when they are involved in fuzzy rules.

3.5 A FIS for the video quality factor

As an example of using a FIS for a continuous topic, the following FIS_5 is the one concerning the quality factor for the video codec, which takes values in

$[0, 100]$:

Rule 1: IF $interest_{qfvideo}$ is Low AND $conf_{qfvideo}$ is $High$
THEN $utility$ is $Medium$
ALSO
Rule 2: IF $interest_{qfvideo}$ is Low AND $conf_{qfvideo}$ is $Medium$
THEN $utility$ is $Medium$
ALSO
Rule 3: IF $interest_{qfvideo}$ is Low AND $conf_{qfvido}$ is Low
THEN $utility$ is $Medium$
ALSO
Rule 4: IF $interest_{qfvideo}$ is $Medium$ AND $conf_{qfvideo}$ is $High$
THEN $utility$ is $Medium$
ALSO
Rule 5: IF $interest_{qfvideo}$ is $Medium$ AND $conf_{qfvideo}$ is $Medium$
THEN $utility$ is $Medium$
ALSO
Rule 6: IF $interest_{qfvideo}$ is $Medium$ AND $conf_{qfvideo}$ is Low
THEN $utility$ is Low
ALSO
Rule 7: IF $interest_{qfvideo}$ is $High$ AND $conf_{qfvideo}$ is $High$
THEN $utility$ is $High$
ALSO
Rule 8: IF $interest_{qfvideo}$ is $High$ AND $conf_{qfvideo}$ is $Medium$
THEN $utility$ is $Medium$
ALSO
Rule 9: IF $interest_{qfvideo}$ is $High$ AND $conf_{qfvideo}$ is Low
THEN $utility$ is Low

where $interest_{qfvideo}$ refers to the real interest the user has on the quality factor for video, $conf_{qfvideo}$ refers to the real value for the quality factor and $utility$ is the contribution of the topic (i.e. quality factor for video) to the global utility. The linguistic terms that model the domain of quality factor for video, are shown in figure 3(a). They provide a concrete meaning to the rules in FIS_5. The first three rules lead to a medium utility in order to not interfere with the interest of the other peer for the same parameter. The rest of the rules express that the higher the interest in the quality factor and the higher the quality factor, the higher the utility of the configuration.

3.6 Fuzzy inference systems at work

Figure 4 shows the defuzzified outputs (i.e. real values) of FIS_1. It illustrates the utility variation of a real configuration taking into account only the interest of the user on the audio codec topic. The solid line represents this utility given that the current real configuration has a value of GSM for the audio codec. For low values of interest from the user on the audio codec, the only rules of FIS_1 that fire are $Rule_1$ and $Rule_5$. Hence, when the interest is low then the utility takes middle values but when the interest reach 0.5 value, then the utility decreases to 0.2. At this point the only fired rules are $Rule_5$ and $Rule_9$ maintaining the utility in low level. The slight increment at point 0.5 is due to greater firing strength achieved when the two antecedents of each rule are fulfilled with a 0.5 degree. The dotted line represents the utility given that the current value for the audio codec is PCM. As it is expressed in FIS_1 it can be observed how the utility grows from a medium value to 1 (the higher one) as user interest in the audio codec increases. When the audio codec is G712 (a low-medium quality codec), if user interest in the topic is low or medium then it

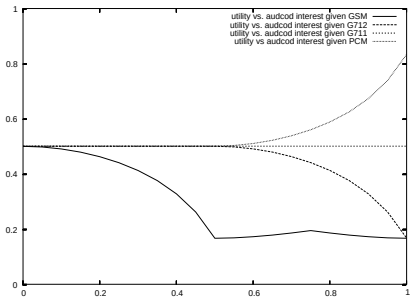
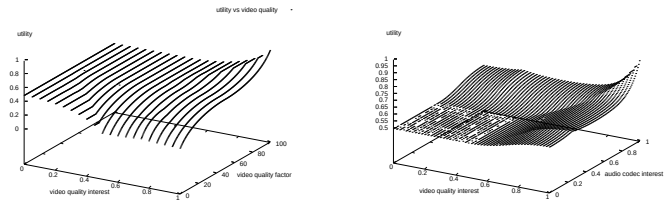


Figura 4: FIS_1 outputs (i.e. partial utility) depending on the audio codec and user interest (x axis) on the topic



(a) (b)

Figura 5: (a) FIS_5 outputs (i.e. partial utility) depending on the quality factor and user interest on the topic, (b) Global utility given that quality factor is 100 and audio codec is GSM, depending on user interest in each topic.

is considered pretty suitable. However, its utility decreases as user interest in the topic grows. Finally, we can see that a G711 configuration has a 0.5 utility whatever the user interest could be.

Graph (a) at figure 5 shows the output of FIS_5 once it is defuzzified. It shows how the utility of a given configuration changes when user interest about the quality factor topic is taken in mind. When the interest in this parameter is low, any value for the quality factor is assigned a medium utility value. As the interest grows the utility of real configurations having a low quality factor decreases to 0. Finally, when user interest in this parameter is high, multimedia session actual configurations with greater values for the quality factor are assigned the highest utility values (1).

Now that it have been shown the relation between utility and interest for a discrete domain, the audio codec, and for a continuous one, the quality factor, let us illustrate how the partial utilities are aggregated to infer the global utility. We have to remark that there is one FIS for each of the five configuration parameters and their outputs must be aggregated to infer the global

utility but, to simplify, in this paper we only consider FIS_1 and FIS_5 tied to the audio codec and quality factor, respectively. Graph (b) at figure 5 shows the variation of the global utility in relation to the variation of the two topics, given a configuration with a PCM audio codec (the best one) and a quality factor of 100 (the highest value). When interest in both parameters is low this configuration is given a medium value of utility. In this way if the other user likes this configuration then the agent of the first user should accept it, since its user does not care about the value of these two parameters. Besides, as user interest on quality factor grows, the utility also grows, since 100 is the highest value for the quality factor. Similarly, as user interest on audio increases then the utility of the given configuration improves because PCM is the best available audio codec. Finally, it can be observed that this configuration is assigned the higher level of utility (1) when then interest in each one of the two parameters is high.

4 Conclusions and future trends

In this paper, the use of fuzzy modeling to allow an user to express his preferences on the different media involved in a multimedia session is introduced. We have demonstrated the usefulness of fuzzy logic to model user interests on the topics which participate in the description of an audio and video session. With a fuzzy inference system, we can assign utility values for multimedia configurations which perfectly reflect what the user consider as important for a session. Utility values are necessary to enable negotiation between agents to decide which is the most convenient multimedia configuration for both.

Future works include considering not only media preferences but also device and generic preferences following the same approach as the one introduced in this paper.

Referencias

[1] Robert Babuska. *Fuzzy Modeling and Identification*. PhD thesis, Technische Universiteit Delft, 1997.

[2] W.A IJsselsteijn G. Riva, F. Davide, editor. *Being There: Concepts, effects and measurement of user presence in synthetic environments*. IOS Press, Amsterdam, The Netherlands, 2003.

[3] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: prospects, methods and challenges. *Int. J. of Group Decision and Negotiation*, 2(10):199–215, 2001.

[4] K. Kamyab, F. Guerin, P. Goulev, and E. Mamdani. Context sensitive virtual sales agents. In *Proceedings of the Conference on Artistic, Cultural and Scientific Aspects of Experimental Media Spaces (cast01), FhG*, pages 79–83, Institut Medienkommunikation (IMK), Germany, 2001.

[5] R. Luce and H. Raiffa. *Games and Decisions*. John Wiley & Sons, Harvard University, 1967.

[6] E. H. Mamdani and S. Assilan. An experiment in linguistic synthesis with fuzzy logic controller. *International Journal of Man-Machine Studies*, (1):1–13, 1975.

[7] M J Owen, L Lee, G Sewell, S Steward, and D Thomas. Multi-agent trading environment. *BT Technol J*, 7(3):33–43, 1999.

[8] H. Raiffa. *The art and Science of Negotiation*. Harvard University Press, 1982.

[9] Anand Ranganathan, Roy H. Campbell, Arathi Ravi, and Anupama Mahajan. Conchat: A context-aware chat program. *IEEE Pervasive Computing*, (2):51–57, 2002.

[10] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. Sip: Session initiation protocol. Request for Comments 3261, June 2002.

[11] Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter. Designing Conventions for Automated Negotiation among Computers*. MIT Press, 1994.

[12] Tuomas W. Sandholm. Distributed rational decision making. In Gerard Weiss, editor, *Multi-Agent Systems. A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.

[13] C. Schmitt, D. Dengler, and M. Bauer. Multivariate preference models and decision making with the maut machine. In *Proceedings of the 9th International Conference on User Modeling*.

[14] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28(11):15–33, 1988.

[15] Stephen Voida, Elizabeth D. Mynatt, Blair MacIntyre, and Gregorio M. Corso. Integrating virtual and physical context to support knowledge workers. *IEEE Pervasive Computing*, (2):73–79, 2002.

[16] Lofti A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning i. *Information Sciences*, 8:199–249, 1975.

DIAGRAMAS UML PARA MODELOS DE TOMA DE DECISIÓN CON INFORMACIÓN HETEROGÉNEA

C. García, F. Herrera
Dpto. de Ciencias de la Computación e I.A.
Universidad de Granada, 18071 - Granada, España
gcarlos@fedro.ugr.es, herrera@decsai.ugr.es

L. Martínez, L.G. Pérez, P. J. Sánchez
Dpto. de Lenguajes y Sistemas Informáticos
Universidad de Jaén, 23071 - Jaén, España
martin,lgonzaga,pedroj@ujaen.es

Resumen

En un problema de Toma de Decisión nos encontramos con un conjunto de alternativas que debe ser valorada por un conjunto de expertos que expresarán sus preferencias sobre las alternativas. Generalmente se viene utilizando un único domino de información para expresar estas preferencias, pero este modo de actuar presenta una serie de inconveniente como la falta de flexibilidad. También puede darse el caso que el experto deba dar sus preferencias en un dominio de información que le es desconocido y por tanto dificulta su valoración. En este artículo se presentan unos diagramas UML para representar modelos de Toma de Decisón que pueden trabajar con información expresada en distintos dominios (numérico, intervalar y lingüístico) cuyo desarrollo teórico se ha presentado en [2, 3, 6]. Estos diagramas permiten una gran flexibilidad a la hora de trabajar con la información así como a la hora de poder definir modelos de Toma de Decisión.

1 INTRODUCCIÓN

En un modelo de Toma de Decisión tenemos un conjunto finito de alternativas $X = \{x_1, ..., x_n\}$ con $n \geq 2$, así como un conjunto finito de expertos $E = \{e_1, ..., e_m\}$ con $m \geq 2$. Los expertos deben valorar las distintas preferencias de cada una de las alternativas utilizando algún modelo de representación disponible como vectores de utilidad [3] o relaciones de preferencia [6]. Generalmente estas preferencias suelen espresarse en un único domino de información como el numérico, es decir, todas las valoraciones vienen expresadas por medio de números [8] o intervalos [12] que nos permiten una mayor flexibilidad. Si las alternativas tienen una característica cuantitativa este modo de expresar las preferencias suele ser apropiado pero al trabajar con alternativas con características cualitativas se vuelve dificil poder dar una valoración apropiada y un enfoque lingüístico se ha demostrado válido para trabajar con este tipo de preferencias [14]. Nuestra propuesta es poder trabajar con cualquier tipo de domino donde las distintas alternativas se puedan evaluar de una forma más conveniente o en el domino donde el experto sule trabajar para que de esta forma sus valoraciones sean lo más precisas posibles. En [3, 6, 7] se han presentado distintos modelo que nos permiten trabajar con información heterogénea (numérica, intervalar y lingüística) para problemas de Toma de Decisión.

El objetivo principal de este artículo es presentar una serie de diagramas UML que nos permitirán definir modelos de Toma de Decisión que trabajan con información heterogénea. En los diagramas se muestran distintos modelos de representación de preferencias con los que puede trabajar nuestros modelos de Toma de Decisión.

La contribución se estructurará como sigue: en la sección 2 realizaremos un repaso al esquema de Toma de Decisión con información heterogénea que luego se representará en los diagramas UML, en la sección 3 presentaremos los diagramas UML prara modelos de Toma de Decisión con información heterogénea. Para finalizar presentaremos las conclusiones.

2 ESQUEMA DE TOMA DE DECISIÓN

En nuestro esquema para la Toma de Decisión tenemos distintas alternativas que han sido valoradas por los distintos expertos en diferentes dominios, independientemente del modelo de representación de preferencias que se utilice. En [6, 7] hemos presentado el esquema de Toma de Decisión con información heterogénea basado en el prodeso de decisión presentado en [11] pero con las modificaciones necesarias para

poder trabajar con las alternativas valoradas en diferentes dominios (numérico, intervalar y lingüístico), incluso el esquema presentado puede servir para un proceso de evaluación [3]:

1. *Fase de Agregación*: para obtener un valor de preferencia colectiva para cada una de las alternativas valoradas por los distintos expertos en los diferentes dominios (numérico, intervalar y lingüístico) debemos seguir los siguientes pasos:
   (a) *Unificar la información.* La información heterogénea se unificará en un conjunto difuso valorado en un dominio lingüístico específico, llamado Conjunto Básico de Términos Lingüísticos (CBTL) y simbolizado por S_T. El CBTL es elegido según las condiciones presentadas en [7]
   (b) *Proceso de agregación.* Para cada alternativa, se obtiene un valor colectivo agregando el conjunto difuso valorado en el CBTL que representa la valoración individual que el experto ha asignado a esa alternativa por medio de un operador de agregación [4, 5, 13].
   (c) *Transformación a 2-tupla.* El valor colectivo obtenido para cada una de las alternativas expresado por medio de un conjunto difuso valorado en el CBTL se transformará en una 2-tupla lingüística [5] valorada en el CBTL utilizando la función presentada en [6].
2. *Fase de Explotación.* Una vez que tenemos las preferencias colectivas de cada una de las alternativas expresadas en 2-tuplas lingüísticas para obtener la mejor alternativa utilizaremos alguna función de selección como las presentadas en [10, 11]. Pero este esquema también es válido para un proceso de evaluación con lo que si este fuera el caso utilizaríamos un valor global [3] expresado por medio de una 2-tupla lingüística.

3 DIAGRAMAS UML PARA MODELOS DE TOMA DE DECISIÓN CON INFORMACIÓN HETEROGÉNEA

UML es un lenguaje gráfico para visualizar, especificar, construir y documentar los elementos de un sistema independientemente del software que se utilizará para su implementación [1, 9]. UML proporciona un estandar para la especificación de componentes en un sistema informático. En este artículo se presentan dos diagramas UML de clases donde se muestran los principales componentes de los modelos de Toma de Decisión con información heterogénea. El diagrama UML de la Figura 1 presenta las clases necesarias para poder trabajar con información heterogénea en los modelos de Toma de Decisión.

Pasaremos a descibir las principales características para cada una de las clases (presentadas en negrita), algunos de sus métodos (presentados en cursiva) y variables de instancia (presentadas en courier) del diagrama UML de la Figura 1:

- **Valoracion**. Las distintas alternativas del modelo de Toma de Decisión estarán valoradas en distintos dominios de información. Inicialmente no sabemos en que dominio de información estará valorada cada alternativa, por lo que necesitamos una clase que nos permita englobar todas las características comunes para todos los dominios de información. De esta forma una vez que especifiquemos un modelo de Toma de Decisión podremos asociar a cada alternativa su correspondiente dominio de información. Todos los dominios de información heredarán de esta clase y deberán implementar todos los métodos de esta clase al estar definidos como *abstractos*. De esta forma nos garantizamos que si se añade un nuevo dominio de información deben implementarse unos métodos que garanticen su correcto uso por parte del resto de clases presentes en el diagrama. El método más importante de esta clase, y que nos permite trabajar con la información representada en distintos dominios, es:
  – *unificacion.* Al trabajar con información heterogénea debemos unificar toda la información de nuestro modelo de Toma de Decisión en un único dominio de información para posteriormente poder aplicar un operador de agregación a las valoraciones de las distintas alternativas. El domino de información elegido es un conjunto difuso valorado en el CBTL. Cada clase que herede de **Valoracion** debe implementar este método para su correcta unificación en un conjunto difuso valorado en el CBTL. En [7] se presentan distintos métodos de unificación para información numérica, intervalar y lingüístistica.
- **Numerico, Intervalo, Lingüistico**. Estas tres clases heredan de **Valoracion** y representan los distintos dominios de información que pueden estar presentes en nuestros modelos de Toma de Decisión (información heterogénea). Al heredar de **Valoracion** haremos uso de la propiedad de polimorfismo para que dependiendo del dominio de información se aplique el correcto método de *unificacion* que permitirá transformar la información de entrada a su correspondiente conjunto

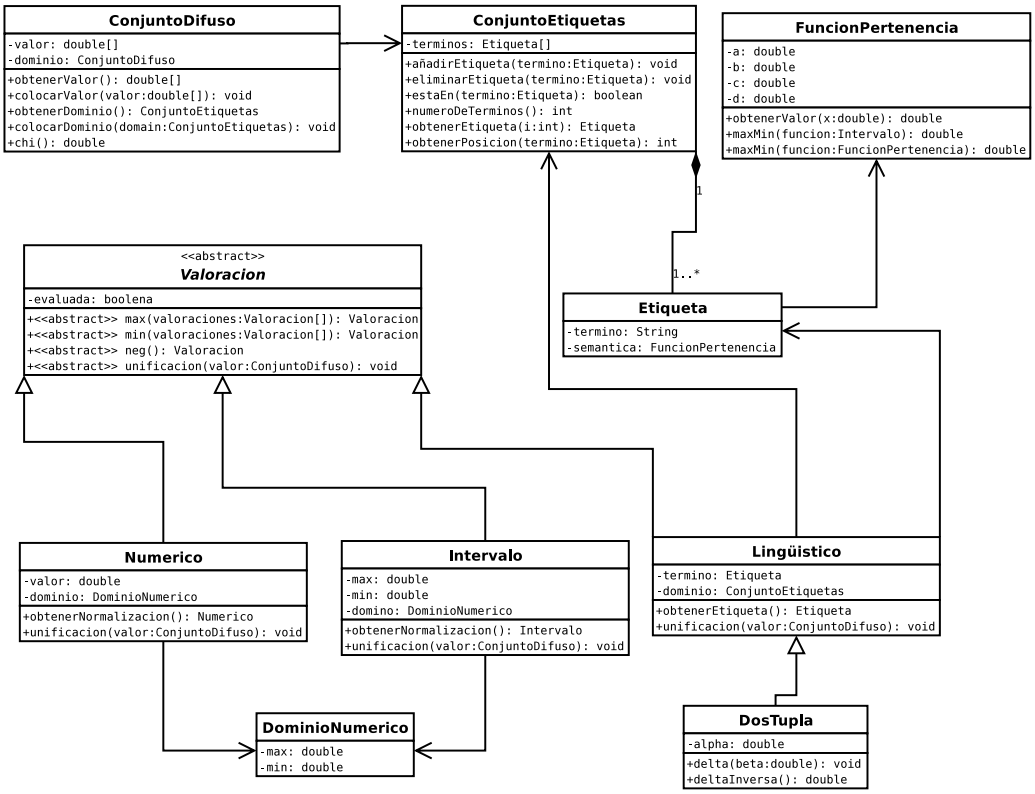


Figura 1: Diagrama UML para trabajar con informacion heterogénea

difuso valorado en el CBTL.

- **DosTupla**. Esta clase contiene los métodos necesarios para implementar el modelo basado en 2-tuplas lingüísticas presentado en [5]. Además también utilizaremos este dominio de información para presentar el resultado final de nuestros modelos de Toma de Decisión.

- **ConjuntoDifuso**. Esta clase se utilizará para representar el domino de la información donde se unificará toda la información de entrada (numérica, intervalar y lingüística). El conjunto difuso viene representado en la variable de instancia `valor[]` y el conjunto de etiquetas que representa el CBTL en la variable de instancia `dominio`.

- bf Etiqueta. Esta clase nos permite representar la sintaxis de una etiqueta lingüística por medio de su etiqueta (`termino`) y su semántica dada por un número difuso perteneciente al intervalo [0,1] (`semantica`).

- bf ConjuntoEtiquetas. Esta clase nos permite representar un conjunto de etiquetas lingüísticas donde serán valoradas las alternativas pertenecientes a un dominio lingüístico.

- **DominioNumerico**. Tanto la información numérica como la intervalar deberá estar expresada por medio de números pertenecientes al intervalo [0,1]. Pero para poder permitir una mayor flexibilidad en nuestro diagrama asociaremos un objeto de esta clase a cada objeto **Numerico** como **Intervalo** para poder permitir cualquier rango numérico (expresado por medio de las variables de instancia `max` y `min`) tanto para valores numéricos como intervalares. Sólo ha de añadirse un método de *normalizacion* en las clases **Numerico** y **Intervalo** que permita transformar los valores iniciales a otros comprendido en el intervalo [0,1].

- **FuncionPertenencia**. Cada objeto de la clase **Etiqueta** tendrá asociada una semántica que viene dada por un número difuso representado de forma paramétrica por una 4-tupla (`a,b,c,d`), donde `b` y `c` indican el intervalo en el cual el valor de la función de pertenencia es 1, con `a` y `d` indicando el límite izquierdo y derecho del dominio donde se encuentra definida la función de pertenencia.

Una vez presentado el diagrama UML que nos permitirá trabajar con información heterogénea en nue-

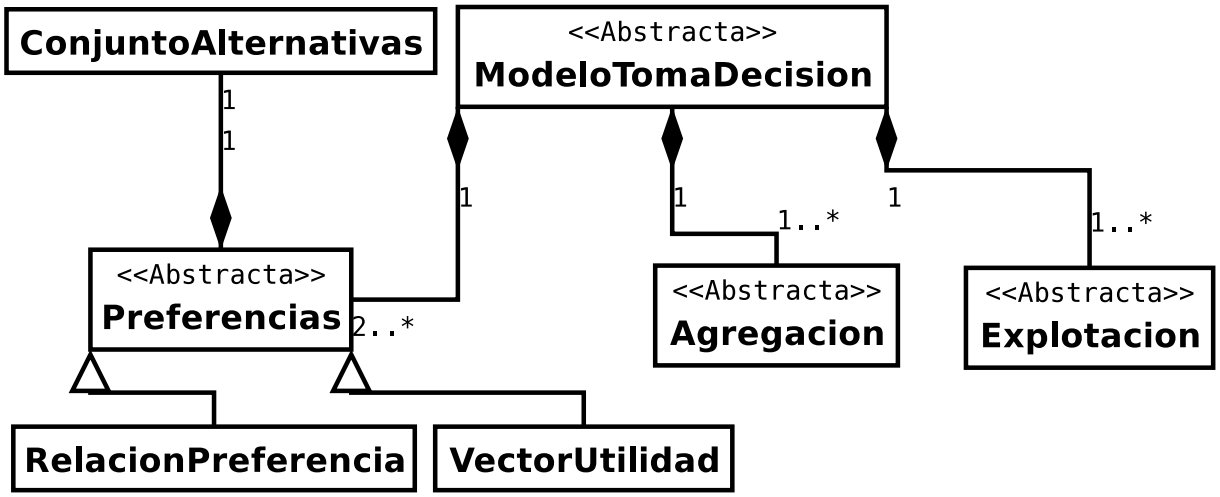


Figura 2: Diagrama UML para los modelos de Toma de Decisión

stros modelos de Toma de Decisión pasamos a presentar el diagrama UML que representa nuestros modelos de Toma de Decisión. El diagrama se presenta en la Figura 2. Pasaremos a descibir las principales características para cada una de las clases (presentadas en negrita) y algunos de sus métodos (presentados en cursiva) del diagrama UML de la Figura 2:

- **ConjuntoAlternativas**. Esta clase nos permitira almacenar las distintas alternativas presentes en un problema de Toma de Decisión. Para un problema de Toma de Decisión sólo habrá un único objeto que representa todas las alternativas que deben se valoradas por los expertos.

- **Preferencias**. Esta clase abstracta nos permitira generalizar los distintos modelos de representación que existen para valorar un conjunto de alternativas. De esta forma nuestro modelo de Toma de Decisión no tiene que cambiar al cambiar la representación en que se valoran las distintas preferencias por parte de los expertos.

- **VectorUtilidad**. Esta es una clase que hereda de **Preferencias** y representa el modelo en el que las preferencias son expresadas por medio de un vector, donde cada valor expresa la preferencia para cada una de las alternativas que se estan considerando. Cuando utilizamos este modelo de representación de preferencias los modelos de Toma de Decisión pasan a denominarse modelos de Toma de Decisión Multicritero [3].

- **RelacionPreferencia**. Esta clase también hereda de **Preferencias** y representa el modelo en el que las preferencias son expresadas por medio de una relación binaria con respecta a cada par de alternativas (x_i, x_j) donde se refleja el grado en que x_i *es preferida sobre* x_j. Cuando utilizamos este modelo de representación de preferencias los modelos de Toma de Decisión pasan a denominarse modelos de Toma de Decisión en Grupo [6].

- **Agregacion**. Esta clase abstracta nos permite generalizar cualquier operador de agregación que queramos definir para nuestro problema de Toma de Decisión. De esta forma podremos aplicar distintos operadores de agregación para un mismo modelo de Toma de Decisión. En [4, 5, 13] podemos encontrar operadores de agregación para distintos problemas de Toma de Decisiones.

- **Explotacion**. Esta clase abtracta nos permite generalizar cualquier proceso de explotación que queramos definir para nuestro problema de Toma de Decisión. De esta forma podemos aplicar distintos procesos de explotación para un mismo modelo de Toma de Decisión. En [10, 11] podemos encontrar distintas funciones de selección de alternativas.

- **ModeloTomaDecision**. Esta clase abtracta nos permite generalizar los distintos modelos de Toma de Decisión que queramos definir para que puedan trabajar todos ellos con los modelos de representación de preferencias que tengamos definidos así como con los operadores de agregación y procesos de explotación con los que contemos. Como uno de nuestros objetivos es poder trabajar con modelos de Toma de Decisión con información heterogénea como los presentados en [6, 7] necesitaremos contar con los siguientes métodos abstractos que deberán ser implementados para cada uno de nuestro modelos:

- *obtenerCBTL*. Este metodo será necesario para poder encontar el conjunto de términos lingüísticos donde de realizará la unificación de la información de entrada. Este proceso de selección podemos encontrarlo en [7].
- *procesoAgregacion*. Como vamos a trabajar con información expresada en distintos dominios no podremos aplicar directamente los operadores de agregación que tengamos definidos. Deberemos seguir los siguientes pasos para realizar una agregación correcta de la información de entrada a nuestro problema:
  1. *Unificar la información*. Las valoraciones de las alternativas de nuestro problema vendrán expresadas en distintos dominos (numerico, intervalar y lingüísticos) y se unificarán por medio de conjuntos difusos expresados en el CBTL.
  2. *Agregación*. Una vez unificada toda la información podemos utilizar algún operador de agregación para obtener un valor de preferencia colectivo para cada una de las alternativas.
  3. *Pasamos a 2-tuplas*. Una vez que hemos utilizado los operadores de agregación pasamos la información a 2-tuplas que es el dominio donde vendrá espresado el resultado final de nuestro modelo de Toma de Decisión
- *faseExplotacion*. Sólo faltan aplicar un proceso de explotación utilizando una función de selección de alternativas (expresadas en 2-tuplas lingüísticas) para poder obtener el resultado final del modelo de Toma de Decisión.

4 CONCLUSIONES

En este artículo hemos presentado los diagramas UML para los trabajos teóricos presentados en [3, 6, 7]. También hemos completado una primera implementación en JAVA para un sistema automático de evaluación de la calidad del profesorado en las Universidades Andaluzas como parte de un proyecto de investigación financiado por La Unidad para la Calidad de la Universidades Andaluzas (UCUA). En la presentación se mostrará un ejemplo de esta primera implementación.

5 Agradecimientos

Este trabajo ha sido financiado por los proyectos de investigación TIC2002-03348, UJA23(UCUA)y PromoJaén un proyecto perteneciente a la iniciativa EQUAL y los fondos FEDER.

Referencias

[1] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Addison Wesley, 1999.

[2] F. Herrera, E. Herrera-Viedma, and L. Martínez. A fusion approach for managing multi-granularity linguistic terms sets in decision making. *Fuzzy Sets and Systems*, 114(1):43–58, 2000.

[3] F. Herrera, E. Herrera-Viedma, L. Martínez, and P.J. Sánchez. A linguistic decision process for evaluating the installation of an ERP system. In *9th International Conference on Fuzzy Theory and Technology*, Cary (North Carolina) USA, 2003.

[4] F. Herrera, E. Herrera-Viedma, and J.L. Verdegay. Direct approach processes in group decision making using linguistic OWA operators. *Fuzzy Sets and Systems*, 79:175–190, 1996.

[5] F. Herrera and L. Martínez. A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Transactions on Fuzzy Systems*, 8(6):746–752, 2000.

[6] F. Herrera, L. Martínez, and P.J. Sánchez. Managing non-homogeneous information in group decision making. *European Journal of Operational Research*, page To appear, 2004.

[7] F. Herrera, L. Martínez, and P.J. Sánchez. *A Multi-Granular Linguistic Decision Model for Evaluating the Quality of Network Services.*, pages 71–92. In: Intelligent Sensory Evaluation: Methodologies and Applications. Springer, 2004.

[8] J. Kacprzyk. Group decision making with a fuzzy linguistic majority. *Fuzzy Sets and Systems*, 18:105–118, 1986.

[9] C. Larman. *Applying UML and Patterns*. Prentice Hall, 1998.

[10] S.A. Orlovsky. Decision-making with a fuzzy preference relation. *Fuzzy Sets Systems*, 1:155–167, 1978.

[11] M. Roubens. Fuzzy sets and decision analysis. *Fuzzy Sets and Systems*, 90:199–206, 1997.

[12] J.F. Le Téno and B. Mareschal. An interval version of PROMETHEE for the comparison of building products' design with ill-defined data on

environmental quality. *European Journal of Operational Research*, 109:522–529, 1998.

[13] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. on Systems, Man, and Cybernetics*, 18:183–190, 1988.

[14] R.R. Yager. An approach to ordinal decision making. *International Journal of Approximate Reasoning*, 12:237–261, 1995.

SISTEMAS DE REGRESIONES POSIBILÍSTICAS: UNA ALTERNATIVA PARA LA INFERENCIA ECOLÓGICA

Sergio Donoso **Nicolás Marín** **M. Amparo Vila**

UTEM
Santiago de Chile – Chile
sdonosos@vtr.net

Dpto. de Ciencias de la Computación e I.A. – Universidad de Granada
18071 – Granada – España
{nicm, vila}@decsai.ugr.es

Resumen

Esta investigación presenta cómo utilizar la regresión posibilística para funciones de pertenencia no simétrica, agrupándolas en un sistema de regresiones, de forma que se puedan incorporar las restricciones propias de cada problema. Se incorpora la minimización de funciones cuadráticas como una manera de hacer ver más claramente la estimación de la incertidumbre que tienen incorporados los datos. Se muestra como aplicar esta metodología en el caso de la inferencia ecológica, en particular en la estimación de las matrices de transición electoral. Varios ejemplos muestran las bondades de la alternativa propuesta

Palabras Clave: regresión posibilística, sistema de regresiones, función de pertenencia no simétrica, función de pertenencia truncada, inferencia ecológica, matriz de transición electoral, minimización de funciones cuadráticas.

1 INTRODUCCIÓN

La primera formulación de la regresión difusa data de 1982 [10]. Muchos trabajos han ido enriqueciendo esta metodología. En esta investigación se propone ampliar su uso de una sola regresión a un conjunto de regresiones, que llamamos sistema de regresiones posibilisticas. Para facilitar la práctica de la regresión difusa, se pone énfasis en las funciones de pertenencia no simétrica y en la minimización de funciones cuadráticas.

La inferencia ecológica [8] tiene como objetivo la estimación de conductas individuales a partir de datos agregados. Un ejemplo lo constituye las preferencias partidistas de los electores entre dos procesos electorales. Esta conducta no es conocida dado el carácter secreto del sufragio, aunque es evidente la gran utilidad que tiene su estimación para la ciencia política en general y los estudios electorales en particular. La inferencia ecológica constituye un problema estadístico que ha sido tratado durante más de medio siglo, sin que se hayan obtenido resultados concluyentes hasta la fecha.

Dado que este problema es muy complejo desde el punto de vista estadístico, los estudios que pueden encontrarse en la literatura se han centrado en la estimación de tablas 2x2 [5, 8, 9].

2 REGRESION POSIBILISTICA

La ecuación (1) muestra el modelo lineal múltiple de la regresión probabilística:

$$Y_i = \beta_0 + \beta_1 X_{1i} + ...+ \beta_m X_{mi} + \varepsilon_i \quad (1)$$

para n observaciones, en que los coeficientes β son desconocidos.

Este modelo fue reformulado por Tanaka [10] para valores Y y X crisp y coeficientes de regresión β difusos con función de pertenencia triangular simétrica $A_i=(a_i,c_i)$. En este nuevo enfoque, el error aleatorio ε_i es reemplazado por la imprecisión en los coeficientes a estimar, resultando el siguiente problema de programación lineal:

$$\text{Minimizar } \sum(\sum c_j|X_{ij}|), \text{ para } j=0,...,n;\ i=1,...,m \quad (2)$$

sujeto a la condición posibilística, de que los extremos de la función de pertenencia de la estimación contengan a la función de pertenencia del dato Y_i:

$$Y_i \ \dot{C} \ \tilde{Y}_i$$

lo que se expresa formalmente mediante el siguiente conjunto de desigualdades:

$$\sum a_i X_{ij} + (1-h)\sum c_i|X_{ij}| \geq y_i; \quad i=1,..., n \quad (3)$$

$$-\sum a_i X_{ij} + (1-h)\sum c_i|X_{ij}| \geq -y_i; \quad i=1,..., n \quad (4)$$

$$c_i \geq 0; \quad i=0,....,m \quad (5)$$

con un grado h de creencia para Y_i estimado, tal que

$$\mu(Y_i) \geq h \quad (6)$$

2.1. COEFICIENTES DE REGRESIÓN CON FUNCIONES DE PERTENENCIA NO SIMÉTRICAS

La formulación anterior, construida a partir del principio de extensión de Zadeh [15], puede ampliarse para estimar coeficientes de regresión difusos con funciones de pertenencia no simétricas (a_i, c^L_i, c^R_i) mediante el siguiente problema de programación lineal:

$$\text{Minimizar } \sum(\sum((c^L_j+c^R_j)|X_{ij}|) \qquad (7)$$

sujeto a las condiciones:

$$\sum a_i X_{ij} + \sum c^R_i|X_{ij}| \geq y_i ; \quad i=1,..., n \qquad (8)$$

$$\sum a_i X_{ij} - \sum c^L_i|X_{ij}| \leq y_i ; \quad i=1,..., n \qquad (9)$$

$$c^R_i \; c^L_i \geq 0 ; \quad i=0,...,m \qquad (10)$$

donde, si se consideran p variables X, se tiene 3p parámetros a estimar.

Para el caso en que la variable dependiente o de salida sea difusa LR con función de pertenencia no simétrica (y_i, p_i, q_i), las restricciones del problema tienen la siguiente formulación:

$$\sum a_i X_{ij} + R^{-1}(h)\sum c^R_i|X_{ij}| \geq y_i + R^{-1}(h)\, q_i ; \quad i=1,..,n \quad (11)$$

$$\sum a_i X_{ij} - L^{-1}(h)\sum c^L_i|X_{ij}| \leq y_i - L^{-1}(h)\, p_i ; \quad i=1,..,n \quad (12)$$

$$c^R_i \; c^L_i \geq 0 ; \quad i=0,...,m \qquad (13)$$

La principal crítica que ha recibido la regresión posibilística es el efecto que producen los puntos extremos. De hecho, se recomienda hacer un análisis de los puntos extremos al momento de hacer la estimación para tener una diagnostico adecuado [12].

2.2 LA REGRESION POSIBILÍSTICA COMO PROBLEMA DE PROGRAMACIÓN CUADRATICA

Para la regresión intervalar, una forma menos exigente de regresión posibilística, Tanaka y Lee plantean que en esté último análisis de regresión, algunos coeficientes tienden a ser exactos por características propias de la programación lineal y sugieren el uso de la programación cuadrática [11][1].

Un modelo exclusivamente posibilistico se plantea mediante la minimización de la suma de los cuadrados de las extensiones de la función de pertenencia sujeto a que los datos y_i deben estar incluidos en la salida estimada.

[1] Las soluciones de ejemplo, formula (38) y Tabla III del trabajo de Tanaka, no están correctos.

Se generaliza la formulación de [11] mediante el uso de funciones de pertenencia no simétricas, quedando formalizado el problema de programación cuadrática con la minimización de la expresión

$$J = c^L XX' c^{L'} + c^R XX' c^{R'} \qquad (14)$$

bajo las mismas restricciones (8), (9) y (10).

Otro modelo, más general, que tiene por finalidad incorporar como objetivo la minimización de la tendencia central de la regresión (objetivo de mínimos cuadrados), junto a la minimización de las extensiones de las estimaciones (objetivo posibilistico), se expresa en la siguiente expresión

$$J = k_1 \Sigma(y_i - a'X_j)2 + k_2 (c^L XX' c^{L'} + c^R XX' c^{R'}) \qquad (15)$$

donde k_1 y k_2 son coeficientes de ponderación, que cumplen un rol muy importante, porque permite plantear como meta una mayor expectativa respecto a la tendencia central ($k_1 > k_2$) o de mayor expectativa en reducir la incertidumbre de la estimación ($k_1 < k_2$). Las restricciones a que esta sujeta esta función objetivo son las mismas que el modelo anterior (14).

3 REGRESION ECOLOGICA

A la hora de hacer estimaciones, hay situaciones en las que no se dispone de información a nivel de individuos, sino que esta información está agregada a cierto nivel geográfico o de otro tipo.

Por ejemplo, se tienen resultados electorales a nivel de comunas, barrios, distritos, etc., pero no se conoce como votó una persona en particular.

Gootmann en 1953 [8] planteó la regresión ecológica como solución para este problema, con el requisito llamado supuesto de homogeneidad, es decir, que todas las probabilidades condicionales son iguales en las diversas unidades de información. Esto se ha aplicado hasta ahora en muchos estudios, por ejemplo, en los trabajos de Engel [6, 7].

Duncan [5] estimó un intervalo como solución, lo que se denomina la solución de bordes. King en 1997 [9] planteó un nuevo enfoque en que los coeficientes de regresión son aleatorios, aplicado por Wellhofer [12] para diversas elecciones italianas y Burden [2] para elecciones simultaneas en Estados Unidos. Cho [3, 4] ha criticado el modelo de King, ya que teniendo supuestos menos estrictos que los de Goodman, también son frecuentemente sobrepasados, y plantea algunas alternativas para mejorar los dos modelos anteriores.

3.1. ESTIMACIÓN DE MATRICES DE TRANSICIÓN ELECTORAL

Se considera el siguiente esquema para nuestro problema de estimación de matrices de transición electoral: se toman dos elecciones, en general consecutivas, a saber, la elección 1 y la elección 2. La elección 2 puede ser una elección simultánea para elegir otro cargo político (Senado y Cámara de Diputados, etc.) o puede ser la segunda vuelta de la primera elección. También puede ser la siguiente elección política como, por ejemplo, las elecciones generales españolas del 2000 y 2004.

- Para la elección 1 se presentan p partidos políticos, cada uno de los cuales obtiene X_{ik} votos en cada unidad k de recepción de sufragios (hay n unidades).

- Para la elección 2 se presentan q partidos políticos, cada uno de los cuales obtiene Y_{jk} votos, en la unidad k.

Teniendo esto en cuenta, para cada unidad i, existe la probabilidad condicional:

β_{jik} = Prob (un elector votó por partido j en la elección 2 / el elector votó por partido i en la elección 1)

Por el carácter secreto del sufragio, no es posible conocer estas probabilidades. Para estimarlas, se pueden escribir las relaciones

$$Y_{jk} = \sum \beta_{jik} X_{ik} \qquad (16)$$

donde se tienen pxqxn valores β a estimar y sólo qxn ecuaciones, con lo cual no se tiene una solución única. Si se supone que los valores β son constantes en las diversas unidades i, es decir,

$$\beta_{ji} = \beta_{jik} ; \quad k=1,...,n \qquad (17)$$

entonces es posible realizar algún tipo de estimación estadística para los coeficientes β_{ji}. Este supuesto se conoce como supuesto de homogeneidad en la regresión ecológica. Si este supuesto no se cumple, entonces la regresión de Gootman anteriormente citada entrega resultados incorrectos.

Si los valores β_{jik} varían de unidad en unidad y el supuesto de homogeneidad no se cumple, entonces el objetivo de las matrices de transición electoral es estimar el promedio ponderado de los β_{jik} para las n unidades de información.

3.2. NOTACIÓN CLÁSICA DE LA INFERENCIA ECOLOGICA

El problema de la inferencia ecológica ha sido planteado como una tabla 2x2, en términos de proporciones de personas en cada unidad de información:

Tabla 1: Problema de la inferencia ecológica

Elección 1	**Elección 2**		
	Partido B	**Otros par**	**Total**
Partido A	β^B	$1 - \beta^B$	X
Otros part	β^W	$1 - \beta^W$	1-X
Total	T	1 – T	1

donde X es la proporción de votos por el partido A en la primera elección, T es la proporción de votos por el partido B en la segunda elección y las probabilidades condicionales β^B y β^W son los valores a estimar, como valor promedio entre todas las unidades de información. β_B representa la proporción de los votantes por el partido A en la primera elección que votan por el partido B en la segunda elección.

De la Tabla 1 se tiene la relación

$$T = \beta^B X + \beta^W (1 - X) \qquad (18)$$

que es estimada efectivamente como

$$T = \beta^W + (\beta^B - \beta^W) X \qquad (19)$$

Gootman [8] propone la estimación de esta regresión (por lo que es conocida con el nombre de regresión de Gootman) basado en el supuesto de que todos los β^W y β^B son constantes en las diversas unidades de información.

Duncan y Davis [5] proponen un método para acotar la los coeficientes β^B y β^W para lo cual transforman la ecuación (16) en una recta en función de los coeficientes para cada dato disponible:

$$\beta_i^W = (T_i / (1-X_i)) - (X_i / (1-X_i)) \beta_i^B \qquad (20)$$

De ésta ecuación se obtiene las cotas para los coeficientes a estimar [9: 79]:

$$\text{Max}(0, (T_i -(1-X_i))/X_i) \le \beta_i^B \le \text{Min}(T_i / X_i, 1) \qquad (21)$$

y

$$\text{Max}(0, (T_i -X_i)/(1- X_i)) \le \beta_i^W \le \text{Min}(T_i /(1- X_i), 1) \qquad (22)$$

El promedio ponderado de estas cotas sobre los n datos [9: 84] parece ser una buena medida de cota inferior y cota superior para el valor central a estimar en la regresión ecológica.

King [9] propone otro método de estimación, que se llamará EI (Ecological Inference), que es el nombre del programa que creó el mismo King para realizar la estimación.

El esfuerzo de King es un avance considerable en el campo de la regresión ecológica, pero de todas maneras descansa en tres supuestos que pueden no cumplirse, a pesar que King fundamenta que la violación de estos

supuestos no invalida el modelo. Cho [3, 4] ha destacado las limitaciones de los enfoques de Goodman y King

La estimación de mínimos cuadrados ordinarios de β^B y β^W puede producir valores fuera del intervalo [0,1] (véase [1, pag.288] y tabla 4 de esta investigación). Es una muestra de lo que se ha llamado "sesgo de agregación".

Cuando los coeficientes de mínimos cuadrados están fuera de rango, es conveniente hacer un análisis de los datos, considerando variables tales como tamaño de las unidades de información, tamaño de X y T, o recurrir a otras variables [4] tales como nivel socio-económico de cada unidad geográfica de información, para una posible división de la estimación en dos grupos. En el ámbito de la regresión difusa, también se encuentra una proposición para determinar los puntos de cambio de tendencia [14].

El hecho de estimar datos solo de tablas 2x2, como en el modelo EI, convierte en tediosa la estimación de muchas matrices de transición electoral como en el estudio de las elecciones italianas [13].

4 SISTEMA DE REGRESIONES POSIBILÍSTICAS

Si se quiere estimar la matriz de transición electoral (16) con la condición que los coeficientes sean conjuntos difusos con función de pertenencia no simétrica, se tiene que

$$Y_j = \sum \beta_{ji} X_i \qquad (23)$$

para cada uno de los q partidos de la segunda elección, en que los coeficientes a estimar son conjuntos difusos. Como los coeficientes representan probabilidades condicionales, entonces su máximo valor de pertenencia a_{ji} debe estar entre 0 y 1. Para lo cual a cada una de las q regresiones debe agregar las condiciones

$$a_{ji} \leq 1 \text{ y } a_{ji} \geq 0 ; \quad j=1,...,q \quad i=1,...,p \qquad (24)$$

Además las q probabilidades condicionales de los votantes de un partido de la primera elección que votaron por los diversos partidos en la segunda elección deben sumar 1, lo que se expresa en la relación

$$\sum_i \beta_{ji} = 1 \qquad (25)$$

lo cual involucra no sólo a una de las regresiones sino al conjunto de ellas (q).

Por esta razón, se debe formar con los q problemas de la forma (7)-(10), un solo problema de programación lineal, que para las p variables X tendrá un total de 3pq valores a estimar.

Como, en general los β_{jik}, no son iguales para las n unidades de recepción de votos, lo que se estima realmente es un β_{ji} promedio. Por lo tanto, es conveniente colocar la condición adicional

$$\sum a_{ji} (\sum X_{ik}) = \sum Y_{jk} \qquad (26)$$

o la relación equivalente

$$\sum a_{ji} (\text{Prom}(X_{ik})) = \sum \text{Prom}(Y_{jk}) \qquad (27)$$

para cada uno de los q partidos de la segunda elección.

Además, si se quiere incorporar las limitaciones de borde para la estimación, en forma más estricta que (24), se puede tener las condiciones adicionales

$$a_{ji} + c^R_{ji} \leq 1 \text{ y } a_{ji} - c^L_{ji} \geq 0 ; \quad j=1,...,q \quad i=1,...,p \qquad (28)$$

5 EJEMPLOS DE MINIMIZACIÓN LINEAL

Se considerará primero el caso 2x2, construido expresamente para conocer los coeficientes a estimar, y para comparar con los resultados de la regresión de Gootman y la de King, se tomarán las siguientes alternativas de regresión difusa para el sistema de ecuaciones:

Difusa SI 1: modelo que solo incluye las restricciones de que los coeficientes están en el intervalo [0,1].

Difusa SI 2: modelo difuso SI 1 más la restricción (25), de que la suma de los coeficientes de cada partido de la primera elección deben sumar 1.

Difusa SI 3: modelo difuso SI 2 más la condición (26)

Difusa SI 4: agregando a las condiciones anteriores el límite de cotas de las relaciones (26).

El ejemplo 1 considera una situación relativamente corriente en que el coeficiente $\boldsymbol{\beta^B}$ es alto y el otro coeficiente es bajo. Se generaron 40 datos con una desviación estandar para la variable dependiente de 0.179.

Tabla2: Comparación de resultados, ejemplo 1

Método	$\boldsymbol{\beta^B}$	$\boldsymbol{\beta^W}$	**error**
VERDADERO	0,9390	0,1231	
Goodman	0,9905	0,0756	0,0990
EI	0,9703	0,0934	0,0610
Difuso SI 1	0,9799	0,1364	0,0542
Difuso SI 2	0,9669	0,1695	0,0743
Difuso SI 3	0,9742	0,0899	0,0684
Difuso SI 4	0,9693	0,1080	0,0454

En este primer ejemplo se aprecia que el método de Goodman es el que entrega una máxima diferencia entre el valor estimado para los coeficientes y el valor verdadero y hay una superioridad de los métodos difusos frente al método EI. El método difuso SI 2 entrega los resultados menos correctos, talvez por forzar la "unicidad" de los coeficientes, "unicidad" que se cumple, prácticamente automáticamente, en el caso que la segunda elección tenga solo dos opciones.

El segundo ejemplo es real, forma una tabla 4x2, del cual no se conocen los verdaderos valores de la matriz de transición electoral. Corresponde al condado de Amoostook, en le Estado de Maine, Estados Unidos, se desarrollaron elecciones simultaneas de Gobernador, Senador, Representantes, y otros cargos, en Noviembre del 2002. Los totales para Senador y Gobernador, por partido político está reflejado en la tabla 3 y la estimación en la tabla 4.

Tabla 3: Totales para Senador y Gobernador por partido

	sen rep	**sen dem**	**TOTAL**
gober rep	?	?	8.401
gober dem	?	?	15.783
gober verde	?	?	1.279
gober indep.	?	?	812
TOTAL	17.867	8.408	26.275

Tabla 4: estimación de Goodman para ejemplo 2

	sen rep	**sen dem**	**TOTAL**
gober rep	1,0996	-0,0996	8.401
gober dem	0,439	0,561	15.783
gober verde	0,6	0,4	1.279
gober indep.	-0,78	1,78	812
TOTAL	17.867	8.408	26.275

Se aprecia la presencia de valores fuera del rango [0,1] lo que invalida abiertamente los resultados. Aún truncando a 0 o 1 los valores fuera de rango, se dispondría de una estimación poco confiable.

Para la estimación de King, realizando también 4 estimaciones 2x2 (King solo recomienda extender su modelo a tablas 3x2, pero tomamos como una prueba de robustez del modelo el forzarlo en tablas 4x2), se tiene los resultados de la tabla 5.

Tabla 5: Estimación de King (EI) para el ejemplo 5

	sen rep	**sen dem**	**TOTAL**
gober rep	0,9495	0,0505	8.401
gober dem	0,5096	0,4904	15.783
gober verde	0,9942	0,0058	1.279
gober indep.	0,0265	0,9735	812
TOTAL	17.867	8.408	26.275

¿Cómo va a ser real que el coeficiente correspondiente al candidato a gobernador republicano sea menor que el coeficiente del candidato a gobernador verde? ¿Cómo va a ser posible que el candidato ganador para senador con una amplísima ventaja, prácticamente no obtenga ningún voto de los que votaron por el candidato independiente a gobernador? La estimación para el total de la votación del candidato a senador republicano es de 17013, lo que significa una subestimación de unos 500 votos.

La estimación de cualquiera de los sistemas difusos, a diferencia de las dos estimaciones anteriores, se realizaron de una sola vez, como tabla 4x2.

Por ejemplo, la estimación con el sistema SI 2 produce una diferencia absoluta de 2703, que es más alta que la anterior, por lo que no se considera interesante detenerse en ella.

Por la calidad de ajuste, se recomienda la estimación del modelo SI 3 como la más adecuada para calcular la matriz de transición electoral. Como se dijo, la estimación SI 1, aunque muy razonable, subestimaba la estimación global del candidato a senador republicano, por lo que tiene plena justificación el aumento de valor en 3 coeficientes, entre las estimaciones SI 1 y SI 3, para compensar la subestimación indicada.

6 EJEMPLO DE MINIMIZACIÓN CUADRATICA

En Diciembre de 1999 se realizó la última elección presidencial en Chile. En Enero del 2000, se realizó, por primera vez en la historia del país, la segunda vuelta de esta elección, en que salió elegido el actual Presidente de la República Ricardo Lagos.

Se estructuró una matriz de transición 4x3, en que para la primera vuelta se consideró los tres candidatos que obtuvieron las mayores votaciones.

El modelo considerado es la minimización cuadrática dada por la relación (15), con k1=1 y k2=0.0001, ya que la meta es lograr la mayor precisión en la estimación de la tendencia central, sujeto a las restricciones posibilisticas.

Ponderando los resultados de estas regresiones se logra finalmente la siguiente matriz de transición (Tabla 6).

Tabla 6: Matriz de transición de elección presidencial de Chile de 1999-2000.

	LAVIN	LAGOS	RESTO	Votos
LAVIN	0,959	0,019	0,022	3.352.199
LAGOS	0,022	0,964	0,015	3.383.339
MARIN	0,074	0,799	0,127	225.224
RESTO	0,164	0,156	0,680	1.123.714
Votos	3.490.561	3.677.968	915.947	8.084.476

Cada uno de estos coeficientes estimados, tiene una función de pertenencia, que restringido al intervalo [0,1] llamamos función de pertenencia "truncada

7 CONCLUSIONES

La regresión posibilística para datos crisps, como también para el caso en que la variable dependiente o de salida sea difusa con función de pertenencia no simétrica, puede organizarse como un sistema de regresiones si el modelo es múltiple en el número de ecuaciones, de manera que la estimación conjunta de todas las ecuaciones permite incorporar condiciones adicionales a las que se pueden agregar a cada regresión independientemente.

La regresión posibilística, que se ha aplicado a la minimización de una función lineal, se puede ampliar a la minimización de funciones cuadráticas, obteniéndose como ventaja la posibilidad de ponderar los objetivos y representar la incertidumbre del modelo en conjuntos difusos.

En este trabajo, se ha demostrado que los sistemas de regresiones difusos permite el cálculo de las matrices de transición electoral, un típico ejemplo de inferencia ecológica. Se han definido diversos grupos de restricciones para ser incorporadas a la estimación del sistema de regresiones posibilísticas.

Mientras que la regresión ecológica ha sido utilizada fundamentalmente para el caso de una tabla 2x2, nuestra formulación para estimar las matrices de transición electoral permite la estimación de tablas pxq.

Los ejemplos numéricos desarrollados muestran que los métodos difusos definidos con minimizaciones lineales son mejores estimadores que el método de Goodman, y que se encontraron estimaciones difusas superiores al método EI. En particular, el modelo difuso SI 3 entrega mejores estimaciones para el ejemplo real estudiado.

Referencias

[1] Anselin, Luc y Cho, Wendy K. Tam. Spatial effects and ecological inference, *Political Analysis*, 10 (3), Pág. 276-297, Summer 2002.

[2] Burden, Barry C. Y Kimball, David C. A New Approach to the Study of Ticket Splitting, American political Science Review, Vol. 92 (3) Pág. 533-544, 1998.

[3] Cho, Wendy K. Tam y Gaines, Brian J. The limits of ecological Inference: The case of Split Ticket Voting, *American Journal of Political Science*, 48 (1) Pág. 152-171, January 2004.

[4] Cho, Wendy K. Tam. Latent groups and cross-level inferences, *Electoral Studies*, Vol. 20, Pág. 243-263, 2001.

[5] Duncan O D y Davis B, An alternatiave to ecological correlation, American Sociological Review, Vol. 18 Pág. 665-666, 1953.

[6] Engel, E. Evolución del comportamiento electoral desde el plebiscito a la elección presidencial, *Colección de Estudios CIEPLAN*, 28, Junio 1990, Pág. 73-83

[7] Engel, E. y Hernando, A. Chile: ¿Dos o más bloques? *Perspectivas en Política, Economía y Gestión*, 2003 vol 6 Pag. 219-230

[8] Goodman, Leo A. "Ecological Regressions and the Behavior of Individuals" *American Sociological Review*, 18 (6), 1953, Pag. 663-666

[9] King, G. A solution to the ecological inference problem: reconstructing individual behavior from aggregate data, Princeton University Press, 1997.

[10] Tanaka, H., Vejima, S., Asai, K. Linear regression analysis with fuzzy model. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 12, N.6 Pág. 903-907, 1982.

[11] Tanaka, Hideo y Lee, Hackwan. Interval Regression Analysis by Quadratic Programming Approach. IEEE Transaction on fuzzy systems, Vol. 6 N. 4 1998

[12] Tsaur, Ruey-Chyn; Wang, Hsiao-Fan. Outliers in fuzzy regression analysis. *International Journal of Fuzzy Systems* Vol. 1, N. 2 pág. 113-119, 1999

[13] Wellhofer, E. Spencer, Party realignment and voter transition in Italy, 1987-1996, *Comparative Political Studies*, Vol 34 (2), Pág. 156-186, March 2001.

[14] Yu Jing-Rung, Tzeng Gwo-Hshiung y Li Han-Lin. General fuzzy piecewise regression analysis with automatic change-point detection. 2001. Fuzzy Sets and Systems, vol 119, Pág 247-257.

[15] Zadeh L. A. The concept of a linguistic variable and its application to aproxímate reasoning I, II, III" Information Sciences, 8, Pag. 199-251, 301-357; 9, Pag 43-80.

APLICACIONES DE REDES NEURONALES ARTIFICIALES EN LA INGENIERIA Y CIENCIA NUCLEAR

Rafael Correa
UTEM. Santiago de Chile
rcorrea@utem.cl

Ignacio Requena
Universidad de Granada. España
requena@decsai.ugr.es

Resumen

En este trabajo se presenta un análisis cuantitativo de espectros PIXE de tipo nuclear, obtenido de muestras orgánicas, mediante el uso de redes neuronales artificiales (RN). Se aplican RN de propagación hacia delante entrenadas con el algoritmo de retropropagación del error, para determinar las concentraciones de P, S, K y Ca. Previamente se realizó un estudio sobre el estado del arte del uso de RN en el ámbito de la ingeniería y ciencia nuclear, siendo analizadas 60 referencias que reportan aplicaciones de interés en este campo.

1. Introducción

La aplicación de nuevas tecnologías de información en la ingeniería y ciencia nuclear, que se enmarcan en el ámbito de la inteligencia artificial (IA) y la neurocomputación se desarrollan con intensidad creciente desde finales de la década de los ochenta en adelante. Los paradigmas de Computación Flexible (Soft Computing) han sido muy usados y aplicados en la ciencia nuclear y física de altas energías. En especial, tienen un gran impacto las aplicaciones de RN en un amplio espectro de problemas.

En este trabajo se describe una aplicación de las RN en el análisis espectral, donde el rol de la red neuronal es identificar la presencia y concentración de determinados elementos de interés (en concreto, P, S, K, Ca), presentes en una muestra orgánica de especies de peces capturados en el Lago Rapel, en la VI Región de Chile. La elaboración de las muestras, su irradiación con haz de protones de (3.0 – 4.0) MeV de energía y la detección de los espectros de emisiones X, se realizaron en Laboratorio de Física Experimental de la Facultad de Ciencias de la Universidad de Chile, en el transcurso del año 2003.

Además y como aspecto previo, se refleja un estudio sobre el estado del arte de la aplicación de RN en el ámbi-to de la Ingeniería Nuclear y la Física de Altas Energías. Se realizó un estudio sobre las metodologías clásicas y los modelos de RN más usados en este campo de la ingeniería y la ciencia, y se incorpora una taxonomía y clasificación de las aplicaciones estudiadas, según los modelos de RN.

Se han considerado como relevantes para este estudios, los siguientes tópicos de la ingeniería nuclear y de la física nuclear y de altas energías:

- Análisis Espectral e Identificación de Isótopos (AE).
- Operación y Control de Procesos en Reactores y Plantas Nucleares de Potencia (RNu).
- Dosimetría y Espectrometría de Neutrones (DEN).
- Física de Partículas de Altas Energías (FAE).

Las aplicaciones de RN en cada uno de los tópicos anteriores de la ingeniería y ciencia nuclear, se clasificaron según los siguientes tipos y modelos:

a) Redes neuronales de propagación hacia delante (RPD)
b) Redes neuronales recurrentes (RNR).
c) Redes neuronales de función radial base (RFB).
d) Redes neuronales autoorganizadas (RAO).
e) Otros tipos de RN. Modelos híbridos (MH).

2.- Revisión de la Literatura

Se revisaron un total de 81 referencias directamente relacionadas con aplicaciones de RN en los tópicos de la ingeniería y ciencia nuclear antes mencionados, que datan desde 1988 en adelante, resaltando algunos trabajos iniciales [17-20, 35, 36, 39, 46, 72].

Tabla 5: Clasificación temporal de los trabajos analizados

Año	AE	RNu	DEN	FAE	TOTAL
88-91	0	2	0	3	5
92-94	2	2	0	2	6
95-96	3	1	0	0	4
97-98	4	4	2	8	18
99-00	7	3	2	5	17
01-02	5	2	5	2	14
03-	1	8	1	7	17

La tabla 1, muestra el número de publicaciones anuales, en un período que abarca los últimos 15 años. De éstas, 33 corresponden al período 1988 – 1998, y 48 de 1999 en adelante, lo que indica la tendencia al creciente uso de nuevas tecnologías de información, y en particular de las RN, en este campo de acción científica y tecnológica. En la sistemática de aplicación de RN en este campo se aprecian 20 tipos y variantes de RN, clasificadas en las clases antes mencionadas y 6 con sistemas híbridos.

El tipo de red más usado es el de propagación hacia delante, entrenadas con el algoritmo de retro propagación del error, en cada uno de los tópicos considerados. Esto coincide con lo indicado en [44] donde se revisan más de

60 publicaciones sobre aplicaciones de paradigmas de la inteligencia artificial en este campo. También se observa que la mayor cantidad y variedad de aplicaciones están relacionadas con la operación y control en plantas nucleares de potencia, y en segundo lugar en el análisis espectral.

La revisión muestra el interés creciente de la comunidad científica, en el ámbito de la ingeniería y ciencia nuclear, por el uso de nuevas tecnologías de información de IA. Esto es plenamente coherente con lo planteado en [65] donde se analizan algunas tendencias de aplicaciones de inteligencia artificial en física nuclear y de altas energías, resaltando el rol de las RN en lo relativo a clasificación y selección de eventos, identificación de partículas, estimación y clasificación de parámetros en análisis espectral, reconstrucción de energías y análisis de trazas en complejos sistemas de detección de partículas de altas energías. Se concluye que las nuevas tecnologías de información de IA operan bastante mejor, en este tipo de análisis científico, que las metodologías tradicionales "no – IA".

En la Tablas 2 y 3 se presenta una clasificación de aplicaciones de RN y sistemas híbridos en el ámbito de la ingeniería y ciencia nuclear, según el modelo de RN o sistema híbrido usado, en cada uno de los tópicos de la ingeniería nuclear considerados. Se puede apreciar que las RN de propagación hacia delante son las más usadas, pero se observa una tendencia a la investigación con nuevos modelos de RN y sistemas híbridos, sobre todo en el ámbito de los reactores y plantas nucleares.

Tabla 2: Sistemática de Aplicaciones referidas de RNs

RN	AE	Rnu	DEN	FAE
RPD	13,14, 32-36, 62,63,64,65, 69,80	3,24,26,27,28 ,38,39,72,56	5,23,42 ,43,44, 53,70	2,7,8,15,22 ,25,29,45, 47,61, 78
RNR	35,36,71	40,41,76	1,15	12
RAO	37,74	4,52,66		6,54
RFB	62,63	66,67		
Otros	21,58,60,75	16,52	79	10,73

Tabla 3: Sistemática de Aplicaciones referidas de SHs

Sistemas híbridos (SH)	AE	RNu	FND	FAE
RN y Alg. Genéticos	49	42, 59,57		24
RN y Sist. Expertos		40		
RNN y Sist. Ddifusos		3,50		

En las aplicaciones de RN citadas en cada tópico de la ingeniería nuclear se utilizan diversos tipos de algoritmos de entrenamiento, casi siempre supervisados, pero hay bastantes aplicaciones con entrenamiento no supervisado, sobre todo si la red actúa en tiempo real y en línea.

Entre los entrenamientos supervisados el algoritmo que más se usa es el de retro propagación (RP) del error, o versiones modificadas del mismo. Otros algoritmos usados son, algoritmos específicos de análisis espectral, el de Marquard – Levenberg o de matriz ortogonal.

3.- Análisis de espectros PIXE con redes neuronales.

La presente aplicación tiene por objeto el uso de técnicas de redes neuronales en el análisis de espectros nucleares de emisiones X inducidas por protones, y la determinación de las concentraciones de los siguientes elementos: P, S, K, Ca, presentes en muestras de sustancias orgánicas provenientes de especies de peces capturados en aguas dulces. Estos estudios de carácter experimental aplicado comprende investigaciones de análisis espectral PIXE en diversas áreas científicas [49, 50] y el objeto del uso de redes neuronales es alcanzar una automatización del reconocimiento espectral, lo que redunda en disminución de costos y tiempos en el proceso de análisis y de investigación en general.

Cabe mencionar que las aplicaciones de redes neuronales en el análisis de radiaciones X, si bien son escasas, debido a las complicaciones que presentan los espectros por la superposición de las líneas espectrales de interés con el ruido presente, y que por ende dificultan cualquier metodología de análisis, tienen antecedentes y perspectivas interesantes en investigaciones de primer nivel, como las enmarcadas en el proyecto XEUS [60].

3. 1) Situación experimental y datos espectrales.

La actual aplicación es parte de un diseño experimental que comprende principalmente las siguientes etapas:

Preparación de muestras.

- Irradiación de muestras con haz de protones de (3 – 4) MeV de energía, producido por un acelerador de partículas Van de Graff.
- Sistema de detección de emisiones X, que tiene como elemento principal un detector Na(I).
- Sistema computacional y multicanal de proceso de espectros PIXE.
- Redes neuronales para el reconocimiento automático de concentraciones de elementos de interés.

El proceso experimental (las cuatro primeras etapas), se realizan en el Laboratorio de Física Experimental de la Facultad de Ciencias de la Universidad de Chile, cuyos resultados son los datos utilizados en la RN aplicada.

La figura 1 muestra un ejemplo de espectro usado en esta aplicación, donde E es la energía y que varía de 0 a 60 KeV y N es el número de emisiones X para cada energía, es decir, el gráfico es una representación de N = N (E).

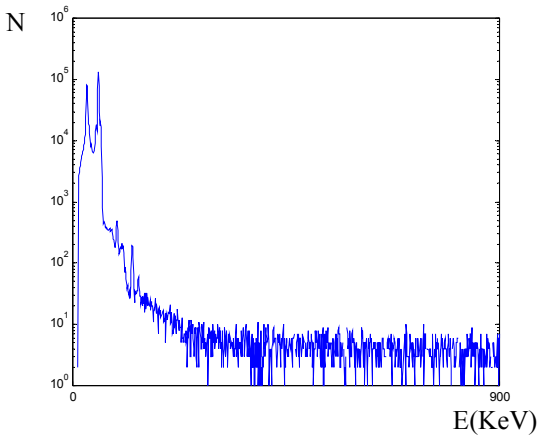


Figura 1: Espectro PIXE

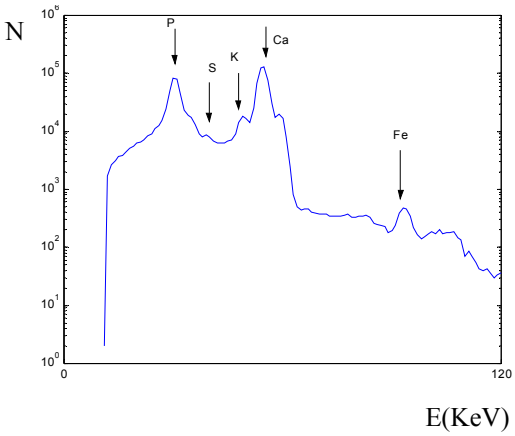


Figura 2: Región Espectral Analizada.

La región espectral relevante en el este análisis (de bajas energías, E ≤ 8.0 KeV), se ve en la figura 2. Las líneas espectrales medidas en este tipo de espectros son características y representan transiciones nucleares de núcleos energéticamente excitados a estados de energía inferior y son propias de isótopos y elementos. Las líneas identificadas en el espectro anterior corresponden a emisiones X, del tipo K_α y K_β, para cada uno de los elementos mencionados, ya que son las más intensas en espectros de emisiones X, y cuyas energías para los elementos P, S, K y Ca se especifican en la Tabla 4:

Tabla 4: Energía espectral de los elementos P, S, K y Ca

Elemento	Nº Atómico Z	K_α (Kev)	K_β (KeV)
Fósforo – P	15	2.0137	2.0127
Azufre – S	16	2.3078	2.3066
Potasio – K	19	3.3138	3.3111
Calcio – Ca	20	3.6917	3.6881

En los espectros experimentales, las líneas K_α y K_β para cada elemento aparecen superpuestas, pero esta situación debe considerarse en la calibración realizada, y no tiene mayor significado en el análisis en sí.

3. 2) Red Neuronal de propagación hacia delante.

Para determinar la concentraciones de los elementos de interés en las muestras investigadas se utilizan RN de propagación hacia delante con el algoritmo de retropropagación del error. Se realizan dos tipos de aplicaciones:

(A.1) – Para determinar la concentración de cada elemento (P, S, K, Ca) se usa una RN de propagación hacia delante (4– 3– 1), para cada elemento. Las entradas a la red son los 4 datos experimentales: masa de la muestra, intensidad del haz de protones, superficie de la muestra perpendicular al haz y número de emisiones X en la línea espectral del elemento investigado. La salida es la concentración del elemento investigado. Se utiliza, en toda la red, una función de activación sigmoide. En esta modalidad se utilizan 11 espectros en el entrenamiento y 1 en el conjunto de test, para cada elemento de interés (esto se hizo así porque se disponía solo de 12 espectros).

(A.2) – Se usan dos redes neuronales de propagación hacia delante (5– 3– 2) y cada una determina las concentraciones de dos elementos, agrupados en forma aleatoria, obteniendo en concreto los pares (S, P) y (K, Ca). Las entradas a la red son las 3 primeras señaladas en (A.1) y el número de emisiones X en las líneas espectrales de los dos elementos considerados. Las salidas son las concentraciones de cada elemento. Se utiliza, en toda la red, una función de activación sigmoide. En el conjunto de entrenamiento hay 33 espectros y 3 para el test, para cada RN, obtenidos a partir de los 12 espectros disponibles.

Los parámetros de la RN usados en el entrenamiento son: tasa de aprendizaje 0.6, momentum 0.8 y un error de 0.01 (para finalizar el entrenamiento). El número de ciclos en el proceso de entrenamiento es diferente para cada red, pero varía en el intervalo (800 – 1200 ciclos) en la modalidad (A.1) y (2000 – 2500) en la modalidad (A.2).

3. 3) Análisis y Resultados

En la Tabla 5 se reflejan los resultados de (A.1), para el test de cada una de las 4 RN usadas.

Tabla 5: Comparación entre concentración real y concentración dada por la red. (A.1)

Elemento	**Nº Atómico**	**Concentración Real %**	**Respuesta de RN %**
P	15	1.7×10^{-2}	1.76×10^{-2}
S	16	1.0×10^{-3}	1.2×10^{-3}
K	19	2.5×10^{-3}	2.47×10^{-3}
Ca	20	1.8×10^{-2}	1.8×10^{-2}

La determinación de la concentración de cada elemento es lo suficientemente precisa como para calificar los resultados como buenos.

En la modalidad (A.2), a cada una de las dos RN utilizadas se le realizaron tres consultas:

- T1.- Determinar la concentración de S (K) en una muestra donde no hay P (Ca) ,
- T2.- Determinar la concentración de P (Ca) en una muestra donde no hay S (K),
- T3.- Determinar las concentraciones de S (K) y P (Ca), en muestra con los dos elementos presentes.

Los resultados a estas consultas (T1, T2 y T3), para cada RN, se reflejan en las tablas 6 y 7, donde se puede ver, que a pesar de lo reducido del conjunto de entrenamiento, la RN realiza una predicción válida de las concentraciones de los cuatro elementos de interés usados.

Tabla 6: Respuesta de la RN del par (S, P)

Consulta	Elemento	NºAtóm.	Concentr. Real %	Respuesta de RN %
T1	P	15	0.0	$\sim 10^{-4}$
	S	16	1×10^{-3}	2×10^{-3}
T2	P	15	1.7×10^{-2}	1.79×10^{-2}
	S	16	0.0	$\sim 10^{-6}$
T3	P	15	1.7×10^{-2}	1.75×10^{-2}
	S	16	1.0×10^{-3}	0.96×10^{-3}

Tabla 7: : Respuesta de la RN del par (K, Ca)

Consulta	Elemento	NºAtóm.	Concentr. Real %	Respuesta de RN %
T1	K	19	2.5×10^{3}	2.24×10^{-3}
	Ca	20	0.0	$\sim 10^{-4}$
T2	K	19	0.0	$\sim 10^{-4}$
	Ca	20	1.8×10^{-2}	1.97×10^{-2}
T3	K	19	2.5×10^{3}	2.06×10^{-3}
	Ca	20	1.8×10^{-2}	2.01×10^{-2}

Podemos resaltar que la metodología de usar una RN, definida y entrenada, para determinar la concentración de determinado elemento de interés, es una idea novedosa que se robustece por los resultados. Las entradas a la red no provienen sólo de datos del espectro detectado, sino que se basan en los antecedentes propios del experimento y se relacionan con sus distintas componentes.

En general, se puede apreciar por los resultados, que la modalidad (A.1) se comporta mejor que la (A.2). Un factor importante a considerar, es el hecho que el número de ejemplos de entrenamiento es bajo en relación al nú-mero de parámetros a definir en el entrenamiento. Esta aplicación, orientada a la utilización de las técnicas de RN en la automatización del análisis de espectros PIXE, posee varios elementos originales, pero también se deben considerar situaciones correctivas como:

- Aumentar el número de ejemplos de entrenamiento para obtener una mejor relación con la cantidad de parámetros a definir en el entrenamiento.
- Aumentar el número de entradas a la red, y la utilización de otros modelos y topologías de redes.

Un elemento a considerar para optimizar los resultados es ejecutar acciones correctivas del ruido, que en las partes bajas del espectro es considerable.

3.-Conclusiones.

Este trabajo muestra el estado del arte de la Aplicación de RN en la Ingeniería y Ciencia Nuclear. Se han revisado 78 trabajos sobre el tema, destacando el incremento del uso de RN en la ingeniería nuclear.

Se ha presentado una aplicación de RN en análisis de espectros PIXE, para determinar la concentración de determinados elementos: P, K, S, y Ca, en muestras orgánicas de peces procedentes del Lago Rapel en Chile. Los primeros resultados, teniendo en cuenta los pocos datos disponibles, son prometedores, y deberán mejorarse en experiencias próximas, con más datos experimentales disponibles, y diseñando la experiencia de forma más completa.

4.-Referencias

[1] S.M. Abel-Halim et al. "The characteristics of proton and induced neutron-emulsion high energy interactions". Chaos, Solitons and Fractals 16 (2003) 691 – 697

[2] M. Ambriola et al. "Supervised algorithms for particle classification by a transition radiation detector". Nuclear Instr. Methods Phys. Res., A 510 (2003) 362–370

[3] W. S. Andrews, B. J. Lewis, D. S. Cox. "Artificial neural networks models for volatile fission product release during severe accident conditions". Journal of Nuclear Materials 270 (1999) 74 – 86.

[4] I.K. Attieh et al. "Pattern recognition techniques for transient detection to enhance nuclear reactor's operational safety". D. Nuclear Eng, U. Tennesse, Knoxville.1996

[5] C.C. Braga, M.S. Dias. "Neural networks for unfolding neutron spectra measured by means of bonner spheres". Nuclear Inst. Meth. Phys. R., A 476 (2002) 252–255.

[6] H. Bischof, R. Fruhwirth. "Recent developments in pattern recognition applications in high energy physics". Nuclear Inst. Meth. Phys. R., A 419 (1998) 259 – 269.

[7] M. Bonesini, et al.. "Application of artificial neural nets to Shashlik calorimetry" .Nuclear Instruments and Methods in Physics Research A 389 (1997) 148 – 153.

[8] E. Boos, L. Dudko, D. Smirnov. "Optimized NN search of Higgs boson production with the Tevatron". Nuclear Inst. Meth. Phys. R., A 502 (2003) 486–488.

[9] M. Boroushaki et al. "An intelligent nuclear reactor core controller for load following operations, using recurrent NN and fuzzy systems". Annals Nuclear Energy 30 (2003) 63 – 80.

[10] P. Busson, R. Nobrega, J. Varela. "Modular neural networks for online event classification in high energy physics". Nucl. Inst. Meth. Phy. R., A 410 (98) 273– 283
[11] Z. Cao, L.F. Miller, M. Buckner. "Implementation of dynamic bias for neutron–photons pulse shape discrimination by using neural network classifiers". Nuclear Instrum. Methods Phys. Rese. A 416 (1998) 438 – 445.
[12] T. Carli, B. Koblitz. "A multi variate discrimination technique based on range searching". Nuclear Inst. Meth. Phys. Res., A 501 (2003) 576 – 588
[13] C. Citterio et al.. "Function appximation – A fast convergence neural approach based on spectral analysis". IEEE Transac. Neural Networks10 (4): 725-740 Jul 1999.
[14] C. Citterio et al.. "Spectral approximation of functions by using three layered feed forward neural network". Dep. Electronics Inform., Politecnico Milano. 1996.
[15] E. Couto. "Separation of gamma and muon particles using multilayer neural networks". Stanford Univ., 2001
[16] Y.J. Choi et al. "Hybrid accident simulation methodology using artificial neural networks for nuclear pwer plants". Information Sciences – 2003 – In press.
[17] B. Denby. "Neural networks and cellular automata in experimental high energy physics". Computer Physics Communications, 49 (1988) 429.
[18] B. Denby. "A bat's ear view of neural nets in physics". Nuclear I. Methods Phys. Res. A 399 (97) 8–11.
[19] B. Denby. "Neural networks in high energy physics: a ten years perspective". Computer Physics Communications, 119 (1999) 219 – 231.
[20] B. Denby. "Neural network tutorial for high energy physics". Int. Workshop of Software Engineering, Artif. Intelligence and Expert Systems for High Energy and Nuclear Physics, Lyon, France, March – 1990.
[21] L. Dolmatova et al. "Hierarchical NN modeling for infrared spectra interpretation modified starches". J. Chemometry Inf. Com. Science, 39 (1999) 1027–1036
[22] D. Dorfan et al. "Gamma/hadron separation with neural networks". Univ. California, Santa Cruz, 2000.
[23] B.P. Dubey, S.K. Kataria, A.K. Mohanty. "Neural networks fits to neutron induced reactions using weighted lest – mean – squares". Nuclear Instruments and Methods in Physics Research A397 (1997) 426 – 439
[24] B.P. Dubey, V. Jagannathan, S.K. Kataria. "Quick and reliable estimation of power distribution in PHWR by ANN". Annals Nuclear En 25 (1998) 567–579.
[25] M.Y. El-Bakry, K.A. El-Metwally. "Neural network model for proton – proton collision at high energy". Chaos, Solitons Fractals 16 (2003) 279 – 285.
[26] M. Embrechts, S. Benedek. "Hybrid identification of unlabeled nuclear power plan transients with artificial neural networks". D. Decision Sciences and Engineering Systems, Rensselaer Polytechnic Inst., New York , 1998.
[27] E. Eryurek, B. R. Upadhyaya. "Sensor validation for power plant using adaptive backpropagation NN". IEEE Trans. Nuclear Science, 37, 2, 1040–1047, 1990
[28] E. Fernandes, C. Pereira. "Nuclear fuel loading pattern optimisation using a neural network". Annals of Nuclear Energy 30 (2003) 603 – 613
[29] M.T. Hagan, M.B. Menhaj. "Training feedforward neural networks with Marquardt algorithm". IEEE Trans. Neural Networks 6 (1994)
[30] F. Hakl, M. Halvacek, R. Kalous. "Application of neural networks optimized by genetic algorithms to higgs boson search". M. Trade Ind., Czech Rep, RP 4210/69/97
[31] D. Horn. "Neural computation methods and applications: Summary talk of the AI session". Nuclear Inst. Meth. Phys. Res., A 389 (1997) 381–387
[32] W.W. Hsieh, A. Wu. "Nonlinear singular spectrum analysis by neural networks". Neural Networks, Jul 2001.
[33] P. Inakollu et al. "Effectiveness of using artificial neural networks for elemental concentrations from LIBS data". XV Int. C. Computer Appl. Indus. Eng, 2002, USA
[34] L.J. Kangas et al. "Alpha Spectral Analysis via Artificial Neural Networks". IEEE Nuclear Science Symposium and Medical Imaging Conf. – 1994
[35] P.E. Keller, R.T. Kouzes. "Gamma Spectral Analysis via Neural Networks". IEEE Nuclear Science Symposium and Medical Imaging Conf. – 1994.
[36] P.E. Keller et al. "Nuclear Spectral Analysis via Artificial Neural Networks for Waste Handling". IEEE Tran. Nuclear Science 42 (4), Aug 1995.
[37] S. Kelly et al. "Parametric Models and Spectral Analysis for Classification in Brain Computer Interfaces". Proc. 14ª Int. C. Design Signal Proc., Greece, July 2002
[38] S. Keyban, X. Song, M. Kelly. "Nuclear fuel pellet inspection using artificial neural networks". Jour. Nuclear Materials 264 (1999) 141– 154
[39] C.C. Ku, K.Y. Lee, R.M. Edwards. "Neural network for adapting nuclear power plant control for wide rang operation". Tran. Am. Nuclear Soc., 63, 114–115, 91
[40] C.C. Ku et al.. "Improved nuclear reactor temperature control using diagonal recurrent neural networks". IEEE Tans. on Nuclear Science, Vol. 30, Nº6, 1992.
[41] C.C. Ku, K.Y. Lee, R.M. Edwards. "Diagonal recurrent neural network for dynamic systems control". IEEE, Trans. on Neural Networks, 1992.
[42] S.Y. Lee, J.L.Kim, K.J. Lee. "Feasibility of using neural networks to unfold the response of multi-element TLD – for mixed field dosimetry". Korea Atomic Energy Research Institute, Taejon – Korea. 2000
[43] S.Y. Lee, B.H. Kim, K.Y. Lee. "An application of artificial neural intelligence for personal dose assessment using a multi-area OSL dosimetry system". Radiation Measurments, 33 (2001) 293 – 304.
[44] J. Loskiewics, J. Swakon, K. Kulczykowska. "Estimation of thermal neutron absorption cross-section from K, U, Th concentrations for Miocene rocks from Piedmont in Poland using artificial neural networks". Applied Radiation and Isotopes 52 (2000) 1459 – 1465
[45] B. Majorovitz, H.V. Klapdor-Kleingrothaus. "Digital Pulseshape analysis by neural networks for the Heidelberg-Moscow double beta decay experiment". Max Planck Inst. fur Kernphysik, Heidelberg – Germany, 2000
[46] M. Marseguerra et al. "Artificial neural networks applied to multiple signals in nuclear technology". Progress Nuclear Energy, Vol. 27, Nº 4, 297 – 304, 1992.

[47] M. Milek, P. Patel. "Neural network tagging in toy model". Nuclear I. Me. Phy. R., A 425 (1999) 577–588
[48] A.C.A. Mol, A.S. Martínez, R. Schirru. "A neural model for transient identification in dynamic proceses with "don´t know"response". Annals of Nuclear Energy 30 (2003) 1365 – 1381
[49] J.R. Morales, M. I. Dinator. "Characterization of colour pigments in pre-columbian chilean potieries by PIXE elemental analysis". Journal of Radioanalytical and Nuclear Chemistry, V. 140 Nº 1 (1990) 133 – 139 .
[50] J.R. Morales et al. "Sample preparation of archaelogical materials for PIXE análisis". J. Radio-analytical Nuclear Chemistry, V. 187 Nº 1 (1994) 79 – 89
[51] U. Muller. "Artificial intelligence – applications in high energy and nuclear physics". Nuclear Instruments and Methods in Physics Research A 502 (2003) 811 – 814 [52] K. Nabeshima et al. "Nuclear reactor monitoring with the combination of neural network and expert system". Math. Computers Simulation, 60 (2002) 233-244
[53] W.V. Nunes et al. "Explosives detection using prompt-gamma neutron activation and neural networks". Applied Radiation and Isotopes 56 (2002) 937 – 943
[54] R. Odorico. "Learning vector quantization with training count ". Neural Networks 10, 6, 1083-1088, 97.
[55] J.J. Ortiz, I. Requena. "Using neural networks to predict core parameters in BWR". Nuclear Science and Engineering. 143 (2003) 1 – 14.
[56] J.J. Ortiz, I. Requena. "Using a Multi-State Recurrent NN to Optimize Loading Patterns in BWRs". Accepted in Annals of Nuclear Energy.2004.
[57] J.J. Ortiz, I. Requena. "Order Coding Genetic Algorithms to Optimize Fuel Reload in a Nuclear Boiling Water Reactor". Nuclear Scie. Engine. 2004
[58] M. Panda. "The Spectral Support System: A Software Product with Embedded Artificial Neural Networks in the Analysis of LIBS Spectral Data". Master Project Report, Mississippi State University, Aug 1998
[59] C. Pereira, C. Lapa. "Coarse-grained parallel genetic algorithm applied to nuclear reactor core design optimization problem". Annals Nuclear Ener. 30 (2003) 555–565.
[60] L.L. Perlovsky, et.al. "Einstenian NN for Spectrum Estimation". Neural Net., 10,9, 1541–1546, 97
[61] A.L. Perrone et al. "Offline analysis of HEP events by "dynamic perceptron" neural network". Nuclear Inst. Methods Physics Res., A 389 (1997) 210 – 213
[62] T. Philip et al. "Application of Artificial Neural Networks in the Analysis of LIBS Spectral Data". XI Inter. C. Computer Applic. Industry Engineering, 1998.
[63] T. Philip et al. "Appllying neural networks to ana-lyze spectra to verify element". Depart. Computer Science Mississippi State University, Mississsippi – USA, 1999.
[64] V. Pilato et al. "Application of Neural Networks to Quantitative Spectrometry Análisis". Nuclear Instr. Meth. Physics Res., A 422(1-3), 423-427, Feb 11 1999
[65] S. Pozzi, F.J. Segovia. "Application of Stochastic and Artificial Intelligence Mtehods for Nuclear Material Identification". D. Nuclear Engineering, Polyt. Milan, 99
[66] D. Roverso. "Soft computing tools for transient classification". Information Science127 (2000) 137 – 156
[67] M. Sadighi, S. Setayashi, A.A. Salehi. "PWR fuel management optimization using neural networks". Annals of Nuclear Energy 29 (2002) 41 – 51
[68] H.T. Siegelmann, E. Nissan, A. Galperin. "A novel neural/symbolic hybrid approach to heuristically optimized fuel allocation automated revision of heuristic nuclear engineering". Adv. Engin Softw. 28 (97) 581–592
[69] R. Tagliaferri et al. "Spectral Análisis of Stellar Light Curves by Means of Neural Networks". Astronomy and Astrophysics Supp. Series 137 (2) 391–405. 1999
[70] T. Tambouratzis, M. Antonopoulos. "Instability locali-zation with artificial neural networks ". Annals of Nuclear Energy 29 (2002) 235 – 253
[71] V. Tchistiakov et al. "Neural network modelling for very small spectral data sets : reduction of the spectra and hierarchical approach". Chemometry Intell. Laborat. Systems, 54 (2000) 93– 10
[72] R.E. Uhrig. "Potential application of neural networks to the operation of nuclear power plants". Nuclear Safety, Vol. 32, Nº 1, 68 – 79, 1991
[73] A.Vaiciulis. "Support vector machines in analysis of top quark production". Nuclear Instruments and Methods in Physiscs Research A 502 (2003) 492 – 494
[74] E.F. Vieira, J.D. Ponz. "Automated Spectral Classification Using Neural Networks". Astronom. Data Analysis Softw. Systems VII, ASP Conf Series, 145, 1998
[75] V. Vigneron et al. "Statisical Modelling of Neural Networks in Gamma-Spectrometry". Nuclear Instr. Meth. in Physics Research – A 369 (2 - 3) 642 – 647. 1996
[76] M. Weerasinghe, J.B. Gomm, D. Williams. "Neural networks for fault diagnosis of a nuclear fuel processing plant at different operating points". Control Engineering Practice 6 (1998) 281 – 289 .
[77] D.O. Whiteson, N.A. Neumann. "Support vector regression as a signal discriminator in high energy physics". Neurocomputing, in press (april – 2003).
[78] M. Wolter. "Measurement of physical quantities in the Bayesian framwork using neural networks". Institute Nuclear Physics, Krakow, Poland. 1999
[79] P.C. Womble al. "Detection of explosives PELAN system". American Inst. Physics 0-7354-0015(2001)
[80] E. Yoshida et al. "Application of NN for the analysis of gamma-ray spectra measured with Ge spectrometer". Nuclear Inst. Meth. Physics Res.A 484 (2002) 557-563.
[81] J. Zimmermann, C. Kiesling, P. Holl. "Class separation and parameter estimation with NN for the XEUS wide field imager". Max Planck I. Phy., 2001.

REGLAS DE ASOCIACIÓN APLICADAS A LA DETECCIÓN DE FRAUDE CON TARJETAS DE CRÉDITOS

María-Amparo Vila Miranda vila@decsai.ugr.es
Daniel Sánchez Fernández daniel@decsai.ugr.es
Luis Cerda Leiva lcerda@cientec.com

Resumen

Las reglas de asociación han sido reconocidas como los modelos mejor estudiados para minería de datos. En este documento, planteamos el uso de este modelo de extracción de conocimiento, para obtener patrones de comportamiento fraudulento desde bases de datos transaccionales de tarjetas de crédito, a fin de detectar y prevenir el fraude.

Palabras Clave: Reglas de Asociación, Minería de datos, Fraude con tarjetas de crédito.

1. INTRODUCCION

Como se ha planteado en [29], la minería de datos cuyo objetivo es la obtención de conocimiento útil desde datos almacenados en grandes repositorios, es reconocida como una necesidad básica en muchas áreas, especialmente aquellas relacionadas con los negocios.
La teoría de conjuntos difusos contribuye ciertamente a este propósito. Es ampliamente reconocido que muchas de las relaciones que se dan en el mundo real son intrínsicamente difusas y las decisiones que se toman en las áreas comerciales habitualmente utilizan información incompleta o imprecisa.
Uno de los modelos mejor estudiados para minería de datos es el de reglas de asociación [1], este modelo asume que el objeto básico de nuestro interés es un ítem, y que el dato aparece en la forma de conjunto de ítem llamado transacción. Las reglas de asociación son implicaciones que relacionan la presencia de ítem en las transacciones. Las transacciones son la estructura básica desde donde las reglas de asociación son obtenidas.

En nuestro caso las transacciones son compras que se realizan utilizando como medio de pago tarjetas de crédito emitida por la misma casa comercial que es afectada por la transacción fraudulenta.

2. LA SITUACION BAJO ESTUDIO

Los datos de la operación diaria se encuentran dispersos en distintos sistemas de tratamiento y representación de información en diversas plataformas de hardware y software, lo cual no facilita la extracción de conocimientos.

La carencia de un efectivo sistema que permita detectar y prevenir la actividad fraudulenta obedece fundamentalmente a la falta de confianza que los sistemas permitan en forma fehaciente detectar la actividad fraudulenta, la relación de transacción fraudulenta versus transacciones lícitas en el caso bajo estudio es de 3 por cada mil, lo que se traduce en una alta probabilidad de obtener falsos positivos lo que produciría un efecto contrario al deseado cual es contribuir a la fidelización del cliente.

Se ha detectado como en otros ámbitos donde se utilizan como medios de pago las tarjetas de crédito, un rápido incremento de la actividad fraudulenta, una análisis estadístico preliminar de los datos, permitió conocer que existe un importante incremento de la actividad fraudulenta que alcanza al 15% en el último año, presentando una mayor incidencia las transacciones generadas fuera del ámbito de los puntos de ventas propios de la empresa bajo estudio, es decir, aquellas efectuadas en los locales de venta de sus socios comerciales.

Lo anterior respalda la urgente necesidad de pasar de una posición reactiva o una pro-activa, la tasa de crecimiento de la actividad fraudulenta conlleva a un rápido deterioro de la relación con los socios comerciales y clientes, como también de un incremento de costos de operación para todos los involucrados, lo que significa, pasar a una política real de detección y prevención del fraude con un sistema en línea que resguarde los intereses de todos los involucrados.

3. LOS SISTEMAS INTELIGENTES EN LA DETECCIÓN DE FRAUDE

Según Bhatla et al [2], mientras ninguna de las herramientas y tecnologías revisadas en su estudio puede por sí misma eliminar el fraude, cada técnica agrega valor a la habilidad de detectar el fraude, y postula que una buena práctica ha sido la combinación de varias de ellas, como por ejemplo la combinación de procesos manuales aplicados a transacciones calificadas como sospechosas

de fraude por herramientas inteligentes, por ejemplo revisar el uso múltiple de una tarjeta en diferentes sitios físicos o virtuales durante un corto período de tiempo, en su estudio Bhatla incluye un conjunto de técnicas implementadas por los software de detección de fraude, de todas ellas nosotros hemos decidido utilizar las reglas de asociación.

4. REGLAS DE ASOCIACIÓN

Reglas de Asociación: se trata de buscar las posibles relaciones existentes entre la presencia o no de ítem en determinados conjunto de transacciones, por ejemplo se utilizan para identificar asociaciones entre la presencia o ausencia de artículos en transacciones comerciales, un ejemplo típico de regla de asociación es: el 90% de la gente que compra pan, compra leche, en lenguaje natural se puede expresar como: "la mayoría de la gente que compra pan compra leche", pudiéndose expresar mediante la notación pan $\Rightarrow$ leche, en esta notación la parte izquierda se denomina **antecedente** y la parte derecha **consecuente**.

En problemas del mundo real no es fácil hallar implicaciones como las del ejemplo típico expuesto en el párrafo anterior, que cumplan con total certeza en un conjunto de transacciones y que ocurran un número de veces que las hagan interesante de considerar, con este propósito Agrawal et al, 1993, proponen las medidas de confianza y soporte, mientras que Silverstein, et al, 1998 proponen como medidas alternativa a la confianza el interés.

La definición de Agrawal de una Regla de asociación es una implicación de la forma $X \Rightarrow Y$, donde $X,Y \subset I$ y $X \cap Y = 0$, entendiendo con ello que toda transacción que satisface a X también satisface a Y.

El problema de extraer reglas de asociación se puede dividir en dos subproblemas:

1. Generar todas las combinaciones de ítems, entendiendo éstas como ítemsets, con un soporte por encima de cierto umbral, previamente definido, el llamado soporte mínimo (minsupp). Esas combinaciones suelen encontrarse en la literatura con el nombre de ítemsets frecuentes.

2 Dado un ítemset frecuente Y = i1; i2; : : : ik; $k \geq 2$, generar todas las reglas que contengan todos los ítems de ese ítemset. Para ello, se toman todos los subconjuntos no vacíos X de Y y se generan las reglas $X \rightarrow Y - X$ que cumplan que su confianza es mayor que cierto umbral al llamado confianza mínima (minconf). El valor de la confianza viene dado por soporte(Y) / soporte (X).

Como Y es frecuente, cualquier subconjunto suyo también será frecuente ([Agrawal et al., 1993]), así se tendrá almacenado su soporte. Las reglas que se deriven de Y satisfarán la restricción del soporte, pues Y así lo hace.

Mejoras al problema de la extracción de Reglas de Asociación

El principal inconveniente que presenta la extracción con reglas de asociación estriba en que, en bases de datos lo suficientemente voluminosas, los costes tanto en tiempo como en espacio pueden resultar inviables. Por un lado, para lograr nuestro objetivo hemos de trabajar con todos los ítemsets posibles. Si tenemos m ítems, esto quiere decir que habremos de considerar 2^m posibles ítemsets. Afortunadamente, los algoritmos existentes, aplican técnicas heurísticas para reducir en la medida de lo posible el número de ítemsets que se considerarán, de acuerdo a la estimación de si podrán o no ser frecuentes.

Aún contando con esa capacidad de los algoritmos para disminuir los requerimientos en tiempo de proceso y en espacio de memoria, mejorando en definitiva la eficiencia del procedimiento, todavía se puede encontrar con otros problemas, ésta vez asociados a la aplicación que les pueda dar el usuario final. El conocimiento obtenido es muy dependiente del contexto al que pertenece la información contenida en la base de datos original. Por este motivo, suele ser conveniente y necesaria la intervención de un experto humano que pueda dar una interpretación de las reglas obtenidas, indicando cuáles de ellas son potencialmente útiles y cuáles no, debido por ejemplo a su trivialidad.
Pero la labor del experto humano puede verse entorpecida si el conjunto de reglas obtenido es demasiado amplio. Es por eso que, de cara a optimizar la obtención y posterior interpretación de reglas de asociación dentro de una base de datos se pueden establecer ciertas restricciones:

Restricciones sintácticas. Estas restricciones limitan los ítems que pueden aparecer en una regla. Por ejemplo, podemos estar interesados sólo en las reglas que tengan un ítem específico en el consecuente o en el antecedente, o en una combinación de restricciones.

Restricciones de soporte. Podemos estar interesados sólo en las reglas cuyos ítems aparezcan en un porcentaje de las tuplas de T por encima de un soporte mínimo. Esto quiere decir que para que la información que nos da la regla tenga cierto peso es necesario que aparezca con cierta frecuencia en la base de datos.

Restricciones de cumplimiento. La confianza nos da una medida de la fuerza de una regla. Nos informa sobre la dependencia entre la aparición del consecuente si aparece el antecedente. En general, interesa que la confianza supere un mínimo.

5. EXTENSIONES DE LAS REGLAS DE ASOCIACIÓN

Reglas de Asociación Generalizadas

Reglas de Asociación Generalizadas: es otro tipo de reglas de asociación que permiten evitar perder información relevante cuando se ha construido una taxonomía (conjunto de jerarquías del tipo es-un) que impide conocer cuanto es el soporte conjunto de una familia de ítems, por ejemplo que esos ítems se encuentren registrados en la base de datos por sus marcas las cuales individualmente no cumplen el umbral mínimo establecido, pero en conjunto si lo hacen.

Reglas de Asociación Cuantitativas: cuando los atributos del tipo numérico como por ejemplo la estatura, edad, el sueldo, o categóricos como color, artículo, tienen pocos valores (sus correspondientes dominios son finitos y reducidos), estamos ante un problema similar al planteado en el apartado anterior porque la transformación es directa y poco problemática, pero cuando son muy numeroso o cuando estos atributos son de tipo numérico continuo (real, racional), no es posible especificar en forma sencilla jerarquías para disminuir el nivel de granuralidad, en este caso se habla de trabajar con atributos de tipo cuantitativo y a las reglas de asociación que involucran ítem sobre estos atributos se las denomina reglas de asociación cuantitativa, los problemas de obtener reglas de asociación cuantitativas se denominan QARP: Quantitative Association Rules Problem.

La solución que se ha dado a este problema según Srikant y Agrawal[26], es particionar el dominio en intervalos, sin embargo existen dos problemas a la hora de obtener reglas cuantitativas mediante estos procedimientos:

Problema de mínimo soporte (Minsupp): si el número de intervalos en el que se particiona el dominio de un atributo cuantitativo es grande, el soporte para cada intervalo en solitario puede ser muy bajo. Si no se utilizan intervalos mayores, es posible que no se encuentren reglas que involucren a este atributo por no alcanzar el soporte mínimo necesario.

Problema de mínima confianza (Minconf): Existe cierta pérdida de información cuando dividimos los dominios en intervalos. Algunas reglas pueden tener confianza mínima sólo cuando un ítem del antecedente consiste en un sólo valor o en un intervalo pequeño. Esta pérdida de información se incrementa conforme crece el tamaño de los intervalos.

Estos dos problemas interactúan, si disminuye el tamaño del intervalo se podría resolver el problema de la mínima confianza, pero agravar el problema del mínimo soporte, en el texto de Sánchez[24], se explica que la solución propuesta por Srikant y Agrawal, consiste en dividir los dominios numéricos en intervalos del mismo tamaño, usando un tamaño pequeño, y considerar distintos rangos sobre el dominio mediante el agrupamiento de intervalos en intervalos de nivel superior, estos autores demuestran que esta partición minimiza la cantidad de información que se pierde. Este procedimiento introduce dos nuevos problemas:

Aumento del tiempo de ejecución (ExecTime): Si un atributo cuantitativo tiene n valores (o intervalos), hay una media de $O(n^2)$ rangos que contienen un valor o intervalo específico. Por tanto, el número de ítems por tupla se dispara, lo que también hará mucho mayor el tiempo de ejecución.

Aumento del número de reglas (ManyRules): Si un valor (o intervalo) de atributos cuantitativos tiene soporte mínimo, también lo tendrá cualquier rango en el que esté. Así, el número de reglas también se dispara. Muchas de estas reglas no serán interesantes y en consecuencia, constituirán un gasto innecesario de tiempo y memoria.

La solución que propone Sánchez para el problema del tiempo de ejecución consiste en introducir un umbral máximo de soporte mismas.

6. PROBLEMÁTICA QUE PRESENTA LA EXTRACCIÓN DE CONOCIMIENTO

6.1 PRIMER PROBLEMA QUE PRESENTA LA EXTRACCIÓN DE CONOCIMIENTO.

Las medidas de soporte y confianza evalúan el interés y el grado de cumplimiento de las reglas de asociación con un enfoque meramente estadístico, si bien es cierto, el uso del soporte es bastante generalizado y se acepta como la mejor opción para medir la importancia ya que según [25], reúne las siguientes ventajas:

- Es adecuado para la tarea de medir la relevancia estadística de una regla.
- Es una medida con significado intuitivo, y por tanto la interpretación de los valores de soporte es relativamente sencilla para el usuario.
- El uso del soporte contribuye al diseño de algoritmos eficientes para la búsqueda de reglas de asociación.

No ocurre lo mismo con la confianza, como medida del grado de asociación, implicación o dependencia entre antecedente y consecuente, ya que ha recibido diversas

críticas. Los argumentos en contra de esta medida son los siguientes:

- La confianza no mide adecuadamente el grado de independencia estadística entre el antecedente y el consecuente.
- La confianza no refleja la dependencia negativa entre antecedente y consecuente.
- Por último, la confianza es una medida de probabilidad confeccionada. La probabilidad condicionada no es intuitiva, y por esta razón resulta difícil para un usuario no experto establecer umbrales mínimos de confianza semánticamente significativos a la hora de obtener reglas de asociación.

Para intentar resolver el problema, Sánchez ha propuesto el uso del Factor de certeza

Factor de Certeza: es una representación de la incertidumbre asociada al conocimiento, su principal objetivo es proporcionar unas medidas de incertidumbre más intuitivas que las medidas de probabilidad condicionada y sin algunos problemas de la teoría de la probabilidad, la certeza de un hecho en el intervalo discreto [-1,1], donde la certeza de hecho (H) será 1 si H es un hecho cierto; H será -1 si es un hecho falso y H será 0, si se desconoce el grado de verdad de H.

6.2. SEGUNDO PROBLEMA QUE PRESENTA LA EXTRACCIÓN DE CONOCIMIENTO.

Tal como se plantea en 6.1, el primer problema a resolver en la extracción conocimiento, es la cantidad y calidad de las reglas de asociación en bases de datos relacionales que surgen al aplicar los modelos existentes a repositorios con presencia de atributos con dominios de un alto nivel de granularidad.

El segundo problema, que Sánchez plantea en [24], es que la semántica de las reglas en sí y los parámetros que permiten al usuario interactuar con el proceso de extracción, sabiendo que el ser humano trabaja con descripciones del conocimiento con un menor nivel de granularidad, genera una dificultad de asignar un concepto claro al usuario, esto aumenta cuando se usa como criterio obtener reglas con el mayor soporte y confianza posible, aumentando además la complejidad del mecanismo de extracción.

En síntesis, por una parte, la alta fragmentación de las bases de datos, es decir, la presencia de atributos con alto nivel de granularidad, representa una incompatibilidad con la forma habitual del razonamiento humano y por otra parte, y tal como se planteó anteriormente en 5, al revisar el concepto de Reglas de Asociación Cuantitativas, al intentar resolver este problema de incompatibilidad entre el alto nivel de granularidad y nuestra forma de razonar, genera otros problemas como el del mínimo soporte (minsupp), minima confianza (minconf), tiempo de ejecución (ExecTime) y aumento del número de reglas (ManyRules), para resolver esta problemática de una forma eficiente e intuitiva se sugiere el uso de las Reglas de Asociación difusa.

7. PLANTEAMIENTO DE LA SOLUCION

7.1. SOLUCIÓN PLANTEADA A LA EXTRACCION DE CONOCIMIENTO.

Para Delgado et al [10], las reglas de asociación difusa son uno de los mejores modelos estudiados para la obtención de conocimiento en el campo de la minería de datos. Durante los últimos años, el modelo de encontrar reglas de asociación entre valores de atributos cuantitativos en bases de datos relacionales fue introducido en el campo de la minería de datos por Agramal, Imielnsky y Swami [1].

Como conclusiones de la investigación de Sánchez en [24], se plantea que para resolver las problemáticas que presenta la extracción de conocimiento se debería hacer lo siguiente:

- Usar un nuevo método para definir en conjunto con el experto las etiquetas lingüísticas, obteniendo como resultado una reducción de la granularidad de los dominios de los atributos.

- Utilizar para la generación de reglas de asociación el nuevo concepto de asociación difusa usando en este proceso el soporte y los factores de certeza en lugar de la confianza.

- El uso de factores de certeza que sustituye a la confianza, evita los problemas que representa la confianza sin aumentar la complejidad del proceso de extracción.

- Aplicar el nuevo concepto de regla de asociación muy fuerte, evitando que las reglas de asociación no reflejen la realidad, resultado que se obtiene al tener en cuenta no solamente el soporte de la asociación entre la presencia del antecedente y el consecuente, sino también considerando el soporte de la asociación entre la ausencia del consecuente y la ausencia del antecedente.

Haciendo uso de la metodología recomendada por Sánchez, se elimina la gran cantidad de reglas falsas, obteniendo reglas de mayor calidad, basadas en casos positivos y negativos, que permitan discriminar si la asociación es correcta o no, y el uso de etiquetas

lingüísticas que ayuden a reducir la segmentación del dominio conduce a otorgar reglas más intuitivas por su proximidad a la forma de razonar del experto humano.

7.2. EL TRABAJO A DESARROLLAR.

Habiendo recopilado la información relacionada con las técnicas de lógica difusa y conocida una de las problemáticas más contingentes en el ámbito de la toma de decisiones comerciales, el trabajo futuro estará orientado a la construcción de un sistema en línea que procese la información procedente de una transacción en ejecución, realice una estimación de sus atributos característicos y evalúe el potencial riesgo de fraude, para ello se aplicará técnicas de lógica difusa para generar reglas de asociación obtenidas a partir del conocimiento extraído de las bases de datos fuzzificadas con etiquetas lingüísticas modelados usando los criterios planteados por Sánchez en [24]. Las reglas de asociación se generaran usando los conceptos de reglas de asociación muy fuertes y la nueva definición de dependencia aproximada de Sánchez

7.2.2. Síntesis los objetivos de la propuesta de trabajo futuro son:

- Modelar un sistema inteligente que permita discriminar entre las transacciones genuinas y transacciones fraudulentas [23], utilizando reglas de asociación muy fuerte.

- Implementar un sistema en línea utilizando el modelo anterior que al detectar transacciones con características fraudulentas notifique a los analistas de riesgo con antecedentes suficientemente intuitivos que reduzcan al mínimo la incertidumbre en la toma de decisión.

- Implementar y mantener una base de datos relacional difusa que almacene las transacciones históricas y mantenga vigente el conocimiento adquirido mediante un proceso de extracción de conocimiento de las nuevas conductas fraudulentas.

7.2.3. Beneficios de la Solución propuesta

Este proyecto utilizará las bondades que la lógica difusa provee para enfrentar la problemática de la detección y prevención de actividad fraudulenta, lo que permitirá visualizar entre otros los siguientes beneficios:

Resultados Intuitivos: Hacer más intuitivo al analista de riesgo los resultados entregados por el sistema, esto es un resultado inmediato de la incorporación de cuantificados lingüísticos, los cuales permiten que la extracción de conocimiento se realice en términos más próximos a la forma en que el ser humano resuelve sus problemas.

Extracción de conocimiento útil: la extracción de conocimientos en los términos explicados anteriormente, permite generar un conjunto de reglas de asociación para contribuir a detectar sobre la marcha la actividad fraudulenta con resultados fácilmente entendibles por los analistas de riesgo.

Labor Pro activa: la fuzzificación de las bases de datos operacionales, permitirá contar con los datos necesarios para detectar nuevos comportamientos fraudulentos y actualizar el proceso de verificación en línea con la marcha la actividad fraudelenta.

Solución Integral: generar una solución tecnológica que permita tanto a la empresa bajo estudio como a sus socios comerciales detectar y prevenir la actividad fraudulenta.

8. CONCLUSIONES

Algunas de las conclusiones producto de este trabajo son las siguientes:

1. Las principales necesidades de obtención de conocimiento, tienen relación con las nuevas exigencias que plantea el nuevo escenario competitivo, el cual demanda conocimientos prácticamente en línea del comportamiento de sus clientes, productos, proveedores, vendedores y socios comerciales.

2. Los motivos que han impedido obtener buenos resultados haciendo uso de las herramientas inteligentes tienen su origen en lo heterogéneo de los sistemas de tratamiento y representación de la información junto con las imperfecciones que los datos presentan producto del modelo tradicional que sustenta estos sistemas, todo ello constituye las principales dificultades que han impedido obtener buenos resultados.

3. Es importante y urgente resolver estas necesidades e impedimentos haciendo uso de técnicas de inteligencia artificial, ejemplo de ello es la necesidad de prevención y detección de fraude, derivada del impacto que el incremento de la actividad fraudulenta puede llegar a tener en la relación con los clientes y los socios comerciales.

Finalmente, el fuerte incremento de la complejidad de la gestión comercial producto de la globalización, requiere proveer de herramientas inteligentes que resuelvan en forma más integral la problemática de la extracción de conocimientos desde las bases de datos operacionales para apoyar la toma de decisiones, para ello se deben proveer interfases intuitivas y eficientes como las que el trabajo futuro pretende entregar, y de acuerdo al resultado

de esta investigación preliminar, la lógica difusa se presenta como una alternativa válida y vigente para tal propósito, sobre todo que la metodología propuesta busca simplificar los resultados, reduciendo la cantidad de asociaciones, descartando mediante algoritmos aquellas poco relevantes y haciendo más intuitivos los resultados a través del uso de etiquetas lingüísticas mas naturales al experto humano, por todo lo anterior, se concluye que es factible otorgar una solución más integral, pro-activa, en línea, para resolver una necesidad de conocimiento para la toma de decisiones comerciales como es el caso de la prevención y detección del fraude, mediante la extracción de conocimientos utilizando técnicas de lógica difusa aplicadas a las bases de datos operacionales.

Agradecimientos

Agradecemos la oportunidad de dar a conocer el resultado de nuestra investigación preliminar que plantea una alternativa de solución para resolver mediante la lógica difusa una problemática de urgente solución en los ámbitos comerciales.

Referencias

[1] Agrawal R., Mielinski T., and A.Swami:" Mining association rules between set of items in large databases", In Proc. Of the 1993 ACM SIGMOD Conference, pages 207-216, 1993.

[2] Bhatla Tej Paul, Prabhu Vikram, Dua Amit:"Understanding Credit Card Frauds", 2003.

[3] Bolton, Richard J.; Hand, David J. :"Unsupervised Profiling Methods for fraud detection".

[4] Brause,R; Langsdorf,T; Hepp,M.:"Neural Data Mining for Credit Card Detection".

[5] Cahill, Michel H; Lambert, Diane; Pincheiro, Jose; Sun, Don:"Detecting fraud in the real world".

[6] Castillo, Enrique; Gutierrez, José Manuel; Hadi, Ali S.:"Sistemas Expertos y Modelos de Redes Probabilísticos", 2003.

[7] Cybersource, 5th Annual on line fraud report, 2003.

[8] Chan, Philip K.:"Toward Scalable Leaning with Non-Uniform class an d cost Distribution: A Case Study in Credit Card Fraud Detection".

[9] Delgado Miguel, Sánchez Daniel, Vila María Amparo, Marín Nicolás : "Fuzzy Association Rules: General Model and Aplication", 2003.

[10] Delgado Miguel, Sánchez Daniel, Vila María Amparo: "Acquisition of Fuzzy Association Rules from Medical Data", 2002.

[11] Delgado Miguel, Sánchez Daniel, Vila María Amparo:"Fuzzy Quantified Dependencies in Relational Databases", 1999.

[12] Delgado Miguel, Sánchez Daniel, Vila María Amparo: "A logic based definition of fuzzy cardinality, 1999.

[13] Delgado Miguel, Sánchez Daniel, Vila María Amparo: "Fuzzy cardinality based evaluation of quantified sentences, 1999.

[14] Delgado Miguel, Sánchez Daniel, Vila María Amparo: "Un enfoque lógico para calcular el grado de cumplimiento de sentencias con cuantificadores lingüísticos, 1997.

[15] Delgado Miguel, Sánchez Daniel, Vila María Amparo: "Un método para la evaluación de sentencias con cuantificadores lingüísticos, 1998.

[16] Faughnan, John G.:" International Net-Baset Credit Card/Check Card Fraud with Small Charges".

[17] Frenkel, Brian:"Using Artificial Intelligence to Detect Fraud In Credit Cards",

[18] Fu Y. and Han. J. Meta-rule-guide of asociation rules in relational database. In Prec. First Int. Workshop Integration of KKD with deductive and Object-Oriented databases (KD00D'95).

[19] Galindo, José: "Tratamiento de imprecisión en Bases de Datos Relacionales" , 1999.

[20] Goebel,Michael; Gruenwald, Le:"A Survey of datamining and knowledge software tools".

[21] Klebe Steven W.: "Evaluating on line credit fraud fraud witch artificial intelligence", 1998.

[22] Prodromidis, Andreas L.; Stolfo, Salvatore:"Agent-Based Distributed Learning Applied to Fraud Detectión", 1999.

[23] Sam, Maes; Tuys, Kart; Vanschoenwinkel,Bram; Manderick, Bernad:"Credit Card Detection Using Bayesian and Neural Networks".

[24] Sánchez Fernandez Daniel: "Adquisición de relaciones entre atributos en base de datos relacionales", 1999. (http://decsai.ugr.es/~daniel/).

[25] Serrano José María, Knowledge Fusion in Relational Databases: Aggregation and Summarization Measures, 2003.

[26] Srikant R. and Agrawal. R. Mining quantitative association rules in large relational tables. In Proc. 1996 ACM SIGMOD Int´l. Conf. Management Data, Pages 1-12, 1996.

[27] Stolfo, Salvatore; Fan, David W.; Lee, Wenke; Prodromidis, y Andreas L. "Credit Card Detection Using Meta-Learning Issues and Initial Results".

[28] Trillas, Enric, et al, "Fundamentos e Introducción a la Ingeniería FUZZY, Omrom", 1994.

[29] Vila, Amparo: "Modelos de datos avanzados: Bases de Datos orientadas a objetos", 2002.

[30] Zadeh L.: Fuzzy Sets. Information and Control. 8, 338-353., 1965

COMPUTACIÓN FLEXIBLE APLICADA AL WEB MINING

José M. Benítez
Dpto. Ciencias de Computación
Universidad de Granada
jmbs@decsai.ugr

Juan L. Castro
Dpto. Ciencias de Computación
Universidad de Granada
castro@decsai.ugr.es

Ricardo Valenzuela G.
Universidad Tecnológica Metropolitana
Santiago-Chile.
rivagat@vtr.net

Resumen

La unión de la ***Lógica Difusa (Fuzzy Logic), Redes Neuronales Artificiales, Algoritmos Genéticos, Conjuntos de Aproximación (Rough Set) y los Sistemas Neurodifusos***, forman la denominada **Computación Flexible**. Estas herramientas nos permitirán establecer un método para la obtención de conocimiento a partir de la información generada por programas que registran la actividad y los eventos ocurridos en la navegación de un usuario sobre un sitio web; información que es almacenada en archivos de registro de eventos conocidos como "archivos de logs". Dado que los datos a utilizar son del tipo numérico, se emplearan los conjuntos difusos para establecer variables lingüísticas que permitan crear un índice de comportamiento, por medio del cual se puedan obtener aquellas páginas más relevantes empleadas por los usuarios en su navegación a través del sitio. Se utilizara una red neuronal artificial con la finalidad de establecer un motor de procesamiento de patrones automatizado, motor que ordene la información de interés eliminando aquellas referencias sin importancia

Palabras Clave: Lógica Difusa, Redes Neuronales Artificiales, Web Mining, Archivos Logs.

1 INTRODUCCION

Es evidente que la eficiente recuperación de patrones y la obtención de conocimiento desde el Web se ha convertido en un tema de vital importancia para todos aquellos que emplean este dominio virtual. La información almacenada en distintas computadoras que están físicamente distribuidas en Internet, puede ser asociada conceptualmente con una gran **base de datos distribuida que contiene información no estructurada**. Esta base de datos conceptual o ***B_Internet***, contiene distintos tipos de información la cual puede ser clasificada como: ***distribuida, sin etiquetas, heterogénea, medianamente mezclada, semi-estructurada, cambiante en el tiempo y de alta dimensionalidad***; información disponible para un usuario en la forma de una ventana virtual a la cual se accede por algún programa de búsqueda y un medio físico o sistema de comunicación. Conceptualmente el análisis de los datos contenidos en ***B_Internet*** puede ser referido al desarrollo de técnicas basadas en sistemas inteligentes, que permitan determinar las necesidades potenciales de un cliente, su comportamiento, como también el descubrimiento de conocimiento desde el web, empleando como fuente de origen los datos proporcionados por los programas o procesos que registran los eventos ocurridos en la navegación por Internet, eventos que son almacenados en archivos de registro de eventos o "web logs"

En este trabajo se estudiara un método para obtener tendencias de comportamiento de usuarios de un sitio web o ***Web Mining de Usuario***, a partir de la información que es posible de generar a partir del empleo del sitio por parte de un usuario, información disponible en los archivos de registro de sucesos del sitio o web logs.

2 BASE TEORICA:WEB MINING

Según Sankar, Varun, Mitra [7] "El Web Mining puede ser definido en grandes rasgos como el descubrimiento y análisis de información útil desde el denominado Word Wide Web". Las principales tareas o procesos del web mining son descritas por Etzioni [8] de acuerdo a la Figura 1. La información o "Web Data" a analizar proviene de Internet o el "web", y sus fuentes de origen pueden ser clasificadas desde la perspectiva de los

servidores web, desde el lado de los clientes, o desde los servidores proxy utilizados para mejorar el rendimiento, desde los equipos de comunicación de datos u obtenidas desde bases de datos disponibles en el web que contienen información de negocios o datos consolidados de una empresa; es evidente por tanto que dependiendo de la fuente, la información a recolectar cambia

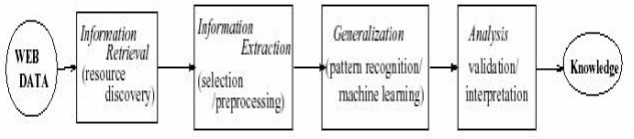


Figura 1 Principales Tareas el Web Mining

La principal fuente de información para nuestro estudio, corresponden a conjuntos de datos provenientes de los procesos que capturan y registran los eventos ocurridos sobre un Servidor Web, eventos almacenados en archivos denominados "web logs". Estos datos son recopilados en distintos formatos y no son del todo confiables entre otras razones debido a los varios niveles de captura de información; por ejemplo, las páginas visitadas y almacenadas en "cache" no son registradas. Estos conjuntos de datos provenientes del cliente, pueden ser recopilados adicionalmente con el empleo de un agente remoto, modificando el código fuente del motor de búsqueda, por medio de técnicas colaborativas voluntarias basadas en "incentivos a los usuarios del browser" (gane ya..!!!.); técnicas que en su conjunto emplean applets java o códigos especializados y están orientadas a una mejor identificación del usuario permitiendo realizar un seguimiento de uso del sitio web, una desventaja de estos métodos es que consumen recursos en el equipo del usuario afectando el rendimiento del mismo. Un browser modificado y aceptado voluntariamente por el usuario tiene una mayor eficacia dado que aumenta la versatilidad de registrar distintos tipos de información del uso del sitio web, teniendo la desventaja de que debe de ser aceptado por el usuario.

Esta variedad de tipos de datos disponibles en el web permiten clasificar su explotación o Web Mining de acuerdo a lo planteado por Srivastava, Cooley, Deshpande y Tan [8]; en tres tipos: ***explotación minera del contenido, explotación minera del uso y explotación minera de la estructura.***

La explotación minera del contenido, apunta a los datos reales que contienen las páginas que fueron diseñadas para satisfacer a los usuarios. Estos datos consisten generalmente en texto y gráficos., pero no se limitan tan solo a este tipo de información.

En ***la explotación minera de la estructura***; la información a emplear es aquella que describe la organización del contenido de la pagina web. La información de la estructura interna de una página incluye por ejemplo el arreglo de las varias etiquetas HTML o de XML dentro de esta. Esto se puede representar como una estructura de árbol en donde una etiqueta (HTML) se convierte en la raíz de este. La clase principal de información en la estructura interna de una página son los "hyper-links" que conectan una página con otra.

La explotación minera del uso esta orientada a descubrir el uso de las páginas web, direcciones IP, referencias a páginas, la fecha y la hora de accesos, datos que proporcionan la información demográfica sobre los usuarios del Sitio Web. Esto incluye a información de datos que permita determinar comportamiento de clientes o información de perfil del cliente.

Sankar, Talwar, Mitra [7] Lee, Kim, Cheng y Kwon [4] definen a la explotación minera de datos en el web en tres categorías generales a partir de la definición anterior denominadas ***Web Content Mining (WCM), Web Structure Mining (WSM) y Web Usage Mining (WUM)***, con una topología similar a la de la Figura 2



Figura 2 Tipos de Web Mining

Web Content Mining (WCM) asocia el concepto de descubrir información útil desde el contenido de los documentos en el web, el contenido de un documento no es solo texto, como es fácil intuir para cualquier persona que utilice Internet; en la red global se encuentran documentos cuyos contenidos son del más variado tipo, audio, video, símbolos, datos, meta-datos, link, hiperlink, textos y otros. Si nos detenemos en el análisis de datos cuyo formato es texto, encontraremos tres tipos: ***datos no estructurados en textos libres, datos semi-estructurados como el código HTML y estructurados como las bases de datos y tablas.***

Web Structure Mining (WSM) en cambio explora la estructura de los hiperenlaces contenidos en una página web, estructuras similares a la indicada en el ejemplo de la ***Figura 3***, ejemplo que se refiere al URL "wharton.upenn.edu", esta estructura posee siete niveles de anidamiento, siendo posible determinar por ejemplo; el documento de mayor uso por parte de los usuarios asignándolo a un conjunto difuso, con la finalidad de establecer reglas de inferencia para un sistema inteligente de adecuación de la estructura de búsqueda del sitio web.

El denominado Web Usage Mining (WUM), extrae información de datos secundarios generados por los usuarios en su interacción por un sitio web (navegación).

WUM incluye datos provenientes de accesos a servidores web registrados en archivos del tipo logs, logs de servidores proxy, logs de motores de búsqueda, perfiles de usuario, archivos de enrolamiento o registro, sesiones de usuarios y transacciones, consultas de usuarios, carpetas marcadas, clic del ratón, desplazamientos (scroll) y otros tipos de información.

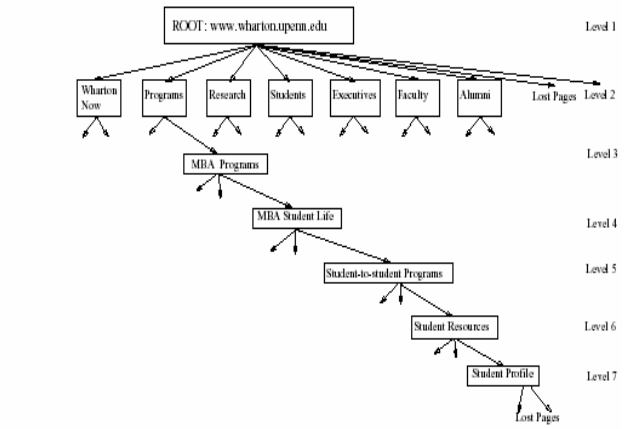


Figura 3 Estructura de una página Web

3 APLICACIONES RELACIONADAS CON WEB MINING

Se describen a continuación algunos proyectos relativos a la Minería de Uso Web o Web Usage Mining que emplean técnicas tradicionales a fin de establecer un parámetro de comparación con respecto a las ventajas de modelar con técnicas de la computación flexible. En el proyecto "WebSIFT" [3], Chen introduce el concepto ***"medida máxima de referencias en la entrega, que caracterizan el recorrido de patrones en un episodio"***, esta medida de las referencias en el envío de paquetes, se refiere a las secuencias de páginas requeridas por un usuario en una sesión en un servidor web. El proyecto denominado "The SpeedTracer" de IBM [10], está construido para trabajar con el índice creado por Chen, "SpeedTracer" emplea estas referencias y adiciona un agente de pre-procesamiento para identificar usuarios y sesiones en el servidor, en ausencia de información del lado del cliente.

Agrawal y Srikant en "Mining Sequential Patters" (IBM A. Research Center) [1], proponen dos algoritmos, para minar patrones secuénciales desde una base de datos de transacciones, base que contiene información como la identificación del cliente, el matasellos de la transacción o base de tiempo, y los productos de la compra realizada por el cliente. A partir de esta información Agrawal y Srikant, proponen un método algorítmico para resolver el problema de la "explotación minera" de patrones, el cual se refiere a determinar el comportamiento de un cliente para una posterior toma de decisión. El comportamiento puede ser entendido como una tendencia de compra; por ejemplo aquellos clientes "hombres" que compran "pañales" también compran "cerveza" en una sola transacción. La solución esta planteada por medio de dos algoritmos que denominan AprioriSome y AprioriAll. Estos algoritmos consideran de manera general las fases o sub-procesos siguientes: ***(1) Fase De ordenamiento:*** *La base de datos es ordenada.* ***(2) Fase de fijación de limites para conjuntos:*** *Se determinan grandes conjuntos de productos y se fijan sus limites.* ***(3) Fase de transformación:*** *Se determina cuales de los conjuntos de secuencias, están contenidos en una secuencia de cliente.* ***(4) Fase de secuencias:*** *Se encuentran las secuencias deseadas; se repite proceso. En cada pasada se analizan los conjuntos determinados.* ***(5) Fase de determinación de Máximos.***

Este algoritmo en sus pasadas, solo cuenta secuencias de un cierto largo, adicionalmente se emplea una función **next**; función que cumple el trabajo de tomar como parámetro el largo de las secuencias que contó en la última pasada, para retornar este largo en la próxima pasada. Esta función debe de ser aplicada en conjunto con una *"heurística que es el porcentaje de los candidatos contados en la pasada actual que tiene incrementos mínimos .."*

Es posible concluir de esta solución que el tiempo de procesamiento es critico dado que el solo hecho de determinar aquellos ítems o grandes conjuntos y relacionarlos con el ***"id"*** del cliente, consume una gran cantidad de recursos como cpu y memoria, recurso que es proporcional al tamaño de los registros a analizar contenidos en la base de datos de transacciones del ***"servidor de retail".***

4 COMPUTACION FLEXIBLE APLICADA AL WEB MINING

Plantearemos un caso de estudio similar en características a los trabajos de Data Mining referidos como [1][2][9], siendo nuestro interés establecer una comparación en las soluciones crisp y las soluciones que emplean técnicas asociadas a la computación flexible para el problema de: "obtener conocimiento", "obtener patrones lingüísticos" u "obtener secuencias de patrones". La lógica difusa reduce la complejidad del problema de detección de patrones de manera significativa al emplear variables lingüísticas cuantificadoras como ***"al menos", "la mayoría", "cerca de la mitad"***, variables que permiten establecer una arquitectura para analizar información y determinar por ejemplo grados de relevancia de los patrones "minados" o extraídos.

Un trabajo que emplea lógica difusa para la detección de patrones lingüísticos, corresponde al desarrollado por Hong, Lin y Wang [5], en donde se emplean conjuntos

difusos para detectar patrones lingüísticos en un proceso de Data Mining orientado a archivos web logs. Esta solución emplea los denominados archivos logs, depurándolos previamente para su posterior análisis, reduciendo el tamaño del conjunto de datos a analizar a solo aquellos ítems de interés, reflejados en este caso en una selección de aquellos eventos que contengan archivos del tipo .asp, .jva, .htm, .html. De estos ítems se obtiene la información como la fecha, tiempo, archivo empleado, dirección ip del cliente, y por medio de la utilización de conjuntos difusos se obtiene una solución que comparativamente reduce la complejidad de las soluciones algorítmicas, como las planteadas en [1][2][9]. La solución planteada por Hong, Lin y Wang tiene como base definir tres conjuntos difusos que expresen la duración de la navegación en un sitio web por parte de un usuario, y asociar este tiempo al archivo empleado y cliente. Estos conjuntos referidos a la navegación en el sitio se definen como se indica en la Figura 6:

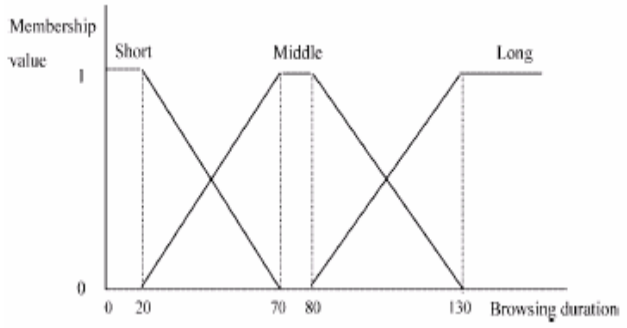


Figura 6 Conjuntos difusos usados por Hong, L y W

Hong, Lin y Wang emplean el concepto de máxima cardinalidad aplicados a los conjuntos difusos definidos para obtener una secuencia de patrones lingüísticos empleados por un usuario (asp, htm u otra). Si bien este planteamiento reduce significativamente la complejidad, plantea nuevos problemas al considerar como fuente de los datos a analizar los referidos "archivos logs crudos", archivos que contienen variada información y son de difícil depuración o preparación para un proceso de minería de datos.

5 UN EJEMPLO DE COMPUTACIÓN FLEXIBLE APLICADO AL WEB USAGE MINING

Un problema no considerado en la solución de Hong, Lin y Wang para minar patrones relacionados con páginas web y construir secuencias de empleo de las mismas o comportamiento de un usuario, corresponde a que la mayoría de los sitios disponibles en Internet son generados dinámicamente; esta técnica que reporta grandes beneficios al momento de explotar la "usabilidad" de un sitio web y mostrar datos en tiempo real extraídos desde diversas fuentes; provoca desventajas al momento de analizar los datos contenidos en los "archivos logs", dado que estos contienen información de eventos que están cambiando en el transcurso del tiempo. El planteamiento anterior establece la necesidad de disponer de una arquitectura que permita el análisis en línea o simultáneo de los eventos que se generan en la navegación de un sitio web, eventos que son capturados y almacenados por un programa activo en la memoria del servidor (proceso).

Una arquitectura posible se indica en la Figura 8, en donde se emplea una red neuronal y un sistema difuso, operando de manera colaborativa, con la finalidad de construir un motor de procesamiento automático que opere en línea. La finalidad de este sistema es ordenar la información de interés, eliminando aquellas referencias sin importancia. Clasificar aquellos datos filtrados previamente dando un formato de mejor calidad a la información recopilada. Sincronizar la captura de la información desde distintas fuentes o procesos, a fin de reducir el tiempo de limpieza de los datos capturados.

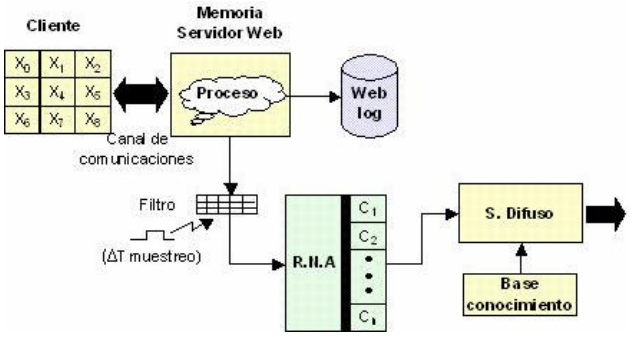


Figura 8 Sistema para Análisis Simultáneo en Línea

De la ***Figura 8***, se puede observar que la entrada de datos proviene de un proceso que registra la actividad del sitio web. Este proceso genera datos del tipo numérico y almacena la información recopilada en archivos de registros de eventos, también conocidos como "archivos logs", información disponible para su análisis estático [5].

En nuestro caso se modificara este proceso con la finalidad de conectar la captura de los eventos a una arquitectura similar a la indicada en la figura 8, sistema que permite un análisis dinámico de los datos de manera similar a la disposición de páginas en el lado del cliente. La arquitectura planteada, tiene por misión básica determinar secuencias de empleo de los elementos dispuestos en el sitio web dinámico, elementos representados en este caso en el lado del cliente como zonas $X_{0-->N}$ de la ventana virtual. El objetivo de este sistema es realizar un seguimiento del usuario durante su interacción con el sitio web e inferir las secuencias de patrones empleados en su interacción con el sitio, guardando las zonas empleadas por el usuario de forma de aplicar sobre cada una de estas heurísticas que determinen su comportamiento o preferencias

5.1. DETERMINACION DE LAS SECUENCIAS DE PATRONES

En general un sitio web dinámico es construido a partir de lenguajes como JSP/ASP, JAVA, XML, HTML y otros, en conjunto con herramientas de software especializadas; la finalidad de esta técnica es disponer (agregar) en una ventana virtual objetos en tiempo real factibles de ser desplegados o activados por los usuarios del sitio. Un ejemplo de una ventana virtual que contiene elementos C_i agregados dinámicamente a ciertas zonas X_J delimitadas de la ventana se indica en la Figura 9; estos elementos u objetos C_i pueden ser agrupados en clases de archivos, imágenes, link u otros objetos factibles de ser contenidos en el diseño del sitio y ser agregados a zonas previamente mapeadas con algún diseño preestablecido.

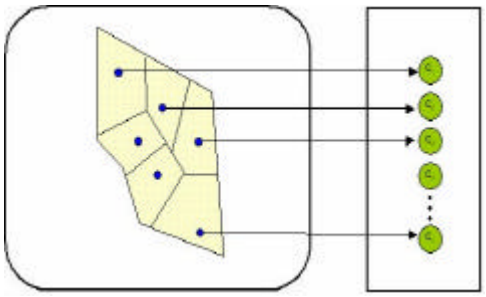


Figura 9 Zonas o Regiones de Ventana Virtual

Se puede observar de la figura, que cada región en la cual a sido subdividida la ventana virtual del sitio, puede ser asociada con una clase de ítems agregados de forma dinámica. Un usuario al recorrer el sitio dejara un rastro de navegación similar al de la **Figura 10**, rastro que se establece por medio del monitoreo de la navegación del usuario.

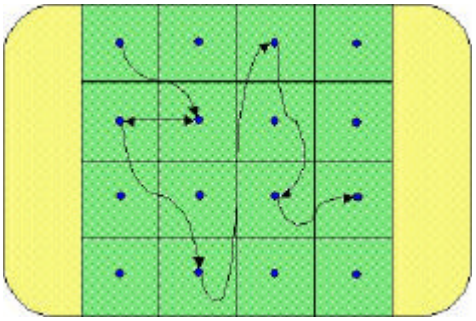


Figura 10 Objetos de una Sesión de Usuario

El monitoreo referido, puede ser realizado desde el lado del cliente por medio de un proceso que capture los movimientos del puntero de navegación (mouse), del teclado u otros dispositivos; o por programas especiales que revisen u "olfateen" (sniffers programs) los paquetes usados en el servidor; estos programas o sniffers capturan y analizan el tráfico de la red; examinando los paquetes TCP/IP hacia el servidor y generando distintos reportes.

El comportamiento del usuario queda determinado por tanto por aquellas secuencias de patrones SP = {CX_0, CX_A,....., CX_z}, construidas a partir de los elementos requeridos por el usuario sobre la ventana virtual en una sesión de navegación. El problema puede ser entendido por tanto como la necesidad de determinar un mapa o topología de usabilidad del sitio en referencia, este mapa puede ser precisado por medio de una red neuronal Kohonen.

La arquitectura planteada considera dos etapas que son el núcleo del sistema: ***un motor de procesamiento, constituido por una red neuronal autoorganizada*** y ***un motor de inferencia conformado por un sistema difuso***. La misión de la red neuronal es actuar como un clasificador de los datos entrantes provenientes del proceso que monitorea el trafico en el servidor web; los datos de entrada a esta red son vectores ***(x, y)*** asociados a la zona activada por el cliente en la ventana virtual y la salida corresponde a una zona, vecindad o clase previamente establecidas en la etapa de aprendizaje de la red neuronal. En la etapa de aprendizaje la red neuronal artificial se entrena con aquellos espacios vectoriales que corresponden a los objetos a estudiar, generando un mapa similar al de la figura 11; se puede entender por tanto que al recibir un patrón de entrada en la etapa de funcionamiento, se asocia este patrón a una clase o primitivas que cumplan con pertenecer a ciertos grupos posibles de archivos u objetos agregados dinámicamente a zonas mapeadas del sitio.

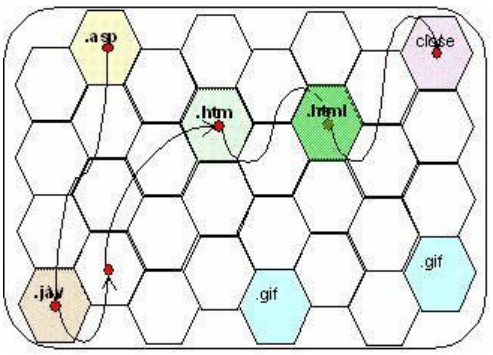


Figura 11 Mapa Estandarizado de una Ventana Virtual

La etapa constituida por el sistema difuso permite establecer reglas de asociación entre los patrones seleccionados sean estos archivos, link, objetos, etc.; dependiendo de la clase de conocimiento que se desea obtener. Es posible por ejemplo crear conjuntos difusos y reglas de inferencia que permitan determinar por ejemplo las zonas de la ventana virtual con mayor preferencia de los usuarios, con la finalidad de disponer en aquellas zonas los objetos.

La arquitectura planteada opera de acuerdo a las siguientes etapas:

A. Entrenamiento del motor de procesamiento (fuera de línea o "stand alone")

   1. Suministro de conjuntos de patrones de entrenamiento.

2. Determinación del mapa o topología,
3. Aprendizaje de red, ajustes y adaptación al mapa o topología requerida.

B. Estudio de los eventos suministrados por el proceso de captura de información (en línea o procesamiento simultáneo)

1. Captura de datos en tiempo real y filtrado de los mismos.
2. Procesamientos de datos filtrados por Red Neuronal Artificial Autoorganizada (Kohonen)
3. Se obtienen secuencias de patrones empleados por un cliente en una sesión.

C. Descubrimiento de Conocimiento: utilización del motor de inferencia.

1. Análisis de las secuencias de patrones.
2. Agregación a vistas: Índices de zonas, objetos de mayor utilización, otros índices.

6 CONCLUSIONES

Un problema del análisis estático de archivos logs y en términos generales de cualquier archivo de reportes de eventos, corresponde a la alta dimensionalidad de los sucesos que forman parte de los mismos; el estudio de estos archivos con la finalidad de obtener conocimiento con técnicas algorítmicas tradicionales es una tarea compleja, de alto costo computacional y no ajena a conclusiones sesgadas, ya sea por fallas en el almacenamiento de los eventos o por algoritmos que impiden abordar de manera global los datos almacenados. La aplicación de la lógica difusa al análisis de estos archivos con la finalidad de obtener conocimiento, simplifica de manera significativa las tareas a realizar sobre el conjunto de datos, por medio de la utilización de conjuntos difusos y variables lingüísticas como MEDIO, ALTO, BAJO. Un problema que no resuelve esta técnica estática de análisis de archivos de registro de eventos, corresponde a las crecientes necesidades de disponer de herramientas de análisis en tiempo real con la finalidad por ejemplo, de mejorar el rendimiento de un servidor web, predecir carga, adaptar el sitio de acuerdo a las preferencias de los usuarios o detectar ataques a la disponibilidad del sitio web.

Desde este punto de vista la utilización de herramientas como las redes neuronales artificiales, junto a la lógica difusa en un sistema integrado tiene por objetivo dotar de inteligencia y adaptabilidad a las urgentes necesidades de análisis en tiempo real de los sucesos y eventos relacionados con la navegación sobre un sitio web, permitiendo simplificar las tareas de monitorización dinámica del sitio web de; como también mejorar el ajuste o refinación de las funciones de pertenencia de los conjuntos difusos y las reglas de inferencia asociadas..

Un sistema integrado por una red neuronal y un sistema difuso operando en línea, permite reducir la dimensionalidad del problema, adecuar la solución a distintas necesidades como por ejemplo estudiar el comportamiento de una parte del portal o sitio web, detectar ataques a la seguridad del sitio, predecir carga.

La redes neuronales autoorganizadas constituyen un pilar básico en el análisis de información en línea o tiempo real, un caso especial de estas redes corresponde a la arquitectura de red de Kohonen o en palabras simples Red de Kohonen, la cual permite construir mapas o topologías, mapas que pueden ser aplicados en soluciones como las expuestas en este trabajo.

Agradecimientos

Al cuerpo de profesores del programa de Doctorado en Informática de la Universidad de Granada España.

Referencias

[1] Agrawal R., Srikat R.; “Mining Sequential Patterns”. IBM Almaden Research Center.

[2] Bayardo R.; Gunopulus D.; Agrawal R. “Constraint Based Rule Mining in Large, Dense Databases. IBM Almaden Research Center.

[3] Chen M.S., Jan J., Yu P.S. “Data Mining for path traversal patterns in a web environment”.

[4] Cooley R, Tan P-N, Srivastava J; “Discovery of interesting usage patterns from web data”.

[5] Hong T.-P.,Lin K.-Y., Wang S.-L.; “Mining linguistic browsing patterns in the world wide web”. Springer-Verlag 2002 . Soft Computing 6 (329-336)

[6] Piramuthu S., “On learning to predict Web traffic”. Elsevier Decision Support System 35 (2003), 213-229

[7] Sankar K. Pal, Varun Talwar, Pabitra M.; “Web Mining in Soft Computing Framework”

[8] Srivastava J, Cooley R., Deshpande M., Tan P-N., Web Usage Mining

[9] Swami A.; Imielinsh T.; Agrawal R. “Mining Association Rules between Sets of Items in large Database. IBM Almaden Research Center.

[10] Wu K-L, Yu P. S., Ballman A. “SpeedTracer: A web usage mining and analysis tool”. IBM Systems Journal, 37(1), 1998..

Metodología de tratamiento inteligente del correo electrónico mediante técnicas de *soft-computing*.

Francisco P. Romero
Centro I+D Soluziona – UCLM
Ronda de Toledo, s/n
13071-Ciudad Real, SPAIN
fpromero@soluziona.com

José A. Olivas
Departamento de Informática,
Universidad de Castilla-La Mancha,
Paseo de la Universidad 4,
13071-Ciudad Real, SPAIN
JoseAngel.Olivas@uclm.es

Pablo J. Garcés
Departamento de CC. de la
Computación e IA.,
Universidad de Alicante
Ctra. S. Vicente del Raspeig s/n
03080 – Alicante, SPAIN
pgarces@dccia.ua.es

Resumen

Este trabajo presenta una metodología de clasificación automática de grandes volúmenes de mensajes de correo electrónico con el fin de posibilitar un aprovechamiento óptimo de la información y/o conocimiento existente en los mensajes. Para alcanzar este objetivo nos basamos en tres pilares fundamentales: la representación conceptual de los mensajes, el *clustering* jerárquico - borroso y la representación del conocimiento obtenido mediante Prototipos Deformables Borrosos. La representación conceptual de los mensajes, basada en las relaciones borrosas de sinonimia y generalidad permite analizar de forma más completa el texto del mensaje y su semántica. Mediante la utilización de un algoritmo de *clustering* borroso modificado se obtienen la agrupación jerárquica de los mensajes en categorías conceptuales borrosas. Con el fin de facilitar el tratamiento de los mensajes entrantes y una sencilla actualización del modelo obtenido, la representación de los elementos "carpetas" se ha realizado mediante Prototipos Deformables Borrosos. Se ha comprobado la bondad del método mediante estudios experimentales utilizando estás técnicas sobre listas de distribución de diferentes temáticas e idiomas.

Palabras Clave: Correo Electrónico, *Clustering* Borroso, Prototipos Deformables Borrosos.

1 INTRODUCCIÓN

El correo electrónico es posiblemente la herramienta de mayor provecho individual que ha aportado Internet. La WordTalk Corporation [3] estima que 60 millones de profesionales utilizan el correo electrónico El uso del correo electrónico se ha convertido en algo imprescindible, tanto en el ámbito empresarial como en el doméstico. Mediante un e-mail, cualquier usuario puede comunicar sus ideas con la claridad de una carta y la rapidez de una llamada telefónica.

La mayoría de programas de gestión de correo electrónico guardan los mensajes recibidos en el buzón de entradas. Dependiendo del uso que se haga del mismo habrá mensajes de diferentes procedencias y temas. Con el tiempo, el número de mensajes tenderá a crecer con lo que resultará difícil localizar un mensaje concreto. Para evitar este fenómeno se incluye la posibilidad de crear diferentes carpetas donde almacenar los mensajes recibidos. Los criterios de organización dependen de los gustos personales del usuario. De esta manera se puede establecer una clasificación por áreas, temas o fechas, etc. Como resultado un usuario posee definidas entre 10 y 100 carpetas organizadas en una jerarquía. El trabajo de clasificar el mensaje en su carpeta correspondiente supone un esfuerzo y una pérdida de tiempo considerable para el usuario. Algunos programas comerciales permiten la definición de reglas simples para ayudar a esta tarea, pero estas reglas cubren únicamente un porcentaje muy pequeño del espectro de mensajes que se reciben, situando mensajes en carpetas incorrectas.

En este trabajo se propone una metodología basada en diversas técnicas de *soft-computing* [7] con el propósito de gestionar de forma eficiente grandes volúmenes de correo electrónico. Los objetivos principales a perseguir son la obtención de una organización jerárquica y borrosa basada en los conceptos tratados en los mensajes, la clasificación automática de los correos entrantes de forma eficaz y facilitar las diferentes búsquedas que se realicen sobre el buzón de correo electrónico.

Muchos de los estudios que relacionados con éste se realizan entre la comunidad investigadora provienen o bien del análisis clásico de documentos o bien del estudio de páginas Web. La mayoría de estos están basados en clasificadores Naive-Bayes aplicados a la categorización documental como propone Turéne [14]. También es necesario mencionar los ya clásicos de procesamiento de

correo electrónico [13][6], o de aplicación del aprendizaje supervisado en este problema [4].

Para comprobar la bondad de las técnicas y realizar los ajustes precisos se ha utilizado material experimental de diversa naturaleza. De esta manera, se han realizado estudios sobre listas de distribución referidas tanto a dominios tecnológicos específicos como a temas históricos o generales. Con esta selección se pretende mostrar las diferentes problemáticas presentes en la gestión del correo electrónico y la aplicabilidad general del método propuesto.

El artículo está organizado en dos apartados principales. En el primero se explica con detalle la metodología de trabajo que se propone en este artículo centrándose en los aspectos principales de cada una de las fases que comprenden la construcción de la organización jerárquica a partir del estado inicial del buzón de correo. También se incluye en este apartado el proceso de tratamiento de los mensajes de correo electrónico entrantes al sistema. En la segunda parte se muestra, también fase a fase, todos los experimentos realizados para comprobar la validez del método así como sus resultados extrayendo las conclusiones más relevantes. Por último se incluyen las conclusiones del trabajo realizado y las líneas de investigación futuras que es posible abordar.

2 METODOLOGÍA DE TRATAMIENTO DE CORREO ELECTRÓNICO

Normalmente el volumen de mensajes de partida de un buzón "plano" de correo electrónico normal (trabajaremos con 1500 mensajes aproximadamente según los casos) es alto. Esto provoca que la organización a construir debe de tener la mayor calidad posible para que permita un aprovechamiento posterior óptimo por parte del usuario. Las características del resultado a obtener serán las siguientes:

1. Organización jerárquica de los mensajes en carpetas siguiendo criterios basados tanto en el contenido "conceptual" del mensaje como en los campos estructurados de éste. La representación conceptual de los mensajes permitirá la posterior búsqueda basada en conceptos.
2. Definición de las carpetas mediante términos/conceptos relevantes y Prototipos Deformables Borrosos.
3. Permitir que un mensaje pueda almacenarse según su contenido en más de un grupo de entre los que forman la jerarquía.
4. Cada mensaje debe de tener un grado de afinidad (o pertenencia) a cada grupo en el que esté clasificado. Los mensajes se ordenarán según este grado de afinidad.
5. Sencillo mecanismo de comparación entre documentos y la definición de cada una de las carpetas. Los resultados de este "*matching*" se utilizarán para almacenar el mensaje en una o varias carpetas.

2.1. OBTENCION DE LA ORGANIZACIÓN INICIAL

El proceso de construcción se basa en las siguientes etapas: Preproceso lingüístico, Representación Conceptual, *Clustering* jerárquico - borroso, Representación del Conocimiento y obtención de los resultados finales. En los siguientes subapartados se explican con detalle.

2.1.1. Preproceso Lingüístico

Dentro de cualquier metodología de análisis de documentos es ineludible el realizar inicialmente el preproceso lingüístico del material con el que se va a trabajar. Se persigue con ello la eliminación de elementos no relevantes dentro de los textos partida, con lo cual se focaliza mejor el problema de forma que se mejora la eficiencia de los procesos posteriores. El correo electrónico presenta una cierta estructura de campos que tras una sencilla transformación previa es posible aprovechar con el fin de apoyar con más información a las etapas de análisis posteriores. En este caso la transformación previa de los mensajes se ha realizado en tres pasos principales:

a) Separación de los campos estructurados (asunto, origen, fechas) del contenido propio del mensaje.
b) Agrupar los mensajes en "conversaciones". Una conversación es la agrupación de aquellos mensajes del mismo asunto, distribuidos uniformemente en el tiempo y con ciertos aspectos comunes en origen/destino con lo que pueden ser considerados parte de una misma unidad documental.
c) Interpretación de la estructura existente: Procesamiento de las reglas y carpetas existentes para que pueda ser aprovechada en la generación de un resultado final de mayor adecuación a las necesidades del usuario.

Una vez realizada la transformación previa se procesan los contenidos de los mensajes mediante técnicas clásicas de procesamiento del lenguaje natural [2]. Los objetivos de este proceso son: primero eliminar toda la información implícitamente sobrante del mensaje (*stop words* y *stop zones*) y segundo reducir el vocabulario de términos a analizar mediante técnicas de extracción de raíces léxicas (*stemming*).

- *Stop zones*: Se han denominado "*stop zones*" a aquellas secciones de un correo electrónico que representa publicidad, identificación completa del origen u otra información agregada que no aporta nada al contenido del mensaje. Para poder detectar y

eliminar estas partes del contenido del mensaje se ha construido un diccionario de patrones en el que se han almacenado secciones en HTML utilizadas para este fin.

- *Stop Words*: La eliminación de *stop words* es una tarea clásica dentro del preproceso lingüístico de cualquier texto.
- *Stemming*: Para la extracción de raíces semánticas y otros análisis léxicos se ha utilizado una adaptación de algunos algoritmos descritos en [2] adaptándolos a las necesidades específicas que se presentan en los idiomas de trabajo.

Una vez que ya se han realizado todas las fases del preproceso puramente lingüísticas y se poseen términos dentro de mensajes es necesario realizar un recorte más para reducir el espectro de términos de estudio a los verdaderamente relevantes. Para ello se ha realizado un ranking de la frecuencia de los términos en todos los mensajes objeto de estudio y se han suprimido para posteriores fases todas aquellas que o bien no aparecían en el 3% de los mensajes o bien aparecían en una cantidad mayor que el 33% de los mensajes a estudiar.

2.1.1. Representación conceptual

Para poder trabajar con los mensajes de forma abstracta es necesario obtener una representación lógica de éstos. Con este propósito se utiliza generalmente el modelo espacio-vectorial o extensiones de éste [10]. FIS-CRM [12] puede considerarse como una extensión de este modelo clásico con el objetivo de representar, dentro del vector que corresponde al documento, los conceptos inherentes a las palabras que aparecen utilizando para ellos las interrelaciones borrosas de sinonimia y generalidad. De esta forma se puede decir que FIS-CRM está basado en dos puntos principales:

a) Si una palabra aparece en un documento, los términos que sean sinónimos suyos también aparecen de alguna forma ya que representan al mismo concepto.
b) Si una palabra aparece en un documento, también deben de reflejarse los términos que la representan de forma más general el concepto al que se refiere.

La base fundamental de FIS-CRM es repartir la semántica de las ocurrencias de una palabra entre los sinónimos borrosos que representan al mismo concepto, y proveer de un peso borroso a las palabras que representan de forma más general el concepto contenido en la palabra que realmente aparece en el documento. Para lograr esto los documentos, serán representados en primer lugar mediante los vectores base cuyos coeficientes se basan en las ocurrencias de las palabras contenidas. Posteriormente, se procede a realizar un proceso de reajuste de pesos para obtener un vector basado en ocurrencias conceptuales. De esta manera, una palabra puede tener un peso en el nuevo vector no están incluida explícitamente en el documento al que representa, si de alguna forma (mediante sinonimia o generalidad) referencia a los conceptos tratados en el mensaje. Para llevar a cabo el proceso de reajuste, se han tenido en cuenta las relaciones borrosas de sinonimia y generalidad. Los grados de sinonimia borrosa para cualquier par de palabras se ha obtenido mediante las expresiones disponibles en el diccionario borroso de sinónimos presentado en [1] para el idioma castellano. Para la obtención de las ontologías temáticas que permitan la utilización de la relación borrosa de generalidad se ha utilizado como base el algoritmo propuesto en [16] que establece una medida borrosa del grado de generalidad entre cada par de palabras que aparecen en la colección de mensajes. En resumen el proceso de representación conceptual de los mensajes de correo electrónico mediante FIS-CRM consiste en los siguientes pasos:

0. Indexación de todos los términos resultantes del preproceso.
1. Construcción de las matrices de sinonimia y ontología almacenando los grados de sinonimia y generalidad entre cada par de palabras en el índice.
2. Representación de los mensajes utilizando el modelo espacio vectorial clásico.
3. Reajustar el vector de pesos utilizando el conjunto de formulas de FIS-CRM.
    a. Debido a la dificultad que presentan el tratamiento de palabras polisémicas, para realizar el ajuste basándonos en la sinonimia borrosa se han utilizado diferentes fórmulas según nos encontremos con sinonimias directas, con sinonimias dependientes del contexto del documento, o con sinonimias dependientes del contexto general de la colección de los documentos.
    b. El reajuste del vector utilizando la interrelación borrosa de generalidad es lineal y proporcional ante el grado de generalidad del término A frente al término B.
4. Generar la matriz de similaridad que almacenará el grado de similitud entre cada par de mensajes en la colección. La matriz será la entrada al posterior proceso de *clustering*. La función de similitud contempla la co-ocurrencia de las palabras y las frases, el tamaño de cada mensaje, la rareza de las palabras y la co-ocurrencia de los conceptos de forma implícita.
5. Almacenar la información necesaria en forma de meta-datos para permitir el posterior tratamiento de mensajes entrantes.

2.1.2. *Clustering*

Todo proceso de *clustering* persigue agrupar todos los elementos del espectro de estudio, en este caso mensajes, en un número reducido de grupos que estén formados por los más similares en cuanto a un criterio de comparación

previamente establecido. En este caso no se intenta obtener una simple agrupación de mensajes si no que se intenta conseguir una organización arborescente de carpetas de mensajes almacenado en cada carpeta o grupo los elementos que conceptualmente sean más similares según lo explicado en el apartado anterior. También se persigue que la estructura obtenida sea borrosa, es decir, que los diferentes grupos no sean disjuntos sino que los mensajes puedan pertenecer a varios grupos con un grado de afinidad a cada uno de ellos. Para conseguir a partir este resultado una organización jerárquica se realizan repetidamente otros procesos de *clustering* dentro de los grupos más numerosos de entre los obtenidos. Esta evolución es la realizada en el metabuscador FISS basado en [5], demostrándose muy útil cuando el número de elementos (mensajes o *snippets*) y el número de términos sujetos de análisis es reducido; pero cuenta con el inconveniente de tener que establecer previamente un número de grupos de partida y que la organización jerárquica solo se establece en los grupos suficientemente numerosos. Si se quiere evitar tener que tomar decisiones sobre el número de grupo de partida es necesario utilizar algoritmos de *clustering* jerárquico del que se parte de cada mensaje como un grupo individual y se realiza un proceso iterativo hasta llegar a conseguir una estructura arborescente que agrupe a todos los elementos.

En este trabajo se ha optado por una mezcla de este tipo de algoritmos para aprovechar su potencia de forma óptima. De esta forma se ha realizado un primer paso de *clustering* jerárquico para obtener el número de grupos resultarían a partir del corte de un umbral de similitud y a partir de ahí se opera con esos grupos de partida mediante el algoritmo de *Soft-Clustering* modificado utilizado en FISS. Los procesos de borrosificación de resultados y de adaptación de los dos algoritmos se han introducido en pasos muy concretos y unitarios del proceso con lo cual resulta un algoritmo de *clustering* único que presenta una combinación de técnicas para la obtención más eficiente de los resultados.

2.1.3. Representación del Conocimiento y Resultados

La organización jerárquico-borrosa obtenida del proceso de *clustering* no conforma todavía los resultados definitivos del proceso sino que deben de ser refinados y completados para poder obtener una nueva organización de carpetas que cumpla las características requeridas. Este proceso se divide en diferentes tareas:

- Obtener la definición completa de cada carpeta. Mediante el algoritmo utilizado en FISS para los *snippets* se extraerán los términos más relevantes de cada grupo con un grado de relevancia que servirá para pasos posteriores. A su vez, se identificará la carpeta mediante él o los términos con grado de relevancia máxima con un máximo de 5.
- Utilizar el conocimiento extraído de la estructura existente para completar tanto nuestra estructura de carpetas como la definición de cada carpeta. Para ello se detectarán las partes de la estructura antigua asimilables a la nueva así como el aprovechamiento de reglas de acción (*crisp*) más importantes.
- Procesar los resultados para lograr la definición completa de la carpeta utilizando Prototipos Deformables Borrosos [11] lo cual facilitará el análisis posterior de los mensajes entrantes y la actualización de la estructura.

Se definen los *Prototipos Deformables Borrosos*, como una combinación lineal de Prototipos Borrosos [15, 16] con la capacidad de adaptarse a una situación real, cuyos coeficientes son los grados de pertenencia a cada uno de estos Prototipos Borrosos. Cada prototipo viene definido por un número borroso (principalmente triangular) y un marco de definición de la situación de datos de los que no se dispone a priori (en este caso los marcos de acción). Dentro de la representación del conocimiento que define cada carpeta se han construido números borrosos a partir de los términos que definen cada carpeta por lo que las funciones de pertenencia y comparación entre mensajes y carpeta se operarán a partir de la inferencia con los Prototipos Deformables Borrosos. A su vez, también la actualización de la definición del grupo se operará de la misma manera debido a que el mecanismo es fácil y rápido de aplicar. El marco de definición de cada prototipo determinará las acciones especiales a realizar (priorización, eliminación, auto-respuesta, reclasificación, etc.) con cada mensaje entrante según su afinidad a la carpeta. Para la definición del marco se aprovecharán las especificaciones iniciales que existieran en la estructura y las indicaciones a posteriori que efectúe el usuario.

2.2. TRATAMIENTO DE LOS MENSAJES ENTRANTES

La tarea de clasificar de forma automática y correcta cada mensaje entrante es una tarea compleja, por lo tanto es necesario que las operaciones de análisis y clasificación se realicen en un proceso transparente al usuario, el cual, sólo tiene que ser consciente de los resultados obtenidos, sin retardo alguno ni perdida de funcionalidad. El proceso de tratamiento de cada mensaje entrante en el sistema será el siguiente:

1. Preproceso lingüístico del mensaje: Eliminar *stop words*, *stop zones* y realizar *stemming*. Determinar si está dentro de alguna conversación abierta.
2. Construir la representación conceptual del mensaje utilizando las técnicas de FIS-CRM y las matrices calculadas en el proceso anterior.
3. Comparación entre las características del mensaje y las características de cada una de las carpetas. Si el mensaje estuviera dentro de una conversación esta comparación sería con un subconjunto de las carpetas existentes y no con todas ellas. Para calcular la afinidad con cada

carpeta se utilizará la inferencia con Prototipos Deformables Borrosos.

4. Almacenar el mensaje en las carpetas en las que haya conseguido una afinidad positiva, destacando su entrada en la carpeta con mayor afinidad. Realización de las acciones pertinentes si se ha indicado alguna en los marcos de definición de las carpetas.
5. Actualización del modelo, si la cantidad de mensajes entrantes o la situación del buzón (carpetas demasiado grandes) lo requiere.

3 EXPERIMENTACIÓN

Las colecciones de trabajo utilizadas para los experimentos han sido las siguientes: **Servlet-JSP** (Lista de correo sobre las tecnologías Java para Internet); **MS-Java** (Newsgroup de Microsoft sobre JAVA), **Capitán Alatriste** (Lista de correo los libros de Arturo Pérez-Reverte), **Send Box** (Carpeta de mensajes enviados), **Intersystems** (Grupo de News sobre las tecnologías de bases de datos de esta empresa). En este estudio el diccionario de sinónimos expuesto en [1] y puesto en práctica en FISS se ha utilizado para las colecciones de mensajes en español. Para las colecciones de mensajes en inglés el tratamiento de la sinonimia se ha basado en WordNet [8] no habiendo obtenido un resultado de tan alta calidad como el logrado en el tratamiento de los mensajes en español. Para la parte del estudio que recurre a las ontologías para el tratamiento de las listas de distribución cuya temática giraba sobre la tecnología JAVA se han creado unos diccionarios de interrelaciones de generalidad de forma semi-automática utilizando como base el algoritmo de construcción de ontologías propuesto en [15].

a. Preproceso Lingüístico y Representación Conceptual.

Los resultados de esta fase que se pueden observar en la Tabla 1 dan lugar a concluir que se obtiene una gran reducción del material de estudio sin detrimento de la calidad de los resultados que posteriormente se obtendrán ya que de esta forma se posibilita un tratamiento posterior más rápido de los aspectos relevantes de los mensajes.

Tabla 1. Resultados del Preproceso Lingüístico

Colección	Mensajes de Partida	Agrupaciones por conversaciones.	Términos Sin Procesar	Términos Resultantes
Servlet-Jsp	1468	521	3958	246
MS-Java	516	192	4960	2309
Capt. Alatriste	270	81	1545	887
Send Box	1218	881	1890	467
Intersystems	436	176	2376	941

Se ha comprobado mediante la comparación de los resultados de los algoritmos de *clustering* que la representación conceptual aporta un aumento significativo de la similitud entre los documentos ya que tiene en cuenta todos los conceptos tratados en los mensajes y no sólo palabras literales.

b. Clustering

Según se puede observar en la tabla 2 el número de grupos obtenidos y su profundidad media viene dada en la siguiente tabla.

Tabla 2. Resultados del Clustering.

Colección	Grupos Raiz	Grupos totales	No clasificados
Servlet-Jsp	12	35	65
MS-Java	6	11	43
Capt. Alatriste	6	14	12
Send Box	24	51	75
Intersystems	10	18	24

d. Resultados finales.

Las carpetas de mensajes obtenidas resultan ser más manejables y significativas que las carpetas que se puedan construir mediante reglas de usuario. Cada carpeta posee una definición completa de sus características mediante la utilización de Prototipos Deformables Borrosos, lo cual permitirá procesar los mensajes entrantes.

e. Gestión de Mensajes Entrantes.

Para realizar las pruebas de utilidad real de la estructura obtenida se han utilizado el resto de las colecciones de mensajes utilizadas con alrededor de la tercera parte de mensajes que las utilizadas en la realización del proceso.
Para evaluar la bondad del proceso de clasificación automática se ha utilizado como métrica discriminante el Número de clasificaciones correctas (NC). Diferenciando entre numero de clasificaciones correctas al nivel 1 de la jerarquía (NC1), a nivel 2 (NC2) o a nivel de grupos finales (NCH). Para todos ellos se emplea el número de mensajes bien clasificados dividido entre el total de mensajes procesados. Los resultados obtenidos para cada una de las colecciones se pueden observar en la tabla 4.

Tabla 4. Resultados de los mensajes de prueba.

Colección	Mensajes	NC1	NC2	NC3
Servlet-Jsp	387	81,8%	78,8%	62,73%
MS-Java	163	75,4%	66,7%	54,12%
Capt. Alatriste	113	85,6%	79,8%	64,56%
Send Box	320	83,2%	66,7%	63,19%
Intersystems	143	77,4%	57,6%	43.29%

Por otro lado, el rendimiento del proceso de tratamiento del mensaje entrante es aceptable en cuanto se realice en el intervalo de tiempo en el que el mensaje entra al servidor y el usuario .lo descarga. La clasificación obtenida prácticamente no genera cambios en la estructura de carpetas obtenida en el proceso inicial; esto significa que el conjunto de mensajes de partido era relevante y la organización obtenida cercana a la óptima.

4 CONCLUSIONES

En este trabajo se ha presentado un método de trabajo con fin de solucionar los problemas que conlleva la gestión de buzones de correo electrónico con gran cantidad de mensajes. Como resultado del proceso se ha obtenido un buzón de correo estructurado jerárquicamente y que permite el máximo aprovechamiento con un esfuerzo mínimo. Además se obtienen una serie de metadatos que permitirán unos sencillos mecanismos de actualización de la estructura y del tratamiento eficiente de los de mensajes nuevos. Así como con esta nueva organización obtenido se pueden realizar además de las búsquedas clásicas, una búsqueda navegacional más cómoda, búsquedas conceptuales como en FISS, y búsqueda mixtas (buscar carpetas a partir de las cuales comenzar la navegación mediante sus términos más relevantes). Se ha comprobado la validez del método mediante la experimentación con los mensajes procedentes de listas de distribución de correo electrónico de diversas naturalezas e idiomas. La diversidad de los contenidos tratados y la bondad de los resultados nos reafirma en la creencia de que este método, aunque mejorable, puede ser útil para la organización eficiente del correo electrónico tanto personal como profesional de un usuario. Por otro lado, existen diferentes puntos en los que el proceso necesita mejorar, por lo que será necesario un trabajo futuro de ampliación. Alguno de estos puntos son los siguientes: Funciones lingüísticas para el tratamiento de características específicas de los mensajes y corrección ortográfica de términos; identificación automática de idioma de los mensajes; explotación del contexto y otros factores en la representación conceptual; mejorar la representación del conocimiento mediante la asimilación de conceptos de representación avanzados como los Conceptual Fuzzy Sets [9].

En comparación con otros métodos similares, la propuesta realizada en este trabajo aporta una estructura jerárquica más rica al introducir la lógica borrosa y las interrelaciones borrosas en su construcción. En cuanto a rendimiento, la utilización de un algoritmo *clustering* iterativo evolucionado y los procesos de representación conceptual de FIS-CRM[12] hacen que sea prácticamente similar a otros procesos basados en algoritmos inspirados en FCM o k-means.

Agradecimientos

Parcialmente subvencionado por los proyectos PBC-03-004 PREDACOM, JCCM, y TIC2003-08807-C02-02 DIMO-CLUST, MCYT.

Referencias

[1] S. Fernández-Lanza (2001): Una contribución al procesamiento automático de la sinonimia usando Prolog, Tesis Doctoral, Universidad de Santiago de Compostela, España.

[2] Frakes W. B. & Baeza-Yates R. (1992): Information Retrieval: Data Structures & Algorithms. Prentice Hall. Englewood Clifss, N. J.

[3] Harrys, D. & Heather C.: Wordtalk releases first Internet e-mail corporate usage report; concludes e-mail abuse at epidemic levels. http://www.wordtalk.com

[4] Joachims T., "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", Proceedings of the European Conference on Machine Learning,(Chemnitz, Germany), 1998.

[5] King-ip L. & Ravikumar, K. (2001): A similarity-based soft clustering algorithm for documents. Proc. of the Seventh Int. Conf. on Database Sys. for Advanced Applications.

[6] Kiritchenko S. & Matwin S. "Email Classification with Co-training", CASCON'02 (IBM Centre for Advanced Studies Conference), Toronto, 2002.

[7] Nikravesh, M. & Azvine, B. (eds.) (2001): Proceedings of the 2001 BISC International Workshop on Fuzzy Logic and the Internet FLINT 2001, University of California-Berkeley.

[8] Miller G.A., WordNet: A lexical database for English, Communications of the ACM 11, (1995) 39-41.

[9] Ohgaya R. , Takagi T, y otros., Conceptual fuzzy sets- based navigation system for Yahoo!, Proceedings of the 2002 NAFIPS annual meeting, (2002) 274-279.

[10] Pasi G. (2002): Flexible information retrieval: some research trends. Mathware & Soft Computing 9, 107-121

[11] Olivas, J. A. (2000): Contribution to the experimental study of the prediction based on Fuzzy Deformable Categories, PhD Thesis, University of Castilla-La Mancha, Spain.

[12] Olivas, J. A.; Garcés, P.; Romero, F. P. (2003): An application of the FIS-CRM model to the FISS metasearcher: Using fuzzy synonymy and fuzzy generality for representing concepts in documents. Int. Jour. of Approx. Reasoning (Soft Computing in Recognition and Search) 34, 201-219.

[13] Sahami M., Dumais S., Heckerman D., Horvitz E. "A Bayesian Approach to Filtering Junk E-Mail", in proceedings of the AAAI Symposium 1998.

[14] Turene, Nicolas, Learning Semantic Classes for improving Email Classification, Proceeding of the Text Link, 2003

[15] Widyantoro D., Yen J., Incorporating fuzzy ontology of term relations in a serach engine,Proceedings of the BISC Int. Workshop on Fuzzy Logic and the Internet, (2001), 155-160.

[16] Zadeh, L. A note on prototype set theory and fuzzy sets. Cognition, 291-297, 1982.

Improving the performance of ordinal fuzzy linguistic IRSs*

E. Herrera-Viedma **O. Cordón** **M. Luque**
Dept. of Computer Science and A.I, Dept. of Computer Science and N.A.,
University of Granada, Granada (Spain) University of Córdoba, Córdoba (Spain)
viedma@decsai.ugr.es ocordon@decsai.ugr.es mluque@uco.es

Abstract

Information Retrieval Systems (IRSs) based on an ordinal fuzzy linguistic approach present some problems of loss of precision and information when working with discrete linguistic expression domains or when applying approximation operations in the symbolic aggregation methods.

In this paper, we present an IRS based on a 2-tuple fuzzy linguistic approach which allows us to overcome the problems of ordinal fuzzy linguistic IRSs and improve the performance.

KeyWords: Information Retrieval, Fuzzy Linguistic Modelling.

1 Introduction

Information Retrieval involves the development of computer systems for the storage and retrieval of (predominantly) textual information (documents). The use of linguistic variables [15] to represent the input and output information in the retrieval process of Information Retrieval Systems (IRSs) considerably improves the IRS-user interaction. In the literature we can find different models of linguistic IRSs based on a *fuzzy linguistic approach* [1, 2, 3, 6, 7, 8, 9].

Linguistic IRS Models proposed in [3, 6, 7, 8] are based on an ordinal fuzzy linguistic approach [4], and they are affected by the two characteristic problems of ordinal fuzzy linguistic modelling [5]:

- *The loss of precision*: The ordinal fuzzy linguistic approach works with discrete linguistic domains and this implies limitations in the representation of the information, e.g. to represent the relevance degrees.
- *The loss of information*: The aggregation operators of ordinal linguistic information use approximation operations in their definitions (e.g. *rounding* operation), and these cause loss of information.

The main aim of the paper is to present a model of a linguistic IRS based on a 2-tuple fuzzy linguistic approach [5]. The 2-tuple fuzzy linguistic modelling solves the problems of ordinal one, and therefore, allows us to improve the performance of ordinal fuzzy linguistic IRSs.

This contribution is set out as follows. The 2-tuple fuzzy linguistic approach is presented in Section 2. The new model of linguistic IRS is defined in Section 3. Finally, Section 4 includes our conclusions.

2 The 2-tuple fuzzy linguistic approach

The 2-tuple fuzzy linguistic approach is a special kind of fuzzy linguistic approach that was introduced in [5] to overcome the problems of the ordinal one [7].

Let $S = \{s_0, ..., s_T\}$ be a linguistic term set with odd cardinality ($T+1$ is the cardinality of S and usually is equal to 7 or 9[1]), where the mid term represents an assessment of approximately 0.5 and with the rest of the terms being placed symmetrically around it.

Definition 1. [5] Let β be the result of an aggregation of the indexes of a set of labels assessed in S, i.e., the result of a symbolic aggregation operation, $\beta \in [0, T]$ and $\beta \notin \{0, ..., T\}$,. Let $i = round(\beta)$ and $\alpha = \beta - i$ be

*This research has been supported by CICYT under project TIC2003-07977

[1] Several studies have demonstrated that people can not handle more than 7+/-2 levels of quantification [11].

two values, such that, $i \in \{0, ..., \mathcal{T}\}$ and $\alpha \in [-.5, .5)$ then α is called a *Symbolic Translation*.

The 2-tuple fuzzy linguistic approach is developed from the concept of symbolic translation by representing the linguistic information equivalent to β by means of 2-tuples (s_i, α_i), $s_i \in S$ and $\alpha_i \in [-.5, .5)$. This model defines a set of transformation functions between numeric values and 2-tuples.

Definition 2. [5] Let S be a linguistic term set and $\beta \in [0, \mathcal{T}]$, then the 2-tuple that expresses the equivalent information to β is obtained with the following function: $\Delta : [0, \mathcal{T}] \longrightarrow S \times [-0.5, 0.5)$,

$$\Delta(\beta) = (s_i, \alpha), \ with \ \begin{cases} s_i & i = round(\beta) \\ \alpha = \beta - i & \alpha \in [-.5, .5) \end{cases}$$

For all Δ there exists Δ^{-1}, defined as $\Delta^{-1}(s_i, \alpha) = i + \alpha$. Obviously, the conversion of a linguistic term into a linguistic 2-tuple consists of adding a symbolic translation value of 0: $s_i \in S \Longrightarrow (s_i, 0)$.

The 2-tuple linguistic computational model operates with the 2-tuples without loss of information and is based on the following operations [5]:

1. Negation operator of 2-tuples:

$$Neg((s_i, \alpha)) = \Delta(\mathcal{T} - (\Delta^{-1}(s_i, \alpha))).$$

2. Comparison of 2-tuples: The comparison of linguistic 2-tuples is carried out according to an ordinary lexicographic order. Let (s_k, α_1) and (s_l, α_2) be two 2-tuples, with each one representing a counting of information:

- If $k < l$ then (s_k, α_1) is smaller than (s_l, α_2)
- If $k = l$ then
  1. if $\alpha_1 = \alpha_2$ then (s_k, α_1) and (s_l, α_2) represent the same information,
  2. if $\alpha_1 < \alpha_2$ then (s_k, α_1) is smaller than (s_l, α_2),
  3. if $\alpha_1 > \alpha_2$ then (s_k, α_1) is bigger than (s_l, α_2).

3. Aggregation of 2-tuples: Using Δ and Δ^{-1} any numerical aggregation operator can be easily extended for dealing with linguistic 2-tuples.

Definition 3. [14] Let $A = \{a_1, \ldots, a_m\}, a_k \in [0, 1]$ be a set of assessments to be aggregated, then the Ordered Weighted Averaging (OWA) operator, ϕ, is defined as

$$\phi(a_1, \ldots, a_m) = W \cdot B^T$$

where $W = [w_1, \ldots, w_m]$, is a weighting vector, such that $w_i \in [0, 1]$ and $\Sigma_i w_i = 1$; and $B = [b_1, \ldots, b_m]$ is a vector associated to A, such that, $B = \sigma(A) = \{a_{\sigma(1)}, \ldots, a_{\sigma(m)}\}$, where $a_{\sigma(j)} \leq a_{\sigma(i)} \ \forall \ i \leq j$, with σ being a permutation over the set of labels A.

A 2-tuple linguistic definition of ϕ would be as follows:

Definition 4. Let $A = \{(a_1, \alpha_1), \ldots, (a_m, \alpha_m)\}$ be a set of assessments in the 2-tuple linguistic domain, then the 2-tuple linguistic OWA operator, ϕ_{2t} is defined as:

$$\phi_{2t}((a_1, \alpha_1), \ldots, (a_m, \alpha_m)) = \Delta(W \cdot B^T)$$

$$B = \sigma(A) = \{\Delta^{-1}(a_1, \alpha_1))_{\sigma(1)}, \ldots, (\Delta^{-1}(a_m, \alpha_m))_{\sigma(m)}\}.$$

3 A 2-tuple fuzzy linguistic IRS model

In this Section, we present a fuzzy linguistic IRS model based on a 2-tuple fuzzy linguistic approach which overcomes the problems of loss of precision and information of the ordinal fuzzy linguistic IRS models. The main properties of this model linguistic IRS model are: i)users can express their information needs by means of multi-weighted linguistic Boolean queries using different semantics, even, simultaneously, and ii) the Boolean connectives are modelled in a flexible way by means of soft computing operator, 2-tuple linguistic OWA operator.

3.1 Multi-weighted linguistic Boolean queries

We consider a set of documents D= $\{d_1, \ldots, d_m\}$ represented by means of index terms T= $\{t_1, \ldots, t_l\}$, which describe the subject content of the documents. A numeric indexing function F : DxT $\rightarrow [0, 1]$ is defined, called *index term weighting*. F maps a given document d_j and a given index term t_i to a numeric weight between 0 and 1. Thus, $F(d_j, t_i)$ is a numerical weight that represents the degree of significance of t_i in d_j. $F(d_j, t_i) = 0$ implies that the document d_j is not at all about the concept(s) represented by the index term t_i and $F(d_j, t_i) = 1$ implies that the document d_j is perfectly represented by the concept(s) indicated by t_i.

To retrieve documents from D users can use multi-weighted linguistic Boolean queries as in [6, 7, 8]. With such queries a information need is expressed as a combination of the index terms which are connected by the logical operators AND ($\wedge$), OR ($\vee$), and NOT ($\neg$) and can be weighted with three ordinal linguistic values [2] taken from a label set S associated to three different semantics as in [7]: symmetrical threshold semantics, relative importance semantics, quantitative semantics.

[2]The weights are defined as ordinal values but they are transformed to 2-tuple values, adding a symbolic translation value of 0, in order to process the query.

As in [2] we use the linguistic variable *Importance* to express the linguistic weights, but defining it with an ordinal fuzzy linguistic approach [7]. Thus, we consider a set of labels S to express the query weights. Then, we define a multi-weighted linguistic Boolean query as any legitimate Boolean expression whose atomic components (atoms) are quadruples $< t_i, c_i^1, c_i^2, c_i^3 >$ belonging to the set, $\mathrm{Tx}S^3$; $t_i \in \mathrm{T}$, and $c_i^1,\ c_i^2,\ c_i^3$ are ordinal values of the linguistic variable *Importance*, modelling the symmetrical threshold semantics, the quantitative semantics, and the importance semantics, respectively. Accordingly, the set Q of the legitimate queries is defined by the following syntactic rules:

1. $\forall q =< t_i, c_i^1, c_i^2, c_i^3 >\in \mathrm{Tx}S^3 \rightarrow q \in \mathrm{Q}$.
2. $\forall q, p \in \mathrm{Q} \rightarrow q \wedge p \in \mathrm{Q}$.
3. $\forall q, p \in \mathrm{Q} \rightarrow q \vee p \in \mathrm{Q}$.
4. $\forall q \in \mathrm{Q} \rightarrow \neg(q) \in \mathrm{Q}$.
5. All legitimate queries $q \in \mathrm{Q}$ are only those obtained by applying rules 1-4, inclusive.

3.2 Evaluating multi-weighted linguistic Boolean queries

The evaluation of a multi-weighted linguistic Boolean query is carried out by means of a constructive bottom-up process based on the *criterion of separability* [12] and at the same time as supporting all the semantics of query weights considered. The evaluation of a query is developed in the five subsequent steps:

1.- Preprocessing of the query.

The user query is preprocessed to put it into either conjunctive normal form (CNF) or disjunctive normal form (DNF), in such a way that every Boolean subexpression must have more than two atoms. Weighted single-term queries are kept in their original forms.

2.- Evaluation of atoms with respect to the symmetrical threshold semantics.

According to a symmetrical threshold semantics, a user may search for documents with a minimally acceptable presence of one term in their representations, or documents with a maximally acceptable presence of one term in their representations [7, 6]. Given a request $< t_i, c_i^1, c_i^2, c_i^3 >$, this means that the query weights that imply the presence of a term in a document $c_i^1 \geq s_{\mathcal{T}/2}$ (e.g. *High, Very High*) must be treated differently to the query weights that imply the absence of one term in a document $c_i^1 < s_{\mathcal{T}/2}$ (e.g. *Low, Very Low*). Then, if $c_i^1 \geq s_{\mathcal{T}/2}$, the request $< t_i, c_i^1, c_i^2, c_i^3 >$ is synonymous with the request $< t_i, at\ least\ c_i^1, c_i^2, c_i^3 >$, which expresses the fact that the desired documents are those having F values as high as possible; and if $c_i^1 < s_{\mathcal{T}/2}$, the former request is synonymous with the request $< t_i, at\ most\ c_i^1, c_i^2, c_i^3 >$, which expresses the fact that the desired documents are those having F values as low as possible. This interpretation is defined by means of a parameterized 2-tuple linguistic matching function $g^1 : D\times$ T $\times(S \times [-.5, .5)) \rightarrow (S \times [-.5, .5))$. Given an atom $< t_i, c_i^1, c_i^2, c_i^3 >$ and a document $d_j \in D$, g^1 obtains the linguistic RSV of d_j, called $RSV_j^{i,1}$, by measuring how well the index term weight $F(d_j, t_i)$ satisfies the request expressed by the linguistic weight c_i^1 according to the following expression:

$$RSV_j^{i,1} = g^1(d_j, t_i, (c_i^1, 0)) =$$

$$\begin{cases} \Delta(0) & \text{if } (s_b,0) \geq (s_{\mathcal{T}/2},0) \text{ and } (s_a,0) = (s_0,0) \\ \Delta(i_1) & \text{if } (s_b,0) \geq (s_{\mathcal{T}/2},0) \text{ and } (s_0,0) < (s_a,0) < (s_b,0) \\ \Delta(i_2) & \text{if } (s_b,0) \geq (s_{\mathcal{T}/2},0) \text{ and } (s_b,0) \leq (s_a,0) < (s_{\mathcal{T}},0) \\ \Delta(\mathcal{T}) & \text{if } (s_b,0) \geq (s_{\mathcal{T}/2},0) \text{ and } (s_a,0) = (s_{\mathcal{T}},0) \\ \Delta(\mathcal{T}) & \text{if } (s_b,0) < (s_{\mathcal{T}/2},0) \text{ and } (s_a,0) = (s_0,0) \\ Neg(\Delta(i_1)) & \text{if } (s_b,0) < (s_{\mathcal{T}/2},0) \text{ and } (s_0,0) < (s_a,0) \leq (s_b,0) \\ Neg(\Delta(i_2)) & \text{if } (s_b,0) < (s_{\mathcal{T}/2},0) \text{ and } (s_b,0) < (s_a,0) < (s_{\mathcal{T}},0) \\ \Delta(0) & \text{if } (s_b,0) < (s_{\mathcal{T}/2},0) \text{ and } (s_a,0) = (s_{\mathcal{T}},0) \end{cases}$$

such that

$$\begin{array}{l} i_1 = Max\{0, (b - \frac{(b-a)}{\mathcal{K}})\} \\ i_2 = Min\{\mathcal{T}, (b + \frac{(a-b)}{\mathcal{K}})\} \end{array} \quad \mathcal{K} \in \{1, 2, 3, ..., b\}.$$

such that, (i) $s_b = c_i^1$; (ii) s_a is the linguistic index term weight obtained as $s_a = Label(F(d_j, t_i))$, being $Label : [0, 1] \rightarrow S$ a function that assigns a label in S to a numeric value $r \in [0, 1]$ according to the following expression:

$$Label(r) = Sup_q\{s_q\ \in S : \mu_{s_q}(r) = Sup_v\{\mu_{s_v}(r)\}\}.$$

3.- Evaluation of atoms with respect to the quantitative semantics.

In this step, documents go on being evaluated with regard to their relevance to individual atoms of the query, but considering the restrictions imposed by the quantitative semantics.

The linguistic quantitative weights are interpreted as follows [7]: when a user establishes a certain number of documents for a term in the query, expressed by a linguistic quantitative weight, then the set of documents to be retrieved must have the minimum number of documents that satisfies the compatibility or membership function associated with the meaning of the label used as linguistic quantitative weight. Furthermore, these documents must be those that better satisfy the threshold restrictions imposed on the term.

Therefore, given an atom $< t_i, c_i^1, c_i^2, c_i^3 >$ and assuming that $RSV_j^{i,1} \in (S\times[-.5, .5))$ represents the evaluation according to the symmetrical threshold semantics

for d_j, we model the interpretation of a quantitative semantics by means of a 2-tuple linguistic matching function, called g^2, which is defined between the $RSV_j^{i,1}$ and the linguistic quantitative weight $c_i^2 \in S$. Then, the evaluation of the atom $< t_i, c_i^1, c_i^2, c_i^3 >$ with respect to the quantitative semantics associated with c_i^2 for a document d_j, called $RSV_j^{i,1,2} \in (S \times [-.5, .5))$, is obtained by means of the linguistic matching function $g^2 : D \times (S \times [-.5, .5)) \times S \rightarrow (S \times [-.5, .5))$ as follows $RSV_j^{i,1,2} =$

$$g^2(RSV_j^{i,1}, c_i^2, d_j) = \begin{cases} (s_0, 0) & \text{if } d_j \notin \mathcal{B}^{\mathcal{S}} \\ RSV_j^{i,1} & \text{if } d_j \in \mathcal{B}^{\mathcal{S}} \end{cases} \quad \text{where}$$

$\mathcal{B}^{\mathcal{S}}$ is the set of documents such that $\mathcal{B}^S \subseteq Supp(\mathcal{M})$ where $\mathcal{M} = \{(d_1, RSV_1^{i,1}), \ldots, (d_m, RSV_m^{i,1})\}$, is a fuzzy subset of documents obtained according to the following algorithm:

1. $K = \#Supp(\mathcal{M})$
2. REPEAT
   $M^K = \{s_q \in S : \mu_{s_q}(K/m) = Sup_v\{\mu_{s_v}(K/m)\}\}$.
   $s^K = Sup_q\{s_q \in M^K\}$.
   $K = K - 1$.
3. UNTIL $((c_i^2 \in M^{K+1})$ OR $(c_i^2 \geq s^{K+1}))$.
4. $\mathcal{B}^{\mathcal{S}} = \{d_{\sigma(1)}, \ldots, d_{\sigma(K+1)}\}$, such that $RSV_{\sigma(h)}^{i,1} \leq RSV_{\sigma(l)}^{i,1}, \forall l \leq h$.

According to g^2, the application of the quantitative semantics consists of reducing the number of documents to be considered in the evaluation of t_i in the later steps.

4.- Evaluation of subexpressions and modelling of the relative importance semantics

We consider that the relative importance semantics in a single-term query has no meaning. Then, in this step we have to evaluate the relevance of documents with respect to the subexpressions of queries composed of two atomic components..

Given a subexpression q_v with $\mathcal{I} \geq 2$ atoms, we know that each document d_j presents a partial $RSV_j^{i,1,2} \in (S \times [-.5, .5))$ with respect to each atom $< t_i, c_i^1, c_i^2, c_i^3 >$ of q_v. Then, the evaluation of the relevance of a document d_j with respect to the whole subexpression q_v implies the aggregation of the partial relevance degrees $\{RSV_j^{i,1,2},\ i = 1, \ldots, \mathcal{I}\}$ weighted by means of the respective relative importance degrees $\{c_i^3 \in S,\ i = 1, \ldots, \mathcal{I}\}$.

In each subexpression q_v we find that the atoms can be combined using the AND or OR Boolean connectives, depending on the normal form of the user query. The restrictions imposed by the relative importance weights must be applied in the aggregation operators used to model both connectives. These aggregation operators should guarantee that the more important the query terms, the more influential they are in the determination of the RSVs. To do so, these aggregation operators must carry out two activities [4]: i) the transformation of the weighted information under the importance degrees by means of a transformation function h; and ii) the aggregation of the transformed weighted information by means of an aggregation operator of non-weighted information f. As it is known, the choice of h depends upon f. In [13], Yager discussed the effect of the importance degrees on the MAX (used to model the connective OR) and MIN (used to model the connective AND) types of aggregation and suggested a class of functions for importance transformation in both types of aggregation. For the MIN aggregation, he suggested a family of t-conorms acting on the weighted information and the negation of the importance degree, which presents the non-increasing monotonic property in these importance degrees. For the MAX aggregation, he suggested a family of t-norms acting on weighted information and the importance degree, which presents the non-decreasing monotonic property in these importance degrees. Similarity, in [10], Lee analizes the behavioral aspects of the fuzzy operators and address important issues to affect retrieval effectiveness.

Following the ideas shown above, we use the extended definition of OWA operators ϕ_{2t}^1 (with orness(W)$\leq$ 0.5) and ϕ_{2t}^2 (with orness(W)$>$ 0.5) to model the AND and OR connectives, respectively. [3] Hence, when $h = \phi_{2t}^1$, then $f = \max(Neg(weight), value)$, and when $h = \phi^2$, then $f = min(weight, value)$.

Then, given a document d_j, we evaluate its relevance with respect to a subexpression q_v, called $RSV_j^v \in (S \times [-.5, .5))$ as follows:

1. if q_v is a conjunctive subexpression then $RSV_j^v = \phi_{2t}^1(max(Neg(c_1^3, 0), RSV_j^{1,1,2}), \ldots, max(Neg(c_{\mathcal{I}}^3, 0), RSV_j^{\mathcal{I},1,2}))$, *and*
2. if q_v is a disjunctive subexpression then $RSV_j^v = \phi_{2t}^2(min((c_1^3, 0), RSV_j^{1,1,2}), \ldots, min((c_{\mathcal{I}}^3, 0), RSV_j^{\mathcal{I},1,2}))$.

5.- Evaluation of the whole query.

In this step, the final evaluation of each document is achieved by combining their evaluations with respect

[3] In order to classify OWA operators in regards to their location between '"and" and "or" Yager [14] introduced a *orness measure*, associated with any vector W as follow: $orness(W) = \frac{1}{(m-1)} \sum_{i=1}^{m} (m - i) w_i$

to all the subexpressions using, again, the extended definition of OWA operators ϕ^1_{t2} and ϕ^2_{t2} to model the AND and OR connectives, respectively.

Then, given a document d_j, we evaluate its relevance with respect to a query q as $RSV_j \in (S \times [-.5, 5))$ where $RSV_j = \phi^1_{2T}(RSV^1_j, \ldots, RSV^{\mathcal{V}}_j)$ if q is in CNF, and $RSV_j = \phi^2_{2T}(RSV^1_j, \ldots, RSV^{\mathcal{V}}_j)$ if q is in DNF, with $\mathcal{V}$ standing for the number of subexpressions in q.

Remark 1: *On the NOT Operator.* We should note that, if a query is in CNF or DNF form, we have to define the negation operator only at the level of single atoms. This simplifies the definition of the NOT operator. As was done in [7], the evaluation of document d_j for a negated weighted atom $< \neg(t_i), c^1_i, c^2_i, c^3_i >$ is obtained from the negation of the index term weight $F(t_i, d_j)$. This means to calculate g^1 from the linguistic value $Label(1 - F(t_i, d_j))$.

3.3 Example of Application

In this subsection, we present an example of performance of the proposed IRS let us suppose a small database containing a set of seven documents $D = \{d_1, \ldots, d_7\}$, represented by means of a set of 10 index terms $T = \{t_1, \ldots, t_{10}\}$. Documents are indexed by means of an indexing function F, which assigns the following weights to each of them:

$$\begin{aligned}
d_1 &= 0.7/t_5 + 0.4/t_6 + 1/t_7 \\
d_2 &= 1/t_4 + 0.6/t_5 + 0.8/t_6 + 0.9/t_7 \\
d_3 &= 0.5/t_2 + 1/t_3 + 0.8/t_4 \\
d_4 &= 0.9/t_4 + 0.5/t_6 + 1/t_7 \\
d_5 &= 0.7/t_3 + 1/t_4 + 0.4/t_5 + 0.8/t_9 + 0.6/t_{10} \\
d_6 &= 1/t_5 + 0.99/t_6 + 0.8/t_7 \\
d_7 &= 0.8/t_5 + 0.02/t_6 + 0.8/t_7 + 0.9/t_8.
\end{aligned}$$

In the same way, let us suppose the following label set to express the values of the linguistic variable *Importance*.

$S = \{N = (0, 0, 0, 0), EL = (0.01, 0.02, 0.01, 0.05),$
$VL = (0.1, 0.18, 0.06, 0.05), L = (0.22, 0.36, 0.05, 0.06),$
$M = (0.41, 0.58, 0.09, 0.07), H = (0.63, 0.80, 0.05, 0.06),$
$VH = (0.78, 0.92, 0.06, 0.05), EH = (0.98, 0.99, 0.05, 0.01)$
$, T = (1, 1, 0, 0)\}$.

Finally, consider that a user formulates the following query $q = ((t_5, VH, VL, VH) \wedge (t_6, L, L, VL)) \vee (t_7, H, L, H)$. Its evaluation is as follows:

1.- Preprocessing of the query.

The query q is in DNF, but it presents one subexpression with only one atom. Therefore, q must be preprocessed and transformed into normal form with every subexpression having more than two atoms. Then, q is transformed into the following equivalent query $q = ((t_5, VH, VL, VH) \vee (t_7, H, L, H)) \wedge ((t_6, L, L, VL) \vee (t_7, H, L, H))$, which is expressed in CNF.

2.- Evaluation of atoms with respect to the symmetrical threshold semantics.

In this step we obtain the documents represented in a linguistic form using the translation function Label:

$$\begin{aligned}
d_1 &= H/t_5 + M/t_6 + T/t_7 \\
d_2 &= T/t_4 + M/t_5 + H/t_6 + VH/t_7 \\
d_3 &= M/t_2 + T/t_3 + H/t_4 \\
d_4 &= VH/t_4 + M/t_6 + T/t_7 \\
d_5 &= H/t_3 + T/t_4 + M/t_5 + H/t_9 + M/t_{10} \\
d_6 &= T/t_5 + EH/t_6 + H/t_7 \\
d_7 &= H/t_5 + EL/t_6 + H/t_7 + VH/t_8.
\end{aligned}$$

Let us set the sensitivity parameter $\mathcal{K} = 2$ which gives a large importance to the closeness between linguistic values in g^1. Then, the evaluations of atoms according to the symmetric threshold semantics modelled by g^1 are:
$\{RSV^{5,1}_1 = (VH, -.5), RSV^{5,1}_2 = (H, 0), RSV^{5,1}_5 = (H, 0)$
$, RSV^{5,1}_6 = (T, 0), RSV^{5,1}_7 = (VH, -.5)\}$
$\{RSV^{6,1}_1 = (H, -.5), RSV^{6,1}_2 = (M, 0), RSV^{6,1}_4 = (H, -.5)$
$, RSV^{6,1}_6 = (L, 0), RSV^{6,1}_7 = (VH, 0)\}$
$\{RSV^{7,1}_1 = (T, 0), RSV^{7,1}_2 = (VH, -.5), RSV^{7,1}_4 = (T, 0)$
$, RSV^{7,1}_6 = (H, 0), RSV^{7,1}_7 = (H, 0)\}$

3.- Evaluation of atoms with respect to the quantitative semantics.

The evaluation of the atoms of the query according to the quantitative semantics modeled by g^2 are:
$\{RSV^{5,1,2}_6 = (T, 0)\}$

$\{RSV^{6,1,2}_7 = (VH, 0), RSV^{6,1,2}_1 = (H, -.5)\}$

$\{RSV^{7,1,2}_1 = (T, 0), RSV^{7,1,2}_4 = (T, 0)\}$

We should note that the quantitative semantics decreases the number of documents associated to be considered in each query term.

4.- Evaluation of subexpressions and modelling the relative importance semantics.

The query q' has two subexpressions and each of them presents two atoms, $q'^1 = (t_5, VH, VL, VH) \vee (t_7, H, L, H)$ and $q'^2 = (t_6, L, L, VL) \vee (t_7, H, L, H)$. Each subexpression is in disjunctive form, and thus we must use an OWA operator ϕ^2_{2t} with $orness(W) > 0.5$ (for example, with $(W = [0.8, 0.2])$).

$\{RSV^1_1 = (M, 0), RSV^1_4 = (M, 0), RSV^1_6 = (H, -.2)\}$

$\{RSV^2_1 = (M, .4), RSV^2_4 = (M, 0), RSV^2_7 = (VL, -.4).\}$

5.- Evaluation of the whole query.

We obtain the document evaluation with respect to the whole query using an OWA operator ϕ^1_{2t} with $orness(W) < 0.5$ (e.g. with $(W = [0.4, 0.6])$).

$$RSV_1 = (M, .16) \qquad RSV_4 = (M, 0)$$

$$RSV_6 = (VL, -.38) \qquad RSV_7 = (EL, -.36).$$

We should point out that in an ordinal context we could not distinguish the best document, however, in our linguistic IRS based on 2-tuple linguistic representation model this is possible, the best document is d_1.

4 Conclusions

In this paper, we have presented a linguistic IRS based on a 2-tuple fuzzy linguistic approach. Such a linguistic approach allows to avoid problems of loss of precision and information in the IRS results, and consequently improves its performance . This improvement is reflected in the fact that the evaluation of the documents is not only a label, but also it has associate the value of the translation from the original result to the closest index label in the linguistic term set.

Besides, we are conscious that the proposed model is complex to be used by the user. It is just a theoretical model. In future works, a graphic user interface allowing the user to utilize this model easily will be developed.

References

[1] G. Bordogna, P. Carrara, and G. Pasi. Fuzzy Approaches to Extend Boolean Information Retrieval. In P. Bosc and J. Kacprzyk, editors, *Fuzziness in Database Management Systems*, pages 231–274. 1995.

[2] G. Bordogna and G. Pasi. A Fuzzy Linguistic Approach Generalizing Boolean Information Retrieval: A Model and its Evaluation. *Journal of the American Society for Information Science*, 44:70–82, 1993.

[3] G. Bordogna and G. Pasi. An Ordinal Information Retrieval Model. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(1):63–75, 2001.

[4] F. Herrera and E. Herrera-Viedma. Aggregation Operators for Linguistic Weighted Information. *IEEE Transactions on Systems, Man and Cybernetics; Part A: Systems and Humans*, 27:646–656, 1997.

[5] F. Herrera and L. Martínez. A 2-tuple Fuzzy Linguistic Representation Model for Computing with Words. *IEEE Transaction on Fuzzy Systems*, 8(6):746–752, 2000.

[6] E. Herrera-Viedma. An Information Retrieval System with Ordinal Linguistic Weighted Queries based on Two Weighting Elements. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(1):77–88, 2001.

[7] E. Herrera-Viedma. Modeling the Retrieval Process for an Information Retrieval System using an Ordinal Fuzzy Linguistic Approach. *Journal of the American Society for Information Science and Technology*, 52(6):460–475, 2001.

[8] E. Herrera-Viedma, O. Cordón, M. Luque, A. G. López, and A. M. Múñoz. A Model of Fuzzy Linguistic IRS Based on Multi-Granular Linguistic Information. *International Journal of Approximate Reasoning*, 34:221–239, 2003.

[9] D.H. Kraft, G. Bordogna, and G. Pasi. An Extended Fuzzy Linguistic Approach to Generalize Boolean Information Retrieval. *Information Sciences*, 2:119–134, 1994.

[10] J.H. Lee, W.Y. Kin, M. Kim, and Y.J. Lee. On the Evaluation of Boolean Operators in the Extended Boolean Retrieval FrameWork. In *Proceedings of the 16th Annual International ACM-SIGIR*, pages 291–297, 1993.

[11] G. Miller. The Magical Number Seven, Plus mInus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review*, 63:81–97, 1956.

[12] W.G. Waller and D.H. Kraft. A Mathematical Model of a Weighted Boolean Retrieval System. *Information Processing & Management*, 15:235–245, 1979.

[13] R.R Yager. A Note on Weighted Queries in Information Retrieval Systems. *Journal of the American Society for Information Science*, 38:23–24, 1987.

[14] R.R. Yager. On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making. *IEEE Transactions on Systems, Man, and Cybernetics*, 18:183–190, 1988.

[15] L.A. Zadeh. The Concept of a Linguistic Variable and its Applications to Approximate Reasoning. *Part I, II & III, Information Science*, 8:199–249, 8:301–157, 9:43–80, 1975.

Minería de textos: aplicaciones con lógica difusa

C. Justicia
Servicio de Información Científica de Andalucía
Universidad de Granada
consuelo@ugr.es

M.J. Martín-Bautista, D. Sánchez
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Granada
{mbautis, daniel}@decsai.ugr.es

Resumen

El presente artículo trata sobre el Descubrimiento de Conocimiento en Textos y de la Minería de Textos como fase principal de este nuevo área de trabajo. Se presenta un resumen de los diferentes modelos y de técnicas para llevar a cabo tareas de minería de textos en el ámbito documental y extraer patrones. Para modelar dichos patrones, una de las herramientas aplicadas con más exito es la teoría de conjuntos difusos. Concretamente, extendemos una de las técnicas más utilizadas en minería, las reglas de asociación al ámbito difuso. Las reglas de asociación difusas nos permiten relacionar términos que representan un conjunto de textos mediante grados de aparición intermedios además de los clásicos de presencia/ausencia.

Palabras Clave: Minería de textos, descubrimiento de conocimiento en textos, lógica difusa, reglas de asociación difusas.

1. Introducción

Desde los años 60, la evolución de los sistemas de soporte a la decisión ha sido completamente exponencial, motivada por la necesidad de las organizaciones de tener la información necesaria para que los directivos puedan tomar decisiones oportunas y, a veces, vitales para la empresa.

De este modo, surge un área de estudio, que denominamos Descubrimiento de Conocimiento en Datos (KDD ó *Knowledge Discovery from Databases*), que podemos definir como el *Proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y en última instancia comprensibles a partir de los datos.* [7].

El proceso de KDD se divide en varias fases tales como: Limpieza, Integración, Selección y Transformación de Datos, Minería de Datos (*Data Mining) ó DM*, Evaluación de Patrones o Representación del conocimiento. De todas ellas, la principal y más conocida es la Minería de Datos que descubre o deriva nueva información de los datos, encontrando patrones en conjuntos de datos.

Con la llegada de las autopistas de la información e Internet, el volumen de datos disponible en la web y en bases de datos documentales es cada vez mayor, y la necesidad de explotar esta información se incrementa. De este modo, surgen áreas de estudio paralelas a KDD y DM, denominadas *Knowledge Discovery from Text* (KDT)y *Text Mining* (TM), que extienden los modelos y técnicas de las áreas originales para aplicarlos a documentos y textos.

La aplicación de KDD y DM se lleva a cabo sobre datos estructurados almacenados generalmente en bases de datos relacionales. Cuando esta misma tecnología se extiende a otro tipos de datos tales como textos y documentos, la falta de estructura y homogeneidad de los mismos dificultan enormemente su procesamiento y explotación. Para encontrar representaciones de los documentos susceptibles de ser procesadas automáticamente, podemos recurrir a la teoría de conjuntos difusos [23]. Asumiendo que un documento se puede representar mediante un conjunto de términos extraídos del mismo, la teoría de conjuntos difusos nos permite dar diferentes grados de presencia de los términos en los documentos en función de la frecuencia de aparición de dichos términos en los documentos.

El uso de este tipo de representaciones difusas hacen que, a su vez, las técnicas utilizadas para la minería de textos puedan manejarlas. De este modo, se aplican técnicas tan comunes como las reglas de asociación, ya extendidas al ámbito difuso en el DM tradicional,

denominadas reglas de asociación difusas.

En este artículo comenzaremos resumiendo los conceptos básicos en las áreas de KDT y TM, presentando los modelos y técnicas más relevantes. A continuación, recogemos algunas de las principales aplicaciones de la teoría de conjuntos difusos a dichas áreas. Finalmente comentaremos las conclusiones obtenidas de dicho estudio y las tendencias futuras de investigación en el campo. Para ello, una vez introducidos los términos KDD y DM, abordaremos la extracción de trozos de información, desconocida a priori, en texto. Para ello, y del mismo modo que hemos hecho con los datos, comenzaremos con la definición del proceso de Descubrimiento de Conocimiento en Textos.

2. Descubrimiento de conocimiento en textos (Knowledge discovery in text (KDT))

Tradicionalmente, el descubrimiento de conocimiento se ha realizado sobre los datos almacenados en bases de datos estructuradas; sin embargo, la mayoría de la información de la que disponen las empresas está en formato de texto, bien sea en Intranets, páginas web, informes de trabajo, publicaciones, correo electrónico..., lo cuál hace necesario una revisión del proceso de descubrimiento de conocimiento.

La información textual difiere de los datos estructurados principalmente en la ausencia de estructura o, mejor dicho, en la compleja estructura implícita del texto. El principal problema a resolver, por tanto, será encontrar una representación intermedia del texto que permita que se apliquen las técnicas de descubrimiento.

Aunque hay autores que hablan indistintamente de KDD o de KDT, nosotros realizaremos una clara distinción: cuando se habla de descubrimiento de conocimiento aplicado a datos, estaremos trabajando con KDD y cuando este descubrimiento se quiere hacer sobre texto nos referiremos a KDT. En la Tabla I vemos la comparativa realizada por [17].

La primera fase a llevar a cabo en KDT es el preprocesamiento, que depende en gran medida del uso que se le va a dar a los resultados. Puede implicar análisis de Lenguaje Natural y se puede realizar con técnicas de Extracción de Información, Recuperación de Información o Análisis de texto, lo que nos permitirá colocar el texto en una estructura útil para ser procesada computacionalmente. A dicha estructura la llamaremos Forma Intermedia. El conocimiento obtenido dependerá, en parte, de la técnica utilizada para obtener dicha estructura. Algunas de las técnicas utilizadas en el preprocesamiento son [17]: Técnicas de PLN, Filtrado de Términos, Motores de Texto (basados en palabras o en conceptos), Grafos Conceptuales, Categorización,...

Después de la fase de preprocesamiento, se lleva a cabo la fase principal del proceso de descubrimiento, esto es, la minería de textos, que detallaremos en la sección 3.

Finalmente, una de las últimas tareas fundamentales en el descubrimiento de conocimiento, es la visualización de los datos y resultados, es decir, proporcionar un entorno de exploración de datos guiado para el usuario que sea lo más amigable posible. Se pueden utilizar diferentes herramientas tales como los grafos dirigidos para llevar a cabo esta fase.

Hemos mencionado con anterioridad que, como fases previas a la minería de textos podemos utilizar técnicas importadas de la Recuperación de Información. A continuación explicaremos más detalladamente dichas técnicas por ser de las más utilizadas en KDT.

2.1. La Recuperación de Información en KDT

La Recuperación de Información (RI) trata con la representación, almacenamiento, organización, y acceso a las unidades de información y los textos, a partir de una colección de documentos escritos que deben satisfacer una información que necesita el usuario y cuya solicitud ha sido expresada en Lenguaje Natural.

La relación entre la RI y KDT se puede ver desde dos puntos de vista. En primer lugar, cuando pretendemos realizar minería de textos, hay que determinar el conjunto de documentos que van a ser analizados y habrá que estudiar el método de conseguirlos del modo más óptimo. Necesitaremos la RI para obtener el conjunto de documentos con el que trabajar. En segundo lugar, para obtener un Forma Intermedia del texto que nos permita su proceso computacional, se pueden utilizar las mismas técnicas de indexación de RI para obtener dicha estructura. Sin embargo, aunque este preprocesamiento de KDT se base en la descomposición del texto en palabras, igual que en IR, el fin de ambos procesos dista mucho: el objetivo de la IR es conseguir optimizar las consultas del usuario, agilizar las búsquedas de documentos en la base de datos o incluso analizar grandes cantidades de documentos, en ningún momento descubre información que no haya sido escrita por los autores de los documentos. Por el contrario, la meta del KDT es extraer conocimiento nuevo y desconocido.

Pasos	KDD	KDT
1	Comprender el dominio de la aplicación.	El usuario define qué conceptos le interesan.
2	Seleccionar un conjunto de datos objetivo.	Los textos se obtienen con herramientas de Recuperación de Información o de forma manual.
3	Limpieza, preprocesamiento y transformación de datos.	Se describen los conceptos y los textos se analizan y representan mediante una Forma Intermedia.
4	Desarrollo de modelos y construcción de hipótesis.	Identificación los conceptos en la colección de textos.
5	Tareas de *Data Mining*.	Tareas de *Text Mining*.
6	Interpretación de resultados y visualización.	Interpretación de los resultados por un humano.

Cuadro 1: Comparativa entre los procesos KDD y KDT.

3. Minería de textos (Text mining (TM))

Como hemos visto en apartados anteriores, la toma de decisiones, que se ha venido realizando exclusivamente en base a los datos estructurados almacenados generalmente en bases de datos relacionales, necesita en los últimos tiempos nutrirse de los documentos que se acumulan en las empresas y que contienen una información aún más rica que dichos datos. Por lo tanto, la necesidad de Minería de Textos surge ante:

- La diversidad de documentos y la naturaleza heterogénea de los mismos.
- La toma de decisiones se suele basar en experiencias ya ocurridas, almacenadas en papel.
- La diversidad de idiomas en los que se puede presentar el texto.
- La ausencia de estructura del texto, que lo hace difícil de tratar computacionalmente.

Hay que tener en cuenta, no obstante, que no basta con utilizar técnicas de procesamiento de textos avanzado, que pueden llevarse a cabo mediante lingüística computacional, sino que necesitamos encontrar patrones o tendencias útiles a la hora de tomar decisiones. Podemos definir el *Text Mining* como el proceso que descubre información útil que no está presente explícitamente en ninguno de los documentos objeto de análisis y que surge cuando se estudian adecuadamente y se relacionan dichos documentos.

Este proceso se ha denominado de muy diversas formas [10]: *Information Mining, Information Archeology, Knowledge Management, Data Mining, Knowledge Distillation, Document Mining, Text Data Mining* o *Web Document Mining*, y en él podemos diferenciar básicamente dos fases diferenciadas:

- *Text refining:* en esta fase se transforma texto sin forma ni estructura aparente en una Forma Intermedia (FI) que dota de alguna estructura al texto.
- *Knowledge distillation:* en esta segunda fase, se deducen patrones o conocimiento a partir de la Forma Intermedia

Como hemos comentado anteriormente, la parte de Text refining y la generación de la FI [16], [12], [9] viene determinada por la necesidad de una representación de los documentos que pueda ser tratado computacionalmente. La FI debe conservar la riqueza del documento y a su vez facilitar el análisis del mismo. Una Forma Intermedia, puede obtenerse en estructuras diversas tales como términos indexados [8],[9], bolsas de palabras (conjuntos de términos donde un elemento puede aparecer repetido) [19], reglas de episodios [14], ... De todas ellas, la más extendida es la de los términos indexados, propiciada especialmente por el uso previo que se ha hecho de la misma en otros ámbitos del tratamiento de textos tales como la Recuperación de Información. En esta Forma Intermedia de indexación de elementos del texto, para describir un conjunto de documentos $D = d_1, ..., d_n$, se emplean un conjunto de términos $T = t_1, ..., t_m$, ya sean simples o compuestos, con pesos $\{w_1, \ldots, w_m\}$. Aunque tradicionalmente los pesos de los términos se determinan por un valor de ausencia/presencia del término en el documento (0/1, respectivamente), modelos más evolucionados han permitido establecer el peso del término en función de su frecuencia de aparición.

3.1. Modelos y Técnicas en Minería de Textos

La propia aportación al campo textual de las tecnología de minería de datos clásica nos lleva a la aplicación y extensión de los modelos y técnicas que podemos encontrar en la literatura. Los algoritmos de Minería de Datos que se emplean en la actualidad se pueden agrupar en dos modelos, el descriptivo y el predictivo. En la literatura, encontramos aplicaciones a texto que cubren ambos modelos y la mayoría de las técnicas.

Modelo descriptivo: informa acerca de la relación entre los datos y sus características. Las técnicas usadas serán de aprendizaje no supervisado: Clustering[13],[11],[15], correlación y asociaciones[13] y reglas de asociación [1],[8],[4].

Modelo predictivo: responde a preguntas formuladas acerca de los datos. Las técnicas predictivas de aprendizaje supervisado más utilizadas son: Clasificación [20] [18] (k-nn, clasificadores de bayes, reglas CN2), árboles de decisión [22](ID3, C4.5, CART).

4. Aplicaciones de minería de textos y lógica difusa: reglas de asociación difusas

De todas las técnicas comentadas anteriormente, una de las más utilizadas es la de la búsqueda de reglas de asociación, debido sobre todo a que la mayoría de las aplicaciones relacionadas con el procesamiento de textos implican relaciones entre elementos del texto tales como términos, frases, etc...

Esta técnica asume un modelo transaccional de base de datos en el que cada transacción dada por un conjunto de *items*, en el ámbito textual, se identifica con un documento dado por un conjunto de términos indexados, tal y como se ha definido la forma intermedia basada en indexación en la sección 3. Las reglas de asociación que se buscan en la base de datos son, por lo tanto, relaciones de presencia de un término ante la aparición de otro en un conjunto de transacciones, es decir, qué términos de la colección de documentos implican la aparición de otros términos en la misma colección.

Formalmente, si T es un conjunto de transacciones que contiene items de un conjunto de items I. Un regla de asociación se define como una relación de la forma $A \Rightarrow C$ tal que $A, C \subset I$, y $A \cap C = \emptyset$, donde A es el antecedente y C es el consecuente de la regla, siendo ambos subconjuntos de items de I. Para medir el interés y la precisión de la regla y así poder seleccionar el mejor subconjunto de reglas de todas las extraídas de un conjunto de transacciones, se definen el *soporte* y la *confianza*. El soporte es el porcentaje de transacciones en el que se cumple la regla, calculado como el porcentaje de transacciones donde la unión del antecedente y el consecuente aparece. La confianza se define como la cardinalidad relativa del consecuente respecto al antecedente, es decir, la probabilidad condicionada del consecuente respecto al antecedente.

En el caso de una colección de documentos $D = \{d_1, \ldots, d_n\}$, obtenemos una forma intermedia consistente en un conjunto de términos $I = \{t_1, \ldots, t_m\}$ formado por los términos indexados de todos los documentos de la colección, cuyos pesos asociados son $W = \{w_1, \ldots, w_m\}$. Puesto que estamos tratando esta representación en un entorno documental, llamamos a cada estructura representando a un documento *transacción de texto* [5].

Ahora bien, cuando intentamos modelar la relación entre estos términos teniendo en cuenta, además, un valor de peso del término diferente a ausencia/presencia, es decir, reflejar la capacidad de representación de ese término respecto al tema sobre el que está escrito el documento, debemos utilizar algún esquema de cálculo basado en la frecuencia de aparición del término en el documento [21]. Este tipo de esquemas, no sólo es recomendable, si no que además refleja la realidad, ya que no todos los términos tienen la misma importancia en un documento. De hecho, la relación de los términos con los documentos, entendida como un grado de pertenencia de un término a un documento, es intrínsicamente difusa, lo cuál nos permite utilizar esquemas de pesos en el intervalo [0, 1] que puedan apoyarse en la Teoría de Conjuntos Difusos [23]. Por lo tanto, podemos extender las transacciones de texto binarias a aquellas que pueden contener grados difusos de pertenencia de los términos (items) a los documentos. A estas nuevas estructuras las denominamos *transacciones difusas de texto*.

En esta misma línea de razonamiento, tenemos que considerar reglas de asociación que permitan manejar antecedentes y consecuentes con grados asociados diferentes a los binarios. Este precisamente es el marco de trabajo de las *reglas de asociación difusas*, denominadas así porque manejan una vaguedad implícita proveniente del origen de extracción de las propias reglas, esto es, un conjunto de transacciones difusas y/o porque también pueden tener asociado un grado difuso de cumplimiento de la regla. El significado de una regla de asociación difusa $A \Rightarrow C$ es, por tanto, el mismo que el de las reglas de asociación convencionales, es decir, el consecuente C aparecerá en toda transacción donde aparezca A. La única diferencia es que el conjunto de transacciones donde se cumple la regla es un conjunto de transacciones difusas, es decir, tanto A como C están en toda transacción difusa $\tilde{\tau} \subset T$ con

un cierto grado de pertenencia ($\tilde{\Gamma}_A$ y $\tilde{\Gamma}_C$, respectivamente). De este modo, podemos afirmar que una regla de asociación difusa $A \Rightarrow C$ ocurre con total precisión en T cuando $\tilde{\Gamma}_A \subseteq \tilde{\Gamma}_C$. No obstante, hay que tener en cuenta que en datos reales, las reglas perfectas y exactas de esta forma son difíciles de encontrar. A menudo se encuentran excepciones para la mayoría de las reglas, básicamente por la gran cantidad de datos con la que se debe tratar en aplicaciones de minería de datos, y en concreto de textos.

En la literatura podemos encontrar varios estudios y propuestas sobre reglas de asociación difusas [3],[6],[2]. De todas ellas, cabe destacar la propuesta de [6] por la modelización mediante evaluación de sentencias cuantificadas para llevar a cabo la generalización de las medidas de soporte y confianza en el ámbito difuso. Además, también se propone el uso de factores de confianza como alternativa al uso de la confianza, que presenta algunos elementos poco intuitivos en su uso al no tener en cuenta el soporte del consecuente de la regla. Una aplicación de este tipo de extensión difusa al refinamiento de consultas se puede consultar en [5]. La idea general es descubrir nuevos términos que se puedan añadir a las consultas realizadas a un sistema de recuperación de información para optimizar el conjunto de documentos más adecuado a las necesidades del usuario. Para ello, un conjunto reglas difusas del tipo $A \Rightarrow C$ con suficiente precisión se extraen de transacciones difusas de texto provenientes de un conjunto inicial de documentos. Partiendo de una consulta inicial con el término A se puede, por ejemplo, particularizar substituyendo A por C. Para que esta particularización se lleve a cabo, es necesario que la regla $C \Rightarrow A$ no tenga mucha precisión, ya que en ese caso A y C serían intercambiables por lo que el conjunto de documentos recuperados sería prácticamente el mismo.

5. Conclusión

Las grandes colecciones de documentos disponibles hoy en día gracias a las autopistas de la información presentan una estructura rica y de una gran complejidad inherente. Para descubrir las relaciones existentes en dichas estructuras, es necesario aplicar técnicas provenientes del descubrimiento de patrones en datos estructurados, pero utilizándolas de forma que puedan manejar información textual.

Ya no se trata de manejar únicamente datos estructurados almacenados generalmente en una base de datos relacional, si no de procesar datos textuales cuya falta de estructura y homogeneidad dificultan enormemente su procesamiento y explotación. Para ello, la creación de formas intermedias estructuradas es esencial para la automatización y el tratamiento computacional de los documentos.

Es en esta labor en la que la teoría de los conjuntos difusos desempeña un papel fundamental para captar la semántica de la representación de los documentos mediante un conjunto de términos contenidos en los mismos. Además, una de las aplicaciones más relevantes de esta teoría es precisamente la extensión de las reglas de asociación hacia las reglas de asociación difusas, que permiten manejar antecedentes y consecuentes con diferentes grados de pertenencia al conjunto de transacciones del que se extraen.

Como trabajo futuro, pretendemos realizar un estudio más detallado de las formas intermedias y desarrollar otras aplicaciones con reglas de asociación difusas a documentos.

Referencias

[1] R. Agrawal, T. Imielinski y A. Swami. Mining association rules between sets of items in large databases. In *Proc. of the 1993 ACM SIGMOD Conference*, pp. 207-216, 1993.

[2] W.H. Au, K.C.C. Chan. FARM: A data mining system for discovering fuzzy association rules. En *Proc. of IEEE International Conference on Fuzzy Systems*, Seúl, Corea, 1999.

[3] G. Chen, Q. Wei. Fuzzy association rules and the extended mining algorithms. *Information Sciences* 147(1-4), pp. 201-228, 2002.

[4] M. Delgado, M.J. Martín-Bautista, D. Sánchez, M.A. Vila. Mining Text Data: Special Features and Patterns. Pattern Detection and Discovery. Proceedings ESF Exploratory Workshop 2002, D.J. Hand, N. Adams and R. Bolton (Eds.), Lecture Notes in Artificial Intelligence Series 2447, pp. 140-153, Springer-Verlag, 2002.

[5] M. Delgado, M.J. Martín-Bautista, D. Sánchez, J.M. Serrano, M.A. Vila. Association Rule Extraction for Text Mining. *Lecture Notes in Artificial Intelligence (LNAI) 2522*, pp. 154-162, 2002.

[6] M. Delgado, N. Marín, D. Sánchez, M.A. Vila. Fuzzy Association Rules: General Model and Applications. *IEEE Transactions on Fuzzy Systems 11(2)*, 214-225, 2003.

[7] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth. The KDD Process for Extracting Useful Knowledge from Volumes of Data. *CACM 39(11)*: 27-34, 1996.

[8] R. Feldman, H. Hirsh. Mining associations in text in presence of background knowlege. In *Proc. 2nd*

Int. Conf. on Knowledge Discovery an Data Mining (KDD'96), pp. 343-346, 1996.

[9] R. Feldman, M. Fresko, Y. Kinar, Y. Lindell, O. Liphstat, M. Rajman, Y. Schler, O. Zamir. Text Mining at the Term Level. In Proc. of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98), Nantes, France, 1998.

[10] M.A. Hearst. Untangling Text Data Mining. In *Proceedings of fue 37th Annual Meeting of fue Association for Computational Linguistics*, University of Maryland, 1999.

[11] T. Hoffman. The Cluster-Abstraction Model: Unsupervised Learning of Topic Hierarchies from Text Data. *International Joint Conference of Artificial Intelligence (IJCAI'99)*, pp. 682-687. 1999.

[12] S. Iiritano, M. Ruffolo. Managing the Knowledge Contained in Electronic Document: Clustering Method for Text Mining. IEEE 2001.

[13] D. Landau, R. Feldman, O. Zamir. TextVis: An integrated visual environment for text mining. *Principles of Data Mining and Knowledge Discovery*, 1998.

[14] H. Mannila, H. Toivonen. Discovering generalized episodes using minimal occurrences. In *Proc. 2nd Int. Conf on Knowledge Discovery and Data Mining (KDD'96)*, pp. 146-151, 1996.

[15] D. Merkl, A.M. Tjoa. Data mining in large free text document archives. In *Proc. of the Int. Symposioum on Cooperative Database Systems for Advanced Applications (CODAS'96)*, pp. 69-76, Greece, 1996.

[16] J. Paralic, P. Bednar. Knowledge Discovery in Texts Supporting e-Democracy. In Proc. of the 6th IEEE International Conference on Intelligent Engineering Systems, Opatija, Croatia, pp. 327-332, 2002.

[17] J. Paralic. Knowledge Discovery in Databases. In *Proc. of the 2nd Internal Scientific Conference of the Faculty of Electrical Enginering and Informatics*, TU Kosice, pp. 33-34, 2001.

[18] S. Park, B. Zhang. Large Scale Unstructured Document Classification using Unlabeled Data and Syntactic Information. En *K.Y. Whang, J. Jeon, K. Shim, J. Srivatava (Eds.): PAKDD 2003, LNAI 2637*, pp. 88–99, Springer-Verlag, 2003.

[19] R. Rajman, R. Besançon. Text mining: Natural language techniques and text mining applications. In *Proc. of the 7th IFIP working conference on database semantics(DS-7)*. Chapmam & Hall, 1997.

[20] J.D.M. Rennie. ifile: An applicacion of machine learning to mail filtering. *Proc. of the KDD-2000 Workshop on Text Mining*, 2000.

[21] G. Salton, C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management 24(5)*, 513-523, 1988.

[22] L. Yi, B. Liu. Web Page Cleaning for Web Mining through Feature Weighting. *Proc. of Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, Mexico, 2003.

[23] L.A. Zadeh. Fuzzy sets. *Information and Control, 8*, 338-353, 1965.

UN MODELO DE RECOMENDACIÓN CON PERFILES DE USUARIO LINGÜÍSTICOS

L.G. Pérez, L. Martínez[0], F. Mata, M. Barranco
Dept. Informática
Universidad de Jaén
23700 - Jaén, Spain
email: lgonzaga@ujaen.es, martin@ujaen.es, fmata@ujaen.es,barranco@ujaen.es

Resumen

Hoy en día Internet se caracteriza por la gran cantidad de información que pone a disposición de los usuarios. Esta cantidad de información llega a ser abrumadora e incluso molesta. En algunos ambitos de internet, principalmente en el comercio electrónico intentan resolver el problema de la sobreinformación con el uso de sistemas de recomendación, un software que ayuda a los usuarios en la compra de productos a través de Internet. En este trabajo proponemos el diseño de un sistema de recomendación basado en contenido que usa perfiles de usuarios lingüísticos y que emplea medidas de similaridad y ranking difuso para realizar las recomendaciones.

Palabras Clave: sistemas de recomendación, ranking difuso, similaridad, perfiles de usuario, teoría de conjuntos difusos

1 Introducción

Entre otras muchas cosas Internet se caracteriza por el gran volumen de datos e información disponibles para los usuarios. No es difícil que un usuario se sienta abrumado ante la gran cantidad de información recibida sobre una consulta y tampoco es difícil que mucha de esta información no sea útil y no se corresponda con las verdaderas necesidades del usuario. En el área del comercio electrónico [1] este problema se ha agudizado debido a la gran cantidad de productos que pueden ser devueltos ante una consulta simple aún cuando el usuario esté buscando un producto en concreto. Dicho usuario puede sentirse frustrado e incluso molesto al recibir como respuesta a su consulta la recomendación de decenas de productos muchos de los cuales no satisfacen sus necesidades . Para evitar este problema, han surgido los sistemas de recomendación que permiten facilitar y personalizar las búsquedas de productos así como aumentar la fidelidad de los clientes y la calidad de las respuestas de los sistemas de comercio electrónico. [1].

Los sistemas de recomendación ayudan a los usuarios en los procesos de búsqueda en la red. Su propósito es recomendar los objetos o productos más adecuados, de un conjunto de éstos, de acuerdo con las necesidades y/o preferencias del usuario o del cliente. Desde este punto de vista, los sistemas de recomendación podrían ser vistos como un tipo de proceso de toma de decisiones, el cuál decide qué productos son los más adecuados para cada usuario.

Las principales estructuras de información que se emplean para la formación de recomendaciones son los perfiles de usuario y las representaciones de los objetos. El proposito de los perfiles de usuario es almacenar información relativa sobre los usuarios como pueden ser sus necesidades, gustos o áreas de interés y el proposito de las representaciones de los objetos es almacenar las descripciones de los productos que se pueden recomendar.

Los sistemas de recomendación actuales se clasifican atendiendo al tipo de información y fuentes que utilicen para sus recomendaciones, distinguiendose tres tipos:

1. *Los sistemas de filtrado basado en contenido* [13]: filtran y recomiendan los elementos emparejando los términos de la consulta con los términos utilizados en la representación de los elementos, ignorando datos de otros usuarios
2. *Los sistemas de filtrado colaborativo* [9]: utilizan

[0]Soportado parcialmente por los proyectos de Investigación TIC2002-03348

información sobre las preferencias o los gustos de los usuarios, obtenidas de forma explícita y/o implícita, para filtrar y recomendar objetos a un usuario dado, ignorando la representación de los objetos.

3. *Sistemas híbridos basados en contenido y colaborativos* [5]: esta nueva clase de sistemas de recomendación ha surgido entre los sistemas basados en contenido y los colaborativos y su objetivo es minimizar las desventajas de cada uno. Un modo habitual de hibridar ambas clases es construir un algoritmo de filtrado de dos niveles, donde en el primer nivel, utilizamos el algoritmo de filtrado basado en contenido para poder obtener un primer conjunto de elementos y después, usamos el segundo algoritmo, el algoritmo de filtrado colaborativo, para filtrar y recomendar los elementos de este conjunto.

El propósito de este trabajo es presentar un modelo de sistema de recomendación basado en contenido, apoyado en medidas de similaridad y en perfiles de usuario lingüísticos y que utilizará ranking difuso para realizar recomendaciones sobre la compra de productos. Nuestro modelo de recomendación ha sido diseñado para tratar problemas en donde necesitamos recomendar una opción (o varías) de compra de productos de una forma rápida dentro un conjunto de posibles alternativas. En muchos casos la información de partida que proporcionan los usuarios es subjetiva y llena de imprecisión. El uso de la Aproximación Lingüística Difusa [19] ha dado buenos resultados al modelar este tipo de información mediante variables lingüísticas cuyos valores son términos lingüísticos en lugar de valores numéricos. Con lo cual los usuarios podrán expresar sus preferencias mediante etiquetas lingüísticas.

El modelo de sistema de recomendación basado en contenido propuesto en esta contribución seguirá el siguiente esquema de funcionamiento:

1. *Adquisición de perfiles de usuario:* Recogemos las preferencias del usuario y construimos su perfil de usuario utilizamos un vector de utilidades lingüísticas, $P_u = \{p_1^u, p_2^u, \ldots, p_l^u\}$, donde cada p_i^u representará una preferencia del atributo i del usuario u.
2. *Cálculo del producto más adecuado para cada usuario de acuerdo a sus preferencias mediante el uso de medidas de similaridad* [4] *con lo que obtendremos un conjunto difuso R_j^u que representará la similaridad.*
3. *Ordenación los productos:* para ello emplearemos un ranking difuso[8, 14] y devolveremos los n mejores.

En la sección 2 revisaremos las herramientas utilizadas para la construcción de nuestro modelo de recomendación y en la sección 3 desarrollaremos nuestro modelo. Finalmente en la sección 4 expondremos las conclusiones obtenidas en nuestro estudio.

2 Preliminares

En esta sección hacemos un breve repaso a las herramientas utilizadas en el desarrollo de nuestro modelo de recomendación. En primer lugar revisaremos la aproximación lingüística difusa. A continuación explicaremos las medidas de similaridad que utilizaremos en nuestro modelo y finalmente hablaremos del ranking difuso, el cual usaremos para realizar las recomendaciones.

2.1 Aproximación lingüística difusa

Muchos aspectos de las actividades del mundo real no pueden ser descritas de forma cuantitativa adecuadamente pero sí de forma cualitativa, por ejemplo, cuando tratamos con información vaga o imprecisa. En estos casos, es mejor emplear valoraciones lingüísticas en vez de numéricas. La Aproximación Lingüística Difusa representa los aspectos cualitativos con valores lingüísticos por medio de variables lingüísticas [18]. Esta aproximación ha sido aplicado con resultados satisfactorios en diferentes problemas.

La elección del conjunto de términos lingüísticos junto a su semántica es el primer punto que debemos realizar para resolver un problema desde el punto de vista del enfoque lingüístico. Uno de los aspectos más importantes que tenemos que analizar en este punto es la "*granularidad de la incertidumbre*", o lo que es lo mismo, el nivel de discriminación entre los distintos grados de incertidumbre. Los valores de cardinalidad mas utilizados son los impares, tales como el 7 y el 9, donde el término medio representa la valoración de "aproximadamente 0.5" y el resto de términos son colocados simétricamente alrededor de él [2]. Estos valores clásicos de cardinalidad parecen satisfacer las observaciones recogidas por Miller sobre el hecho de que los seres humanos pueden trabajar razonablemente teniendo en cuenta 7 o más objetos [12].

Una vez hemos establecido la cardinalidad del conjunto de términos, debemos proporcionar los términos lingüísticos y la semántica de los mismos. Existen diferentes posibilidades para realizar esta tarea [10]. Una de estas posibilidades implica proporcionar directamente el conjunto de términos considerando a todos los terminos distribuidos en una escala con un orden total definido [11]. Por ejemplo, un conjunto S de 7 términos, podría ser:

$$S = \{s_0 = Nada, s_1 = Muy\,bajo, s_2 = Bajo, s_3 = Medio, s_4 = Alto, s_5 = Muy\,Alto, s_6 = Perfecto\}$$

en donde $s_a < s_b$ si y solo si $a < b$. Normalmente, en estos casos, es necesario que los términos lingüísticos satisfagan las siguientes condiciones adicionales:

1. Debe de existir un operador de negación. Por ejemplo, $Neg(s_i) = s_j$, $j = g - i$ (donde $g + 1$ es la cardinalidad de S).
2. Un operador de maximización: $\max(s_i, s_j) = s_i$, si $s_i \geq s_j$.
3. Un operador de minimización: $\min(s_i, s_j) = s_i$, si $s_i \leq s_j$.

La semántica de los términos se presentan por medio de números difusos, definidos en el intervalo $[0, 1]$, descritos por funciones de pertenencia. Una forma de caracterizar un número difuso es utilizar una representación basada en parametros de su función de pertenencia [2]. Las valoraciones lingüísticas dadas a los usuarios son solo aproximaciones, y por lo tanto, las funciones de pertenencia trapezoidales lineales son lo suficientemente buenas para capturar la vaguedad de estas valoraciones lingüísticas, [6]. Esta representación se consigue mediante una 4-tupla (a, b, d, c), donde b y d indican el intervalo en el cual la función de pertenencia vale 1, con a y d indicamos los límites derecho e izquierdo del dominio de definición [2]. Un caso particular de esta representación es la función de pertenencia triangular, donde $b = d$, y que representaremos mediante una 3-tupla (a, b, c). Por ejemplo, podemos asignar la siguiente semántica al conjunto de siete términos anteriormente visto:

$N = s_0 = (0, 0, 0.17)$ $\quad MB = s_1 = (0, 0.17, 0.33)$
$B = s_2 = (0.17, 0.33, 0.5)$ $\quad M = s_3 = (0.33, 0.5, 0.67)$
$A = s_4 = (0.5, 0.67, 0.83)$ $\quad MA = s_5 = (0.67, 0.83, 1)$
$P = s_6 = (0.83, 1, 1)$

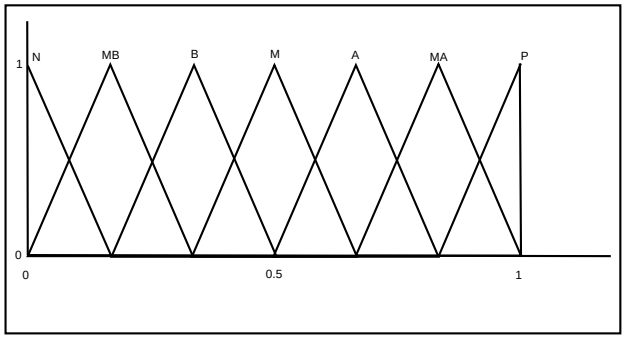


Figura 2. Representación gráfica de las funciones de pertencia asignadas al conjunto S.

2.2 Medidas de similaridad

La comparación de descripciones de objetos es una operación habitual en muchos dominios: psicología, analogía, ciencias físicas, procesado de imágenes, clustering, razonamiento deductivo, razonamiento basado en casos. Estas comparaciones frecuentemente se basan en medidas que intentan determinar que puntos tienen en común ambos objetos y en cuales difieren. Las medidas de comparación tiene varias formas dependiendo del propósito de su utilización. En [4] se consideran cuatro tipos de medidas de comparación:

1. Medidas de satisfiabilidad: La satisfiabilidad se corresponde a una situación en la cual consideraremos un objeto referencia y una clase y decidiremos si el nuevo objeto es compatible o satisface la referencia.
2. Medidas de semejanza: Se utilizan para realizar una comparación entre las descripciones de dos objetos del mismo nivel de generalidad para decidir si tienen muchas características comunes.
3. Medidas de inclusión: también se ocupa de situaciones en donde tenemos un objeto referecia, en este caso, medimos si los puntos en común entre A y B son importantes con respecto a A.
4. Medidas de disimilaridad: la disimilaridad entre objetos evalúa hasta que punto son diferentes.

Como hemos indicado anteriormente trabajaremos en un entorno difuso con valores imprecisos e inexactos. Las tres primeras medidas de comparación son medidas de similitud. En nuestro modelo de recomendación utilizaremos una medida de semejanza para medir la similaridad entre las preferencias de los clientes y las características de los productos a recomendar. La medida que utilizaremos es la siguiente:

$$D(A, B) = \sup_x \min(f_A(x), f_B(x))$$

Ya que cumple las propiedades definidas en [3] para ser medida de semejanza y es ampliamente utilizada en la literatura para llevar a cabo estos procesos de medida de similaridad.

2.3 Ranking difuso

En nuestro modelo de recomendación una vez se aplique una medida de similaridad a los distintos productos con respecto a las preferencias de los clientes

se obtendrán conjuntos difusos que nos indicarán el grado de similaridad

Para poder establecer un orden entre varios elementos definidos por conjuntos difusos debemos poder establecer relaciones entre ellos. Para establecer estas relaciones utilizaremos las siguientes definiciones [8, 14]:

Definición 1. *Dadas dos alternativas* **a** *y* **b**, *la sentencia "a es preferida a b" refleja la presencia de argumentos suficientes para apoyar la sentencia de que "***a** *es preferida a* **b***" pero no "***b** *es preferida a* **a***". La credibilidad de que* **a** *es preferida a* **b** *se define como:*

$$P(a,b) = \max\{S(a,b) - S(b,a), 0\}$$

Donde la función S representa el concepto de cuando **a** domina a **b**, ocurriendo esto solo si hay suficientes evidencias para creer que el concepto **a** es mejor que **b** o al menos **a** es tan bueno como **b**.

Definición 2. *Dadas dos alternativas* **a** *y* **b**, *la sentencia "***a** *y* **b** *son indiferentes" reflejan la presencia de argumentos suficientemente fuertes como para apoyar que "***a** *domina a* **b***" y "***b** *domina a* **a***". La credibilidad de que* **a** *y* **b** *son indiferentes se define:*

$$I(a,b) = \min\{S(a,b), S(b,a)\}$$

Definición 3. *Dadas dos alternativas* **a** *y* **b**, *la sentencia "***a** *y* **b** *son incomparables" refleja la ausencia de argumentos suficientemente fuertes para apoyar la afirmación de que "***a** *domina a* **b***" y "***b** *domina a* **a***". La credibilidad de que* **a** *y* **b** *son incomparables se define:*

$$J(a,b) = \min\{1 - S(a,b), 1 - S(b,a)\}$$

Gracias a estos tres índices podemos establecer un ranking de alternativas, de forma que:

- **a** es preferido a **b** $\Leftrightarrow P(a,b) > I(a,b)$ y $P(a,b) > J(a,b)$
- en otro caso, **a** es indiferente o incompatible con **b**.

Definición 4. *Dada dos alternativas* **a** *y* **b**, *la sentencia "***a** *domina a* **b***" significa que el decisor tiene suficientes razones para admitir que* **a** *es al menos tan buena como* **b**. *Una relación de dominación indica el grado de dominación, denotado por* $S(a,b)$, *asociado con cada par de alternativas* **a** *y* **b**, *donde* $S(a,b) \in [0,1]$.

Si $S(a,b) = 1$ sabemos que **a** domina a **b** con certeza. En el caso contrario, $S(a,b) = 0$ implica que no hay ninguna evidencia de que **b** sea dominada por **a**. Si $S(a,b) \in (0,1)$ entonces tenemos un índice de credibilidad de que **a** domina a **b**.

Nosotros emplearemos un enfoque de dominación difusa que es capaz de manejar las valoraciones representadas con conjuntos difusos de forma que podamos construir relaciones de dominación difusa entre las distintas alternativas. Para realizar esto nos basaremos en los trabajos de Dubois y Prade [7] que desarrollaron un conjunto de cuatro índices de dominancia, PG, PSG, NG y NSG dentro del marco de la teoría de la posibilidad de Zadeh [19].

Los cuatro índices pueden no obtener el mismo orden por lo que obligaría al decisor a realizar una última elección. Para solucionar este defecto del método de ranking y obtener una conclusión sin la necesidad de la colaboración de un decisor externo hemos empleado una función de agregación, el operador OWA (ordered weighted averaging) [17].

Para determinar el grado de dominancia entre dos conceptos necesitamos definir dos índices: *el índice de concordancia y el índice de discordancia.* Para dos conceptos **a** y **b**, el índice de discordancia, DI_k, expresa la credibilidad de la hipótesis de que **a** es al menos tan bueno como **b** con respecto a cierto criterio k. El índice de discordancia es utilizado para expresar dudas sobre la hipótesis de que **a** no es al menos tan bueno como **b** con respecto a algún criterio. El índice de concordancia, CI_k, se utiliza para construir el índice de concordancia global. Ambos índices, el índice de concordancia global GCIy el índice de discordancia los podemos encontrar definidos en [16].

Ahora, a partir de estos índices podemos construir el grado de dominancia $S(a,b)$[15]:

$$S(a,b) = \begin{cases} GCI(a,b) \ si\, GCI(a,b) \geq DI_k(a,b), \\ \quad \forall c_k \in C, \\ \prod_{k^*}[1 - DI_{k^*}(a,b)] \frac{GCI(a,b)}{1-GCI(a,b)} \ para \\ \quad \{k^* | GCI(a,b) < DI_{k^*}(a,b)\} \end{cases}$$

donde c_k es el criterio k-ésimo del conjunto de criterios C.

3 Modelo de recomendación

A continuación mostraremos en detalle el modelo de recomendación que proponemos en esta contribución. Debemos recordar que el contexto en el que trabajaremos es un modelo de sistema de recomendación basado en contenido con perfiles de usuario lingüísticos y donde la información sobre los objetos está almacenada mediante información lingüística o números difusos. Nuestro modelo lleva a cabo 3 fases para realizar las recomendaciones (figura 2):

1. Adquisición de preferencias, en donde construiremos el perfil de usuario.

2. Cálculo del producto más adecuado al usuario de acuerdo a sus preferencias mediante el uso de medidas de similaridad.

3. Ordenación de productos y recomendación de los n mejores teniendo en cuenta que la similaridad de cada producto con el perfil de usuario estará representada mediante un conjunto difuso.

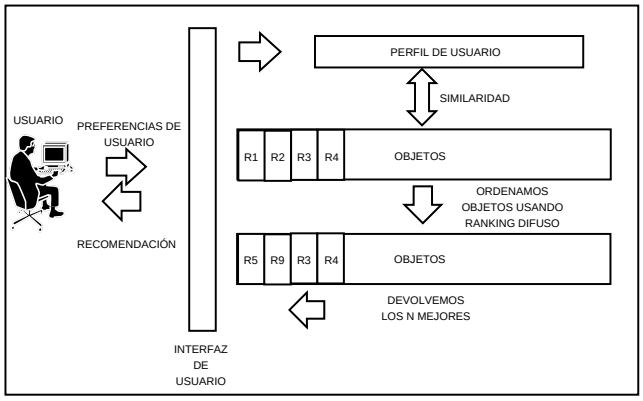


Figura 2. Funcionamiento del modelo de sistema de recomendación

En el diagrama anterior podemos ver el funcionamiento del modelo propuesto. A continuación plantearemos el problema a resolver y explicaremos más detenidamente cada una de estas fases.

3.1 Adquisición de los perfiles de usuario

Sea u el usuario que está interaccionando con el sistema y desea obtener una recomendación. Sea $C = \{c_1, \ldots, c_k, \ldots, c_l\}$ el conjunto de criterios/atributos que emplearemos a la hora de describir un objeto y la hora de describir las preferencias del usuario. Sea $A = \{a_1, \ldots, a_j, \ldots, a_n\}$ el conjunto de alternativas o productos que pueden ser recomendados por el sistema, donde $a_j = \left\{v_1^j, \ldots, v_k^j, \ldots, v_l^j\right\}$ y v_k^j será el valor que se le asigne al objetivo a_j en el criterio c_k. Este valor podrá ser una etiqueta y/o un número difuso que no tienen por qué coincidir con los utilizados en el perfil de usuario. Sea $P_u = \{p_1^u, \ldots, p_i^u, \ldots, p_l^u\}$ el perfil del usuario u donde p_i^u es el valor que le asigna el usuario u al criterio c_i. Este valor será una etiqueta cualquiera de las propuestas por el sistema para dicho criterio.

El objetivo de nuestro modelo es: Dado un conjunto de productos $A = \{a_1, \ldots, a_n\}$ y un perfil de usuario P_u queremos recomendar aquel(los) producto(s) a_j que más se asemeje(n) a las necesidades del usuario expresadas por medio de su perfil P_u.

En primer lugar obtendremos la información del perfil de usuario P_u del usuario u al que queremos realizar una recomendación. El usuario podrá expresar sus opiniones o preferencias o necesidades sobre el conjunto de criterios/atributos $C = \{c_1, \ldots, c_k, \ldots, c_l\}$ y al final de este proceso obtenremos su perfil $P_u = \{p_1^u, \ldots, p_i^u, \ldots, p_l^u\}$.

3.2 Cálculo del producto más adecuado.

Llegados a este punto, tenemos un perfil de usuario P_u con las preferencias del usuario descritas mediante etiquetas lingüísticas y un conjunto de productos $A = \{a_1, \ldots, a_n\}$ valorados por medio de números difusos y/o etiquetas lingüísticas en una serie de criterios/atributos $C = \{c_1, \ldots, c_k, \ldots, c_l\}$:

	c_1	$\ldots$	c_k	$\ldots$	c_l
a_1	$(v_1^1$	$, \ldots,$	v_k^1	$, \ldots,$	$v_l^1)$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
a_j	$(v_1^j$	$, \ldots,$	v_k^j	$, \ldots,$	$v_l^j)$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
a_n	$(v_1^n$	$, \ldots,$	v_k^n	$, \ldots,$	$v_k^n)$

Tabla 1. Descripción del conjunto de productos A.

Para obtener una medida de cual es el producto más adecuado usaremos una medida de similaridad [4], mas exactamente las de semejanza, entre el perfil de usuario P_u y cada alternativa a_i, $i = 1, \ldots, l$. De este cálculo obtendremos un conjunto difuso $R_i^u = \left(r_1^i, \ldots, r_k^i, \ldots, r_l^i\right)$ en donde cada componente r_k^i será un grado de semejanza que mide la similaridad (semejanza) entre las etiquetas que describe la valoración del usuario del criterio c_k, p_i^u , y la etiqueta o número difuso que describe el valor del criterio c_k en la alternativa a_i, v_k^j. La medida de similaridad que hemos empleado ha sido:

$$D\left(A, B\right) = \sup_x \min\left(f_A\left(x\right), f_B\left(x\right)\right)$$

Así, para calcular el grado grado de similaridad R_j^u de la alternativa a_i con respecto al perfil de usuario P_u deberemos realizar lo siguiente:

$$\begin{aligned} R_j^u &= D\left(P_u, a_j\right) = \left(r_1^i, \ldots, r_l^i\right) = \\ &\left(\sup_x \min\left(p_1^u, v_1^j\right), \ldots, \sup_x \min\left(p_l^u, v_l^j\right)\right) \end{aligned}$$

3.3 Ordenamos los productos y devolvemos los n mejores.

Una vez calculada la similaridad de cada producto a_j con el perfil de usuario P_u queremos devolver aquella alternativa más cercana a las necesidades del usuario.

El proceso que tenemos que seguir en este punto para devolver los n mejores es encontrar algún método de ordenación de forma con si $R_i^u \leq R_j^u$ quiere decir que la alternativa a_j satisface mejor las necesidades del usuario expresadas mediante el perfil P_u que la alternativa a_i. Para esta tarea emplearemos el método de ranking difuso expuesto en la sección 2.4.

4 Trabajos futuros

En el futuro nos proponemos mejorar este modelo de recomendación desde 2 puntos de vista:

- Incorporación de diferentes modelos de preferencias de distinta naturaleza (cuantitativa y cualitativa) y de distintos tipos de estructuras (ordenes, relaciones de preferencias,...).
- Estudiar el comportamiento del modelo cuando empleamos otros mecanismos de ranking difuso.

Referencias

[1] A. Ansari, S. Essegaier, and R. Kohli. Internet recommendation systems. *Journal of Marketing Research*, XXXVII:363–375, August 2000.

[2] P. Bonissone and K. Decker. *Selecting Uncertanty Calculi and Granularity: An Experiment in Trading-Off Precision and Complexity*. North-Holland, 1986.

[3] B. Bouchon-Meunier and M. Rifqi. Resemblance in database utilization. *6th IFSA World Congress*, 1995.

[4] B. Bouchon-Meunier, M. Rifqi, and S. Bothorel. Towards general measures of comparison of objects. *Fuzzy Sets and Systems*, (84):143–153, 1996.

[5] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining content-based and collaborative filters in an online newspaper. *Proceedings of the ACM SIGIR Workshop on Recommender Systems - Implementations and Evaluations*, 1999.

[6] M. Delgado, J. Verdegay, and M. Vila. Linguistic decision making model. *International Journal of Intelligent Systems*, (7):479–492, 1992.

[7] D. Dubois and H. Prade. Ranking fuzzy numbers in the setting of possibility theory. *Information Science*, (30):183–224, 1983.

[8] J. Fodor and M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publisher, Dordrecht, 1994.

[9] D. Golberg, D. Nichols, B.M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[10] F. Herrera, E. Herrera-Viedma, and L. Martínez. A fusion approach for managing multi-granularity linguistic term sets in decision making. *Fuzzy Sets and Systems*, (114):43–58, 2000.

[11] F. Herrera, E. Herrera-Viedma, and J. Verdegay. A sequential selection process in group decision making with linguistic assessment. *Information Science*, (85):223–239, 1995.

[12] G. Miller. The magical number sever or minus two: Some limits on our capacity of processing information. *Psychological Review*, (63):81–97, 1956.

[13] M. Pazzani, J. Muramatsu, and D. Billsus. Syskill and webert: Identifying interesting web sites. *In Proceedings of the Thirteenth National Conference on Artificial Intelligence AAAI*, 96:54–61, 1996.

[14] B. Roy. *Partial Preference Analysis and Decision Aid: The Fuzzy Outranking Relation Concept*. O.E. Bell. R.L. Keeny, H. Raiffa, Chichester, New York, 1977.

[15] J.L. Siskos, J. Lochard, and J. Lombard. *A Multicriteria Decision Making Methodology under Fuzzyness: Applications to the Evaluation of Radiological Protection in Nuclear Power Plants*. H.J. Zimmermann, North-Holland, Amsterdam, 1984.

[16] J. Wang. Ranking engineering design concepts using a fuzzy outranking preference model. *Fuzzy Sets and Systems*, (119):161–170, 2001.

[17] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics*, (18):183–190, 1988.

[18] L.A. Zadeh. The cocept of a linguistic variable and its application to approximate reasoning. *Information Science*, (8 and 9):(Part I and II) 8, pp 199–249 and pp. 301–357, (Part III) 9, pp. 43–80, 1975.

[19] L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, (1):3–28, 1978.

APLICACIONES DE LA LÓGICA BORROSA EN SISTEMAS DE VIGILANCIA UTILIZANDO VISIÓN ACTIVA [1]

J. M. Molina, J. García
Dpto. Informática
Avda. Unv. Carlos III, 22
28270, Colmenarejo, Madrid
molina@ia.uc3m.es

J. Carbó
Dpto. Informática
Avda. Univ. Carlos III, 22
28270, Colmenarejo, Madrid
jgherrer@inf.uc3m.es

A. Berlanga
Dpto. Informática
Avda. Univ. 30
28390, Leganés, Madrid
jcarbo@inf.uc3m.es

J. Portillo
ETSI Telecomunicación
Ciudad Universitaria s/n
28040, Madrid
javierp@gprss.ssr.upm.es

Resumen

En este trabajo se presentan avances en la aplicación de la lógica borrosa en sistemas de seguimiento utilizando visión activa. A lo largo del artículo se referencias diversos trabajos llevados a cabo en la aplicación de la lógica borrosa en el terreno de la asociación para el seguimiento de blancos mediante cámaras. En particular se detallará la aplicación de un sistema borroso para mejorar el proceso de seguimiento.

Palabras Clave: Seguimiento, Asociación, Vigilancia, Visión Activa.

1 INTRODUCCIÓN

El conjunto de nuevas tecnologías y procedimientos de soporte a la gestión de tráfico en operaciones de aeropuerto constituyen el concepto A-SMGCS (Advanced Surface Movement, Guidance and Control Systems).

En general, todos los sistemas, operativos o no, presentan como principal debilidad el depender excesivamente de una tecnología de detección concreta, sin cubrir satisfactoriamente los requisitos de continuidad y cobertura, y con un nivel de automatización insuficiente. Las tecnologías de detección convencionales utilizadas en este entorno se agrupan en no cooperativas (radar de superficie, bucles magnéticos, sensores de aparcamiento, etc.), con capacidad de detectar blancos sin necesidad de cooperación, y sensores de tipo cooperativo (redes de triangulación, radar secundario, ADS-B con GPS diferencial, etc.), que proporcionan identificación de los blancos detectados pero requieren que éstos estén equipados con un sistema de comunicaciones radio y/o alguna facilidad de navegación. De esta manera, aparece con frecuencia un problema de falta de cobertura en algunas áreas, que podría ser paliado con nuevos sensores que complementen las deficiencias de los ya existentes. Este problema se sufre especialmente en aeropuertos de gran tamaño.

Existen sistemas ópticos experimentales desarrollados en EE.UU. que ponen de manifiesto la aplicación de este tipo de sensor en el área de aeropuerto. La compañía Tri-Space Inc. dispone de un sistema de seguimiento óptico (IRAS, Infrared Augmentation System) (TriSpace) mediante cámaras infrarrojas con el fin de alertar de incursiones en pistas (otras aeronaves, vehículos, personas, animales) o detectar violaciones en los procedimientos de rodadura. Esta misma compañía ha desarrollado el sistema ALIS (Airport Landing Information System), para dar información a los pequeños y medianos aeropuertos de la actividad de tráfico que registran. El sistema ALIS realiza el reconocimiento de las aeronaves, en tráfico de entrada y salida, obteniendo la matrícula (tail number) mediante el procesado OCR de las imágenes capturadas con video digital. En Europa el DLR (Deutschen Zentrum für Luft- und Raumfahrt) ha desarrollado el prototipo ARMI (Aircraft Registration Mark Identification) (DLR), un sistema de identificación de características similares al anterior, destinado a integrarse en el sistema TARMAC.

La posibilidad de aplicar este tipo de tecnología en los aeropuertos españoles ha sido analizada por AENA (Aeropuertos Españoles y Navegación Aérea) en el estudio de viabilidad (MARIA: "Mecanismo Automatizado de Reconocimiento de Imágenes de Aeronaves en Superficie de Aeropuerto") realizado conjuntamente por los grupos de Procesado de Datos y Simulación (http://www.grpss.ssr.upm.es) y el de Inteligencia Artificial Aplicada (http://www.giaa.inf.uc3m.es). En este estudio se analizó un sistema compuesto de dos

[1] Financiado por los proyectos CICYT (TIC2002-04491-C02-02) y CAM (07T/0034/2003 1)

subsistemas, uno centrado en el seguimiento y otro en la identificación [15]. El subsistema de seguimiento se definió con una arquitectura distribuida basada en seguimiento local y se probaron algoritmos de seguimiento basados en técnicas basadas en medidas no extensas (plots) [7]. En el subsistema de identificación se analizó la capacidad para el reconocimiento de matrículas utilizando un clasificador bayesiano tradicional a partir de patrones ideales.

Las cámaras presentan algunas particularidades frente a los sistemas tradicionales pero en general el conjunto de técnicas utilizadas para las funciones de vigilancia (seguimiento e identificación) son directamente aplicables. En este trabajo presentamos una aproximación para los problemas de asociación en las tareas de seguimiento que aparecen al fragmentarse los blancos en las imágenes. La aproximación tradicional es asignar cada fragmento a un blanco, la aproximación borrosa permite pesar la asignación de manera que cada fragmento se asigne parcialmente a varios blancos. Otra de las aplicaciones de la lógica borrosa es la gestión cooperativa de los sensores ([11, 12, 14, 17, 18] mediante técnicas de agentes [16].

2 EL PROBLEMA DE LA ASOCIACIÓN EN SEGUIMIENTO

Los resultados del proyecto MARIA permiten concluir la viabilidad técnica de un sistema de vigilancia basado en cámaras, tanto para seguimiento como para identificación [7]. Tomando como base este trabajo se han analizado sistemas basados en el conocimiento utilizando lógica borrosa para resolver el problema de la asociación en las tareas de seguimiento. Puede decirse que uno de los problemas más importantes que debe resolver el seguimiento local, aparte de la detección correcta de las regiones de la imagen correspondientes a blancos reales, es el de la correspondencia entre las regiones extraídas de las imágenes y los blancos, puesto que se dan frecuentemente una serie de anomalías en los escenarios de interés, entre las que destacan las siguientes:

• Con mucha frecuencia el proceso de segmentación de los blancos no es perfecto, sino que su imagen se fragmenta en una serie de regiones inconexas. Esto puede darse de forma aleatoria, especialmente cuando los blancos son compuestos (como los convoyes de vehículos de superficie, o ante la presencia de sombras irregulares generadas por los propios blancos), o de forma sistemática si existen obstáculos u otros blancos situados delante del blanco de interés.
• Existen frecuentes ocultaciones totales o parciales de los blancos de interés, ya sean debidos a elementos fijos incluidos en el fondo o a blancos que se aproximan hasta que el más próximo a la cámara oculta al más lejano.
• Cuando las trayectorias de blancos diferentes se aproximan a distancias muy cortas, o por efecto de la perspectiva la imagen de un blanco se superpone parcial o totalmente a la de un blanco moviéndose detrás, la lógica de segmentación tiende a mezclar las regiones de sus imágenes, no siendo posible extraer cada blanco por separado. Esta situación se da frecuentemente en áreas del aeropuerto con alta densidad de tráfico, como son las áreas de aparcamiento, con presencia de aeronaves y múltiples vehículos de superficie moviéndose a baja velocidad y distancias próximas.

Por tanto, la correspondencia adecuada entre las regiones extraídas de cada imagen con los blancos reales será la base que permita la estimación correcta del número de blancos reales en el escenario y de sus trayectorias. A la vista de los efectos mencionados, se trata de un problema complejo de asociación de datos entre múltiples blobs y múltiples pistas, donde varios blobs pueden potencialmente ser asignados a cada pista (debido a la división de la imagen de un blanco en subregiones), y al mismo tiempo las pistas pueden compartir blobs en situaciones de solapamiento. Este problema no se puede tratar directamente con las técnicas de seguimiento que generalmente se aplican a sensores convencionales [20], [2], [22]. Esto se debe a que estos algoritmos realizan un seguimiento puntual y modelan cada sensor como una fuente que proporciona una única medida por blanco, con una distribución de errores centradas en un punto. Sin embargo, el problema real presenta una distribución espacial de cada blanco bajo seguimiento, que en general es muy superior a la resolución del sensor (tamaño del píxel).

Así, existe una amplia variedad de algoritmos de asociación en la bibliografía de procesado de datos, aplicables en función de la complejidad del problema. Las soluciones más simples, los filtros de vecino más próximo (Nearest Neighbour, NN) [20], determinan la solución de mínima distancia estadística entre pistas y medidas en cada instante de procesado, y actualizan la información de las pistas suponiendo que estas decisiones son correctas. Estos algoritmos son vulnerables a la presencia de blancos próximos o de falsas alarmas, y para mejorar la continuidad de los sistemas ante estas situaciones, se han desarrollado las técnicas de seguimiento con múltiples hipótesis (Multiple Hypothesis Tracking, MHT[20]). Estas técnicas evalúan la probabilidad de que diferentes secuencias de medidas se asignen a cada pista, incrementando notablemente la robustez ante la presencia de errores de asociación en instantes puntuales, pero incrementando fuertemente los requisitos de memoria y carga computacional. Las técnicas de asociación multi-scan se han propuesto y estudiado recientemente [21] como una implementación eficiente de un sistema MHT, que evitan la enumeración exhaustiva de hipótesis. En todo caso, estos algoritmos explotan las restricciones de correspondencia uno a uno entre pistas y medidas, lo que

deja de ser válido cuando un blanco ocupa una extensión espacial y proporciona potencialmente más de una medida por iluminación, apareciendo los problemas mencionados antes. Recientemente, y ante problemas similares, algunos autores proponen una extensión que facilite una asignación de cada pista con múltiples blobs, mediante una secuencia iterada de asignaciones uno a uno, aplicando técnicas convencionales. Este procedimiento puede requerir excesivos recursos de computación e impedir la operación en tiempo real del sistema con un hardware moderado (el sistema final debería procesar secuencias de imágenes, con alta densidad de blancos, en el orden de hasta 30 imágenes por segundo). Además, no resuelve el problema de la asignación de blobs incorrectos, que engloban partes de la imagen procedentes de más de un blanco.

Una posible alternativa a estas técnicas para resolver la asociación de forma eficiente puede ser un sistema de decisión suave, que considere todas las detecciones en el entorno de cada pista sin tomar decisiones de asignación excluyentes, y evalúe la verosimilitud de que la información realmente represente a cada blanco bajo seguimiento. Es una aproximación similar a los sistemas de asociación probabilística (Joint Probabilistic Data Association, JPDA [20], que decicen la asignación ponderando todas las medidas posibles para construir una pseudo-medida que actualice las pistas. Sin embargo, los coeficientes de ponderación con un sistema JPDA se calculan en función del residuo estadístico entre pistas y medidas, lo que de nuevo supone una simplificación del problema real con blancos puntuales y distribuciones de error mono-modales.

Una aproximación más adecuada para este problema será utilizar de forma explícita en la asociación la representación espacial de los blancos bajo seguimiento, incluyendo parámetros que describan las propiedades de forma (contorno) o patrones bidimensionales en la imagen junto a la trayectoria descrita. El conocimiento de los contornos predichos para cada uno de los blancos en seguimiento permitirá facilitar la asociación de datos, acotando las regiones de búsqueda iniciales en cada imagen y anticipar algunos de los aspectos mencionados anteriormente, como son las oclusiones o solapamientos entre múltiples blancos para resolver el problema de la asociación. Por tanto, se van a abordar técnicas de asociación directamente relacionadas con los procedimientos de estimación del contorno de los blancos existentes. Como se ha mencionado, la evolución adaptativa de la forma estimada del blanco es uno de los aspectos críticos de estos métodos no basados en modelos, siendo imprescindible llegar a un compromiso aceptable entre los aspectos siguientes para alcanzar los requisitos de continuidad y robustez deseables del sistema:

• El contorno servirá de base para reagrupar las diferentes regiones en las que se pueda dividir un blanco, evitando multiplicidad de pistas asociadas a regiones diferentes. Este contorno deberá adaptarse a la evolución del aspecto bidimensional de cada blanco proyectado en el plano de la cámara, cuya forma y tamaño varía con la distancia y el ángulo de aspecto.
• El reagrupamiento debe estar limitado para no mezclar regiones correspondientes a elementos diferentes de las pistas, llegando a distorsiones de la forma estimada o a discontinuidades y pérdidas.

La estimación dinámica del contorno por tanto debe tener en cuenta la información extraída de cada imagen y así mismo información de alto nivel generada por el propio proceso de seguimiento (presencia de otros blancos en la imagen y sus formas predichas). Esto permitirá razonar sobre la situación a resolver e interpretar mejor la información disponible. Las técnicas bayesianas (filtros de Kalman o variantes similares) de combinación de la información observada y predicha para actualizar los parámetros de contorno, como las mencionadas en [3], proporcionan una solución precisa y robusta cuando se trata de correlar atributos de bajo nivel extraídos de la imagen con el modelo (bordes, color, textura, etc.), en situaciones de un solo blanco a segmentar de la imagen. Sin embargo, son soluciones rígidas que no pueden modelar de forma analítica el conocimiento de alto nivel disponible cuando se presentan situaciones de alta complejidad, como los solapamientos u oclusiones entre blancos distintos. Es muy difícil plantear el compromiso de reagrupamiento y distinción de regiones de pistas distintas mencionado anteriormente con un planteamiento puramente estadístico, tal y como se propone en un problema similar pero con un sensor de tipo radar, y en [9] se propone utilizar heurísticos, calculados con análisis geométricos sencillos, como información adicional para resolver la situación. Estos heurísticos indican algunas características de los problemas a resolver, como pueden ser el grado de densidad de grupos de regiones potencialmente representando un mismo blanco, el solapamiento entre regiones observadas y contornos predichos, o la presencia de conflictos de oclusión entre imágenes correspondientes a blancos distintos. Allí se propone una metodología de seguimiento condicionada a estos valores para tomar las decisiones de agrupamiento y actualización que mejor se ajusten el compromiso mencionado.

Esta opción de seguimiento de las trayectorias y contornos condicionado a reglas que dependan de información de más alto nivel para cumplir unas prestaciones adecuadas abre importantes líneas a explorar. La aplicación flexible de diferentes niveles de conocimiento (atributos de bajo nivel, información de seguimiento, heurísticos) para generar un comportamiento satisfactorio en términos de unas métricas de prestaciones puede ser abordada con técnicas de Inteligencia Artificial. Así, en [9] se propone la utilización de conjuntos borrosos para representar una serie de heurísticos básicos, junto a un sistema de reglas que indica la confianza asignada a la información extraída

de cada imagen, para actualizar los parámetros estimados de cada blanco. Estas reglas resumen el conocimiento generado del análisis de prestaciones en una serie de situaciones representativas, sirviendo el sistema borroso para generalizar y extender el razonamiento a todos los casos intermedios que puedan aparecer. Puede ser interesante explorar la aplicación de esta metodología a las técnicas de seguimiento con regiones y contornos, explorando la representación, atributos y heurísticos que lleguen a comportamientos más adecuados a las condiciones de este problema. Esta metodología puede extenderse incluso para incorporar información de nivel superior al plantear el sistema distribuido final. En ese caso, la información de más alto nivel generada en el nodo de fusión de pistas locales puede realimentarse y tenerse en cuenta en los algoritmos de asociación de cada cámara. Además es posible aplicar técnicas de aprendizaje para la extracción automática de las reglas a partir de ejemplos como puede verse en [10] y [13].

La aplicación de sistemas basados en el conocimiento [4] para integrar información de distinta naturaleza y grados de incertidumbre en el análisis de imágenes ha sido explotado con anterioridad. En [19] se indican sistemas representativos que utilizan diferentes paradigmas de conocimiento (redes bayesianas, razonamiento con evidencias, etc.) para interpretar a alto nivel la información contenida en imágenes en distintas aplicaciones (escenas urbanas, reconocimiento de blancos en aeropuerto, identificación de objetos 3D, etc.). Más específicamente, la lógica borrosa [6] se ha aplicado con éxito a problemas de procesado de imágenes, modelando la incertidumbre o ambigüedad que puede aparecer a diferentes niveles. En [8] se presentan un desarrollo extenso de estas técnicas. En el caso de la aplicación a problemas de asociación para fusión de datos, en [1], se proponen sistemas borrosos para resolver problemas de asociación en condiciones de baja observabilidad y alta densidad de blancos en aplicaciones de vigilancia naval. Dentro de la metodología de seguimiento de vídeo basado en atributos, en [5] se utilizan grafos relacionales borrosos para abordar el problema del agrupamiento de atributos y su coherencia entre imágenes sucesivas.

3 SISTEMA BORROSO DESARROLLADO

En el sistema desarrollado las pistas se representan mediante la posición y la velocidad y se complementan con atributos que definen la extension y la forma del blanco. El contorno predicho del blanco es lo que permite enventanar los blobs del siguiente frame. Por simplicidad se considera una caja rectangular para representar el blanco como se puede observer en la Figura 1. Sobre la posición predicha, $(\hat{x}_p, \hat{y}_p)$, se define la ventana rectangular, (x_{min}, x_{max}, y_{min}, y_{max}), con el tamaño estimado $(\hat{l}_H, \hat{l}_V)$. Además se considera una ventana externa con parámetros Δ_H, Δ_V, que se usa para realizar el enventanado final para seleccionar los blobs que actualizan la pista.

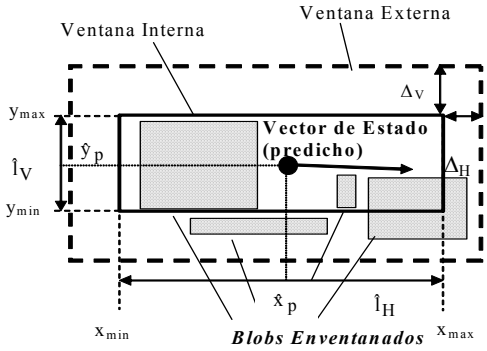


Figura 1. Segmentación del blanco y enventanado

El peso de cada blob para la actualización se calcula mediante un sistema borroso que tiene como entradas:

1. Solapamiento: este componente es una especie de enventanado blando que calcula la fracción de blob que está dentro del área de la ventana.

2. Densidad de Grupo y Distancia a la pista: este heurístico, ρ (Eq. 1), evalua la realción entre las areas de las regions detectadas y los agujeros en la representación final al reconectar los blobs.

$$\rho = \sum_i \text{Area}_i / (l_x l_y) \qquad (1)$$

3. Conflicto con otras pistas: esta componente evalúa la pertenencia de un blob que se encuentra en conflicto con otras pistas. Este problema aparece cuando tenemos varias trayectorias que están cercanas y sus ventanas se solapan permitiendo que los blobs puedan pertenecer simultáneamente a varias pistas. Este heurístico es como el primero donde cuando hay varias pistas se elige el máximo grado de solapamiento.

4. Proximidad a los bordes de la imagen: los bordes de una imagen es donde normalmente se inicializan las pistas por lo que suelen ser zonas con pistas no estables. El heurístico evalúa lo cerca que está el blob del borde.

Estos heurísticos representan la confianza en cada uno de los blobs, se definen también un conjunto de heurísticos para representar la confianza en la pista, si se desvía más o menos de la trayectoria real del blanco:

5. ***Número de actualizaciones perdidas:*** representa el número de frames consecutivos donde ningún blob ha sido incluido en la ventana interna de la pista.

6. ***Area detectada de seguimiento:*** del mismo modo que el heurístico de solapamiento de los blobs este heurístico da la proporción entre el área de la ventana interna y el area llena con blobs detectados.

7. ***Proximidad a los bordes de la imagen:*** este valor es equivalente al de los blobs.

El sistema borroso calcula el nivel de confianza (*correlation*) para los blobs y las pistas. A modo de ejemplo se presentan las reglas ad-hoc utilizadas:

overlap is small => correlation is small
density is small => correlation is small
conflict is large => correlation is small
conflict is medium => correlation is small
(overlap is medium) & (density is medium) & (conflict is small) => correlation is medium
(overlap is medium) & (density is large) & (conflict is small) => correlation is medium
(overlap is large) & (density is medium) & (conflict is small) => correlation is medium
(overlap is large) & (density is large) & (conflict is small) => correlation is large
border is large => correlation is large

Una vez obtenidos los valores de los blobs y las pistas, se actualiza la forma de la pista mediante la confianza de los blobs (varía muy poco) y la actualización de la posición se realiza utilizando los niveles de confianza de los blobs y de las pistas.

5 RESULTADOS

En este apartado se presentan los resultados de precisión en un escenario (de una batería de trayectorias estraídas del aeropuerto de Madrid-Barajas) comparando el sistema borroso frente a un sistema que toma decisiones *hard*. La decision *hard* utiliza dos reglas:

- Actualizar con todos los blobs incluidos en la ventana si la densidad del grupo es mayor de 0.7, en caso contrario se eliminan los blobs más lejanos.
- Si dos o más pistas comparten blobs se predicen sin actualización.

La precisión se evalúa como la raíz cuadrada del error (RMS) en la estimación de cada coordenada. Al no poder generar trayectorias de referencia se ha seleccionado una trayectoria lineal para estimar razonablemente una aproximación mediante mínimos cuadrados para utilizarla como referencia de los estimadores. La trayectoria es un avión que se mueve de derecha a izquierda. Los errores se normalizan según el tamaño del blanco sacado de los blobs extraídos de las imágenes, por lo que se expresan en fracciones del tamaño del blanco en cada coordenada (%). En este caso se producen oclusiones parciales debido al cruce con un autobús y un avión parado en frente. La figura 2 muestra las trayectorias estimadas en X y en Y. Mediante círculos aparece el resultado del sistema borros, mediante triángulos el sistema *hard* y la línea punteada representa la trayectoria aproximada mediante LS. El error normalizado aparece en las figuras 3 y 4 comparando los dos sistemas y el ideal. Los valores medios de RMS también se representan, y se puede observar que el sistema borroso presenta una mejora del 35% en la precisión vertical debido a su mayor estabilidad.

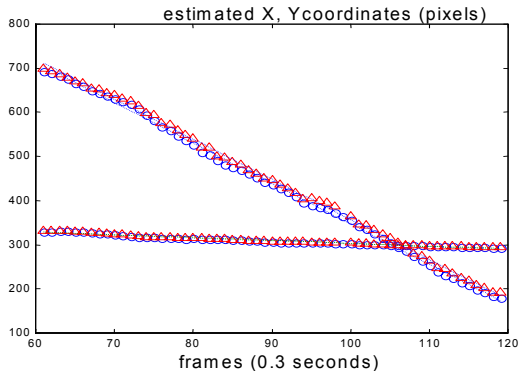


Figura 2. Trayectorias Estimadas y Aproximadas

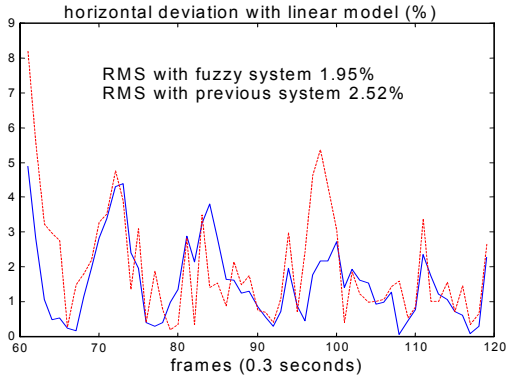


Figura 3. Error Horizontal

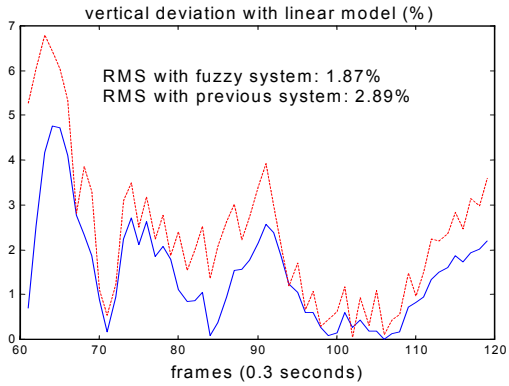


Figura 4. Error vertical.

6 CONCLUSIONES

En este trabajo se han presentado sistemas basados en lógica borrosa aplicados en los sistemas de vigilancia, tanto a bajo nivel, en el sistema de seguimiento, como a más alto nivel, en el sistema de gestión del conjunto de cámaras desplegadas. La aplicación de estos sistemas explota la capacidad de interpolación de los sistemas borrosos junto con la representación explícita del conocimiento.

Referencias

[1] C. Helleur. Fuzzy Clustering for Data Fusion in a Recognized Maritime Picture. FUSION 2001.

Montreal, Canada. August 2001.

[2] D. L. Hall, J. Llinas. *Handbook of multisensor data fusion*. Boca Raton CRC Press. 2001.

[3] F. Dellaert, C. Thorpe. Robust car tracking using Kalman filtering and Bayesian templates. Proc. of SPIE. Vol. 3207. Intelligent Transportation Systems. Pittsburgh. Oct. 1997.

[4] G. Guida, C. Tasso. *Design and Development of Knowledge-Based Systems*. John Wiley&Sons. 1994.

[5] H. Borotschnig, A. Pinz, D. Sinclair. Fuzzy Graph Tracking 5th International Symposium on *Intelligent Robotic Systems*, KTH Stockholm. 1997

[6] H.J. Zimmermann. *Fuzzy Set Theory and its Applications*. Kluwer Academic Publishers, 2nd rev. edition, 1991

[7] J. Besada, J. García, A. Varona, G. González, J. M. Molina, J. Portillo. Image-Based Automatic Surveillance For Airport Surface, 4th International Conference on Information Fusion, Fusion 2001, pp (WeA1) 11-18. Montreal, Canadá, Agosto 2001.

[8] J. C. Bezdek. *Fuzzy models and algorithms for pattern recognition and image processing* Kluwer Academic, Boston 1999. Handbooks of fuzzy sets series

[9] J. García, J. A. Besada, J. M. Molina, J. I. Portillo, G. de Miguel. Fuzzy data association for image-based tracking in dense scenarios, IEEE International Conference on Fuzzy Systems dentro del IEEE World Congress on Computational Intelligence (IJCNN, FUZZ-IEEE, ICEC). pp 902-907. Hawai. Mayo, 2002.

[10] J. García, J. M. Molina, J. A. Besada, J. Portillo. Fuzzy Approach for Data Association in Image Tracking, *Mathware and SoftComputing*. vol X, nº 2-3 pp 117 – 129, 2003 .

[11] J. M. Molina, F. J. Jiménez, J. R. Casar. Negotiation in a MultiAgent System of Surveillance Sensors based on Fuzzy Priorities, 4th International Conference on Information Fusion, Fusion 2001, pp (ThB2) 25-32. Montreal, Canadá, Agosto 2001.

[12] J. M. Molina, J. García, F.J. Jiménez, J.R. Casar. Fuzzy Reasoning in a MultiAgent System of Surveillance Sensors to Manage Cooperatively the Sensor-to-Task Assignment Problem, *Applied Artificial Intelligence*. (in press)

[13] J. M. Molina, J. García, J. de Diego, J. I. Portillo. Neuro-Fuzzy Techniques for Image Tracking, Artificial Neural Nets Problem Solving Methods. Ed. J. Mira y J. R. Álvarez. Lecture Notes in Computer Science 2687, pp 504-511, Springer-Verlag, 2003.

[14] J. Manyika, Durrant-Whyte H., Data Fusion and Sensor Management a decentralized information-theoretic approach, Ellis Horwood, 1994.

[15] J. Portillo, J. Besada, J. García, J. M. Molina, A. Varona, G. González. MARIA: Preliminary Results of an AENA R&D Project on Airport Surface Surveillance Based on CCTV, International Conference on Airport Surveillance Sensors, pp 138-145. París, Francia, Diciembre 2001.

[16] J.M. Corchado, J.M. Molina. *Introducción a la Teoría de Agentes y Sistemas Multiagente*. Publicaciones Científicas, Editorial Catedral Publicaciones. Salamanca, 2002, España.

[17] J.M. Molina, J. García, F.J. Jiménez, J.R. Casar, Cooperative Management of a Net of Intelligent Surveillance Agent Sensors,. International Journal of Intelligent Systems. vol 18, nº 3 pp 279 – 307, 2003.

[18] J.M. Molina, J. García, F.J. Jiménez, J.R. Casar, Surveillance Multisensor Management with Fuzzy Evaluation of Sensor Task Priorities. Engineering Applications of Artificial Intelligence. vol 15 nº6 pp 511 – 527, Diciembre 2002.

[19] R.Haralick D. Dori. A Pattern Recognition Approach to Detection of Complex Edges, *Pattern Recognition Letters*, May, 1995, Vol. 16, No. 5, pp. 517-529. 1995

[20] S. Blackman, R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House. 1999

[21] S. Deb, M. Yeddanapudi, K. Pattipati, Y. Bar-Shalom. A Generalized S-D Assignment Algorithm for Multisensor-Multitarget State Estimation. *IEEE Transactions on Aerospace and Electronic Systems*. Vol. 33, Nº 2. April 1997. pp. 523-536.

[22] Y. Bar-Shalom, X. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. Editado por Yaakov Bar-Shalom. 1995.

On Agregation-based Membership Functions for Fuzzy Image Segmentation

J. Chamorro-Martínez, D. Sánchez and B. Prados-Suárez*
Department of Computer Science and Artificial Intelligence. University of Granada (Spain)
e-mail: {jesus,daniel,belenps}@decsai.ugr.es

Abstract

In this paper we introduce a methodology for image segmentation by means of fuzzy subsets of connected pixels. Unlike the clustering-based techniques, which classify each pixel without considering its neighbourhood, the approach presented in this paper uses information about fuzzy connectivity between pixels. The connectivity degree measurement is then obtained by searching the least cost path linking both pixels, being the cost choice an important issue to take into account. Our study involves the analysis of different alternatives to path-cost functions, for which a set of desirable properties are defined. To determine the performance of the proposed functions, a set of experiments is carried out with both synthetic and real images. Finally, the cost functions are used to obtain fuzzy regions in natural images.

Keywords: Image segmentation, fuzzy segmentation, path-based segmentation, fuzzy connectivity.

1 Introduction

Image segmentation is defined as a partition of the image into connected subsets of pixels, called regions, on the basis of some homogeneity criterion. Usually, this is the starting point in a wide number of applications like image database retrieval [1], or robot vision [2].

Many types of segmentation techniques have been proposed in the literature, for example those based on histogram analysis [3], clustering [4], split and merge [5], region growing [6] and edge based algorithms [7]. It is certain that most of the proposals falling in the aforementioned categories, provide a crisp segmentation of images, where each pixel has to belong to an unique region. However, separation between regions is usually imprecise in natural images, as occours in shadows, brights and colour gradients, where crisp techniques are not often appropriate. To solve this problem, some approaches propose the definition of *region* as a fuzzy subset of pixels, in such a way that every pixel of the image has a membership degree to that region [8, 9].

The majority of the fuzzy techniques are based on fuzzy clustering, like C-means algorithms, which defines a set of centroids and compute the membership value for all the pixels in the image to each centroid [10, 11]. Another examples are those based on the definition of fuzzy histograms [12] or fuzzy rule-based systems [13]. Nevertheless, a drawback of most of these fuzzy approaches is that they don't take into account that a region must be topologically connected. As a consequence, pixels belonging to separate and different regions could be assigned to the same cluster.

To face up to above-mentioned problem, arise path-based techniques, which incorporate spatial information related to adjacency between pixels. These approaches are based on the idea of fuzzy topology, introduced by Rosenfeld [14], and the use of fuzzy connectivity to measure the relationship between any pair of pixels. Thus, given a region represenative point (called seed point), the path-based methods obtain the associated fuzzy region on the basis of the connectivity between the seed point and the rest of pixels in the image [6, 8, 15].

In the context of path-based image segmentation, an open problem is how to measure the cost of a "path" connecting two pixels and, based on this cost, the membership degree of its fuzzy connectivity. Most of the approaches in the literature calculate this cost from the set of distances between consecutive points in the path, being the most common solution the use of a simple aggregation function (sum or maximum) over this set of distances [6, 8]. However, to our knowledge,

*This work has been supported by the MCYT (Spain) under grant TIC2003-01504

there is not a study about the goodness of different cost functions applied to image segmentation.

In this paper an experimental study of different cost functions is performed. For this purpose, a set of desirable properties are defined and a set of candidates which verify that properties are proposed. To analyze the performance of the studied functions, a set of experiments is presented with synthetic and real images. Using the cost functions analyzed in the experiment, the growing region algorithm proposed in [2] is used to obtain fuzzy regions by means of a fuzzy subset of connected pixels.

The rest of the paper is organized as follows: In section 2 we introduce the path-based segmentation techniques. Section 3 describes the desirable properties for the cost function and the candidates studied in this paper. Finally, the experimental results and the main conclusions are showed in sections 4 and 5 respectively.

2 Path-based image segmentation

Using the notation introduced in [16], in this section we present a methodology to obtain image regions as fuzzy subsets of connected pixels. The connectivity between any pair of points is calculated on the basis of a distance between pixels which take into account not only the pixel features (grey level, colour, etc.), but also the "topology" of the image.

2.1 Connectivity between pixels

To measure differences between pixels, we first chararecterize each pixel as follows:

Definition 2.1 *Let $\overrightarrow{f_p}$ be a vector of features associated to the pixel p*

$$\overrightarrow{f_p} = [f_p^1, f_p^2, \ldots, f_p^n] \tag{1}$$

where a feature f_p^i, with $i \in \{1, 2, \ldots, n\}$, is defined as any relevant characteristic that may be obtained for p.

As an example of features vector, let's think about a three band colour representation like HSI or RGB

Given two pixels p and q, a difference $\triangle(p, q)$ between its features vectors is needed. In general, the function $\triangle$ may correspond to the Euclidean distance:

$$\triangle(p, q) = \sqrt{\sum_{i=1}^{n} \left(f_p^i - f_q^i\right)^2} \tag{2}$$

but it is not always the case (it may depend on the nature of the features).

In path-based image segmentation, to measure how different are two pixels we do not use directly the difference between its features given by $\triangle$, but we use information about the path joining them, according to next definition:

Definition 2.2 *A path between two pixels p and q is a sequence*

$$\pi_{pq} = (r_1, r_2, \ldots, r_k) \tag{3}$$

where $k \geq 1$, such that $r_1 = p$ and $r_k = q$ and r_i is connected to r_{i+1} $\forall i \in \{1, \ldots, k-1\}$

We will note $\prod_{pq}$ the set of possible paths linking the pixels p and q through pixels of the image I, being the cost of each path in the set defined as follows:

Definition 2.3 *The cost of a given path $\pi_{pq} \in \prod_{pq}$, is defined as a function*

$$cost : \Pi_{pq} \rightarrow [0, 1] \tag{4}$$

calculated on the basis of distances between consecutive points on the path.

To measure the difference between consecutive points, the function $\triangle$ will be used. In section 3 we shall study desirable properties of *cost* and we shall propose some candidate functions.

Taking into account the *cost* function, we define the optimum path between p and q, $\hat{\pi}_{pq}$, as the path that links both points with minimum cost, in the following way:

Definition 2.4 *The optimum path between p and q is:*

$$\hat{\pi}_{pq} = \underset{\pi_{pq} \in \Pi_{pq}}{argmin} \{cost(\pi_{pq})\} \tag{5}$$

Based on this optimum path, we can get the measure of the distance between two pixels as given by:

Definition 2.5 *The distance between two pixels p and q is the cost of the optimum path from p to q:*

$$d(p, q) = cost(\hat{\pi}_{pq}) \tag{6}$$

Let us remark that the distance defined in (6) uses topographic information (paths linking the pixels) and distances between pixel features.

2.2 Membership function for fuzzy regions

In the above section we have introduced the use of paths to measure the distance between any pair of pixels (without a fuzzy analysis). Now this distance will

be used to obtain fuzzy regions in an image I by means of the growing region algorithm proposed in [2]. Concretely, in this approach a region is defined as a fuzzy subset of connected pixels, so it is necessary to define a measure that indicates the degree in which each pixel in the image I belongs to each region. Under the assumption that a fuzzy region $\widetilde{R_s}$ has a representative seed point r_s (as in the region growing techniques), we introduce the following membership function associated to each region:

Definition 2.6 *The membership degree $\mu_{\widetilde{R_s}}(p)$ of a pixel p to a fuzzy region $\widetilde{R_s}$ is defined as:*

$$\mu_{\widetilde{R_s}}(p) = 1 - \frac{d(p, r_s)}{M} \tag{7}$$

where r_s is the seed point of $\widetilde{R_s}$, and M is a normalization factor given by

$$M = \max\left\{d(q, r_s)\ ,\ q \in I\right\} \tag{8}$$

Using equation (7) we can calculate the membership degree of every point $p \in I$ to each region $\widetilde{R_s}$. This allows us to obtain a set of fuzzy regions $\widetilde{\Theta} = \left\{\widetilde{R}_1, \widetilde{R}_2, \ldots, \widetilde{R}_m\right\}$ from a set of seed points $\Theta = \{r_1, r_2, \ldots, r_m\}$. An algorithm to calculate $\widetilde{\Theta}$ is proposed in [2] with a computational complexity of $O(mn)$, where n is the number of pixels and m is the number of seeds.

3 Measuring the path cost

In previous section we have described a methodology for path-based fuzzy segmentation. In this section we discuss about possible ways to measure the cost of a path.

3.1 Path Cost Intuition

We want to come up with a function $cost(\pi_{pq})$ measuring the cost of a given path π_{pq}. As was introduced in previous section, the cost of a path depends on the set of distances between consecutive pixels in the path (measured by means of the distance $\triangle$ defined in equation 2). Hence, it seems natural to define $cost(\pi_{pq})$ as a aggregation of the distances between consecutive points in the path π_{pq}.

Let $\mathcal{P}(\pi_{pq})$ be the set of pixels in the path π_{pq}, and let π_{pq}^{rs} be the subpath of π_{pq} that connects r and s with $r, s \in \mathcal{P}(\pi_{pq})$, and r appearing before s. The *cost* function should verify the following properties:

1. $cost(\pi_{pq}) \geq \max_{k\in\{1,\ldots,n-1\}}\{\Delta(r_k, r_{k+1})\} \geq 0$
2. If p, q are two neighbour pixels. Then we request $cost(\hat{\pi}_{pq}) \leq \Delta(p, q)$.
3. Monotony: $cost(\pi_{pq}^{rs}) \leq cost(\pi_{pq})$.
4. Reflexivity: Let $\pi_{pp} = p, p$. Then $cost(\pi_{pp}) = 0$.
5. Symmetry: $cost(\pi_{pq}) = cost(\pi_{pq}^{-1})$
6. Transitivity: $cost(\hat{\pi}_{pq}) \leq min_{r\in P_I}\{cost(\hat{\pi}_{pr} \boxplus \hat{\pi}_{rq})\}$.

where $\boxplus$ stands for path concatenation defined as follows: if $\pi_{pq} = \{p_1, \ldots, p_n\}$ and $\pi_{qr} = \{q_1, \ldots, q_m\}$ are two paths with $p \neq q \neq r$ (obviously, $p_n = q_1 = q$), then

$$\pi_{pq} \boxplus \pi_{qr} = p_1, \ldots, p_n, q_2, \ldots, q_m$$

We shall rewrite this path as $p_1, \ldots, p_n, p_{n+1}, \ldots, p_{n+m-1}$.

These properties satisfy the intuitive idea of using a fuzzy t-conorm to aggregate the distances between consecutive pixels into the final cost of the whole path, i.e., to define the cost function as

$$cost(\pi_{pq}) = \bigvee_{r_i, r_{i+1}\in\mathcal{P}(\pi_{pq})} \Delta(r_i, r_{i+1}) \tag{9}$$

were $\vee$ is a fuzzy t-conorm.

3.2 Some candidates

The choice of the t-conorm to be used in equation 9 depends on the application at hand. In order to discuss about this issue we have performed some experiments using the following t-conorms:

- Maximum
- Algebraic sum $S_A(a, b) = a + b - ab$.
- Bounded sum $S_B(a, b) = \min(1, a + b)$.
- Dubois-Prade's parametric t-conorm:

$$S_{DP}(a, b) = 1 - \frac{(1 - a)(1 - b)}{\max\{(1 - a), (1 - b), \alpha\}} \tag{10}$$

  with several parameters $\alpha \in [0, 1]$.

- Weber's parametric t-conorm:

$$S_W(a, b) = \min\left(1, \frac{a + b - \lambda ab}{1 - \lambda}\right) \tag{11}$$

  with several parameters $\lambda > -1$.

- Frank's parametric t-conorm:

$$S_F(a, b) = 1 - \log_s\left(1 + \frac{(s^{(1-a)} - 1)(s^{(1-b)} - 1)}{s - 1}\right) \tag{12}$$

  with several parameters $s > 0$, $s \neq 1$.

In the next section we show some results and we discuss about using these t-conorms to calculate the cost of a path.

4 Experimental results

Experiments presented in this section have been performed on gray level images, so the features vector defined in equation (1) has only one component: the illumination level of the pixel (a value in [0,255]). Thus, the distance defined in (2) is specified in this case as $\triangle(p,q) = |I(p) - I(q)|$.

4.1 Synthetic images

To test cost functions we have used synthetic images of size $256x256$ simulating, by means of a gaussian function, a toning down from a light incidence point (Figure 1). This point is placed in (0,0) in image Fig.1(A1) while Fig.1(B1) has, in addition to it, two more points located in (128,128) and (255,255).

In this experiment we have fixed a path over the shade to study the evolution and behaviour of cost functions along it. This path is, in both images (Figure 1(A1 ,B1)), the diagonal path from the pixel (0,0) to the pixel (255,255), that is, the path $\pi_{r_0 r_{255}} = (r_0, r_1 \dots r_{255})$ with r_i being the pixel (i,i) $\forall i \in \{0,1,\dots,255\}$. The graphics in figure 1 show the value of $cost(\pi^{r_0 r_i}_{r_0 r_{255}})$ $\forall i \in \{0,1,\dots,255\}$, with the X axis representing the parameter i, using different cost functions. To reader clarify, results are presented as membership values $(1 - cost(\pi^{r_0 r_i}_{r_0 r_{255}}))$, so the graphic may be interpreted as the fuzzy connectivity degree of each pixel r_i in the path respect to the seed point r_0.

Since there are several graphic representations, we have used bounded sum as reference to be able to make comparisons between different cost functions. In all the examples presented in Figure 1, image 1 is the original image (with the analyzed path); graphic 2 represents the membership values obtained along the path with maximum, bounded sum, algebraic sum and Dubois Prade t-conorm with several values for its parameter, and graphics (3) and (4) show the results for Weber a Frank t-conorms, respectively, with different values for their parameters.

In figure 1(A2), bounded sum shows a higher slope than Dubois and Prade t-conorm, whose parameter varies in [0,1]. As can be observed, when its parameter takes value 1 this t-conorm matches Algebraic Sum and gets more plain and more different to the reference as its parameter decreases, becoming the maximum function when $\alpha = 0$.

Graph 1(A3) shows degrees obtained with Frank's

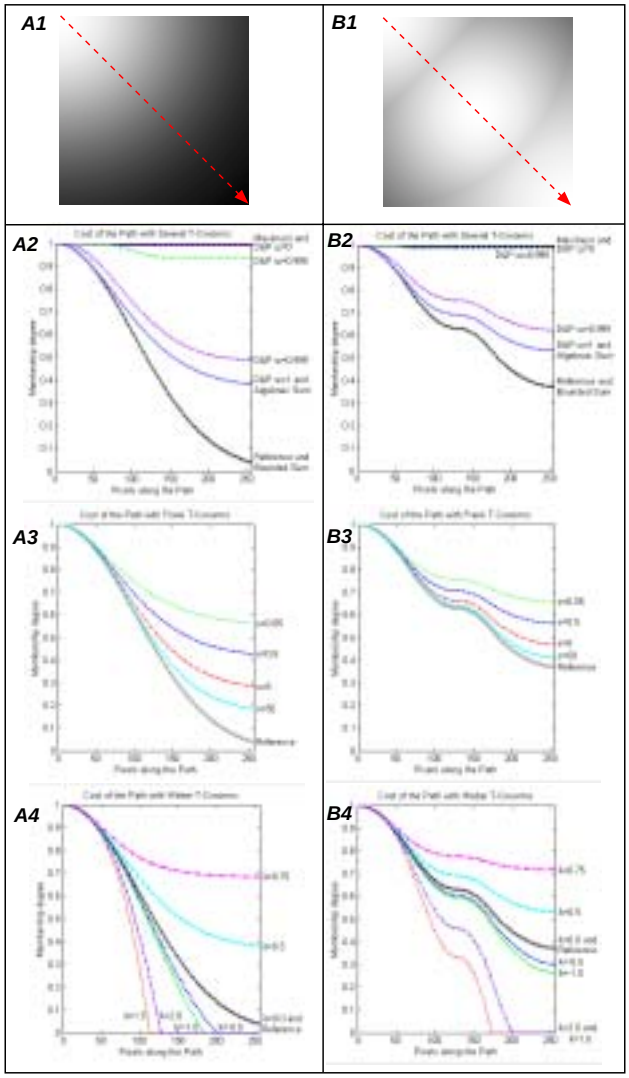


Figure 1: Results with synthetic images generated by means of a Gaussian function simulating a toning down from a light incidence point. (A1) Light point placed in (0,0). (B1) Three light points located in (0,0), (128,128) and (255,255). The graphics (A2-A4) and (B2-B4) show the result of computing the membership value along the diagonal path with different cost functions.

t-conorm. In this case, parameter s increase, makes the function assign higher values to pixels in the path and capture low decreasing shades.

In the last graph 1(A4), we can see how when Weber's function parameter takes its smallest value ($\lambda = -1$), function has a higher slope than reference function. Weber's function also has a saturation point after which function keeps constant, meaning the point from where this function considers the shade ends. While λ is negative and as it increases, Weber function gets softer, matching bounded sum when it takes value 0. From this point, while λ tends to 1, function gets more horizontal, which means that shade has very little variation from one point to de following one in the path (it is very soft). Functions also assign higher membership values to each pixel. So, for example, when $\lambda = 0.75$ the membership values are over 0.7 for all the pixels in the path. In the case of $\lambda = 1$, there is a discontinuity that makes Weber's functions get slopes higher than for negative values of λ. Functions obtained with $\lambda > 1$ capture shades decreasing fast and limit the shade to the point in which they become constant. This point gets farther from the origin of the shade as λ increases.

In Figure1(B2-B4) can be observed how all the functions depict the same behavior, keeping the relation between them even through the two inflexion points due to the change from a tone down to another one. Every time the path goes through the intersection of two shades, there is a point where the slope of the function changes to adapt to the new gray fade; and all the studied functions shows it.

4.2 Real images

Figure 2 shows an example of the usefulness of this variety of behaviors, applied to real images, like cells in image 1. In graphics of Figure 2(2-6) we have computed the membership value from the points marked with white crosses in Figure 2(1) to all the pixels in the image, using the bounded sum and Weber t-conorm with $\lambda = 1.5$. Figures 2(2,3) shows how both cost functions detect the bounds of the well delimited cell, having a big decreasing membership value behavior, when cell's contour is reached. What makes different both functions is the magnitude of the decrease, and how far influence area of the cell spreads through the intercellular background. This difference may also be seen in Figures 2(4,5) where there are three superposed cells and there are no evident and clear limits. In this case both functions have a soft decrease inside both cells and the superposed area, and a bigger decrease in membership values when their frontier with the background is reached. In the case of Weber's function, this t-conorm considers just both cells as belonging to

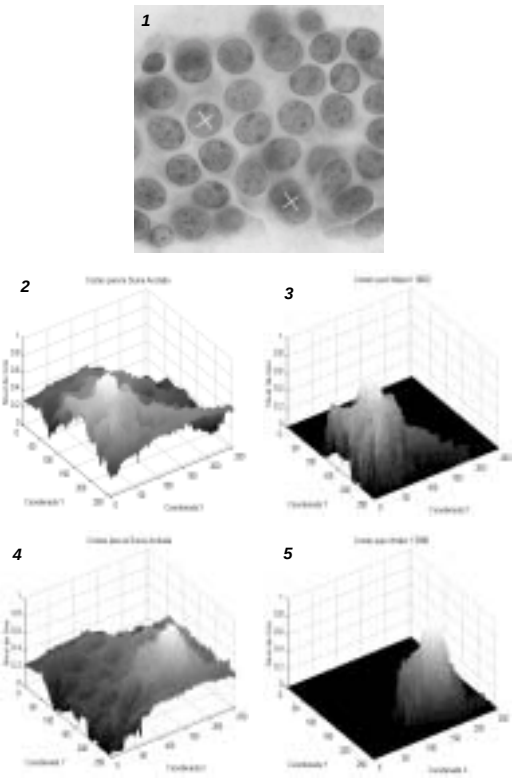


Figure 2: Results with real image. Original image (1) and fuzzy regions (2)

the superposing cells, while bounded sum lets membership value keep higher than 0 through a quite bigger surrounding area (almost the whole image).

5 Conclusions

We have proposed the use of different t-conorms as cost functions since they present a wide range of behaviors, giving us a powerful and flexible tool to adapt membership functions to any kind of image, depending of its purpose and characteristics. This range of different behaviors, which derives from the order relation existing between t-conorms, goes from the maximum to the bounded sum. It allows us to see the same shade as a soft fade or a fast decreasing tone down, depending on how sensible to the shade the chosen function is.

Although all the studied functions adapt properly to tone downs, functions like maximum don't seem to perform well, since they get saturated very near to the origin of the shade. On the other hand, bounded sum seems to capture better the variation in gray levels, being the most similar function to shape of the shade, and spreading through the whole extent of the tone down. Meanwhile some Weber's functions are able to find a limit in the tone down. So, all of the t-

conorms are valid, and the choice has only influence on the magnitude of the membership value, in the interval in which membership function actually takes value and, mainly, in the slope of the shade: in how fast the fade is and how far it reaches.

These seen representations lead us to find two different behaviors in cost functions. On one hand we have those with a small slope, like bounded sum, which are suitable for representing soft shades with small variation. On the other hand, we find functions, like Weber with $\lambda = 1.5$, where the fade of the shade is fast and has a big slope. These functions find a limit in the shade from which membership value is 0, and so there is no shade. This is useful in images where it's necessary to find a frontier between regions whose bounds are fuzzy.

References

[1] J. Fuertes, M. Lucena, N. Perez de la Blanca, and J. Chamorro-Martinez, "A scheme of colour image retrieval from databases," *Pattern Recognition Letters*, vol. 22, pp. 323–337, 2001.

[2] J. Chamorro-Martinez, D. Sanchez, and B. Prados-Suarez, *A Fuzzy Color Image Segmentation Applied to Robot Vision*. In Advances in Soft Computing - Engineering, Design and Manufacturing. J.M. Benitez, O. Cordon, F. Hoffman, and R. Roy (ed.). Springer., 2003, pp. 129–138.

[3] A. Gillet, L. Macaire, C. Botte-Lococq, and J. Postaire, "Color image segmentation by fuzzy morphological transformation of the 3d color histogram," *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 824 –824, 2001.

[4] D. Zhong and H. Yan, "Color image segmentation using color space analysis and fuzzy clustering," *Proceedings of the 2000 IEEE Signal Processing Society Workshop*, vol. 2, pp. 624– 633, 2000.

[5] G. Borges and M. Aldon, "A split-and-merge segmentation algorithm for line extraction in 2d range images," *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 1, pp. 441 – 444, 2000.

[6] A. Moghaddamzadeh and N. Bourbakis, "A fuzzy region growing approach for segmentation of color images," *Pattern Recognition*, vol. 30, no. 6, pp. 867–881, 1997.

[7] A. Shiji and N. Hamada, "Color image segmentation method using watershed algorithm and contour information," *Proceedings of International Conference on Image Processing*, vol. 4, pp. 305 – 309, 1999.

[8] B. Cavalho, C. Gau, G. Herman, and T. Kong, "Algorithms for fuzzy segmentation," *Proceedings of the International Conference on Advances in Pattern Recognition*, pp. 154–63, 1999.

[9] J. Chamorro-Martinez, D. Sanchez, B. Prados-Suarez, E. Galan-Perales, and M. Vila, "A hierarchical approach to fuzzy segmentation of colour images," in *IEEE International Conference on Fuzzy Systems*, vol. 2. St. Louis, Missouri (USA), May 2003, pp. 966–971.

[10] J.-F. Yang, S.-S. Hao, and P.-C. Chung, "Color image segmentation using fuzzy c-means and eigenspace projections," *Signal Processing*, vol. 82, pp. 461–472, 2002.

[11] Y. Zhao and M. Li, "A modified fuzzy c-means algorithm for segmentation of mri," *Proceedings of the fifth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2003), 27-30 September*, pp. 391–395, 2003.

[12] J. Han and K.-K. Ma, "Fuzzy color histogram and its use in color image retrieval," *IEEE Transactions on Image Processing*, vol. 11, no. 8, pp. 944–952, 2002.

[13] F. Hacbouf and N. Mezhoud, "An improved fuzzy rule-based segmentation system," *Proceedings of the Seventh International Symposium on Signal Processing and Its Applications, July 1-4 2003*, vol. 1, pp. 533–536, 2003.

[14] A. Rosenfeld, "Fuzzy digital topology," *Information and Control*, vol. 40, no. 1, pp. 76–87, 1979.

[15] J. Udupa and S. Samarasekera, "Fuzzy connectedness and object definition: Theory, algorithms and applications in image segmentation," *Graphical Models and Image Processing*, vol. 58, no. 3, pp. 246–261, 1996.

[16] J. Chamorro-Martinez, D. Sanchez, B. Prados-Suarez, E. Galan-Perales, and M. Vila, "Segmenting colour images on the basis of a fuzzy hierarchical approach," *Mathware and Soft Computing*, vol. 10, no. 2, pp. 101–115, 2003.

APORTACIÓN A LA SEGMENTACIÓN AUTOMÁTICA DE IMÁGENES, BASADA EN LA LOCALIZACIÓN DE LOS PÍXELES Y LA INFORMACIÓN PROPORCIONADA POR LOS NIVELES DE ILUMINACIÓN[1]

Angela Grau
Dep. Matemàtica Aplicada II
Universitat Politècnica de Catalunya
angela.grau@upc.es

Pilar Sobrevilla
Dep. Matemàtica Aplicada II
Universitat Politècnica de Catalunya
pilar.sobrevilla@upc.es

Eduard Montseny
Dep. Eng. Sist. Automàtica i Inf. Industrial
Universitat Politècnica de Catalunya
eduard.montseny@upc.es

Resumen

Puesto que las imagines monocromas digitales se muestran como conjuntos discretos de niveles de iluminación, si en el diseño de un algoritmo de segmentación se considera la capacidad del ojo humano para discriminar dichos niveles, los resultados obtenidos pueden mejorar notablemente.

Si bien los niveles de iluminación que aparecen en una imagen aportan mucha información a su segmentación, el uso de esta característica plantea el problema de determinar los niveles de gris que constituyen cada nivel de iluminación. Siguiendo esta línea, en la presente comunicación se propone un sistema para el reconocimiento de los niveles de iluminación de una imagen en el que, a fin de poder simplificar los problemas y automatizar el proceso, al análisis del histograma de frecuencias de niveles de gris de la imagen, se le añade el análisis de la localización de los píxeles asociados a cada pico del histograma.

Palabras Clave: Segmentación de imágenes, Niveles de iluminación, Técnicas difusas.

1 INTRODUCCIÓN

La segmentación es uno de los procesos básicos del análisis de imágenes para su posterior reconocimiento y comprensión. Tradicionalmente se ha intentado segmentar las imágenes analizando características relacionadas con nivel de gris, textura, color, … . Sin embargo, se ha comprobado que el uso individualizado de cada una de estas características es insuficiente para obtener un algoritmo que permita la segmentación automática de cualquier tipo de imagen.

Durante las últimas décadas se ha trabajado en el desarrollo de algoritmos que, teniendo en cuenta la información facilitada por todas o algunas de estas características, permitan obtener una segmentación automática de tipos concretos de imágenes.

Actualmente, en un intento por obtener algoritmos en los que se facilite la fusión de la información proporcionada por las diferentes características, se está mejorando tanto la cantidad como la calidad de la información que se puede extraer del análisis de cada una de dichas características a fin de optimizar su aportación al resultado final. Sin embargo, a pesar de los grandes avances logrados, la segmentación automática de imágenes sigue siendo una línea abierta de investigación.

Dentro de esta línea de investigación, y como continuación a trabajos previos ([3], [7]), en este artículo se presentarán algunos de los resultados obtenidos cuando, junto con la información proporcionada por el análisis de los niveles de luminosidad que aparecen en una imagen monocroma, se considera la información correspondiente a la localización (relativa) de los píxeles dentro de la misma.

El artículo se ha organizado del modo siguiente: En la sección 2 se introducirán diversos conceptos relacionados con los niveles de luminosidad que se pueden identificar en una imagen monocroma. A continuación se explicará el proceso utilizado para segmentar el histograma de niveles de gris, obteniendo los picos relevantes, así como la problemática relacionada con la fusión de dichos picos y su posterior asociación con los niveles de luminosidad existentes en la imagen. En la sección 4 se introducirá la descripción del sistema propuesto, así como un conjunto de experimentos desarrollados. Se finalizará presentando las conclusiones y trabajos futuros.

[1] Financiado por el proyecto CICYT: TIC2002-02791

2 LOS NIVELES DE LUMINOSIDAD EN LA SEGMENTACIÓN DE IMÁGENES

Durante la última década se han empezado a desarrollar algoritmos de propósito general, que permitan emular el comportamiento del sistema de visión humano en el proceso de segmentación de imágenes. En este sentido se puede decir que los niveles de luminosidad presentes en una imagen se pueden considerar como las distintas intensidades/tonalidades que aparecen en la misma, y que corresponden a agrupaciones de niveles de gris. Estas intensidades son las utilizadas por un observador humano para diferenciar los diferentes objetos que aparecen en la imagen. Así por ejemplo, en la imagen de la figura 1, un observador humano apreciaría cuatro niveles de luminosidad, correspondientes a los cuatro elementos (los tres objetos y el fondo) de la misma.



Figura 1: Imagen con cuatro niveles de luminosidad.

El sistema de visión humano se basa en el análisis de los niveles de luminosidad que presentes en las imágenes monocromas a fin de diferenciar los objetos que en ella aparecen. Así mismo, utiliza dichos niveles como una característica que permita identificar las fronteras que existen entre los objetos.

Los niveles de luminosidad de una imagen, L_i ($1 \le i \le n_L$), están determinados por sus niveles de gris $g_j^{L_i}$ ($l_i \le j \le m_i; 0 \le l_i \le m_i \le 255$). Sin embargo, puesto no existen niveles de gris determinados que separen perfectamente los niveles de luminosidad correspondientes a objetos contiguos, las fronteras entre los niveles de gris no son nítidas. Por otra parte, los niveles de luminosidad presentan características muy variables ya que, incluso para imágenes de un mismo tipo, los niveles que en ellas aparecen no son siempre los mismos, ni están definidos por los mismos niveles de gris.

De lo anteriormente expuesto se puede concluir que los problemas principales a tener en cuenta, y resolver, cuando se considera el nivel de luminosidad como característica para la segmentación de imágenes, son:

I. Decidir cuáles son los niveles de gris que constituyen cada uno de los niveles de luminosidad que aparecen en la imagen.
II. Determinar cómo tratar los niveles de gris que constituyen – o están relacionados con- las fronteras entre niveles de luminosidad.

Para resolver el primer problema, que es el que aquí nos ocupa, se va a considerar la siguiente regla:

R_{pres}.: Dada una imagen I, un determinado nivel de luminosidad L_i estará *presente* en I si:

$$\exists P = \{p_{x,y}\} \quad t.q.: \; 1.\text{-} \#(P) \gg k,$$
$$2.\text{-} \; p_{x,y} \in R_l \;\; \text{con} \;\; R_l \in I$$
$$3.\text{-} \; ng(p_{x,y}) = g_j^{L_i}$$

De lo anteriormente expuesto se puede concluir que una solución para detectar los niveles de luminosidad de una imagen consiste en segmentar los *picos relevantes* que aparecen en el histograma de niveles de gris de la misma, para posteriormente agrupar los que deben estar asociados a un mismo nivel de luminosidad.

3 SEGMENTACIÓN DEL HISTOGRAMA DE UNA IMAGEN

Una vez obtenido el histograma de niveles de gris de una imagen, el siguiente paso consiste en realizar un análisis del mismo que permita obtener los intervalos de niveles de gris asociados con las diferentes *Etiquetas de iluminación*. Para ello, una vez obtenidos los mínimos locales del histograma suavizado, se obtienen los *picos relevantes* a partir del análisis del histograma, en el que se aplican un conjunto de reglas que eviten identificar como picos relevantes los asociados al rizado que aparece en el histograma, debido a la presencia de ruido en la imagen.

Los picos obtenidos proporcionan una información global de la apariencia de la imagen, la cual no permite conocer ni el número de *etiquetas* ni los intervalos de niveles de gris asociados a las mismas.

Puesto que se pretende definir un algoritmo que capture la funcionalidad del sistema de visión humano, se considerará que las diferentes *Etiquetas de Iluminación* están determinadas por los intervalos de niveles de gris asociados a los objetos, teniendo en cuenta que los niveles de gris aparecen mezclados en los mismos.

Por otra parte, para poder obtener una segmentación automática del histograma, se han de encontrar métodos que permitan resolver los siguientes problemas:

- Cómo relacionar los picos relevantes detectados con los niveles de luminosidad presentes en la imagen.
- Cómo juntar los picos cuyos niveles de gris están asociados al mismo nivel de luminosidad.
- Cómo validar la segmentación obtenida del histograma.

Una primera aproximación, que proporcione la segmentación resolviendo los problemas anteriores, consiste en realizar el análisis manual de un conjunto de imágenes de entrenamiento que proporcione un conjunto de reglas a partir de las cuales, y de forma automática, se obtengan el número de niveles de luminosidad y los picos relevantes asociados a cada uno de ellos.

Dado que el proceso anterior es lento, costoso y muy difícil de automatizar, se propone un nuevo enfoque para

la resolución del problema. Este enfoque se basa en considerar, junto con la información del nivel de gris de cada píxel, la información proporcionada por su localización, así como la proporcionada por las características de los píxeles en un entorno del mismo.

4 DESCRIPCIÓN DEL SISTEMA

A fin de asociar los picos relevantes del histograma con los niveles de luminosidad, se analizarán las relaciones que existen entre los picos. Dichas relaciones se definirán atendiendo a la localización, dentro de la imagen, de los píxeles cuyos niveles de gris están asociados a los picos.

Se ha considerado que dos picos relevantes pueden estar: *Aislados*, *Mezclados* o ser *Frontera*. Estas relaciones vienen dadas por la siguiente definición:

Definición 1.- Sean P_i y P_j dos *picos relevantes* con niveles de gris asociados I_{P_i} e I_{P_j}. Si denotamos por R_k, L_k, O_k las regiones, los niveles de luminosidad y los objetos de la imagen, respectivamente, diremos que:

1.1.- P_i y P_j son *Aislados* si para todo par de píxeles $p_{x,y}$ y $p_{z,v}$, con $ng(p_{x,y}) \in I_{P_i}$ y $ng(p_{z,v}) \in I_{P_j}$, se cumple que $p_{x,y} \in R_1$ y $p_{z,v} \in R_2$. Donde R_1 y R_2 son regiones de la imagen sin ningún punto de contacto.

1.2.- P_i y P_j están *Mezclados* si dados $p_{x,y}$ y $p_{z,v}$, tales que $ng(p_{x,y}) \in I_{P_i}$ y $ng(p_{z,v}) \in I_{P_j}$, entonces $p_{x,y}, p_{z,v} \in L_k$ o bien $p_{x,y}, p_{z,v} \in O_k$.

1.3.- P_i y P_j son *Frontera* si para todo par de píxeles $p_{x,y}$ y $p_{z,v}$, con $ng(p_{x,y}) \in I_{P_i}$ y $ng(p_{z,v}) \in I_{P_j}$, se tiene: $p_{x,y} \in R_1$ y $p_{z,v} \in R_2$, o $p_{x,y} \in L_1$ y $p_{z,v} \in L_2$ y $p_{z,v} \in L_2$, estando en la imagen R_1 y R_2 (L_1 y L_2) una al lado de la otra, y pudiendo trazar una frontera, o establecer una banda, que separe ambas regiones.

A continuación se presentan los resultados de los experimentos realizados para evaluar las relaciones entre los picos relevantes de un histograma, que permitirán definir los correspondientes conjuntos difusos

4.1. EVALUACIÓN DE LAS RELACIONES ENTRE PICOS DE UN HISTOGRAMA

Para evaluar las relaciones existentes entre los picos relevantes del histograma se han analizado las relaciones de los píxeles correspondientes a dichos picos, en un conjunto de imágenes.

En primer lugar se ha diseñado una imagen sintética (figura 1) con tres objetos de diversos tamaños y formas.

A partir de dicha imagen se han obtenido las siguientes versiones:

- *Imagen real*: Obtenida a partir de la impresión de la imagen sintética y posterior escaneado de la misma.
- *Imagen real con ruido blanco*.
- *Imagen real con ruido blanco y niveles de gris próximos*. Obtenida a partir de la impresión de la imagen sintética modificada de manera que los niveles de gris de los objetos sean más próximos, y realizando posteriormente el escaneado de la misma.

Una vez segmentados los histogramas de estas imágenes, para realizar una evaluación de las relaciones entre los mismos, consideraremos las siguientes definiciones:

Definición 2.- Dado un pico P_i y un píxel $p_{z,v} \in P_j$, se define la *relación de vecindad* "$P_i \boldsymbol{R}\ p_{z,v}$" tal que, si $p_{x,y} \in P_i$, y $\boldsymbol{\mathcal{E}}(p_{z,v})$ un entorno 3x3 del píxel $p_{z,v}$, se tiene:

$$p_{x,y}\ \boldsymbol{R}\ p_{z,v} \Leftrightarrow p_{x,y} \in \boldsymbol{\mathcal{E}}(p_{z,v})$$

De donde, el *número de píxeles de* P_i *vecinos de* $p_{z,v}$ es: $NR_i(p_{z,v}) = \#(P_i \boldsymbol{R}\ p_{z,v}) = \#\{p_{x,y} \in P_i : p_{x,y} \boldsymbol{R}\ p_{z,v}\}$, siendo $0 \leq NR_i(p_{z,v}) \leq 8$.

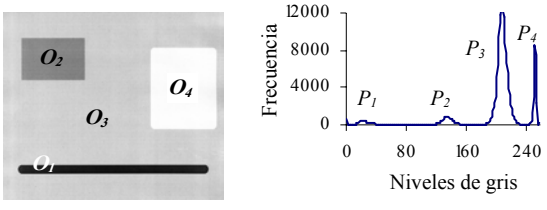
Definición 3.- El número de píxeles del pico P_j que tienen k vecinos del pico P_i viene dado por:

$$NR_i(P_j)_k = \#\{\, p_{z,v} \in P_j : NR_i(p_{z,v}) = k \,\};\ 0 \leq k \leq 8$$

Estas relaciones y conceptos, aplicadas sobre las imágenes de prueba descritas anteriormente, son la base para establecer reglas que permitan formalizar las definiciones de picos *Aislados*, *Mezclados* y *Frontera* dadas en la sección anterior.

4.1.1. Prueba 1: Imagen real.

La imagen de la figura 2-a) corresponde a la imagen real, obtenida a partir de la impresión de la imagen sintética y el posterior escaneado de la misma, en la que los objetos no presentan un único nivel de gris (al contrario de lo que ocurría en la imagen sintética). Este hecho se aprecia más claramente en su histograma de niveles de gris (figura 2-b), donde se observa que cada uno de los picos relevantes está constituido por más de un nivel de gris, correspondiendo cada uno de dichos picos a uno de los objetos que aparece en la imagen de la figura 2-a).



a) b)

Figura 2: a) Imagen real; b) Histograma de niveles de gris.

De la observación de las figuras anteriores y considerando las definiciones dadas en la definición 1, se puede deducir

que la relación entre los picos P_1, P_2 y P_4 es de *Aislados*, mientras que dichos picos son *Frontera* con el P_3.
Así pues, las características de la imagen son:

- Los picos están claramente separados y son fácilmente segmentables.
- Los relaciones entre picos son: *Aislados* o *Frontera*.

Para evaluar formalmente las relaciones entre los picos se han considerado, de acuerdo con las definiciones 2 y 3, las *relaciones de vecindad* entre los píxeles de los mismos,. En la tabla 1 se presentan los resultados de la *relaciones de vecindad* del pico P_2 con respecto a P_1 y P_3. En las columnas Total aparecen los valores $NR_1(P_2)_k$ y $NR_3(P_2)_k$. Los valores $NR_i(P_2)_k/\#(P_2)$, i=1, 3 se dan en la columna %., y en la columna %rel se indica el porcentaje relativo al porcentaje máximo de la columna anterior.

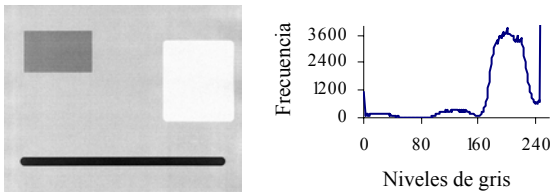
Tabla 1: Resultados de la relación de vecindad del pico P_2 con los picos P_1 y P_3, obtenidos para la figura 2.

	$NR_1(P_2)_k$			$NR_3(P_2)_k$		
k	**Total**	**%**	**%rel.**	**Total**	**%**	**%rel.**
0	14950	95.2	100	14413	91.7	100
1	10	0.1	0	71	0.5	1
2	21	0.1	0	101	0.6	1
3	705	4.5	5	1060	6.7	7
4	17	0.1	0	30	0.2	0
5	4	0.0	0	11	0.1	0
6	0	0.0	0	0	0.0	0
7	0	0.0	0	6	0.0	0
8	3	0.0	0	18	0.1	0
#(P_2)	**15710**			**15710**		

Observando los valores obtenidos, se aprecia que, para la imagen de la figura 2-a), los valores de la *relación de vecindad relativa (columnas %rel)* para el caso de *picos aislados* (P_2 y P_1) y *frontera* (P_2 y P_3) son muy parecidos.

4.1.2. Prueba 2: Imagen real con 20% de ruido blanco.

La imagen de la figura 3-a) corresponde a la imagen real con 20 % de ruido blanco añadido, por lo que los objetos aparecen con un mayor número de niveles de gris que en la figura 2-a). En el histograma de la imagen (figura 3-b), se aprecia que los picos son más bajos y anchos que los que podrían ser sus homólogos para la imagen de la figura 1, habiéndose reducido notablemente la distancia entre los picos relevantes, cuyo rizado ha aumentado notablemente.



a) b)

Figura 3: a) Imagen real con 20% de ruido blanco añadido; b) Histograma de niveles de gris.

En la segmentación del histograma se detectan 7 picos relevantes cuyos intervalos de niveles de gris aparecen en la tabla 2. En este caso la presencia del ruido fuerza al sistema a detectar mayor número de picos que objetos en la imagen, si bien los picos relevantes correspondientes a objetos distintos no se mezclan notablemente.

Tabla 2: Niveles de gris que constituyen cada pico relevante.

Picos	P_1	P_2	P_3	P_4	P_5	P_6	P_7
Niv. Gris	0–28	29–70	71–126	127–153	154–204	205–248	249–255

En la tabla 3 se dan las relaciones entre los picos del histograma según los conceptos dados en la definición 1. En la intersección de las filas y columnas aparece la relación entre los picos correspondientes, donde *A*, *M* y *F* se refieren a *Aislados*, *Mezclados* y *Frontera*. La *X* indica que no se estudia la relación un pico consigo mismo.

Tabla 3: Relaciones entre los picos relevantes del histograma de la imagen de la figura 4.

	P_1	P_2	P_3	P_4	P_5	P_6	P_7
P_1	*X*	*M*	*A*	*A*	*F*	*F*	*A*
P_2	*M*	*X*	*A*	*A*	*F*	*F*	*A*
P_3	*A*	*A*	*X*	*M*	*F*	*F*	*A*
P_4	*A*	*A*	*M*	*X*	*F*	*F*	*A*
P_5	*F*	*F*	*F*	*F*	*X*	*M*	*F*
P_6	*F*	*F*	*F*	*F*	*M*	*X*	*F*
P_7	*A*	*A*	*A*	*A*	*F*	*F*	*X*

Atendiendo a lo observado en la tabla anterior, y a las figuras 2-a) y 2-b), se puede concluir que:

- Los picos no están claramente separados y no son fácilmente segmentables.
- Los picos presentan los tres tipos de relaciones: *Aislados*, *Frontera* y *Mezclados*.

En la tabla 4 se presentan los resultados de la evaluación de las *relaciones de vecindad* del pico P_2 con respecto a P_3 y P_7, con los que, según la tabla 3, tiene una relación de *Aislados*.

Tabla 4: Resultados de la relación de vecindad del pico P_2 con los picos P_3 y P_7, obtenidos para la figura 4.

	$NR_3(P_2)_k$			$NR_7(P_2)_k$		
k	**Total**	**%**	**%rel.**	**Total**	**%**	**%rel.**
0	3247	96.6	100	3291	97.9	100
1	69	2.1	2	42	1.2	1
2	34	1.0	1	21	0.7	1
3	10	0.3	0	7	0.2	0
4	1	0.0	0	0	0.0	0
5	0	0.0	0	0	0.0	0
6	0	0.0	0	0	0.0	0
7	0	0.0	0	0	0.0	0
8	0	0.0	0	0	0.0	0
#(P_2)	**3361**			**3361**		

En esta tabla se observar que, para la imagen 2-a), los las *relaciones de vecindad y vecindad relativa (% y %rel)* son muy parecidas para picos definidos como *Aislados*.

La siguiente tabla presenta los valores obtenidos para las *relaciones de vecindad* de los picos P_2 con respecto a P_1 y de P_5 con respecto a P_6 que, según la tabla 3 están *Mezclados*.

Tabla 5: Resultados de las relaciones de vecindad del pico P_2 con P_1 y de P_5 con P_6.

	$NR_1(P_2)_k$			$NR_6(P_5)_k$		
k	**Total**	**%**	**%rel.**	**Total**	**%**	**%rel.**
0	38	1.1	5	2386	3.1	13
1	119	3.5	16	1788	2.3	10
2	278	8.3	39	6026	7.8	34
3	542	16.1	75	12800	16.5	71
4	663	19.7	92	17912	23.1	99
5	718	21.4	100	17994	23.2	100
6	601	17.9	84	12290	15.8	68
7	312	9.3	43	5359	6.9	30
8	90	2.7	16	1151	1.5	6
#(P_2)	**3361**		**#(P_2)**	**77706**		

Se observa que, al igual que en el caso de picos *Aislados,* para picos *Mezclados* los valores de las *relaciones de vecindad y vecindad relativa (% y %rel)* son muy parecidos.

En los resultados de la siguiente tabla, correspondientes a las *relaciones de vecindad* de los picos *Frontera* P_2 con P_5 y P_3 con P_6, se observa que, como en los casos anteriores, los valores de las *relaciones de vecindad y vecindad relativa* (*% y %rel*) son muy parecidos, teniendo la particularidad de ser muy similares a los obtenidos para picos *Aislados* (tabla 4), pero muy diferentes a los correspondientes a picos *Mezclados* (tabla 5).

Tabla 6: Resultados de la relación de vecindad del pico P_2 con P_5 y de P_3 y P_6.

	$NR_5(P_2)_k$			$NR_6(P_3)_k$		
k	**Total**	**%**	**%rel.**	**Total**	**%**	**%rel.**
0	3350	99.7	100	4855	89.1	100
1	9	0.3	0	157	2.9	3
2	2	0.1	0	244	4.5	5
3	0	0.0	0	193	3.5	4
4	0	0.0	0	1	0.0	0
5	0	0.0	0	1	0.0	0
6	0	0.0	0	0	0.0	0
7	0	0.0	0	0	0.0	0
8	0	0.0	0	0	0.0	0
#(P_2)	**3361**		**#(P_3)**	**5451**		

4.1.3. Prueba 3: Imagen real con 15% de ruido blanco y niveles de gris próximos.

Esta prueba se ha realizado para la imagen real con 15 % de ruido blanco, en la que los niveles de gris de los objetos que aparecen más oscuros son muy parecidos. Este hecho queda reflejado en el histograma de niveles de gris (figura 4) por el pico ancho de la izquierda, correspondiente a los dos niveles de iluminación más oscuros. Tras segmentar el histograma se han obtenido 6 picos relevantes, correspondientes a los intervalos de nivel de gris que aparecen en la tabla 7. Es de resaltar que, a diferencia de lo que ocurría con la prueba anterior, en este caso, los picos relevantes correspondientes a objetos distintos se mezclan notablemente

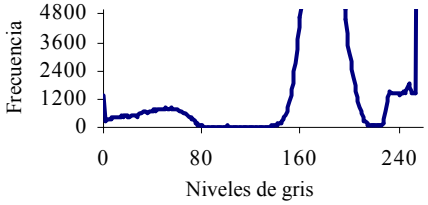


Figura 4: Histograma de la imagen real modificada, con 15% de ruido blanco.

Tabla 7: Niveles de gris de los picos relevantes.

Picos	P_1	P_2	P_3	P_4	P_5	P_6
Niv. Gris	0–26	27–53	54–88	89–219	220–253	254–255

Las relaciones entre cada par de los picos del histograma se ha representado en la tabla 8 aplicando el mismo proceso y notación que en la prueba 2 (tabla 3).

Tabla 8: Relaciones entre los picos relevantes del histograma de la imagen de la figura 4.

	P_1	P_2	P_3	P_4	P_5	P_6
P_1	*X*	*A*	*A*	*F*	*A*	*A*
P_2	*A*	*X*	*M*	*F*	*A*	*A*
P_3	*A*	*M*	*X*	*F*	*A*	*A*
P_4	*F*	*F*	*F*	*X*	*F*	*F*
P_5	*A*	*A*	*A*	*F*	*X*	*M*
P_6	*A*	*A*	*A*	*F*	*M*	*X*

Teniendo en cuenta las especificaciones de las relaciones dadas en la definición 1, lo que se observa en la figura 4, y los resultados de la tabla anterior, se deduce que:

- Los picos están muy solapados.
- Los picos presentan los tres tipos de relaciones: aislados, frontera y mezclados.

La siguiente tabla presenta los valores obtenidos para las *relaciones de vecindad* del pico P_2 con respecto a P_1 y viceversa que, según la tabla 8 son *Aislados*.

Tabla 9: Resultados de la relación de vecindad del pico P_2 con P_1, y de P_1 y P_2.

	$NR_1(P_2)_k$			$NR_2(P_1)_k$		
k	**Total**	**%**	**%rel.**	**Total**	**%**	**%rel.**
0	12890	70.5	100	2081	16.7	63
1	769	4.2	6	3249	26.1	100
2	363	2.0	3	3115	25.1	96
3	717	3.9	5	2182	17.6	67
4	656	3.6	5	10810	8.7	33
5	810	4.4	6	447	3.6	14
6	909	5.0	7	193	1.6	6
7	837	4.6	6	77	0.6	2
8	338	1.8	2	7	0.1	0
#(P_2)	**18289**		**#(P_1)**	**12432**		

Se observa que los valores de las *relaciones de vecindad y vecindad relativa* para $NR_1(P_2)_k$ son parecidos al resto de casos de picos *Aislados* (tabla 4), lo cual no ocurre para $NR_2(P_1)_k$, aunque la relación entre ellos, según la tabla 8, también es de *Aislados*. Se ha de destacar que los valores de las *relaciones de vecindad y vecindad relativa (% y %rel)* para $NR_2(P_1)_k$ se asemejan más a los correspondientes a picos *Mezclados* de la prueba 2 (tabla 5), que a los obtenidos para $NR_1(P_2)_k$.

Dada la vaguedad que se aprecia en los resultados obtenidos, se introducen técnicas difusas que permitan tratar la información obtenida a fin de establecer la relación existente entre los picos del histograma.

5 . ANÁLISIS DIFUSO DE LAS RELACIONES ENTRE PICOS

Dada una imagen, una vez segmentado su histograma, para obtener los conjuntos difusos difusos correspondientes a las relaciones existentes entre cada par de picos P_i, P_j, se procede como sigue:

1. Se obtienen los valores correspondientes a las *relaciones de vecindad relativa* (*%rel*) de $NR_i(P_j)_k$, y $NR_j(P_i)_k$; $0 \le k \le 8$.
2. Estos valores son fuzzyficados de acuerdo con las siguientes funciones de pertenencia:

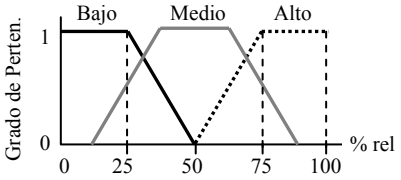


Figura 5: Funciones de pertenencia para los valores de vecindad relativa.

3. Una vez obtenidos los grados de pertenencia de las *relaciones de vecindad relativa* a cada uno de los conjuntos difusos, "$(\%rel_{i,j})_k$", la relación entre los picos vendrá determinada por las siguientes reglas:

3-1.- Picos Mezclados:

RM₀: $(\%rel_{i,j})_0$ *Bajo* $\Rightarrow P_i, P_j$ *Mezclados*.

RM_k: $(\%rel_{i,j})_k$ *Bajo* o *Medio* $\Rightarrow P_i, P_j$ *Mezclados*.
k=1, 2, 7, 8

RM_l: $(\%rel_{i,j})_l$ *Medio* o *Alto* $\Rightarrow P_i, P_j$ *Mezclados*. $3 \le l \le 6$

Aunque es suficiente con que se verifique una cualquiera de las reglas, a mayor número de reglas mayor grado de confianza en la relación.

3-2.- Picos Aislados o Frontera:

RAF₁: Si $(\%rel_{i,j})_0$ *Alto* & $(\%rel_{i,j})_1$ *Bajo o Medio* & $(\%rel_{i,j})_k$ *Bajo* $(2 \le k \le 6) \Rightarrow P_i, P_j$ *Aislados o Frontera*.

RAF₂: Si $(\%rel_{i,j})_{0,2}$ *Bajo o Medio* & $(\%rel_{i,j})_1$ *Alto* & $(\%rel_{i,j})_k$ *Bajo* $(3 \le k \le 6) \Rightarrow P_i, P_j$ *Aislados o Frontera*.

RAF₃: Si $(\%rel_{i,j})_2$ *Alto* & $(\%rel_{i,j})_3$ *Bajo o Medio* & $(\%rel_{i,j})_k$ *Bajo* $(k \ne 2, 3) \Rightarrow P_i, P_j$ *Aislados o Frontera*.

RAF₄: Si $(\%rel_{i,j})_{0,3}$ *Bajo o Medio* & $(\%rel_{i,j})_2$ *Medio* & $(\%rel_{i,j})_2$ *Alto* & $(\%rel_{i,j})_k$ *Bajo* $(k \ge 4) \Rightarrow P_i, P_j$ *Aislados o Frontera*.

A mayor grado de verificación de cualquiera de las reglas mayor grado de confianza en la relación.

6 . CONCLUSIONES Y TRABAJOS FUTUROS

De los resultados obtenidos hasta el momento se deduce la viabilidad de determinar los niveles de gris asociados a cada nivel de iluminación a partir de las relaciones entre los picos relevantes del histograma de niveles de gris. También se ha podido verificar que el proceso presentado puede ser automatizado, aunque para ello necesita algunos pasos adicionales. Por otra parte, el hecho de tener en cuenta la localización de los píxeles en el proceso de análisis, mejorará los resultados obtenidos hasta ahora con procesos más imprecisos y costosos.

En cuanto a trabajos futuros, se están realizando más pruebas para determinar, con más precisión: las reglas de evaluación de picos mezclados y aislados; la necesidad de añadir niveles de fuzzyficación; cómo utilizar la información de picos frontera.

Referencias

[1] R.C. Gonzalez, and R.E. Woods, *Digital Image Processing*, Addison Wesley, 1992.

[2] S.U. Lee et al.. A comparative Performance Study of Several Global Thresholding Techniques for Segmentation. *Comp. Vision, Graphics, Image Proc*. vol. 52, nº. 2. Pág. 171-190, 1990.

[3] E. Montseny, P. Sobrevilla. On the Use of Image Data Information for Getting a Brightness' Perceptual Fuzzy Model. *11th IEEE Int. Conf. on Fuzzy Systems.* Pág. 1086-1091. Honolulu, Mayo 2002.

[4] N. R. Pal and S. K. Pal. A review of image segmentation techniques. *Pattern recognition*, vol. 26, nº. 9. Pág. 1277-1294, 1993.

[5] J. R. Parker. Gray Level Thresholding in Badly Illuminated Images. *IEEE Trans. PAMI*, vol. 13, nº. 8. Pág. 813-819, 1991.

[6] A. Perez, R.C. Gonzalez. An Iterative Thresholding Algorithm for Image Segmentation. *IEEE Trans. PAMI*, vol. 9, nº. 6. Pág. 742-751, 1987.

[7] P. Sobrevilla, E. Montseny. Generation of Brightness Membership Functions for Labeled Image Segmentation. *10th IEEE Int. Conf. On Fuzzy Systems*. Melbourne, Australia, Diciembre 2001.

COLOREADO DE GRAFOS DIFUSOS: VERIFICACIÓN FORMAL Y COMPROBACIÓN PRÁCTICA CON MAPLE

Victoria López
Dept. Estadística e I.O. I
Universidad Complutense de Madrid
vlopez@mat.ucm.es

Lorenzo Martín
Dept. Matemática Aplicada a las TT.II.
Universidad Politécnica de Madrid
lmg@mat.upm.es

Resumen

La utilización de lógicas difusas en algorítmica obliga al tratamiento de asertos difusos en los programas. Entendiendo por algoritmo difuso aquél cuya especificación sea difusa, utilizaremos las técnicas de verificación clásicas para extenderlas al caso difuso y trabajar con este tipo de nuevos algoritmos. Presentamos el caso de la coloración de un grafo difuso para estudiar su verificación formal así como la comprobación de resultados mediante el sistema de cálculo simbólico Maple.

Palabras Clave: Grafos Difusos, Coloración, Clasificación, Verificación, Maple.

1 INTRODUCCIÓN

En Teledetección, el análisis de una imagen digital obtenida vía satélite tiene como objeto la clasificación de los píxeles [6] que la conforman en diferentes grupos de interés.

En algunos casos, a partir del análisis de fotografías aéreas, mapas y la propia experiencia personal sabemos de la existencia de ciertos grupos homogéneos (con ciertas características similares) en los que podremos clasificar los píxeles. Este es el caso de la clasificación supervisada [6, 12], no sólo porque se conocen de antemano las posibles clases o grupos donde vamos a clasificar el píxel sino porque además, a posteriori, podemos realizar comprobaciones sobre el terreno.

Por otro lado, existen características no visibles, detectables mediante la imagen digital, de las que no sabemos a priori ni de qué tipo son ni cuántas hay. En este caso no es posible una clasificación supervisada por lo que se realiza un estudio en el que la clasificación de un píxel y la determinación de los clústers [5] o grupos surgen de forma simultánea. En esta clasificación no supervisada no conocemos a priori las clases y a posteriori no podemos enviar a un explorador del terreno por no ser visible la característica en estudio. Los algoritmos de clasificación no supervisada que empleamos se basan en la identificación de zonas homogéneas y reconocimiento de patrones por lo que además y simultáneamente se tendrá en cuenta el entorno del píxel para su clasificación lo que conduce a una clasificación difusa [1, 2] del mapa de píxeles.

Estamos interesados en la especificación, comprobación y verificación de este tipo de problemas [11, 13]. Para la especificación y verificación formal haremos uso de la lógica de predicados clásica y difusa, y de la semántica axiomática [4, 9]. Para la comprobación de resultados nos apoyaremos en la potencia del sistema de cálculo simbólico Maple [10].

En la segunda sección de este artículo, se plantea el problema de la verificación desde el paradigma funcional; en la tercera, se introducen los elementos necesarios para el problema de coloración propuesto; en la cuarta, se realiza la especificación formal de las funciones que calculan los elementos planteados en la sección anterior; en las secciones quinta y sexta se implementa y verifica respectivamente el algoritmo de coloración de un grafo difuso.

2 VERIFICACIÓN DE ALGORITMOS RECURSIVOS

La estructura general de una función recursiva no final [13] es

fun Rec (X: TDatosEntrada)
ret R: TDatosSalida;
{PRE}
case $B_t(X) \rightarrow$ R:=Triv(X)
case $B_{nt}(X) \rightarrow$

Combinar(X,Rec(Decrem(X)))
endcase;
endfun;
{POST}

donde:

X representa a todos los datos de entrada.

$B_t(X)$ y $B_{nt}(X)$ son las condiciones de los casos trivial y no trivial, respectivamente.

Triv(X) es la solución al caso trivial.

Combinar(X,Y) es la operación principal del caso no trivial.

Decrem(X) es la operación que decrementa el tamaño de la entrada en la llamada recursiva. Corresponde el "sucesor" en un preorden.

Como en el caso iterativo, la verificación de un algoritmo recursivo se fundamenta en el principio de inducción matemática [13, 3] convenientemente adaptado a la recursión. Las condiciones a comprobar son:

- En primer lugar se comprueba si los datos cumplen la precondición, ajustándose a uno de los dos casos: trivial o no trivial.

$$PRE(X) \Rightarrow B_t(X) \vee B_{nt}(X)$$

- Los datos que cumplen la precondición y el caso no trivial deben seguir cumpliendo la precondición al ser 'decrementados', lo que propicia una nueva aplicación de la función recursiva.

$$PRE(X) \wedge B_{nt}(X) \Rightarrow PRE(Decrem(X))$$

- Base de la Inducción: Si los datos cumplen la precondición y el caso trivial, entonces Triv(X) debe cumplir la postcondición.

$$PRE(X) \wedge B_t(X) \Rightarrow POST(X, Triv(X))$$

- Paso inductivo: Si X cumple la precondición y el caso no trivial y Y' es la solución a Decrem(X), entonces X con resultado Combinar(X ,Y') cumple la postcondición.

$$PRE(X) \wedge B_{nt}(X) \wedge POST(Decrem(X), Y') \Rightarrow \\ \Rightarrow POST(X, Combinar(X, Y'))$$

- Existencia de una función f medidora del tamaño del problema y estrictamente decreciente en cada etapa:

$$PRE(x) \Rightarrow f(X) \geq 0$$
$$PRE(x) \wedge B_{nt}(X) \Rightarrow f(Decrem(X)) < f(X)$$

3 INTRODUCCIÓN AL PROBLEMA DE COLORACIÓN

3.1 Imágenes digitales

Una imagen digital puede modelizarse [8] con un grafo no dirigido y valorado cuyos nodos son los píxeles de la imagen y cuyas aristas se establecen entre nodos contíguos en la imagen a modo de cuadrícula. El valor asociado a cada arista será el grado de semejanza entre dos píxeles contíguos.

Definición 3.1 *Una imagen es un par (P, I) donde*

$$P = \{(i, j) : i \in \{1..r\}, j \in \{1..s\}\}$$

es el conjunto de píxeles e

$$I = \{\overline{x_{ij}} = (x_{ij}^1..x_{ij}^k) : (i, j) \in P\}$$

es el conjunto de medidas asociadas que caracterizan a cada píxel.

Así cada píxel tiene asociado un vector de características medibles por lo que podremos determinar la diferencia entre dos de ellos.

Definición 3.2 *Dada una imagen se define una medida de desemejanza o distancia entre cada par de píxeles por*

$$d : P \times P \rightarrow [0, \infty)$$

cumpliendo las propiedades de distancia.

En imágenes digitales dos píxeles son adyacentes cuando comparten una coordenada siendo la otra contigua.

Definición 3.3 *Grafo no dirigido asociado a una imagen (P, I) es terna $G = (P, E, d)$ siendo P el conjunto de vértices del grafo, d la función distancia entre los píxeles de P y E el conjunto de aristas definido por:*

$$E = \Big\{ \big((i, j), (i', j')\big) \in P \times P : \\ \big((i = i') \wedge |j - j'| = 1\big) \vee \big((j = j') \wedge |i - i'| = 1\big) \Big\}.$$

Habitualmente, los grafos valorados se representan mediante una matriz de diferencias D de dimensión $r \times s$. Sin embargo, las características de los grafos asociados a imágenes digitales permiten reducir esta información almacenando sólo los costes de las aristas existentes: E. Las matrices D^1 y D^2 se utilizan para este propósito.

Definición 3.4 *La matriz D^1 de diferencias en aristas verticales es aquella cuyos elementos son*

$d_{ij}^1 = d((i, j), (i+1, j)) \forall (i, j) \in \{1, \ldots, r-1\} \times \{1, \ldots, s\}$.

Análogamente, la matriz D^2 de diferencias en aristas horizontales es aquella cuyos elementos son

$$d^2_{ij} = d((i,j),(i,j+1)) \forall (i,j) \in \{1,\ldots,r\}\times\{1,\ldots,s-1\}$$

Definido el grafo asociado a una imagen digital, es necesario introducir el concepto de c-coloración [8] para un grafo general.

Definición 3.5 *Una c-coloración de un grafo $G = (V,E,d)$ es una función $C : V \to \{0..c-1\}$ tal que*

$$\{u,v\} \in E \Rightarrow C(u) \neq C(v)$$

Una c-coloración induce una clasificación nítida del conjunto de nodos:

$$V_c(k) = \{v \in V / C(v) = k\}$$

que denota el conjunto de nodos a los que se les asigna color k. La clasificación en c clases será

$$\{V_c(0), V_c(1), ..., V_c(c-1)\}$$

Si $c = 2$ se dice que es una coloración binaria.

Definición 3.6 *Un grafo difuso es una tupla $\widetilde{G}$ =$(V,E,\mu_{\widetilde{E}})$ donde V es un conjunto de vértices, E es un conjunto de aristas en $V \times V$ y*

$$\mu_{\widetilde{E}} : V \times V \to M$$

siendo M un conjunto cualquiera totalmente ordenado y

$$\mu_{uv} = \mu_{\widetilde{E}}(u,v)$$

representa el grado de intensidad de la arista $\{u,v\}$.

Este concepto generaliza al concepto de grafo nítido cuando μ_{uv} es siempre 0 o 1.

Habitualmente se escoge $M = [0,1]$ con el orden natural en los números reales.

Si el grado de intensidad de una arista es 1, entendemos que los vértices están conectados perfectamente. Puesto que se miden las diferencias entre píxeles y no su afinidad, las aristas con grado de intensidad 1 representan la posibilidad de asignar igual color a ambos vértices, es decir, los vértices extremos de dicha arista podrán ser clasificados en el mismo grupo.

Una primera aproximación a la coloración del grafo difuso consiste en realizar coloraciones en grafos nítidos, llamados α-cortes, de aproximación al grafo difuso según un parámetro α.

Definición 3.7 *Dado un grafo difuso $\widetilde{G}$ y dado α se define un α-corte como un grafo nítido $G_\alpha = (V, E_\alpha)$ donde $E_\alpha = \{\{u,v\} / \mu_{uv} \geq \alpha\}$*

Los grafos asociados así a una imagen digital son todos bipartitos y por lo tanto 2-coloreables al ser su topología sin ciclos de longitud impar, pero debemos encontrar la forma adecuada de realizar la bipartición.

Un primer coloreado del grafo difuso $\widetilde{G}$ surge a partir del coloreado de un α-corte: G_α. La coloración del grafo nítido G_α debe responder a las ecuaciones:

$$\begin{array}{l} \forall (i,j) \in \{1..r-1\}\times\{1..s\}: \\ \quad col(i+1,j) = \left\{ \begin{array}{ll} col(i,j) & si\ d^1_{ij} < \alpha \\ 1 - col(i,j) & si\ d^1_{ij} \geq \alpha \end{array} \right. \\ \forall (i,j) \in \{1..r\}\times\{1..s-1\}: \\ \quad col(i,j+1) = \left\{ \begin{array}{ll} col(i,j) & si\ d^2_{ij} < \alpha \\ 1 - col(i,j) & si\ d^2_{ij} \geq \alpha \end{array} \right. \end{array}$$

con dato inicial $col(1,1) = 0$.

Una vez coloreados los píxeles (i,j), $(i-1,j)$, y $(i,j+1)$, el píxel $(i+1,j+1)$ puede ser coloreado desde ambas ecuaciones recurrentes por lo que puede provocar inconsistencias si se obtiene una coloración diferente del píxel dependiendo desde donde ha sido coloreado.

Definición 3.8 *Un cuadrado de píxeles con extremo superior izquierdo (i,j) es el conjunto de píxeles*

$$sq(i,j) = \{(i,j),(i+1,j),(i,j+1),(i+1,j+1)\}.$$

Interesa que todos los cuadrados de píxeles en que se divide la imagen digital sean consistentes.

Definición 3.9 *Se dice que un cuadrado de píxeles $sq(i,j)$ es consistente a nivel α si el píxel (i+1,j+1) se colorea de igual forma recurriendo a $col(i,j+1)$ que a $col(i+1,j)$.*

Un grafo G_α puede ser inconsistente para un valor α y consistente para otro.

Definición 3.10 *Se dice que un grafo difuso $\widetilde{G} = (V,E,\mu_{\widetilde{E}})$ es consistente a nivel α si cada cuadrado de píxeles lo es.*

Observamos que si

$$\overline{\alpha} = \max\left\{ \max_{1\leq i\leq r-1, 1\leq j\leq s}\{d^1_{ij}\}, \max_{1\leq i\leq r, 1\leq j\leq s-1}\{d^2_{ij}\} \right\}$$

cualquier $\alpha > \overline{\alpha}$ produce una coloración unaria (todos los píxeles se colorean de igual forma, al no existir aristas en G_α)

Por otra parte si

$$\alpha < \underline{\alpha} = \min\left\{ \min_{1\leq i\leq r-1, 1\leq j\leq s}\{d^1_{ij}\}, \min_{1\leq i\leq r, 1\leq j\leq s-1}\{d^2_{ij}\} \right\}$$

el α -corte posee todas las aristas por lo que se colorea como un tablero de ajedrez.

En consecuencia sólo interesan los valores de α comprendidos en el intervalo $[\underline{\alpha}, \overline{\alpha}]$. Entre ellos, conviene escoger alguno que no presente inconsistencias.

Definición 3.11 *Se define el Nivel de Consistencia asociado al grafo como el mayor valor α del intervalo $[\underline{\alpha}, \overline{\alpha}]$ para el cual el grafo difuso es consistente.*

4 ESPECIFICACIÓN DE ALGORITMOS ASOCIADOS

Para calcular el nivel de consistencia de un grafo asociado a una imagen digital –sabiendo que si la imagen es finita, el nivel de consistencia siempre existe– se emplea un algoritmo que recibe como entrada la dimensión de la imagen ($r \times s$), las matrices de diferencias verticales y horizontales y una matriz $r \times s$ de booleanos que informa de la activación de cada píxel (i,j). Al comienzo todos los píxeles están activados pero puede interesar el estudio de una zona concreta de la imagen por lo que se añade la posibilidad de desactivación.

La función ***initalpha*** calcula el intervalo $[\underline{\alpha}, \overline{\alpha}]$:

fun **initalpha** (r,s:nat;
D_1:*Matriz[1..r-1, 1..s] of real*;
D_2:*Matriz[1..r, 1..s-1] of real*;
act:*Matriz[1..r, 1..s-1] of 0..1*)
ret $(\underline{\alpha}, \overline{\alpha})$:(*real*× *real*);
{PRE: Cierto}
{POST:

$$\overline{\alpha} = \max\left\{ \max_{1\le i\le r-1, 1\le j\le s}\{d^1_{ij}\}, \max_{1\le i\le r, 1\le j\le s-1}\{d^2_{ij}\} \right\}$$

$$\wedge$$

$$\underline{\alpha} = \min\left\{ \min_{1\le i\le r-1, 1\le j\le s}\{d^1_{ij}\}, \min_{1\le i\le r, 1\le j\le s-1}\{d^2_{ij}\} \right\}$$

}

La función ***inconsis*** devuelve un booleano a *Cierto* cuando el cuadrado de píxeles dado es inconsistente respecto a las ecuaciones de coloración dadas:

fun **inconsis** (α: *real*; $d^1_{ij}, d^2_{ij}, d^1_{ij+1}, d^2_{i+1j}$:*real*)
ret b:*bool;*
{PRE: Cierto}
{POST:
$b = ((x_1 + x_2) mod 2) \wedge (x_1 = (y_1 + y_2) mod 2)$
$\wedge\, (x_2 = (y_3 + y_4) mod 2) \wedge (y_1 = eval(d^1_{ij} \ge \alpha))$
$\wedge\, (y_2 = eval(d^2_{i+1j} \ge \alpha)) \wedge (y_3 = eval(d^2_{ij} \ge \alpha))$
$\wedge\, (y_4 = eval(d^1_{ij+1} \ge \alpha))\}$

Para calcular el nivel de consistencia de un grafo, se comprueba si todos los cuadrados del mismo son consistentes a nivel $\alpha = \overline{\alpha}$. Si existen inconsistencias, se rebaja este valor hasta encontrar el máximo α en $[\underline{\alpha}, \overline{\alpha}]$ para el cual todos los cuadrados sean consistentes. Este proceso se realiza mediante la función ***nivelcs***.

function **nivelcs**(r, s: nat;
D_1:*Matriz[1..r-1, 1..s] of real*;
D_2:*Matriz[1..r, 1..s-1] of real*;
act:*Matriz[1..r, 1..s-1] of 0..1*)
ret nivel: *real*;
{PRE: Cierto}
{POST: $\underline{\alpha} \le nivel \le \overline{\alpha} \wedge$
$\forall\, (i,j) : (1 \le i \le r-1) \wedge (1 \le i \le s-1) :$
$\neg$ (**inconsis**($nivel$,$d^1_{ij}, d^2_{ij}, d^1_{ij+1}, d^2_{i+1j}$)
$\wedge\, \forall \alpha > nivel : \exists\, (i,j) : (1 \le i \le r-1) \wedge$
$(1 \le i \le s-1)$: **inconsis**(α,$d^1_{ij}, d^2_{ij}, d^1_{ij+1}, d^2_{i+1j}$) }

La postcondición indica que el valor calculado "*nivel*" es un nivel para el cual todos los cuadrados de la imagen son consistentes, y además que este nivel es el máximo con esta propiedad dentro del intervalo $[\underline{\alpha}, \overline{\alpha}]$.

Así para determinar el nivel de consistencia de todo el grafo hay que determinar los niveles de consistencia de los cuadrados de píxeles en particular. Este cálculo local se realiza desde nivelcs invocando a la función newalpha que resuelve el proceso relativo a un cuadrado de píxeles concreto.

fun **newalpha** (α:*real*;
i,j:nat;
D_1:*Matriz[1..r-1, 1..s] of real*;
D_2: *Matriz[1..r, 1..s-1] of real*)
ret α':*real*;
{PRE: **inconsis**(α,$d^1_{ij}, d^2_{ij}, d^1_{ij+1}, d^2_{i+1j}$)}
{POST: $\neg$ **inconsis**(α',$d^1_{ij}, d^2_{ij}, d^1_{ij+1}, d^2_{i+1j}$) $\wedge$
$(\forall \alpha'' > \alpha'$: **inconsis**($\alpha''$,$d^1_{ij}, d^2_{ij}, d^1_{ij+1}, d^2_{i+1j}$)}

Ejemplo 4.1 *Sea la imagen de 9×12 píxeles:*

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	*	*	*	0	0	*
0	0	*	*	*	0	0	*
0	0	*	*	*	0	0	*
0	0	0	0	0	+	+	0
0	0	0	0	0	+	+	0
+	+	0	0	0	+	+	0
+	+	0	0	0	0	0	0

donde aparecen píxeles de tres tipos marcados con los símbolos: 0, +, y que representan puntos distintos pero muy parecidos.*

Las matrices asociadas D^1 y D^2 son:

$$D^1 = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.2 & 0.2 & 15 & 14 & 15 & 0.2 & 0.1 & 20 \\ 0.1 & 0.2 & 0.2 & 0.1 & 0.2 & 0.1 & 0.2 & 0.2 \\ 0.2 & 0.1 & 0.1 & 0.2 & 0.1 & 0.2 & 0.1 & 0.1 \\ 0.1 & 0.2 & 17 & 16 & 14 & 43 & 50 & 16 \\ 0.1 & 0.2 & 0.1 & 0.2 & 0.1 & 0.1 & 0.2 & 0.1 \\ 47 & 49 & 0.2 & 0.1 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.1 & 0.2 & 0.1 & 0.2 & 49 & 49 & 0.1 \end{pmatrix}$$

$$D^2 = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.1 & 0.2 \\ 0.1 & 15 & 0.1 & 0.1 & 17 & 0.2 \\ 0.2 & 16 & 0.1 & 0.1 & 20 & 0.2 \\ 0.2 & 15 & 0.1 & 0.1 & 18 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 49 & 0.1 \\ 0.2 & 0.1 & 0.2 & 0.2 & 50 & 0.1 \\ 0.1 & 49 & 0.2 & 0.2 & 52 & 0.1 \\ 0.1 & 49 & 0.2 & 0.2 & 0.3 & 0.3 \end{pmatrix}$$

Aproximadamente, la distancia entre los píxeles de los tipos 0, es 16, entre los píxeles 0,+ es 50 y la combinación +,* no se aprecia al no aparecer dos píxeles contiguos de estos tipos. Por simple inspección, el nivel de consistencia debe buscarse en el intervalo* $[\underline{\alpha}, \overline{\alpha}] = [0.1, 52]$.

El comportamiento de algunos cuadrados es:

- *El cuadrado* $sq(1,1)$ *es inconsistente a nivel* $\alpha \in (0.1, 0.2]$ *y consistente para el resto de valores.*
- *El cuadrado* $sq(2,2)$ *es consistente para cualquier valor de* α *porque sus aristas son simétricas:*

$$d^1_{22} = d^2_{22} = 0.2 \text{ y } d^1_{23} = d^2_{32} = 15$$

- *El cuadrado sq(5,8) presenta la clasificación:*

nivel α	consistente?	aristas
$\alpha \leq 0.1$	si	4
$\alpha \in (0.1, 0.2]$	no	3
$\alpha \in (0.2, 16]$	si	2
$\alpha \in (16, 20]$	no	1
$\alpha \in (20, \infty]$	si	0

Por último, puede comprobarse que el nivel de consistencia de esta imagen es $\alpha^* = 43$.

Del análisis del ejemplo 4.1, se desprende que el cálculo del nivel de consistencia no es inmediato. Una estrategia efectiva consiste en fijar $\overline{\alpha}$ como nivel de partida e ir verificando la consistencia de cada cuadrado del grafo para ese nivel. Cuando algún cuadrado sea inconsistente, se rebaja este nivel inicial hasta que el cuadrado sea consistente y se continúa con la comprobación de los restantes cuadrados para el nivel. Una vez inspeccionados todos los cuadrados, se repite el proceso anterior comenzando por el valor rebajado de α hasta encontrar un valor para el que todos los cuadrados sean consistentes.

5 UN COLOREADO DEL GRAFO DIFUSO

La función **Coldif** lleva a cabo el método de clasificación de los píxeles del grafo difuso, llamado coloración cruda. Se trata de un proceso por etapas en las que invoca al subprocedimiento **bincol** que realiza una clasificación binaria del conjunto de píxeles activados en cada etapa.

fun **bincol** (r,s:nat; α:real;
D_1: *Matriz[1..r-1, 1..s] of real*;
D_2: *Matriz[1..r, 1..s-1] of real*;
act: *Matriz[1..r,1..s] of 0..1*)
ret col: *Matriz[1..r,1..s] of nat*;
{PRE: }
{POST: La matriz col representa los colores asignados a los píxeles (i,j) con act(i,j)=1 de forma que los cuadrados de píxeles activados son consistentes para el α dado. }

La herramienta de cálculo simbólico MAPLE permite implementar este algoritmo de forma rápida y visual, lo que resulta muy útil para verificar el comportamiento de estos algoritmos mediante pruebas en las relaciones lógicas entre las variables y los datos.

Suponiendo comprobado el funcionamiento de las funciones auxiliares, puede verificarse el proceso **Coldif**.

fun **Coldif** (r, s, k, itc:nat; α:real;
D_1: *Matriz[1..r-1, 1..s] of real*;
D_2: *Matriz[1..r, 1..s-1] of real*;
act: *Matriz[1..r,1..s] of 0..1*)
ret Col,C: *Matriz[1..r,1..s] of nat*;
{PRE: $(\forall(i,j) : (1 \leq i \leq r) \wedge (1 \leq j \leq s) :$
$(C[i,j] = itc \Rightarrow act[i,j] = 1) \wedge$
$(0 \leq itc \leq 2^{k-1})) \wedge$
$(\alpha = \mathbf{nivelcs}(r, s, D_1, D_2, act) \wedge$
$(\forall(i,j) : (1 \leq i \leq r) \wedge (1 \leq j \leq s) : (C[i,j] \in [0 \ldots itc])\}$

case $itc \neq 2^{k-1} \rightarrow$
let
$newact \leftarrow \mathbf{ActivarDesactivar}(C, itc, act)$
$C \leftarrow \mathbf{Actualizar}(C, itc, act)$
$\alpha \leftarrow \mathbf{nivelcs}(r, s, D_1, D_2, newact)$
in
case $\alpha > 0 \rightarrow$
$\mathbf{Coldif}(r, s, k, itc + 1, \alpha, C,$
$newact, \mathbf{Bincol}(r, s, D_1, D_2, newact))$
case $\alpha = 0 \rightarrow$
$\mathbf{Coldif}(r, s, k, itc + 1, \alpha, C, newact, Col)$
endcase
case $itc = 2^{k-1} \rightarrow nada$
endfun
{POST: $(\forall(i,j) : (1 \leq i \leq r) \wedge (1 \leq j \leq s) :$
$(C[i,j] \in [0 \ldots itc])$ }

La inicialización adecuada para este proceso es:

$\mathbf{Coldif}(r, s, k, 0, \mathbf{nivelcs}(r, s, D_1, D_2, act),$
$D_1, D_2, Unos(act), Ceros(Col), Ceros(C))$

donde las funciones "Unos" y "Ceros" inicializan las matrices argumento a unos y ceros respectivamente.

6 VERIFICACIÓN DEL PROCESO DE COLOREADO

En general, la verificación formal de algoritmos proporciona técnicas de depuración y localización de errores.

El primer paso para la verificación formal es casi trivial: la precondición de **Coldif** contiene la cláusula $(0 \leq itc \leq 2^{k-1})$ que garantiza la ejecución de una de las dos ramas de la función.

El tamaño del input viene determinado por el tamaño del rango de etapas $[itc \ldots 2^{k-1}]$ que gracias a la expresión $itc + 1$ se garantiza su decrecimiento en cada llamada a **Coldif**.

La base de la inducción matemática se cumple: Si consideramos la precondición y el caso trivial tendremos:

$$PRE_{Coldif} \wedge (itc = 2^{k-1})$$

lo que garantiza directamente la postcondición $POST_{Coldif}$ debido a que la precondición contiene una cláusula idéntica a la única cláusula que constituye la postcondición:

$$(\forall(i,j) : (1 \leq i \leq r) \wedge (1 \leq j \leq s) : (C[i,j] \in [0 \ldots itc])$$

Dada la sencillez de la postcondición propuesta para el algoritmo el paso inductivo de la prueba se verifica de forma casi trivial: Si por hipótesis de inducción suponemos que se cumple el predicado de la postcondición tras la ejecución con itc que es la variable de control del número de etapas ocurridas, y si estamos en condiciones de volver a ejecutar una llamada a **Coldif**, realizando las instrucciones de avance: let, $itc + 1, \ldots$ llegamos a un estado de cómputo en el que se produce la postcondición para la etapa siguiente $itc + 1$.

El problema de la parada está garantizado al ser finito el rango de etapas.

7 CONCLUSIONES

Podemos formalizar la corrección de los algoritmos nítidos y difusos sobre coloración-clasificación haciendo uso de la verificación formal de algoritmos extendida a la lógica difusa.

La verificación de algoritmos puede conducir a la depuración de algoritmos iniciales obtenidos de forma intuitiva. Los resultados óptimos de la verificación formal se obtienen a partir de una rigurosa documentación sobre el comportamiento lógico del algoritmo y con la ayuda de un sistema de cálculo simbólico como Maple que proporciona un medio adecuado para la ejecución de este tipo de algoritmos al mismo tiempo que permite un enfoque algebraico del problema basado únicamente en operaciones matriciales.

Referencias

[1] A. Amo, J. Montero y G. Biging: *Relevancia y redundancia en sistemas de clasificación difusa.* En: Proceedings Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF 2000; pp. 521-525 (2000).

[2] A. Amo, D. Gómez, J. Montero and G. Biging: Relevance and redundancy in fuzzy classification systems. Mathware and Soft Computing 8, 203-216 (2001)

[3] J.L. Balcázar, *Programación Metódica*, Ed.: McGraw Hill 1993.

[4] H.K. Berg y otros. Formal Methods of Program Verification and Specification. *Ed. Prentice Hall*, New Jersey, 1982.

[5] J. C. Bezdek y J. D. Harris: Fuzzy Partitons and Relations: an axiomatic basis for clustering, *Fuzzy Sets and Systems 1*, 111-127, 1978.

[6] E. Binaghi, P. Madella, M. G. Montesano y A. Rampini: Fuzzy contextual classification of multisource remote sensing images, *IEEE Transactions on Geoscience and Remote Sensing 35*, 326-339, 1997.

[7] M. Broy y B. Krieg. Derivation Of Invariant Assertions During Program Development by Transformation. *ACM Transactions on Programming Languages and Systems*, Vol. 2, N 3, 321–327 1980.

[8] D. Gómez. *Algunas aportaciones sobre relaciones de preferencia.* Tesis doctoral, UCM, 2003.

[9] A. Kaldewaij *Programming: The Derivation of Algorithms*, Ed.:Prentice Hall International, 1990.

[10] V. López y L. Martín *Coloreado de grafos difusos en teledetección*, Ed.:ICE UPM, 85–100, Madrid 2003.

[11] Z. Manna. The Correctness of Programs. *Journal of Computer and System Sciences*, Vol. 3, N 2, 1969.

[12] F. Melgani, A. R. Bakir, A. Hashemy y S. M. R. Taha: An Explicit Fuzzy Supervised Classification Method for Multispectral Remote Sensing Images, *IEEE Transactions on Geoscience and Remote Sensing 38*,287-295, 2000.

[13] R. Peña Marí, *Diseño de Programas, Formalismo y Abstracción*, Ed.: Prentice Hall, 1999

SOFTWARE GENÉRICO DE CONTROL DIFUSO: APLICACIÓN EN AGRICULTURA INDUSTRIAL

Calixto Escobar
Ingeniero Técnico Industrial en Electrónica por la Universidad de Málaga, España
calixto@auna.com

José Galindo
Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, España
ppgg@lcc.uma.es

Resumen

En el presente documento se describe la aplicación informática Sistema de Control Difuso (SCD), como plataforma que permite simular algunos tipos de sistemas de control basados en lógica difusa. Este software proporciona un entorno intuitivo y claro para la creación y modificación de los elementos que conforman un sistema de control basado en conocimiento difuso. La base de conocimiento de SCD está formada por un conjunto de reglas del tipo condicional <si…entonces…> siendo posible la elección de operadores conectivos y de negación entre antecedentes. El programa es muy flexible, visualizando de forma gráfica las características de cada variable junto a los datos de simulación. Este software puede aplicarse a cualquier ámbito de la simulación industrial. En nuestro caso lo hemos aplicado al control climático en invernaderos industriales, como una posible herramienta para solucionar los distintos problemas que subyacen en el ámbito de los sistemas de control de producción agrícola. En síntesis, aquí presentamos un software genérico de simulación de control difuso, que se ha aplicado al funcionamiento de un sistema particular basado en conocimiento difuso.

Palabras clave: Controlador Difuso, Lógica Difusa, Controlador Basado en Conocimiento Difuso, Software de Control Difuso, Sistema Basado en Reglas, Agricultura.

1. INTRODUCCIÓN

Desde que Mamdani [9] mostrase la primera aplicación de la Lógica Difusa al control de un proceso particular, una extensa gama de aplicaciones, desde electrodomésticos a robots y sistemas de control industrial [6,7,10] han sido desarrollados bajo las premisas del Control Difuso. En la actualidad podemos encontrar una gran variedad de software [1,3,11], capaz de controlar procesos y simularlos (Xfuzzy, FuzzyTech, Matlab Fuzzy Logic Toolbox, por citar sólo algunas de las más recientes, junto a otros mencionados en www.eusflat.org).

El software que aquí presentamos trata de ser una aportación con sustanciales mejoras respecto a otras aportaciones (como la presentada en [3]). Cubre la finalidad de poder crear, de la forma más general posible, sistemas basados en conocimiento difuso. Además, la aplicación de la lógica difusa a los sistemas de producción agrícola [2, 4] es un enfoque adecuado en la búsqueda de soluciones pues serviría como herramienta para "almacenar" conocimiento experto de ayuda a la decisión. Por este motivo hemos aplicado el programa al control climático en un invernadero dedicado al cultivo de pimientos en la zona del sureste español (Almería). Por supuesto, el programa puede también aplicarse al control de otros aspectos de la agricultura (fertirrigación, aplicación de tratamientos fitosanitarios…).

En primer lugar expondremos una descripción del software listando sus características principales y resumiendo su funcionamiento (Sección 2). Seguidamente mostraremos las fases de creación del sistema de control en la aplicación (Sección 3). Posteriormente expondremos los pasos a la hora de simular el sistema y para ello veremos las distintas fases que realiza la aplicación para obtener las acciones de control (Sección 4). Finalmente, presentamos unas conclusiones, algunos trabajos futuros y las referencias bibliográficas.

2. CARACTERÍSTICAS PRINCIPALES DEL SOFTWARE SCD

SCD [5] trata de ser una plataforma genérica para el desarrollo y simulación de sistemas de control difuso. El programa permite definir un sistema (variables, etiquetas, reglas...) y simular dicho sistema para una secuencia de valores de entrada y obtener la correspondiente secuencia de acción de control (salida). SCD ha sido programado como aplicación multidocumento (MDI) de forma que sea posible manipular múltiples ventanas de forma simultánea.

El primer menú, *Archivo*, contiene las opciones típicas de apertura, cierre, guardar… necesarias para la gestión de los ficheros de la aplicación. Seguidamente se encuentra el menú *Ver*, el cual permite al usuario mostrar las ventanas *General* y de *Reglas*. La ventana *General* (Figura 1) muestra, en estructura de árbol, toda la información referente al fichero actualmente cargado. El menú *Variables* proporciona las opciones sobre las variables de entrada y salida (creación, modificación, eliminación o restauración). Igualmente, el menú *Reglas* permite añadir, modificar, eliminar, restaurar u organizar las reglas en la base de conocimiento. En el menú *Inferencia* se encuentran las distintas opciones de simulación que posteriormente comentaremos en la Sección 4 junto a la edición de las opciones de inferencia.

Las principales características de SCD son:

- **Redimensionable en tiempo de ejecución** en el número de variables de entrada, salida, reglas, conjuntos difusos de cada variable y antecedentes y consecuentes en cada regla.
- **Definición rápida, cómoda y flexible** de variables, reglas, etiquetas…
- **7 tipos de Conjuntos Difusos:** Gamma, L, Singleton, Intervalo, Triángulo, Trapecio y Trapecio Extendido. Estas posibilidades son muy superiores a otros programas de este tipo.
- **Trapecio Extendido:** Este novedoso tipo de conjunto difuso es una función lineal a trozos que permite gran flexibilidad (incluso permite conjuntos no convexos) y eficiencia. Es redimensionable en el número de puntos de inflexión de dicha función.
- **5 tipos de Difuminadores** (para los valores de las variables de entrada): Singleton, Intervalo, Triángulo, Trapecio y Trapecio Extendido.
- **Universo de Discurso**: Tras modificaciones del Universo de una variable, existe la opción de mantener la proporcionalidad de los conjuntos difusos previamente definidos (o mantenerlos fijos).
- **Análisis Puntual y por Secuencia de Estados**: Para un valor concreto o para una secuencia temporal de datos.
- **Operador And**: T-normas de la Familia Dubois Prade, Frank, Yager, Mínimo, Producto Acotado, Algebraico, Drástico, Einstein y Hamacher.
- **Operador Or**: S-normas de la Familia Sugeno, Yager, Máximo, Suma Acotada, Drástica y Producto.
- **Operador Implicación**: T-norma Mínimo.
- **Operador Agregación**: S-norma Máximo.
- **Generador de tablas de resultados en formato HTML**: Esto no está incluido en otros programas de este tipo analizados.
- **Repres. gráfica de variables lingüísticas**.
- **Ficheros de datos en formato texto**.

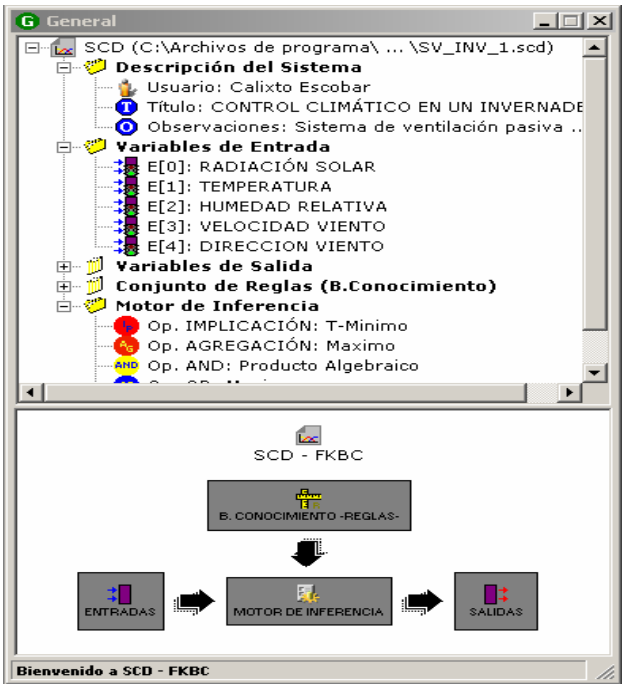


Figura 1: Ventana General de SCD.

- **12 tipos de Concresores** (para los valores de las variables de salida): Media de Máximos (MoM), Centro de Gravedad (CoG), CoG de mayor área, CoG ponderado por el área, CoG ponderado por la altura, CoG de mayor altura, Punto de Máximo Criterio (PMC) ponderado por el área, PMC ponderado por la altura, Media del PMC, Media del mínimo y máximo PMC, PMC de mayor área, y PMC de mayor altura.
- **Operador de Comparación**: T-norma del Mínimo.

3. CREACIÓN DE UN CONTROLADOR EN SCD

Un controlador queda definido en SCD cuando han sido declaradas las *variables de entrada* y de *salida,* el *conjunto de reglas* y las *opciones de inferencia*. Dado que el objetivo en un invernadero es conseguir un microclima idóneo para el desarrollo óptimo del cultivo, nos proponemos con esta idea simular el proceso de control climático del invernadero bajo SCD.

3.1 Declaración de Variables de Entrada

Como primer paso, se establecen las variables de entrada que son según los expertos [9]**:**

- **Radiación Solar.**
- **Temperatura.**
- **Humedad Relativa.**
- **Dirección del Viento.**
- **Velocidad del Viento.**

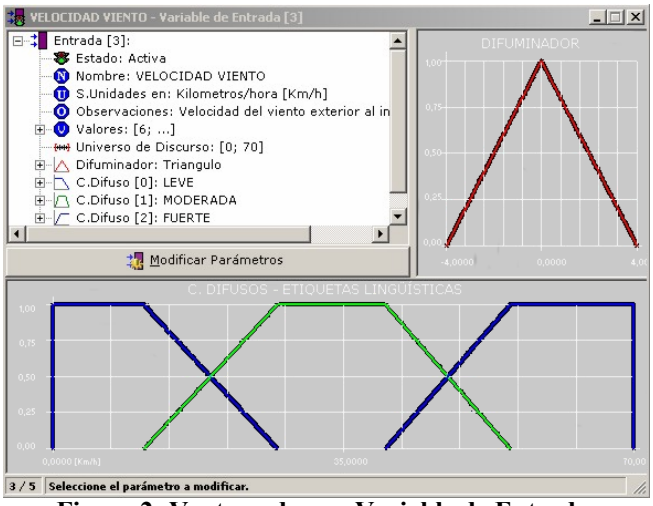


Figura 2: Ventana de una Variable de Entrada.

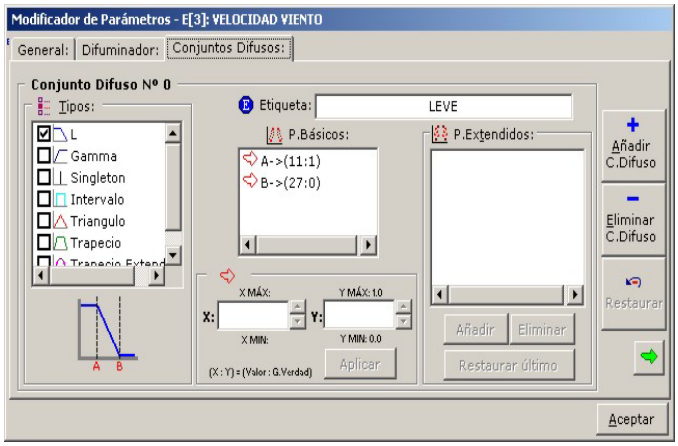


Figura 3: Modificar Parámetros de una Variable.

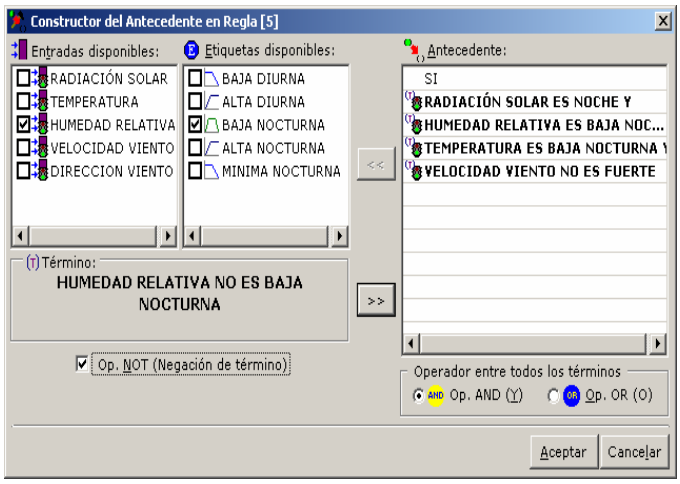


Figura 4: Construyendo el Antecedente en una Regla.

En la creación de una variable de entrada en SCD el usuario podrá optar por crear una nueva variable sin etiquetas lingüísticas o bien a semejaza de una ya existente. Esta última opción copia las etiquetas de otra variable y, posteriormente, podrán modificarse.

Una vez creada, se visualizará una ventana como la Figura 2 que mostrará toda la información concerniente a dicha variable visualizando en su parte superior izquierda a través de una estructura de árbol los valores actuales de sus parámetros. En la parte superior derecha se expone la representación gráfica del tipo de difuminador y en la parte inferior las de las etiquetas lingüísticas.

En este caso la Figura 2 muestra los datos de la variable *dirección del viento* que permiten establecer un nivel de seguridad estructural del invernadero (ante vientos fuertes). A través del botón *Modificar Parámetros* de esa Figura podrá variar el contenido de dicha variable. Para ello SCD visualizará una ventana (Figura 3) que mostrará las características de la variable agrupándolas en tres grupos: *general, difuminador* y *conjuntos difusos*. En la Figura 3 se visualiza la pestaña conjuntos difusos de la variable velocidad del viento, en concreto, la etiqueta lingüística *leve* de tipo L considerada por los expertos como un nivel de velocidad del viento hasta los 11 Km/h decreciendo con una pendiente del 16%.

3.2 Variables de Salida

En segundo lugar se establecerían las variables de salida, que en nuestro caso son las siguientes:

- **Ventilación Superior de Oeste y de Este.**
- **Ventilación Lateral del Oeste y de Este.**
- **Nebulización.**
- **Pantalla Térmica.**

Se ha considerado la ventilación a través de ventanales distribuidos sobre el invernadero para gestionar las renovaciones de aire, la nebulización como sistema de gestión de la humedad interior del invernadero y la utilización de la pantalla térmica como elemento de retención de energía calorífica durante la noche y de sombreo durante el día.

El método de concreción elegido para estas variables ha sido el *Centro de Gravedad ponderado por la Altura*. Este método calcula el centro de gravedad de cada uno de los conjuntos difusos resultantes del conjunto global de salida, recibiendo en base a su altura un peso distinto. Se ha recurrido a éste método, porque creemos que la única característica diferenciable entre los conjuntos difusos del tipo singleton son la altura que puedan tomar según el grado de activación de las reglas de control; al ponderar por la altura las reglas que activen en dichos conjuntos difusos tomarán su grado de importancia en el resultado global.

3.3 Base de Reglas

Una vez definidas las variables de entrada y salida pasamos al tercer paso que es la construcción de la **Base de Conocimiento**. Se han establecido las siguientes agrupaciones:

- **Reglas de Seguridad.**
- **Reglas de Control Nocturno.**
- **Reglas de Control Diurno.**

La Base de Conocimiento es una de las partes más importantes del sistema y es la que más trabajo cuesta a los expertos. Sin embargo, la inclusión de la Base de Conocimiento en el programa es relativamente rápida y sencilla a través de las ventanas constructoras de antecedente (Figura 4) y consecuente de cada regla. Estas ventanas muestran al usuario respectivamente todas las posibles variables de entrada y salida disponibles junto a los correspondientes conjuntos difusos asociados a cada una. Además, en la formación del antecedente existe la posibilidad de elegir entre operadores *and/or* entre términos, como se aprecia en la Figura 4.

3.4 Opciones de Inferencia

Por último se establecerían las **Opciones de Inferencia.** En el presente estudio se han tomado como operadores los mostrados en la Tabla 1. La elección del producto algebraico como operador *and* ha sido en base a lo expuesto en [12] en donde revela, a través de experimentos, que el operador del mínimo no trabaja bien como modelador del conectivo *and*.

Tabla 1: Opciones de Inferencia.

Operador	Tipo
Implicación	Mínimo
Agregación	Máximo
And	Producto Algebraico
Or	Máximo
Comparación	Mínimo

4. SIMULACIÓN DEL CONTROLADOR EN SCD

Una vez definido el controlador, nos disponemos a realizar un análisis y exposición de los resultados obtenidos a través de la simulación del sistema en SCD. Para ello se puede ejecutar la simulación de tres formas distintas. En una primera, la simulación puede ejecutarse paso a paso (Sección 4.1), obteniendo todos los datos intermedios de la inferencia. En segundo lugar, pueden también obtenerse los datos de salida directamente para un instante dado según unos valores de entrada establecidos (Sección 4.2). Así, de forma rápida se podrán comprobar las consecuencias ante la variación de los valores de entrada. Por último, tenemos la posibilidad de evaluar el sistema dada una secuencia de valores de entrada (Sección 4.3). De esta forma podríamos evaluar el sistema dado un registro de las variaciones de las distintas variables que actúan sobre el mismo, obteniendo también una secuencia en los valores de cada variable de salida.

4.1 Simulación Paso a Paso

El proceso de cálculo del controlador basado en conocimiento difuso en SCD realiza cinco fases para la obtención de las acciones de control. Tales fases pueden ser visualizadas a través de sus correspondientes ventanas.

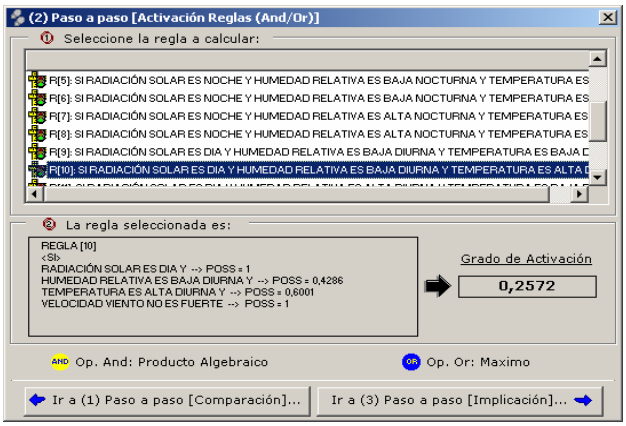


Figura 5: Fase 2, Activación de Reglas.

Estas cinco fases son:

1) **Fase de Comparación:** Este proceso compara los conjuntos difusos. Compara los valores de entrada (difuminados según el difuminador elegido) y los distintos conjuntos difusos que conforman las etiquetas lingüistas de la variable de entrada. Con esto se mide en qué medida el difuminador y las etiquetas se superponen a través del concepto de posibilidad (Poss).
2) **Fase de Activación de Reglas** (Figura 5)**:** Con los valores de posibilidad de cada término que forman los antecedentes en la base de conocimiento, se aplican los operadores and/or y not sobre estos términos en cada regla. Con esto se obtiene el *grado de activación* de cada regla.

   Todo este proceso se ilustra en la Figura 5, la cual muestra en su parte superior el conjunto de reglas de la base de conocimiento. El usuario podrá seleccionar aquella regla sobre la cual desee ver de forma descompuesta el proceso de obtención del grado de activación. Una vez seleccionada el recuadro central izquierdo mostrará las posibilidades de todos los términos que forman el antecedente de la regla y en la parte central derecha se muestra el valor del grado de activación correspondiente a la regla seleccionada junto a los tipos de operadores and/or utilizados.
3) **Fase de Implicación:** El proceso de implicación trabaja sobre las variables de salida del sistema. Aplica el operador de implicación sobre cada uno de los conjuntos difusos de cada variable de salida. Naturalmente, esto tiene en cuenta el grado de activación de las reglas activadas que influyan sobre dicho conjunto difuso de salida.
4) **Fase de Agregación:** Se realiza a través del operador de agregación (unión). En esta fase se unirán los conjuntos difusos resultantes de la fase de implicación. Esta es la fase que unifica los criterios expresados en todas las reglas, generando un conjunto difuso que expresa la información de todas las reglas.

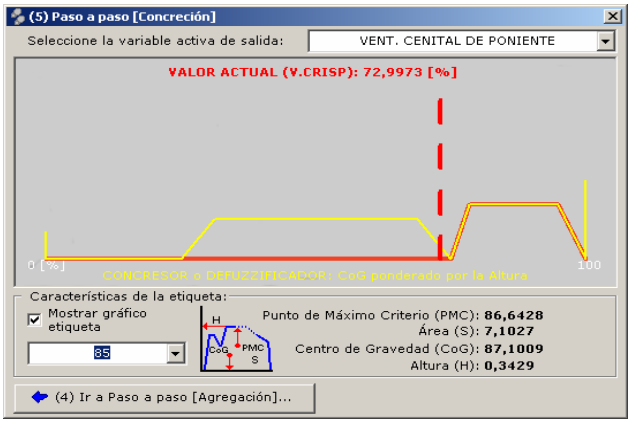


Figura 6: Fase 4, Concreción.

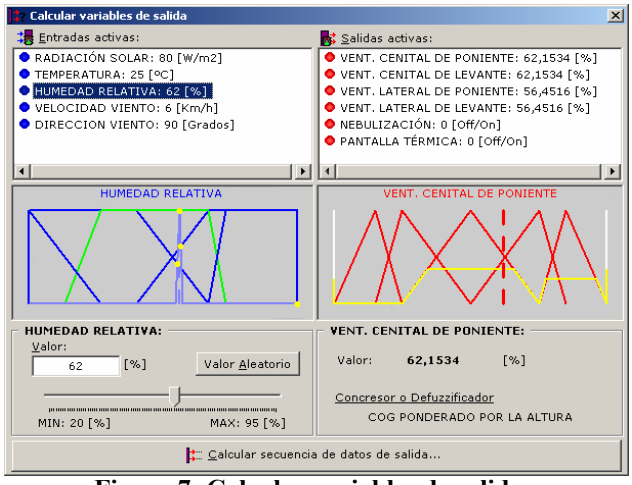


Figura 7: Calcular variables de salida.

5) **Fase de Concreción** (Figura 6)**:** En la última fase se procesa la información difusa (conjunto difuso resultante de la fase de agregación) y se obtiene un valor concreto como resultado físico, real, de la variable de salida. Como se ha dicho, son varios los métodos de concreción que el sistema permite utilizar.

En la Figura 6, una vez seleccionada la variable de salida, se visualizará el conjunto difuso resultante de la acumulación de cálculos de la Fase 4. Junto a esto, se muestra en trazo discontinuo el valor actual de salida obtenido en esta Fase 5 al aplicar el método de concreción a dicho conjunto difuso de salida.

4.2 Cálculo de las Variables de Salida para un Instante Concreto (o para un Valor Concreto de las Variables de Entrada).

La agrupación de todas las fases descritas anteriormente dan lugar al proceso global de cálculo. Una forma rápida de suministrar al usuario la respuesta del sistema a unos valores concretos de entrada, es por medio de la ventana de cálculo de variables de salida (Figura 7).

La ventana dispone de dos listados: la parte superior izquierda contiene las variables de entrada y la parte superior derecha, las variables de salida. Al seleccionar una variable de entrada los datos de la misma son mostrados gráficamente en el marco de la parte inferior. Se muestra el valor actual de la variable, los conjuntos difusos que la forman, el difuminador y los puntos de intersección. El valor actual podrá ser modificado y de forma automática se realizará el proceso de cálculo global que se indicará en el gráfico de salida de la derecha. Igual que se explicó en el apartado anterior, se muestra el valor concreto de salida (trazo discontinuo) junto al conjunto difuso resultante tras la Fase 4.

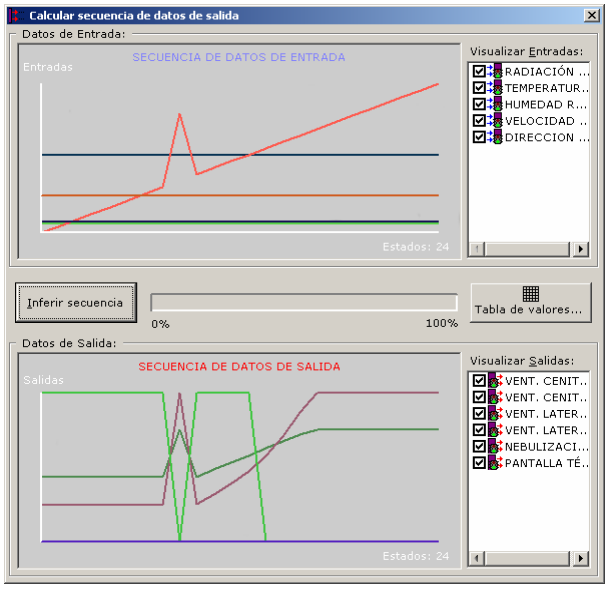


Figura 8: Calcular secuencia de datos de salida.

4.3 Simulación de Secuencias de Valores de Entrada

La posibilidad de evaluar el sistema en distintos instantes de tiempo (estados) va a proporcionar una forma de simular la evolución en la acción de control ante unas determinadas secuencias de datos. Esto nos permite también registrar el comportamiento del sistema a lo largo del tiempo, con el objetivo de tomar decisiones sobre el sistema (cambiar la base de reglas, o la definición de algunas etiquetas...). Esta opción es muy importante para el estudio del sistema, su evaluación y su afinamiento.

Dicha evaluación se realiza a través de la ventana calcular secuencias de datos de salida (Figura 8). La ventana se presenta con dos marcos de elementos. Por un lado, en la parte superior se encuentra el marco de datos de entrada el cual contiene en su parte derecha un listado con las variables de entrada. Aquellas que estén chequeadas serán visibles en el gráfico de la izquierda (secuencia de datos de entrada). El eje X de ese gráfico es el tiempo, medido en "instantes" en los cuales cada variable toma un valor particular. Lo comentado anteriormente para las variables de entrada es aplicable a las de salida (parte inferior).

En la Figura 8 puede verse cómo todas las variables de entrada son constantes menos la velocidad del viento. Un cambio brusco en la velocidad del viento, genera un

cierre inmediato de las ventanas. Tras ese cambio brusco, las ventanas vuelven a abrirse. Como la velocidad del viento sigue aumentando lentamente, llega un momento en el que las ventanas vuelven a cerrarse. Puede verse que el resto de variables de salida toma valores graduales según la velocidad del viento va modificando sus valores de entrada en los distintos instantes de tiempo.

5. CONCLUSIONES

Hemos presentado aquí un software de simulación de control difuso [5]. Una de las características más importantes es su flexibilidad que permite adaptarse a las posibles necesidades de cada usuario dado su carácter flexible y redimensionable. El interfaz permite, de una forma sencilla y amena, introducirse en las nociones básicas sobre controladores difusos y gracias a la evaluación paso a paso conocer de forma más detallada el funcionamiento de un controlador basado en conocimiento difuso. Además, las ventanas de avisos incluyen explicaciones detalladas. Por estos motivos creemos, que SCD podría ser una herramienta de carácter didáctico muy útil. El hecho de ser un programa gratuito y abierto [5] hace que sea, en este punto superior a otros programas, como los citados en la Introducción.

La búsqueda de la implantación de los principios de control difuso en los sistemas de producción agrícola ha sido la gran meta a buscar con la elaboración de este software. El control difuso sería una herramienta más que permitiría a agricultores, científicos… recopilar información de la problemática que los agrupa y obtener resultados productivos. Estos resultados productivos vendrían dados por una capacidad de "almacenamiento" del conocimiento experto de forma que una "maquina" pudiese servir de soporte de ayuda a la decisión.

La aplicación de SCD al control climático de un invernadero de pimientos nos ha permitido su simulación, de forma que podamos obtener información para mejorar el cultivo.

Como trabajos futuros se pretende implementar físicamente el controlador. El objetivo sería no sólo la simulación del sistema sino el control real del mismo por parte de SCD. Para ello se tendría que elaborar un sistema de adquisición de datos en tiempo real de las variables de entrada actuando consecuentemente sobre los dispositivos físicos que gestionen las variables de salida.

Por otra parte el sistema puede ampliarse incluyendo más funciones, especialmente para las opciones de inferencia (Tabla 1). No obstante eso no resulta tan necesario. Quizás es más importante incluir nuevos tipos de reglas que nos permitan expresar la base de conocimiento de forma más cómoda.

El trabajo presentado podría ser utilizado también en el sentido de *aproximador de funciones*. Por ejemplo, supongamos, en el ámbito de la agricultura, que una función *f* calcula la *evapotranspiración* de un cultivo a partir del tiempo transcurrido desde su plantación. Esa función es compleja de calcular matemáticamente. Sin embargo, usando los principios de control difuso podría ser factible calcular una aproximación a esa función a partir de reglas simples.

REFERENCIAS

[1] I. Alvarez López, S. Orestes Llanes. "Una aplicación de control difuso para secado de tabaco en hojas", Proyecto JA00/69 Consejería de Presidencia de la Junta de Andalucía, 2002.

[2] A. M. G. Cornelissen, J. van den Berg, W. J. Koops, U. Kaaymak. "Eliciting Expert Knowledge for Fuzzy Evaluation of Agricultural Production Systems", November 2002, http://www.erim.eur.nl

[3] Oscar G. Duarte, G. Pérez. "Unfuzzy: Fuzzy Logic System análisis, design, simulation and implementation software". EUSFLAT-ESTYLF Joint Conference (European Society for Fuzzy Logic and Technology), pp. 251-254, Palma de Mallorca (Spain), September 1999.Univ. Nacional de Colombia: http://ohm.ingsala.unal.edu.co/ogduarte

[4] E. G. Dunn, J. M. Keller, L. A. Marks. "Extending the application of fuzzy sets to the problem of agricultural sustainability". 3rd International Symposium on Uncertainty Modelling and Analysis, March 17-20, pp. 497, College Park, Maryland, 1995.

[5] C. Escobar. "Software para Control Difuso de Todo Tipo de sistemas (SCD): Aplicación al Control de Invernaderos Industriales". Proyecto Fin de Carrera de Ingeniería Técnica Industrial en Electrónica (Universidad de Málaga), dirigido por J.Galindo, 2003: http://www.lcc.uma.es/~ppgg/PFC

[6] J. Galindo. Curso sobre "Conjuntos y Sistemas Difusos (Lógica Difusa y Aplicaciones)". Informe Técnico de Docencia. LCC-ITI-2001-11, del Dpto. de Lenguajes y Ciencias de la Computación de la Univ. de Málaga, 2001 (www.lcc.uma.es).

[7] J. George, Klir: "Fuzzy Sets and Fuzzy Logics: Theory and Applications, NJ". Prentice Hall, 1995.

[8] P. Lorenzo, Dpto. de Horticultura y Und. Fisiología del Centro de Investigación y Formación Agraria (C.I.F.A. - Almería) Junta de Andalucía, fisiovht@arrakis.es

[9] E. H. Mamdani. "Applications of fuzzy algorithms for simple dynamic plant". Proc. IEE, Vol. 121, No. 12, pp. 1585-1588, 1974.

[10] A. Piegat, "Fuzzy Modeling and Control". Physica-Verlag (Studies in Fuzziness and Soft Comp.), 2001.

[11] Thiang, A. Hannawati, L. Resmana, H. Ferdinando. "PetraFuz: a Low Cost Embedded Controller Based Fuzzy Logic Development System", August 2001. http://ee.petra.ac.id/basiclab/petrafuz.htm

[12] H. J. Zimmermann, P. Zysno. "Latent connectives in human decision making". Fuzzy Sets and Systems 4, pp. 37-51, 1980.

LABORATORIO VIRTUAL DE APRENDIZAJE DE CONTROL FUZZY ADAPTATIVO

Agustín Rodríguez Herrero
arodri@sec.upm.es

Ángel M. Groba González
amgroba@sec.upm.es

José Ángel Herranz Martín
thebestjaherra@yahoo.es

David Galán López
davids79_es@yahoo.es

Escuela Universitaria de Ingeniería Técnica de Telecomunicación
Departamento de Sistemas Electrónicos y de Control
Universidad Politécnica de Madrid

Resumen

Se presenta en este artículo un entorno desarrollado en Labview de un sistema de control de motores mediante técnicas borrosas adaptativas. El entorno es un laboratorio virtual borroso (VFL) que está preparado para la enseñanza de reguladores borrosos (FLC) aplicada a los sistemas de control, para alumnos iniciados en control borroso.

Palabras Clave: fuzzy, adaptativo, motor, aprendizaje, didáctico, Labview.

1 MOTIVACIÓN DEL TRABAJO

La aplicación de la lógica borrosa (FL) a los sistemas de control fue exitosa desde su primera proposición, debido a que resuelve el problema de diseño cuando se desconoce el modelo de la planta totalmente, porque la planta no pueda ser modelada analíticamente; o parcialmente, porque el modelo presente incertidumbres. Un control clásico, con un regulador PID o Truxal [5], se basa para su diseño en el conocimiento de un modelo matemático y el éxito del control reside en la fidelidad de representación del comportamiento de la planta, por parte del modelo. Estas técnicas son dificilmente aplicables cuando se desconoce el modelo o parte de él o simplemente cuando el modelo presenta multitud de incertidumbres. Ante esta adversidad, la FL aplicada al control de sistemas encaja en cuanto a que sustituye el modelo analítico por unas relaciones borrosas entrada-salida y el diseño del regulador se basa en cuantificarlas.

En cuanto al diseño de los sistemas de control invariantes, tan sólo advertir de la gran carga matemática que necesita para su aplicación, (ecuaciones diferenciales o en diferencias, dominios transformados, operaciones de convolución,…). Si a esto se añade la intención de diseñar reguladores variantes, que varíen sus parámetros en base a una serie de objetivos, obviamente la matemática que se ha de manejar requiere del diseñador un buen conocimiento de técnicas de optimización, regresiones lineales, etc… Por tanto surge la necesidad de cómo acercar este tipo de diseño a una persona que no posea un bagaje matemático consolidado, necesidad que se intenta cubrir con el VFL.

Inevitablemente si lo que se intenta es enseñar técnicas de diseño adaptativas basadas en la función de transferencia o espacio de estados, es necesario tener un buen soporte de matemática de variable compleja y álgebra matricial, respectivamente. Pero, ¿se puede enseñar a controlar un sistema sin necesidad de aplicar tanta matemática? No se aventura una respuesta inmediatamente afirmativa, puesto que decir sí, con la aplicación de la FL puede ser contradictorio debido a su compleja formulación matemática.

Otras de las ventajas en el desarrollo de este trabajo fue la utilización del entorno de programación Labview de National Instruments (NI)[1], que posee un toolkit de FL, aunque muy rígido y poco adaptable a los propósitos de este trabajo. También se utiliza como apoyo el *Toolbox de fuzzy logic* de Matlab.

2 INTRODUCCIÓN

La FL permite el control de cualquier tipo de sistema sin necesidad de conocer su modelo analítico de comportamiento temporal, dentro de las limitaciones físicas propias del sistema, como saturación, tiempos de respuesta (dinámica), consumos, etc... conocidas también como **restricciones**. Además reduce los cálculos matemáticos en fase de diseño, lo que indudablemente, lleva a una reducción del tiempo de puesta en servicio del regulador.

El VFL es un entorno orientado al aprendizaje de los controladores borrosos adaptativos y a la consolidación de los conocimientos que el usuario disponga sobre los controladores borrosos invariantes (FLC). El VFL se aplicará sobre motores de corriente continua, para controlar la velocidad de giro, de los que se dispone en el Laboratorio Docente de Sistemas de Control del Departamento de Sistemas Electrónicos y de Control de la Escuela Universitaria de Ing. Téc. de Telecomunicación (a partir de ahora Laboratorio). Asimismo y como el

[1] http://www.ni.com

entorno está pensado para que el alumno pueda hacer sus pruebas fuera del Laboratorio, existe una opción de simulación de los motores a través de la ecuación en diferencias de la discretización de los modelos de los motores.

Para un buen aprovechamiento de la potencialidad didáctica que posee el entorno, el alumno debe tener una base consolidada en cuanto al diseño de sistemas de control invariantes como el método de Truxal, ajuste de Ziegler-Nichols o ubicación de polos a través del lugar de las raíces. Asimismo debe tener una base práctica sobre el álgebra borrosa en cuanto a operaciones básicas con funciones de pertenencia, cortes α y λ, inferencia borrosa y transformaciones de paso de números reales a borrosos y viceversa.

Se pretende con este trabajo, además de introducir al alumno en los sistemas de control adaptativos, presentar una técnica borrosa de control variante para ajustarse adecuadamente a los cambios de la dinámica de los motores. Estos cambios se pueden reflejar como cargas mecánicas a mover. Como objetivo de adaptación (figura 1) se pretende que independientemente de las cargas, los tiempos de respuesta en lazo cerrado sean siempre los mismos.

En primer lugar se realiza una justificación de por qué surgió la idea de este VFL, posteriormente se exponen las posibilidades del entorno en cuanto a FLC, para posteriormente justificar el controlador borroso adaptativo y por último se reflejan los resultados obtenidos.

3 BASE TEÓRICA

En este apartado se muestra el punto de partida del presente trabajo basado en el diseño de un regulador proporcional adaptativo por modelo de referencia (PMRA), cuando se utilizan las técnicas tradicionales de optimización de funciones. Si bien no sería necesario demostrar nada, los autores quieren dejar constancia de la aplicación matemática necesaria para la obtención del regulador.

3.1. CONTROLADOR PMRA

La técnica de diseño óptimo adaptativo permite el cambio de los parámetros del regulador para poder **adaptarse** a las variaciones de la dinámica de un motor y obtener en todo momento los objetivos fijados, utilizando para ello una función de coste donde se penalizan los efectos adversos que se desean controlar. Adaptarse es un cambio continuo del valor de los parámetros del regulador, para conseguir un objetivo. En concreto, en PMRA el objetivo es que la respuesta del sistema de control en lazo cerrado formado por la planta (cambiante) y regulador (cuyo diseño se hace variante), sea exactamente igual a la respuesta fijada por un modelo de referencia (M_0).

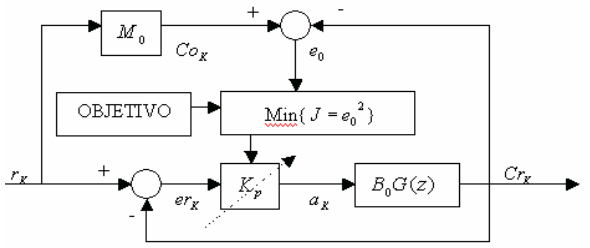


Figura 1: Control de un MCC con PMRA

La topología de la figura 1, incorpora un nuevo lazo respecto al sistema de control invariante, denominado lazo de adaptación, que tiene la misión de obtener un nuevo error e_0 (error de adaptación), que servirá como base para el ajuste de los parámetros del regulador.

Sea el equivalente discreto de la planta:

$$B_oG(z) = \frac{C_r(z)}{A(z)} = \frac{C_r(z)}{K_p \cdot E_r(z)} = \frac{K_g \cdot z^{-1}}{1 - P_g \cdot z^{-1}} \quad (1)$$

Donde $C_r(z)$ es la salida, $A(z)$ la acción de control, $E_r(z)$ el error de control, P_g el polo y K_g la ganancia.

El objetivo es que la planta controlada se comporte como la salida del modelo dado como referencia:

$$M_o(z) = \frac{C_o(z)}{R(z)} = \frac{K_o \cdot z^{-1}}{1 - P_o \cdot z^{-1}} \quad (2)$$

Siendo $C_o(z)$ la salida, P_o el polo y K_o la ganancia.

Hay que tener especial cuidado en elegir un modelo adecuado, puesto que se han de tener en cuenta las restricciones propias de la planta. Así por ejemplo, no podemos exigir un error de control (e_r) nulo ante escalón, puesto que la planta con un regulador P, nunca anularía ese error de control .

La optimización se basa en encontrar un valor de la ganancia proporcional, K_p del regulador que minimice una función de coste J (índice de funcionamiento):

$$J(K_p) = e_0^2 = \left(c_r(k) - c_o(k)\right)^2 \quad (3)$$

Sustituyendo en (3), (1) y (2):

$$J(K_p) = \left(K_g \cdot K_p \cdot e_r(k-1) + P_g \cdot c_r(k-1) - K_o \cdot r(k-1) + P_o \cdot c_o(k-1)\right)^2 \quad (4)$$

Derivándola para calcular su mínimo, y despejando K_p:

$$K_p = \frac{-P_g \cdot c_r(k-1) + K_0 \cdot r(k-1) + P_0 \cdot c_0(k-1)}{K_g \cdot e_r(k-1)} \quad (5)$$

Si se hubiesen realizado los cálculos para un regulador PID, que tiene tres parámetros, el cálculo matemático se vería incrementado considerablemente, una de las técnicas más usadas para su solución es la regla del MIT[2]. Se puede llegar a una situación de no encontrar una solución al problema planteado.

3.2. CONTROLADOR BORROSO (FLC)

Se supone al lector familiarizado con esta técnica de control por lo que sólo se hará una breve referencia sobre el funcionamiento, [3], [7] (pp. 141-150). La primera etapa es transformar el universo de entrada en un dominio borroso donde puedan modelarse el comportamiento como funciones de pertenencia a unos conjuntos borrosos de entrada. La siguiente etapa está formada por la inferencia, que no es más que la aplicación de las reglas diseñadas para controlar la planta, por ejemplo: *If error is alto then salida is menor.* Se aplica la técnica MAX-MIN para obtener la evaluación de la inferencia. La etapa final en la versión Mamdani[3] está formada por unos conjuntos de salida, se obtiene el número real de salida con el método del centro de gravedad (COG).

Se ha confeccionado (figura 2) una primera prueba para que el alumno refresque conocimientos básicos sobre FLC, como es por ejemplo cómo afecta la anchura de los conjuntos a la respuesta en el tiempo y su comparación con un regulador proporcional, que es lo que más conoce.

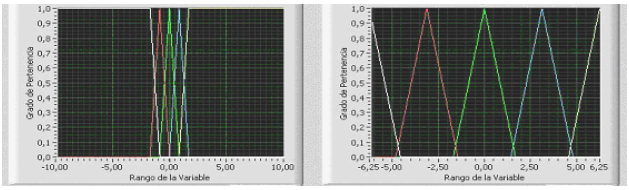


Figura 2: Izquierda: conjuntos de entrada relativos al error. Derecha: conjuntos de salida relativos a la acción de control.

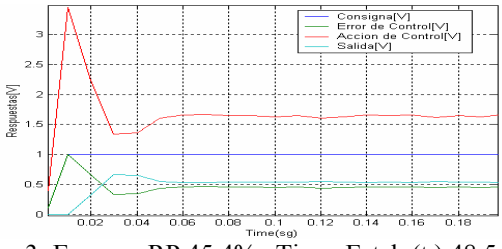


Figura 3: Error en RP 45.4%, Tiem. Estab.(t_s) 48.5 mseg

Según la figura 2, se definen 5 conjuntos de entrada, siendo los de ambos extremos de la mitad de anchura que los 3 centrales, todos son triangulares y se solapan en un 50%. De igual forma se definen 5 conjuntos de salida, con un solapamiento reducido para disminuir el error [1]. El resultado para una entrada escalón de 1V, se presenta en la figura 3.

El reducir la anchura de los conjuntos de entrada reduce el error que produce el sistema en régimen permanente, pero puede comprometer la estabilidad. Esto es debido a que una mínima fluctuación en el error le asignaría un conjunto diferente, cambiando el valor de la salida. Reducir la anchura en dichos conjuntos, se asemeja a aumentar la ganancia en un regulador proporcional.

4. F.M.R.L.C.

Este apartado presenta un sistema de control borroso por modelo de referencia, que ya ha sido planteado y estudiado por otros autores [2], [6]. En este trabajo se utiliza con algunas modificaciones (ver apartado 4.3). Tiene la posibilidad de cambiar su estructura, en función de un modelo de referencia (M_o). El ajuste puede aplicarse sobre cualquier variable del FLC:

- Conjuntos de entrada y de salida: se puede modificar su número, solapamiento, anchura y forma.
- Inferencia: cambiar reglas o realizar una ponderación de importancia de las misma, buscar reglas no obvias.

El diseño del VFL está basado en la técnica "*Fuzzy Model Reference Learning Control (FMRLC)*" [2], [6], la cual realiza el ajuste de la posición de los conjuntos de salida mediante las variables error y su derivada. Hay otros mecanismos de aprendizaje que no se tratan aquí [7], [4].

Para hacer realidad el VFL se ha fijado el autoajuste del regulador solamente en el desplazamiento de los conjuntos de salida, figura 4, que es una variante del FMRLC, por tanto podrán ser aplicables todas las propiedades y conclusiones que otros autores han obtenido sobre el FMRLC [6].

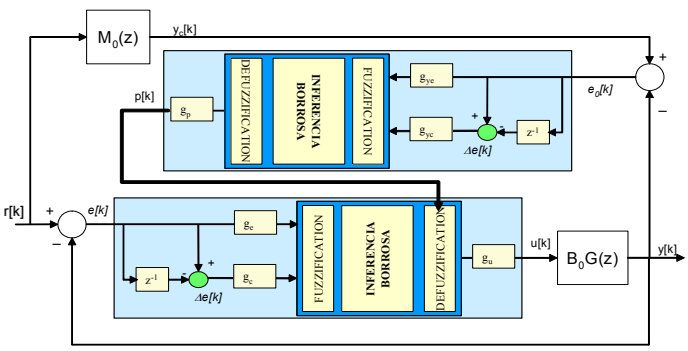


Figura 4: DB del sistema de control FMRLC.

[2] Massatchussets Institute Tecnology

[3] La versión Sugeno obtiene directamente el número real como resultado de aplicar una ecuación lineal en la inferencia.

El término de "aprendizaje" se usa como contraposición al "adaptativo" para distinguirlo del PMRA que necesita el modelo de la planta. La diferencia entre ambos radica en que el FMRLC recuerda los resultados obtenidos en estados anteriores, al mantener la situación de los conjuntos que no han sido evaluados en el instante de muestreo actual, mientras los sistemas convencionales cambian los parámetros del regulador como un acto reflejo. La capacidad de "aprendizaje" puede definirse como el proceso de observación de situaciones pasadas, para que, cuando se vuelva a encontrar en una misma situación, se responda mejor y más rápidamente.

4.1. ESTRUCTURA Y TOPOLOGÍA

El sistema consta de cuatro partes: equivalente discreto de la planta, FLC, modelo de referencia y el mecanismo de aprendizaje. El FMRLC tiene como entradas *e* y Δe. "g_e" se elige de tal forma que el rango de valores que toma *e* no conlleve la saturación de las funciones de pertenencia extremas de los conjuntos de entrada. "g_c" puede determinarse de forma heurística con varios valores de entrada (sin el mecanismo de adaptación), con el fin de determinar el rango de valores que tomará Δe. "g_u" se elige de tal forma que el rango de valores de las salidas sea el máximo posible.

Las funciones de pertenencia de los conjuntos de salida son desconocidas, serán sintetizadas automáticamente por el lazo de adaptación. El mecanismo de aprendizaje ajusta la posición de los conjuntos de salida para que el sistema en lazo cerrado consiga el objetivo. Estos ajustes se hacen observando los datos procedentes del proceso de control, del modelo de referencia y del FLC. Una parte del mecanismo de aprendizaje es un modelo inverso borroso que se encarga de realizar los ajustes pertinentes en el regulador para que e_0 sea el mínimo posible. O sea, como salida se obtiene *p* que es la que indica la variación de la posición de los conjuntos de salida según la siguiente ley:

"En el instante anterior se debería haber evaluado u(k – 1) + p(k), para hacer $e_0(k)$ menor" en lugar de haber evaluado u(k-1)"

El modificador de la base de conocimientos desplazará los centros de las funciones de pertenencia de los conjuntos asociados con las reglas del controlador borroso que contribuyeron a la acción del control del instante anterior *u(k – 1)*.

4.2 ALGORTIMO FMRLC

Basándose en [2] y [6] el VFL está inspirado en el siguiente algoritmo:

1. Seleccionar la ganancia g_{ye} para que e_0 no sature la función de pertenencia de entrada, elegida mediante pruebas, ya que no se sabe a priori el valor de e_0.
2. Seleccionar la ganancia $g_p = g_u$ y $g_{yc} = 0$.
3. Aplicar un escalón a la entrada *r* de un valor típico para condiciones normales.
4. Observar las respuestas de BG y M_o. Hay 3 casos:
   a) Si la respuesta de BG tiene grandes oscilaciones en relación a M_o, incrementar g_{yc} (se necesita mayor acción derivativa en el mecanismo de aprendizaje para reducir las oscilaciones).Volver al paso 3.
   b) Si la respuesta de BG es incapaz de seguir la respuesta de M_o, disminuir g_{yc}. Volver al paso 3.
   c) Si la respuesta de BG es aceptable con respecto a la respuesta de M_o, fin.

El FMRLC está basado en un modelo inverso borroso, que no es más que un FLC cuya salida *p(k)* modula la posición de los conjuntos de salida del regulador de la cadena directa del lazo de control, ver figura 4 parte superior.

4.3. VARIANTE FMRLC-E (VFL)

El VFL que se propone está basado en una variante del regulador FMRLC, denominada FMRLC-E en la que sólo se considera la ganancia g_p y el resto de ganancias $g_c=g_{yc}=0$ y $g_u=g_e=g_{ye}=1$. Con esta simplificación se consigue el control del sistema sólo mediante el análisis de e_0, reduciendo los tiempos de computación, fundamental para obtener una ejecución en tiempo real. De igual forma se facilita la comprensión al alumno del mecanismo de aprendizaje del regulador.

La dinámica de adaptación del sistema se puede variar por la ganancia g_p, definida entre 0 y 1, aunque se puede determinar de forma heurística que se puede ampliar su rango entre 0 y 5; si ésta es pequeña el sistema se adaptará de manera suave; si es grande, la dinámica aumenta, pero se pueden llegar a producir oscilaciones indeseables.

Los conjuntos asignados al controlador borroso se muestran en la figura 5, donde el hecho más significativo y a diferencia del FLC de la figura 2, es que los conjuntos de salida tienen anchuras uniformes y están centrados en el origen. En el momento que haya un e_0, la posición cambiará por acción de la señal *p*, con el propósito de minimizarlo.

El mecanismo de aprendizaje, modelo inverso borroso, es a su vez un algoritmo borroso definido por la distribución de conjuntos de entrada y de salida idénticos a la estructura mostrada en la figura 5 (izquierda). La inferencia que controla el regulador inverso, es la misma que para el regulador directo.

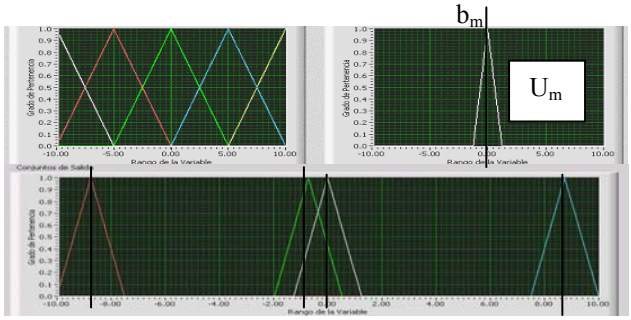


Figura 5: Arriba: Izquierda: conjuntos de entrada. Derecha: posición inicial de conjuntos de salida. Abajo posición final de los conjuntos de salida.

5 ENTORNO DIDÁCTICO. VFL.

Se mostrarán las interfaces del VFL, aplicación realizada con LabView 6.1 de NI. Mediante la *Toolkit Fuzzy Adaptive* (propuesta por los autores) se puede: diseñar el FLC, definir el modelo de referencia, configurar la consigna, salvar las capturas, visualizar el movimiento que introduce el sistema en la distribución de los conjuntos de salida, la estadística de reglas usadas, etc...

5.1. TOOLKIT "FUZZY ADAPTIVE"

La *toolkit fuzzy logic* que distribuye NI para Labview, sólo define FLC invariantes, luego surge la necesidad de implementar una nueva para adaptarla a reguladores variantes [3]. Partiendo de un trabajo [1], donde se realizó una *toolkit* de FLC para Labwindows CVI[4], no sólo se ha hecho el esfuerzo de "traducir" esa *toolkit* a Labview sino que se ha ampliado y adaptado para poder trabajar con FMRLC, esta es la denominada *toolkit fuzzy adaptive*. La aplicación usa una librería dinámica de datos *(*.dll)* en código C basada en el trabajo [1].

5.2. INTERFACE DEL USUARIO

Tanto la aplicación expuesta como el Manual de Usuario se presentan como de libre difusión estando a disponibilidad del público en la dirección de Internet: http://www.sec.upm.es/agustin/fmrlc/fmrlc.htm

Principales ventanas del VFL:

- Panel Principal de la aplicación, figura 6: donde se presentan las señales en tiempo real del sistema y el acceso a todas las opciones del programa.
- Panel de definición del regulador borroso: se accede desde el Panel Principal, mediante este control se pasa a una nueva pantalla "definir conjuntos", donde se definen los conjuntos de entrada, pero no los de salida.
- Panel de visualización de la estadística de las reglas empleadas en el proceso de inferencia, el resultado estadístico puede emplearse, por ejemplo para la clasificación de reglas buenas y/o malas.

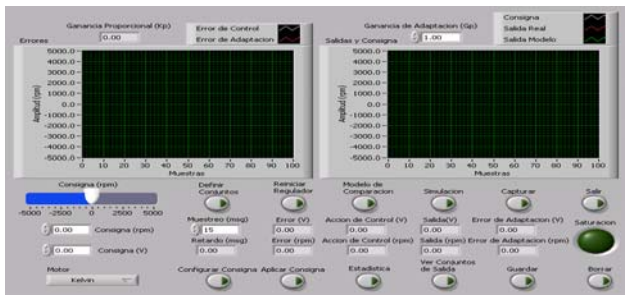


Figura 6: Panel principal (PP)

6 APLICACIÓN DEL VFL

Para comprobar el correcto funcionamiento del VLF se realizan varias pruebas sobre los motores, la consigna es la misma en todas ellas. Como modelo de referencia se elige uno que permite un error en régimen permanente del 10% y t_s=100ms. En las pruebas se muestra el seguimiento del motor al modelo de referencia, mostrando así la capacidad de aprendizaje.

Prueba 1: Motor Kelvin (figura 7): debido a la pequeña ganancia de adaptación el sistema consigue alcanzar al modelo de forma lenta.

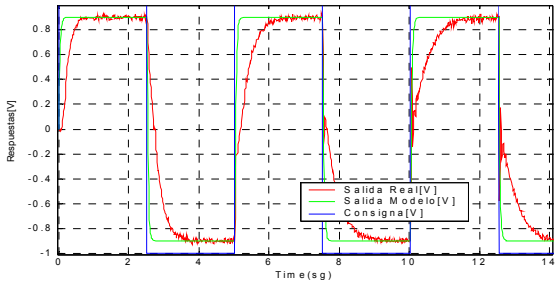


Figura 7: Respuesta del motor Kelvin (para G_p=*0.3*)

Prueba 2: Motor Feedback (figura 8): Ídem. Las oscilaciones son debidas a un sensor más ruidoso.

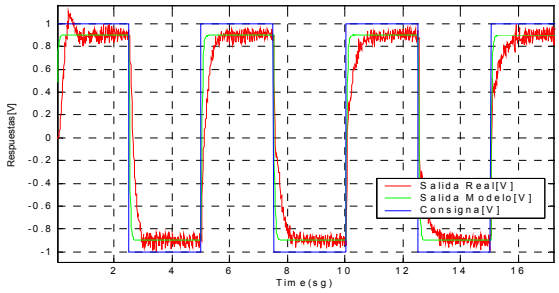


Figura 8: Respuesta del motor Feedback (para G_p=*0.3*)

[4] Es un entorno de NI, con la misma idea que Labview pero orientado a la programación en el Lenguaje C.

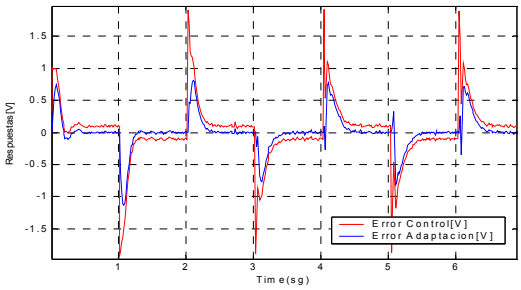


Figura 9: errores del sistema

Para terminar la exposición se muestra en la figura 9 la evolución de los errores presentes en el control del motor, siendo nulo e_0 después del transitorio.

7 CONCLUSIONES

El correcto proceso de adaptación depende en gran medida del modelo de referencia elegido para el sistema. Si presenta una dinámica más rápida que la que posee la planta, el lazo de adaptación del sistema no tiene tiempo suficiente para desplazar los conjuntos de salida al sitio adecuado, deformando la respuesta.

El término de "aprendizaje" claramente hace referencia a que el controlador recuerda lo que pasó en el pasado, utilizándolo en el presente para dar una respuesta más rápida, ya que en esa situación no tiene que volver a adaptar los conjuntos de salida. Por tanto la capacidad de aprendizaje se puede entender como el recuerdo de las situaciones por las que ha pasado para que, cuando se vuelva a encontrar en la misma situación, sea capaz de responder mejor y más rápidamente al evento. El concepto de "adaptativo" hace referencia a que para cada periodo de muestro el regulador cambia para adaptarse a la nueva situación. De estos comentarios se deduce que un regulador con aprendizaje consume menos recursos computacionales una vez adquirida la experiencia.

A diferencia de los trabajos realzados por Cortadillas [2], en las plantas aquí utilizadas, el hecho de utilizar el incremento del error como entrada adicional, era un efecto negativo, esto es debido a la gran rapidez con la que es capaz de responder el motor y un periodo de muestreo más bien grande.

Como respuesta a la pregunta inicialmente formulada *¿se puede enseñar a controlar un sistema sin necesidad de aplicar tanta matemática?* El lector debe juzgar por sí mismo, en cuanto a las "trabas" matemáticas que se ha ido encontrando a lo largo de la exposición del VFL.

8 DISCUSIÓN

Aunque la primera aplicación que se pretende dar a este entorno y en la versión no adaptativa, es en la enseñanza del control borroso que se impartirá en el laboratorio de la asignatura Control Avanzado, todavía no se ha obtenido una experiencia como metodología docente por lo que en el curso próximo se realizará una encuesta a los alumnos que la utilicen para valorar su utilidad como herramienta didáctica. De igual forma, la versión adaptativa se enseñará en clases de proyectos y se pretende actuar de la misma forma.

El FMRL-E puede ser ampliado en muchos de los pasos que hay en el algoritmo borroso, por ejemplo se puede pensar en no solo cambiar los conjuntos de salida de sitio sino modificar su anchura, su forma, las reglas de la inferencia y los conjuntos de entrada. Incluso podría pensarse en utilizar otras técnicas [7], [4] y/o modificaciones.

En cuanto a la simulación de los motores, comentar que puede ampliarse su funcionamiento en cuanto a poder simular cargas, de tal forma que cuando el motor simulado cambie de carga, o sea, de dinámica se observe el movimiento de los conjuntos para adaptarse a la nueva situación.

La estadística de reglas se presenta en este trabajo como un mero contador de uso, pero en posteriores trabajos se tiene la intención de utilizar este recurso para evaluar las reglas eliminar las peores y sustituirlas por otras que en principio no fueron pensadas por el diseñador.

Referencias

[1] Araujo García-Patos, Oscar. *"Aplicaciones de la lógica borrosa a los Sistemas de Control"*. Proyecto Fin de Carrera EUIT Telecomunicación UPM, 2000.
[2] Cortadellas Martínez, Fco Javier. Mayer Pujadas, August. "*Control Adaptativo Borroso (FMRLC)"*. Departamento de Electrónica. Ingeniería La Salle Barcelona. Revista Robótica Diciembre 1998.
[3] Galán López, David y Herranz Martín, José A. *"Controlador Adaptativo Borroso"*. Proyecto Fin de Carrera EUIT Telecomunicación UPM, 2004.
[4] Li-xi Wang. *"Adaptive fuzzy systems and control"*. Prentice Hall, Englewood Cliffs. NJ 1994
[5] Ogata, K. *"Ingeniería de control moderna"*. Ed: Prentice Hall. 1996.
[6] Passino, Kevin M. Yurkovich, Stephen. *"Fuzzy Control"*. Department of Electrical Engineering The Ohio State University. Addison Wesley Longman, Inc., 1998
[7] Ronadl R.Y. and Dimitar P.F. *"Essentials of fuzzy modeling and control"*. John Wiley & Sons, inc. NY 1994

KBCT: Una herramienta para adquisición y representación de conocimiento usando lógica fuzzy

José M. Alonso, Luis Magdalena
ETSI Telecomunicación
Universidad Politécnica de Madrid (España)
{jmam,llayos}@mat.upm.es

Serge Guillaume
Cemagref Montpellier (Francia)
guillaume@montpellier.cemagref.fr

Resumen

Este artículo presenta una herramienta multiplataforma y multilingüe, con una interfaz gráfica amigable, que facilita la extracción y representación de conocimiento en sistemas basados en lógica fuzzy. Su objetivo es la construcción o el refinamiento de bases de conocimiento integrando conocimiento experto e inducido, realizándose todo el proceso bajo el control del experto. La expresividad semántica de la lógica fuzzy, próxima al lenguaje natural usado por el experto, simplifica el proceso de adquisición y aumenta la interpretabilidad.

Palabras Clave: Adquisición, representación, conocimiento experto e inducido, integración, coherencia, interpretabilidad.

1. INTRODUCCIÓN

KBCT son las siglas de *Knowledge Base Configuration Tool*. Este nombre hace referencia a la funcionalidad básica de la herramienta, es decir, la creación o modificación de una base de conocimiento, haciendo uso de lógica fuzzy. Permite inducir conocimiento a partir de datos, facilitando la integración de este conocimiento con el extraído del experto, y garantizando la coherencia a través de mecanismos de análisis de conflictos.

El diseño de KBCT parte de un análisis de la oferta de software para el trabajo con sistemas basados en lógica fuzzy. Existe por un lado software para sistemas expertos y por otro software para inducción (Xfuzzy[1], FisPro[2], Matlab[3],...). KBCT es una herramienta genérica que integra ambos tipos de conocimiento bajo el control del experto, construyendo una base de conocimiento genérica que pueda ser utilizada por diferentes motores de inferencia borrosos. Definiendo un proceso de adquisición y representación del conocimiento en el que lo fundamental es mantener la interpretabilidad del sistema, sin descuidar la eficiencia.

KBCT se ha desarrollado en el marco del proyecto europeo de investigación ADVOCATE II[4] [7]. El objetivo de este proyecto es mejorar la fiabilidad y eficiencia de sistemas robóticos autónomos terrestres y submarinos. KBCT reduce el esfuerzo de extracción y representación de conocimiento, en la construcción de bases de conocimiento adaptadas a los problemas específicos de estos sistemas. Actualmente se está utilizando para tratar problemas de movimiento en robots terrestres [2].

Al margen de su aplicación en ADVOCATE II, KBCT se ha convertido en un proyecto de investigación con identidad propia. Es una herramienta multiplataforma, trabaja sobre Linux o Windows, y multilingüe, permite elegir entre tres idiomas (Francés, Inglés y Español), y dispone de un completo manual de usuario escrito en HTML (*Hypertext Markup Language*) disponible en Español e Inglés.

El resto del artículo se ha estructurado de la siguiente forma. La sección 2 presenta el proceso de adquisición de conocimiento. A continuación, en la sección 3 se describen las tareas necesarias para la extracción y representación del conocimiento experto. Después, en la sección 4 se detalla el proceso de inducción de conocimiento a partir de datos. La integración de ambos tipos de conocimiento se explica en la sección 5. La sección 6 presenta el proceso de análisis de coherencia. Finalmente, el artículo termina con unas breves conclusiones en la sección 7.

[1] http://www.imse.cnm.es/Xfuzzy
[2] http://www.inra.fr/bia/M/fispro
[3] http://www.mathworks.com
[4] http://advocate2.e-motive.com

2. PROCESO DE ADQUISICIÓN DE CONOCIMIENTO

Para manejar sistemas complejos, lo primero que se debe hacer es consultar a los expertos, las personas que más saben del sistema [5]. Sin embargo, parte del conocimiento del experto es implícito, basado en su experiencia y en sus intuiciones. El experto conoce el comportamiento del sistema, pero no siempre será capaz de expresarlo de forma sencilla. Aquí es muy importante la lógica fuzzy, que ayuda a formalizar el conocimiento del experto, de una forma cercana al lenguaje natural empleado por éste. Además, parte del conocimiento puede residir en el subsconsciente del experto, y la interacción entre variables es difícil de cuantificar. Por estas razones, el conocimiento experto debe ser complementado por un conocimiento inducido de datos que reflejen significativamente el comportamiento del sistema. Por tanto, uno de los puntos clave en el proceso de adquisición será la integración de ambos tipos de conocimiento, experto e inducido, manteniendo el equilibrio entre interpretabilidad y precisión. El proceso completo de construcción de una base de conocimiento usando KBCT podría incluir los siguientes pasos:

1. Extraer un primer nivel de conocimiento experto. Se define en un alto nivel de abstracción las principales variables de entrada y salida, con información cualitativa sobre ellas.

2. Inducir particiones a partir de datos experimentales para las variables definidas por el experto.

3. Integrar el conocimieto experto e inducido sobre las variables. Se obtendrá un universo común, definido por variables lingüísticas.

4. Formalizar un segundo nivel de conocimiento experto. El experto expresa su conocimiento sobre el sistema en forma de reglas borrosas, sobre el marco semántico común definido previamente.

5. Inducir reglas a partir de los datos.

6. Integrar ambos tipos de reglas, expertas e inducidas. La comparación de reglas se realiza sólo a nivel lingüístico, ya que todas las reglas usan las mismas etiquetas lingüísticas definidas por los mismos conjuntos borrosos, pudiendo evaluarse la coherencia y la calidad de la base de reglas resultante en cualquier momento.

KBCT es una herramienta dinámica. Todo el proceso de adquisición de conocimiento se realiza bajo la supervisión del experto. Ayuda a construir la base de conocimiento a partir de la información que se pueda extraer del experto y de los datos experimentales disponibles. Los pasos propuestos son orientativos, pudiendo ser alterados si se considera necesario, aunque con algunas restricciones obvias como tener que definir las particiones antes de inducir reglas.

3. CONOCIMIENTO EXPERTO

El experto es capaz de establecer cuáles son las variables fundamentales para entender el sistema, y puede explicar su comportamiento.

KBCT permite realizar el proceso de adquisición de conocimiento en distintos niveles de abstracción. En un primer paso, se definirán las variables sin detallar los conjuntos borrosos asociados con ellas, el experto no necesita saber nada de lógica fuzzy. Se definirá el rango de la variable, el número de etiquetas lingüísticas necesarias para describir su comportamiento, el nombre de las etiquetas, los valores típicos asociados con ellas (prototipos o puntos modales),... KBCT genera de forma automática particiones fuertes con un número de conjuntos borrosos igual al número de etiquetas lingüísticas establecido por el experto. Posteriormente, estas particiones podrán ser sustituidas por las inducidas a partir de datos, o modificadas por el experto, si conoce el formalismo de la lógica fuzzy.

KBCT ofrece una ventana gráfica (Fig. 1) que permite definir todos los elementos de la base de conocimiento. El proceso de adquisición de conocimiento experto implica dos aspectos, la definición de variables y la descripción de reglas.

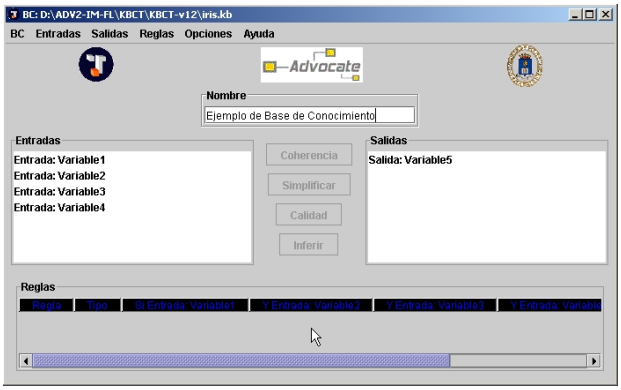


Figura 1: Extracción del conocimiento experto

El primer paso es la definición de variables. Si el usuario crea una variable o hace doble click sobre el nombre de una definida previamente, se abrirá una nueva ventana (Fig. 2) que le permitirá establecer los principales parámetros de dicha variable:

Nombre: Nombre de la variable.

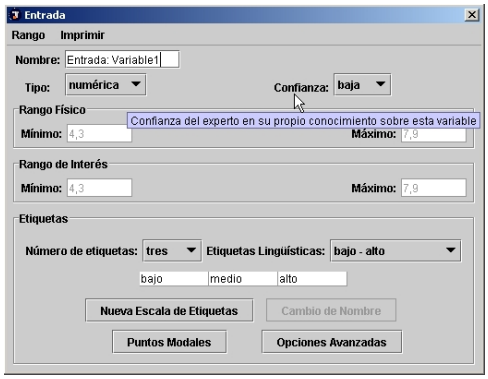


Figura 2: Definición de una variable

Tipo: Naturaleza de la variable.

- *categórica*: Las etiquetas representan clases numeradas en orden creciente (1, 2,...).
- *lógica*: Sólo dos etiquetas son posibles (falso/verdadero, 0/1,...).
- *numérica*: Las etiquetas se definen mediante funciones de pertenencia de distintos tipos (triangular, trapezoidal,...). Por defecto, se generan particiones fuertes que pueden ser modificadas usando *Opciones Avanzadas*.

Confianza: Confianza del experto en su propio conocimiento sobre esta variable.

- *alta*: Conoce la importancia de la variable, es capaz de describir su comportamiento.
- *media*: El conocimiento sobre esta variable tiene que ser confirmado por datos.
- *baja*: Supone que la variable influye pero no es capaz de cuantificar su comportamiento.

Rango: Margen de valores en que está definida una variable numérica: *Rango Físico* y *Rango de Interés* (incluido dentro del *Rango Físico*).

Etiquetas: términos lingüísticos (asociados a conjuntos borrosos).

- *Número de Etiquetas*: Elige un número entre 2 y 7 (Para variables lógicas sólo 2 es válido).
- *Etiquetas Lingüísticas*: poco-mucho, bajo-alto, pequeño-grande, negativo-positivo y definido por usuario.
- *Nueva Escala de Etiquetas*: El experto define su propia escala (*definido por usuario*).
- *Cambio de Nombre*: Modifica el nombre de las etiquetas de la escala *definida por usuario*.
- *Puntos Modales*: Sólo para variables numéricas. Asocia un valor característico (el centro del conjunto borroso) a cada etiqueta. Puede ser un prototipo: por ejemplo, 37°C para la temperatura normal de un adulto.
- *Opciones Avanzadas*: Sólo disponible si la variable es numérica. Define la función de pertenencia asociada con cada etiqueta.

Definidas las características básicas de las variables del sistema, se puede completar la definición de dichas variables a partir de la inducción de particiones, o empezar a describir el comportamiento global del sistema mediante la definición de reglas expertas. En el segundo caso, se invita al experto a expresar su conocimiento en forma de reglas borrosas de la forma "SI condición ENTONCES conclusión". La Fig. 3 muestra cómo definir las reglas seleccionando la etiqueta lingüística asociada a cada variable de entrada o salida, estableciendo respectivamente las premisas y la conclusión de las reglas.

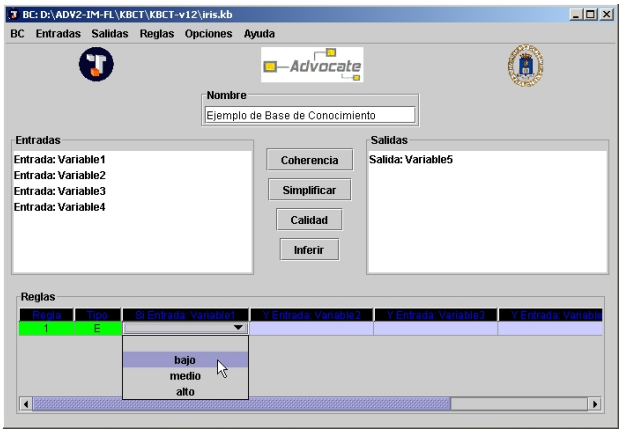


Figura 3: Definición de reglas

Definidas las variables y reglas expertas, el proceso de extracción de conocimiento experto se puede considerar cerrado. Se puede pasar a la inducción de conocimiento a partir de datos. Pero si la inducción contradice al conocimiento experto, siempre se puede volver a modificarlo.

4. CONOCIMIENTO INDUCIDO

La inducción de conocimiento permite validar y complementar el conocimiento experto. Los datos se obtienen en experimentos reales, corresponden a situaciones concretas y raramente cubren todos los casos. Esto hace que las reglas inducidas no sean reglas universales. Son más o menos generales dependiendo de lo representativo que sea el conjunto de datos.

Los datos se almacenan en archivos de texto plano, distribuidos por columnas, cada columna corresponde a una variable. Al abrir un archivo de datos se pregunta al usuario qué variables quiere usar, y cuáles desea que sean utilizadas como salidas. Pueden darse dos situaciones:

- Abrir un archivo de datos antes de crear una base de conocimiento experta. Se genera un archivo experto a partir de la información de los datos. La distribución de cada variable determina su tipo (lógica si hay 2 valores diferentes; categórica si el número de valores diferentes está entre 3 y 7; y numérica en el resto de casos). El rango queda determinado por los valores máximo y mínimo de la distribución. Esta base de conocimiento construida de forma automática a partir de los datos se puede modificar pulsando el botón *Experto*. Alternativamente, se pueden usar *Datos* para inducir particiones o reglas.

- Abrir un archivo de datos cuando está abierto un archivo experto. Se comprueba que el número de variables definidas por los datos es coherente con el número de entradas y salidas definidas por el experto.

El menú *Datos* permite ver la distribución de los datos (histograma) o la relación entre variables (gráfico XY). Esto ayuda a determinar si son o no significativos. Si el rango de los datos es muy diferente al rango de interés definido por el experto, los datos no ayudarán a resolver el problema, tendremos que hacer más experimentos. Sin embargo, la obtención de datos suele ser difícil, la realización de experimentos normalmente es costosa en tiempo y dinero. Sólo si los datos disponibles son significativos, las particiones y reglas inducidas sirven para complementar el conocimiento experto ofreciendo una buena imagen de la interacción entre variables.

Si el usuario selecciona la opción de inducir particiones sobre una variable, KBCT muestra la ventana que se puede ver en la Fig. 4, en ella se presenta la siguiente información:

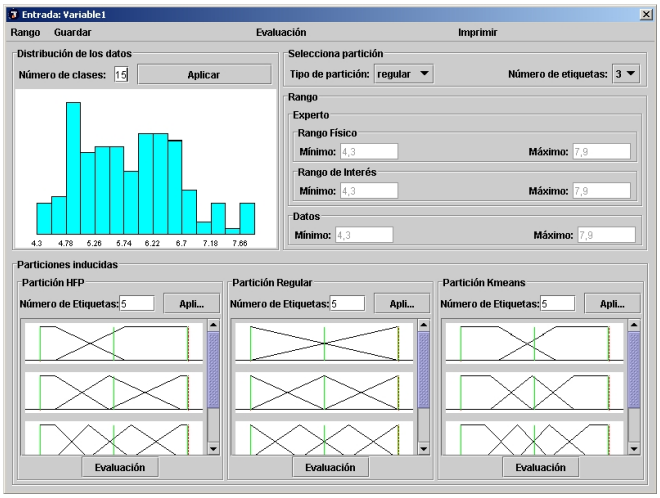


Figura 4: Particiones Inducidas

- **Histograma** con la distribución de datos.

- **Rango**: Los rangos físico y de interés son definidos por el experto, pero el rango de los datos lo determina la distribución.

- **Particiones**: Se puede comparar las particiones regulares, generadas automáticamente cuando se define una variable numérica, con las particiones inducidas mediante los algoritmos *Hierarchical Fuzzy Partitioning* (HFP) descrito en [9, 10] o *k-means* [4]. Las líneas verticales representan a los puntos modales definidos por el experto, se pueden comparar con los centros de los conjuntos borrosos de las particiones inducidas.

- **Evaluación**: Existen múltiples criterios para evaluar las particiones borrosas. KBCT implementa tres, el coeficiente de la partición (PC: *Partition Coefficient*), la entropía de la partición (PE: *Partition Entropy*) y el índice de Chen (Ch: *Chen Index*) explicados en [11]. Si se selecciona *Evaluación* se puede ver una ventana con los índices calculados (Fig. 5). Una Buena partición debe minimizar PE y maximizar PC y Ch.

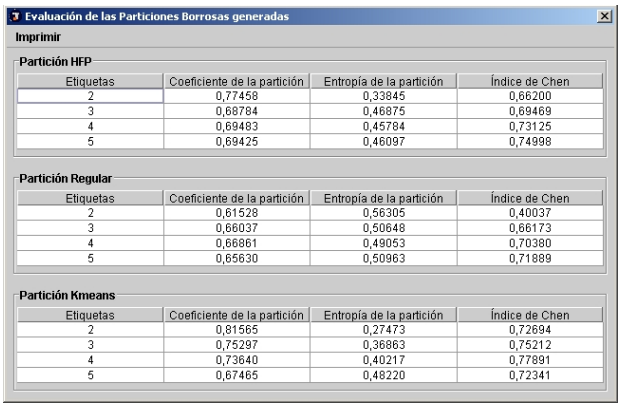


Figura 5: Índices de evaluación de las particiones

- **Guardar**: Se debe seleccionar la partición a guardar eligiendo el tipo de partición y el número de etiquetas. La partición seleccionada sustituirá a la partición de la variable correspondiente en la base de conocimiento.

Para inducir reglas se puede elegir entre tres métodos (*Fuzzy Decision Trees* [3], *Wang and Mendel* [6], *Fast Prototyping Algorithm* [8]) que inducen reglas sobre las particiones definidas previamente. El resultado de la inducción se presenta en la ventana de la Fig. 6. Si se selecciona *Guardar* las reglas inducidas serán añadidas a la base de conocimiento. Como último paso de la inducción, se puede seleccionar la opción *Simplificar* (Fig. 3) que invoca el método de simplificación de la base de reglas inducidas implementado en FisPro [1].

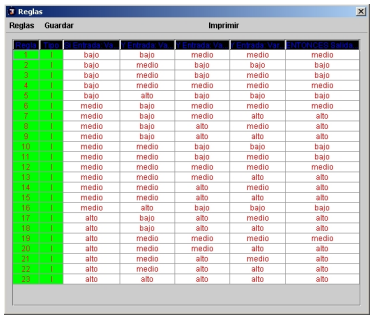


Figura 6: Reglas Inducidas

5. INTEGRACIÓN

Ambos tipos de conocimiento, experto e inducido, son complementarios. La integración se debe hacer sin perder la interpretabilidad y la coherencia. Todo el proceso está supervisado por el experto que es el encargado de resolver los conflictos que puedan surgir, el conocimiento experto podría ser cuestionado por las evidencias extraídas de los datos. Se pueden distinguir dos pasos en el proceso de integración:

- **Integración de particiones**: Se realiza variable por variable. El acuerdo entre ambos tipos de particiones, expertas e inducidas, se debe alcanzar para todos los elementos definidos por el experto:
  - *Rango*: El rango de los datos debe quedar incluido dentro del rango experto.
  - *Granularidad*: El experto indica el número de etiquetas que necesita. A partir de los datos se inducen particiones de distintos tamaños, para cada una de las cuales se calculan índices de calidad (PC, PE, Ch). El experto compara las particiones, y elige la mejor según los índices.
  - *Puntos Modales*: Los prototipos de los conjuntos borrosos deben ser próximos a los puntos modales definidos por el experto.
- **Integración de reglas**: Ambos tipos de reglas usan las mismas etiquetas lingüísticas gracias al universo común definido previamente. Por ello la comparación de reglas sólo se hace a nivel lingüístico. Las dos bases de reglas se mezclan prestando atención a la interpretabilidad, evitando redundancia y garantizando la coherencia.

6. ANÁLISIS DE COHERENCIA

La detección de conflictos lingüísticos puede hacerse en cualquier momento, el análisis sobre la reducción de conflictos se describe en [12]. Se consideran 4 niveles de mensajes, clasificados y presentados en orden decreciente de importancia. El primer nivel muestra errores que deben ser corregidos, el resto son avisos o consejos:

1. *Errores de Coherencia*
   - Regla con conclusión sin definir. Se debe definir la conclusión o eliminar la regla.
   - Reglas duplicadas. Se debe eliminar una de las reglas.
   - Reglas contradictorias (mismas premisas y diferente conclusión). Se debe decidir cuál es la regla correcta y eliminar la otra.
2. *Avisos de Redundancia o Especificidad*
   - El espacio de entrada cubierto por una regla está incluido en el cubierto por otra:
     - Igual conclusión. La regla más específica parece redundante y el experto es invitado a eliminarla.
     - Diferente conclusión. Las reglas son contradictorias en la parte común del espacio de entrada.
   - Dos reglas tienen intersección no vacía en las premisas:
     - Igual conclusión. Existe un efecto de refuerzo dependiente del operador de agregación de reglas. Para evitarlo se debería usar el máximo en lugar de la suma.
     - Diferente conclusión. Las reglas son contradictorias en la parte común. El resultado de la inferencia dependerá del sistema de inferencia usado. El experto debería cambiar las premisas de las reglas para deshacer la contradicción.
3. *Avisos de entradas*: Variables y etiquetas definidas pero no usadas en las entradas, se debe comprobar si son realmente necesarias.
4. *Avisos de salidas*: Variables y etiquetas definidas pero no usadas en las salidas, se debe comprobar si son realmente necesarias.

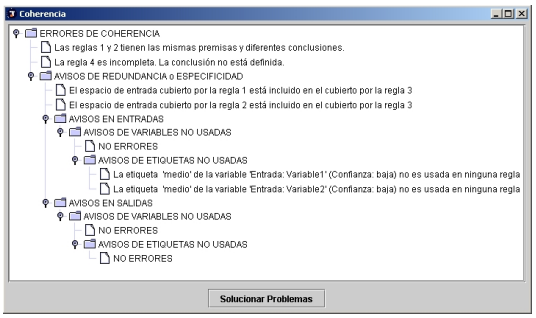


Figura 7: Informe de Análisis de Coherencia

La Fig. 7 muestra la ventana con los mensajes resultantes del análisis de coherencia. El botón *Solucionar Problemas* abre una nueva ventana con sugerencias sobre cómo resolver los conflictos detectados.

7. CONCLUSIONES

Los sistemas de inferencia borrosos son muy usados en procesos de simulación o control. Hay muchas técnicas de diseño para este tipo de sistemas, pero la mayoría construye la base de conocimiento a partir de conocimiento adquirido de expertos o inducido de datos. KBCT facilita la integración del conocimiento inducido con el adquirido del experto, realizando el proceso de integración bajo el control del experto, al que se ofrece gran cantidad de información e índices como ayuda en la toma de decisiones. La base de conocimiento generada es independiente del motor de inferencias que se vaya a usar. En teoría podría ser utilizada con cualquier aplicación que disponga de un motor de inferencias de lógica fuzzy. En la práctica, la herramienta permite convertir la base de conocimiento al formato apropiado para los motores de inferencia de FisPro y Matlab.

KBCT tiene un completo manual de usuario. El entrevistador puede consultarlo en cualquier momento, utilizándolo como guía en el proceso de adquisición y representación del conocimiento. La herramienta es multilingüe, el usuario puede elegir el idioma de las ventanas gráficas. También puede modificar la apariencia de dichas ventanas entre tres estilos diferentes. Y puede ejecutar la aplicación en Windows o Linux. Todo ello hace que la interfaz sea amigable y que el manejo de KBCT resulte claro y sencillo, haciendo más dinámico el proceso de adquisición de conocimiento.

KBCT estará disponible pronto en Internet como una aplicación de software libre.

Agradecimientos

Nos gustaría reconocer la importancia del proyecto europeo de investigación ADVOCATE II en el diseño y desarrollo de KBCT. El consorcio está formado por: Universidad de Alcalá, GETRONICS DECAN, Universidad Politécnica de Madrid, ATL Elektroniks, IFREMER, HUGIN Expert A/S, INNOVA S.p.A, and E-MOTIVE. La Comisión Europea (IST-2001-34508) supervisa este proyecto.

Referencias

[1] Guillaume, S., Charnomordic, B., and Lablée, J.-L. Fispro: an open source portable software for fuzzy inference systems. (2002). http://www.inra.fr/bia/M/fispro.

[2] Guillaume, S., Magdalena, L., Sotelo, M. A., Bergasa, L. M., Flores, R., Ocaña, M. (2003). Expert and induced knowledge for intelligent motion analysis of ground robots. In EUSFLAT'03, Zittau, Germany.

[3] H. Ichihashi and T. Shirai and K. Nagasaka and T. Miyoshi, Neuro-fuzzy ID3: a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning, *Fuzzy Sets and Systems*, 81, 157-167. 1996.

[4] John A. Hartigan and M.A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28:100-108, 1979.

[5] Larichev, O. I. (2002). Close imitation of expert knowledge: the problem and methods. *International Journal of Information technology and Decision Making*, 1(1),27-42.

[6] Li-Xin Wang and Jerry M. Mendel, Generating Fuzzy Rules by Learning from Examples, *IEEE Transactions on Systems*, *Man and Cybernetics*, 22(6), 1414-1427. 1992.

[7] Miguel A. Sotelo, Luis M. Bergasa, Ramón Flores, Manuel Ocaña, Marie-Hélène Doussin, Luis Magdalena, Willi Hornfeld, Michel Perrier, Anders L. Madsen, Aleardo Furlani, and Damien Roland, ADVOCATE II: ADVanced On-board diagnosis and Control of Autonomous sysTEmsII, In Ninth International Conference on Computer Aided Systems Theory, Las Palmas de Gran Canaria, Spain, February 2003.

[8] Pierre-Yves Glorennec, Algorithmes dápprentissage pour systèmes dínférence floue, Editions Hermès, Paris, 1999.

[9] Serge Guillaume and Brigitte Charnomordic. A new method for inducing a set of interpretable fuzzy partitions and fuzzy inference systems from data, volume 128 of Studies in Fuzziness and Soft Computing, pages 148-175. Springer, 2003.

[10] Serge Guillaume and Brigitte Charnomordic. Generating an interpretable family of fuzzy partitions. *IEEE Transactions on Fuzzy Systems*, In Press, 2004.

[11] Serge Guillaume and Luis Magdalena. Expert guided integration of induced knowledge into a fuzzy knowledge base. *Soft Computing* (In Press).

[12] Serge Guillaume and Luis Magdalena. Typologie des conflits dans les systèmes de mandani. In LFA'03, France. Cépaduès editions. 73-80. 2003.

NetFuzzy y FuzzyComponents: un entorno distribuido para el cálculo fuzzy *

Adolfo R. de Soto
E. de Ingenierías Industrial e Informática
Universidad de León
ddears@unileon.es

Conrado Andréu Capdevila
E. de Ingenierías Industrial e Informática
Universidad de León
conrado.andreu@wanadoo.es

Resumen

En el marco del proyecto EDAB (Entorno de Desarrollo de Altas Prestaciones para la Lógica Borrosa) se han desarrollado un conjunto de herramientas software destinadas a posibilitar el cálculo distribuido con Sistemas de Reglas Borrosas. Al realizar su diseño se ha optado por un modelo basado en componentes, con el objetivo de facilitar tanto su uso como su posible ampliación futura, pensando especialmente en entornos en los que se combinen diversas componentes inteligentes para construir agentes software. El objetivo de este trabajo es mostrar los principales elementos que constituyen el sistema así como el estado actual de desarrollo.

Palabras Clave: Sistemas de Reglas Fuzzy, Componentes Software, Programación Distribuida, XML

1. INTRODUCCIÓN

El trabajo inicial de desarrollo del entorno software del proyecto EDAB se orientó hacia la construcción de un conjunto de componentes software distribuidas que permitiese el cálculo remoto con Sistemas de Reglas Borrosas (SRB). Se construyeron prototipos de componentes COM y Corba, implementadas en el entorno Delphi, para el cálculo con SRB [3] y utilizando el lenguaje FTL [1] como lenguaje de definición de SRB (LDSRB). A partir de esos trabajos, se diseñó la arquitectura de un sistema de componentes, denominado FuzzyComponents, más amplia, que permitiese el calculo remoto con SRB, tanto para un cálculo puntual, como para un cálculo masivo, la traducción de especificaciones de SRB de un lenguaje LDSRB a otro y la extracción automática de SRB a partir de un conjunto de datos [4]. El objetivo al desarrollar estas componentes es facilitar la creación de aplicaciones software que tengan la necesidad de solicitar cálculos a SRB independientemente de su ubicación, por ejemplo, agentes inteligentes, sistemas de domótica, personalización de entornos en aplicaciones remotas o de búsquedas inteligentes en la Web, sistemas que tengan la necesidad de extraer automáticamente SRB o de actualizarlos on-line. Al sistema de componentes se le exigieron varios requisitos no funcionales como un buen control de versiones de las componentes desplegadas y la definición o utilización de mecanismos de servicios de nombres. El diseño se realizó considerando siempre un sistema que fuese fácilmente extensible por equipos de desarrollo distintos al original. Se decidió construir el prototipo en esta ocasión en el lenguaje Java, utilizando la especificación Enterprise Java Beans para el desarrollo de las componentes distribuidas. En la sección 3 se detalla el estado actual de desarrollo del sistema.

Uno de los objetivos del proyecto EDAB era la construcción de un entorno software de diseño de SRB que facilitase la aplicación de los algoritmos de optimización desarrollados por el equipo. En un primer momento no se estimó necesario definir un nuevo lenguaje LDSRB, optándose por utilizar lenguajes previamente definidos como FTL de FuzzyTech [1], XFL de Xfuzzy [2], y el lenguaje LIF desarrollado por el equipo en un proyecto previo. Al evaluar las necesidades que fueron surgiendo durante el desarrollo del sistema FuzzyComponents y las aplicaciones a las que podía ir dirigido, con especial énfasis en la instalación y consulta remota de SRB y a la extracción automática de SRB, tanto offline como online, se apreció la necesidad de un len-

*Este trabajo ha sido parcialmente financiado por el proyecto TIC-2000-1420 del Plan Nacional de I+D+I, por el proyecto LE19/02 de la Junta de Castilla y León y por la REd temática nacional de procesos de toma de DEcisiones, Modelado y Agregación de Preferencias. (REDEMAP) Ministerio de Ciencia y Tecnología. Ref. TIC-2002-11942-E.

guaje de definición de SRB que facilitara estas tareas y especialmente, su tratamiento distribuido. Por todo ello se decidió definir un lenguaje basado en XML, y así beneficiarse del conjunto de herramientas software disponibles en múltiples plataformas para el tratamiento de esquemas XML, procesamiento de documentos XML, tanto en su lectura como en su construcción, transformaciones XSL que permiten traducir especificaciones entre distintos lenguajes, validar especificaciones, o generar modelos de clases de objetos en diferentes lenguajes de programación. Además se pensó en dar cabida en el lenguaje a otros campos del Soft Computing, permitiendo, en el futuro, dar especificaciones de Redes Neuronales. La primera versión del lenguaje iXSCL (Extensible Soft Computing Language) apareció publicada en [5], haciendo especial hincapié en su diseño en lograr una buena modularidad y extensibilidad. En la sección 2 se describen las principales características del lenguaje iXSCL.

La definición del lenguaje iXSCL junto con el sistema FuzzyComponents condicionaron en un alto grado el diseño del entorno de desarrollo de SRB. Por una parte había que aprovechar la multitud de herramientas existentes para el tratamiento de lenguajes basados en XML, por otra, había que pensar en una fácil integración con las componentes desarrolladas, persiguiendo facilidades para su definición, construcción y despliegue. El desarrollo como Enterprise Java Beans de las mismas también influyó, puesto que el lenguaje Java jugaba un importante papel, aunque no determinante, en la medida en que se pueden crear envoltorios de forma sencilla para que esas mismas componentes sean componentes Corba, por ejemplo. Todo ello provocó la elección de NetBeans como plataforma de desarrollo. La plataforma NetBeans es la base de un entorno de desarrollo integrado (IDE) para el lenguaje Java, pero también se ha utilizado como base de otras aplicaciones como herramientas para trabajar con UML, editores de otros lenguajes, y diversas utilidades. El entorno de desarrollo puede extenderse mediante módulos, por ejemplo para su integración con otras herramientas o la edición de otros lenguajes. El objetivo fue pues construir uno de esos módulos que facilitase la edición de especificaciones iXSCL, ya sea en forma de documento XML o mediante interfaz gráfica. En la sección 4 se detalla el estado de desarrollo del modulo NetFuzzy.

2. EL LENGUAJE IXSCL

El lenguaje iXSCL (acrónimo de Extensible Soft Computing Language) es un vocabulario XML para la especificación de objetos comunes en el área del Soft Computing. La primera versión del lenguaje sólo consideró Sistemas Fuzzy [5] pero ya se ha ampliado con esquemas para la definición de Redes Neuronales [6], aunque para el propósito que aquí nos ocupa no será de interés esta ampliación, salvo como medio de remarcar que uno de los objetivos de diseño del lenguaje es facilitar el añadido de construcciones del área de la computación flexible.

Por lo que respecta a la especificación de SRB, el lenguaje ha sido fuertemente influenciado por el lenguaje XFL [2] y su definición se ha realizado mediante esquemas y no DTD, para posibilitar el uso de tipos abstractos que permitan su posterior extensión. Los principales constructores del lenguaje son: Contextos Lingüísticos, donde se pueden definir múltiples Variables Lingüísticas; Bases de Reglas, donde se define una base de reglas fuzzy y Ligaduras, que permite asociar los diversos operadores fuzzy, tales como operadores de conjunción, de disyunción o de agregación, con operaciones concretas tales como t-normas, t-conormas y operadores de agregación. Un Bloque Fuzzy se construye a partir de un Contexto Lingüístico, una Base de Reglas y una Ligadura. Los constructores Map y Layout permiten la composición de Bloques Fuzzy con otros tipos de bloques para dar lugar a construcciones más complejas.

Los Contextos Lingüísticos actúan como espacios de nombres y se organizan en una estructura tipo árbol. Se permiten definir tanto Variables Lingüísticas como Variables Nítidas. Dentro de una Variable Lingüística pueden definirse Términos Lingüísticos definidos por su Función de Pertenencia así como Modificadores Lingüísticos opcionales.

Una Base de Reglas estará formada por reglas ponderadas cuyos términos en los antecedentes o consecuentes pueden también modificarse mediante Modificadores Lingüísticos. Una Base de Reglas es una unidad por sí misma de forma que un documento iXSCL que contenga únicamente una de ellas está bien formado. De este modo, y mediante el uso de Contextos Lingüísticos, pueden reutilizarse las Bases de Reglas con diferentes interpretaciones para las variables y términos que involucren.

Un Map es una estructura abstracta que define un interfaz de forma que múltiples Maps puedan combinarse en un Layout. Ejemplos de Maps son los Bloques Fuzzy, los Bloques Defuzzificadores o las Redes Neuronales. Los Bloques Fuzzy se forman mediante uno o varios Contextos Lingüísticos, que dan el significado de las variables, modificadores o términos lingüísticos que aparecen en una Base de Reglas, y un constructor Ligadura que proporciona la semántica de los operadores que aparezcan en la Base de Reglas.

La elección de XML como base para el lenguaje iXSCL debe permitir en el futuro construir aplicaciones que procesen especificaciones en iXSCL en entornos Web

tales como servicios Web basados en consultas remotas a SRB o personalización de entornos mediante generación on-line de SRB. Las transformaciones XSL permiten procesar de forma sencilla archivos iXSCL lo que en la actualidad se está utilizando para sintetizar una especificación iXSCL a lenguaje Java, pero cuyo patrón de funcionamiento puede permitir sintetizar a otro lenguaje con la misma facilidad, generar páginas html o enviar salidas a bases de datos externas.

3. FUZZYCOMPONENTS

Con el objetivo de permitir el cálculo con SRB en entornos distribuidos, al amparo del proyecto EDAB, se han desarrollado un conjunto de componentes software Enterprise Java Beans. En la actualidad el prototipo del sistema FuzzyComponents cuenta con una componente de control de versiones ('VersionProviderEJB'), una componente ('FRS') para el cálculo con SRB a partir de un DataSet, que es un conjunto de datos almacenado en formato XML, una componente para la generación de SRB ('FRSGeneration'), y una componente de síntesis de una especificación de un SRB dado en el lenguaje XFL a clases Java ('XFLDefinitionAnalysis'). El nivel de desarrollo y de pruebas de las diversas componentes es bastante diverso, estando todas implementadas en su funcionalidad básica (excepto 'VersionPRoviderEJB' y 'FRS') pero no habiendo sido sometidas aún a un proceso intensivo de prueba. La funcionalidad de cada una de las componentes se describe a continuación.

El tipo de componente central del sistema FuzzyComponents es la componente FRS (FuzzyRuleSystem) que permite el cálculo con un SRB a partir de un conjunto de datos. En un sistema podrían existir múltiples componentes FRS cada una posibilitando el cálculo con un SRB distinto. Tanto los datos de entrada como los de salida de la componente se presentan como un documento XML, instancia de un esquema definido en el proyecto. Cualquier implementación que respete esa definición es aceptable como conjunto de datos; en este caso, ya que el esquema es simple, se recurrió a JAXB para generar de forma automática un conjunto de clases Java que envuelven al documento XML. La elección de este tipo de formato de datos se debe a la facilidad para gestionar documentos XML en entornos Web y en herramientas de acceso a datos.

Una componente FRS debe generarse a partir de la especificación de un SRB en un lenguaje determinado. Existen componentes de tipo FRSDefinitionAnalysis para realizar esta tarea. Todas ellas deben recibir una implementación de la interfaz 'Envelope' con la especificación del SRB a analizar en un lenguaje dado, generar la componente FRS y registrarla en el sistema. En la actualidad existe una componente IXSCLDefinitionAnalysis que acepta un 'Envelope' con un árbol DOM asociado a un documento iXSCL, y otra componente XFLDefinitionAnalysis que espera un tipo distinto de 'Envelope' que contiene el archivo XFL que define el sistema.

Se han construido componentes de Acceso a Datos que posibiliten una independencia de la fuente de datos que proporcione las entradas a las componentes FRS. Existe un tipo de componente MassiveInputData destinada a tomar datos de una fuente de datos y convertirlos al formato dado por el esquema XML para conjuntos de datos, por ejemplo mediante la implementación basada en JAXB. En la actualidad se encuentran desarrolladas una componente FileMassiveInputData cuya fuente de datos es un fichero texto con un formato predefinido, y una componente BDMassiveInputData que considera que la fuente de datos es una base de datos a la que se accede siguiendo el patrón DAO. Así mismo existe un tipo de componentes denominado Massive que permite realizar cálculos intensivos sobre una componente FRS especificando el conjunto de datos de entrada (MassiveInputData) y el destino de los datos de salida (MassiveOutputData).

Otro tipo de componentes fundamental en la arquitectura de FuzzyComponents son las componentes de extracción de reglas. La interfaz que define el servicio es 'FRSGeneration'; mediante una factoría 'FRSGenerationFactory' se obtiene una implementación de esa interfaz para el algoritmo de extracción deseado. El usuario puede además registrar nuevos algoritmos de extracción de reglas en el sistema que quedan disponibles en la factoría.

4. NETFUZZY

A la hora de elegir un editor para especificaciones iXSCL y asumiendo el sistema de componentes FuzzyComponents descrito en la sección anterior, se tomó la decisión de elegir la plataforma NetBeans (www.netbeans.org). Las herramientas facilitadas por el entorno para manipular ficheros XML, su integración con Java, y las posibilidades de desarrollar módulos que amplíen la funcionalidad de la plataforma fueron las razones básicas para su elección. Además el entorno facilita enormemente el trabajo con componentes JavaBeans lo que posibilita construir el editor gráfico basándose en este tipo de componentes y a la vez que esas componentes puedan ser reutilizadas en otros entornos.

En la actualidad se está desarrollando un módulo NetBeans para la edición de especificaciones iXSCL, al que hemos denominado NetFuzzy, y un conjunto de JavaBeans que permitan la edición gráfica de este tipo de

especificaciones. La integración de las JavaBeans en el módulo NetFuzzy puede realizarse fácilmente.

Uno de los componentes principales del entorno es un árbol donde se presenta la estructura del documento iXSCL en edición; lo que posibilita una visión gráfica de los distintos constructores de iXSCL tales como Contextos Lingüísticos, Ligaduras o Bases de Reglas. El árbol está sincronizado con un editor de documentos XML, por lo que el documento iXSCL puede editarse en la actualidad bien a través de la representación en árbol bien a través el editor de texto. Cada fragmento de iXSCL se asocia a un nodo del árbol que muestra todas sus propiedades y que a su vez permite desplegar todos los elementos iXSCL que son descendientes del nodo en cuestión; estos nodos se denominan BeanViews. Un problema importante relacionado con la definición de BeanViews, es la asociación de fragmentos de iXSCL que representan objetos que en el esquema iXSCL tienen un tipo abstracto con sus BeanViews. Un requisito del entorno es que proporcione métodos sencillos para definir un nuevo tipo, por ejemplo, de función de pertenencia, en el esquema y asociarlo con una nueva BeanView. El usuario debe poder ampliar fácilmente el lenguaje con nuevos tipos de funciones de pertenencia, lo que fuerza a que, simultáneamente, sea capaz de ampliar también, con las menores complicaciones posibles, el módulo NetFuzzy para que reconozca el nuevo tipo de función de pertenencia. En el módulo se recurre para ello a un servicio de directorio, que dado un elemento DOM intenta resolver la clase Java asociada que construye su BeanView.

En cuanto a la edición gráfica de documentos iXSCL, es posible asociar un componente gráfico de edición (customizer) a cada BeanView. En este caso se ha optado porque cada una de esas JavaBeans de edición gráfica sea responsable de un fragmento de XML bastante mayor que en el caso de las BeanViews y por tanto no hay una componente gráfica por cada BeanView. En la actualidad no se encuentran desarrolladas todas las componentes gráficas JavaBeans necesarias para realizar una edición completa de un documento iXSCL, pero se completarán próximamente.

Durante todo este tiempo de desarrollo del sistema, y teniendo en cuenta que la plataforma NetBeans está sometida a un continuo desarrollo, en ocasiones se ha mostrado como demasiado complicada y su base de código un tanto caótica. Un ejemplo puede ser el hecho de que no se ha fijado adecuadamente el uso de hilos de ejecución en el entorno lo que ha provocado problemas con el tratamiento del modelo DOM subyacente a un documento iXSCL. Las implementaciones habituales de las herramientas de manipulación de este modelo no están pensadas para entornos multihilos. Sin embargo, sí se aprecia que el entorno NetBeans propone bastantes buenas ideas, lo que con un poco más de organización y simplicidad podría alcanzar esa potencia que por ahora tan solo se adivina.

5. CONCLUSIONES

En este trabajo se ha pretendido mostrar el estado actual de desarrollo del sistema NetFuzzy y del conjunto de componentes software FuzzyComponents, herramientas software destinadas a facilitar el uso en sistemas distribuidos de Sistemas de Reglas Fuzzy. A pesar de no estar concluido el trabajo de codificación del sistema, su arquitectura y diseño conceptual esta finalizado, así como muchas de sus partes. Cuando concluya el trabajo, el sistema debe ser sometido a pruebas desarrollando aplicaciones que aprovechen sus potenciales.

Referencias

[1] Constantin von Altrock (1997) *Fuzzy Logic and NeuroFuzzy in Business and Finance.* Prentice Hall.

[2] Moreno-Velo, F.J.; Sánchez-Solano, S.; Barriga, A., Baturone, I, López, D.R. (2001) A specification language for fuzzy systems. *Mathware and Soft Computing*, VIII(3):239–253.

[3] de Soto, A.R.; Cuervo Fernández, E.(2002) Componentes Distribuidas CORBA para el Cálculo de Sistemas de Reglas Borrosas, Actas Simposio de Informática y Telecomunicaciones pp. 355–365, Sevilla.

[4] de Soto, A.R.; Cuervo Fernández, E.; Capdevila, A. C.; (2003) *Entorno de componentes software distribuidas para la gestión de sistemas de reglas borrosas*, Proc. X Sigef Congress Vol. 1 pp. 411–422, León, España.

[5] de Soto, A.R.; Capdevila, A. C.; Cuervo Fernández, E. (2003) An XML vocabulary for Soft Computing, *Proc. 3rd Eusflat ConferencePublicacion*, pag. 645–649, Zittau, Alemania.

[6] De Soto, A.R., Capdevila, C.A., Fernández, E.C. (2003) Fuzzy Systems and Neural Networks XML Schemas for Soft Computing, *Math & Soft Computing* Vol. X num. 2-3, pp. 43–56.

[7] XSL Transformations (XSLT) versión 1.0. *Technical Report REC-XSLT-19991116*, W3C (MIT, INRIA, Keio), Noviembre 1999.

[8] Extensible Markup Language (XML) 1.0 (Segunda edición). *Technical Report REC-XML-20011006*, W3C (MIT, INRIA, Keio), Octubre 2000.

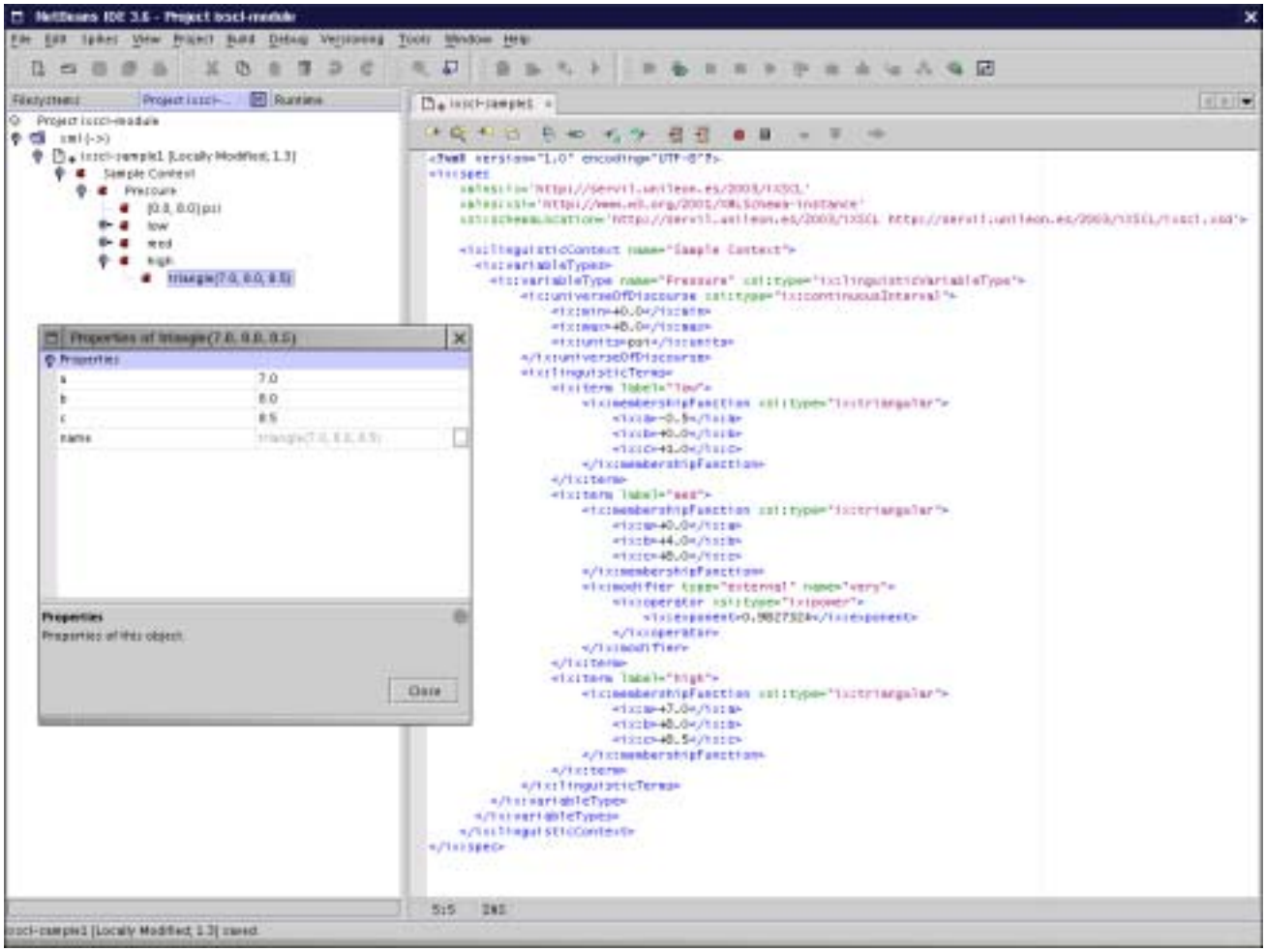


Figura 1: Pantalla ejemplo del módulo NetFuzzy

[9] XML schema part 0: Primer. *Technical Report REC-XMLSchema-0-200110502*, W3C (MIT, INRIA, Keio), Mayo 2001.

LA DEFINICIÓN DE CONCEPTOS LINGÜÍSTICOS EN XFUZZY 3

F. J. Moreno Velo I. Baturone S. Sánchez-Solano A. Barriga
Instituto de Microelectrónica de Sevilla (CNM-CSIC)
Avda. Reina Mercedes s/n, Edif. CICA. 41012-Sevilla. España
xfuzzy-team@imse.cnm.es

Resumen

Este trabajo presenta un estudio sobre las diferentes formas de definir conceptos lingüísticos difusos, ya sean funciones de pertenencia libres o familias de funciones de pertenencia, y describe la capacidad del lenguaje XFL3 (el lenguaje de especificación formal de Xfuzzy 3) tanto para utilizar ambas formas como para extender el conjunto de funciones y familias disponibles. El trabajo incluye un ejemplo de aplicación en el que se pone de manifiesto las ventajas e inconvenientes de cada opción.

Palabras clave: Funciones de Pertenencia, Diseño de Sistemas Difusos, Herramientas CAD, Lenguajes formales.

1 INTRODUCCIÓN

Las funciones de pertenencia, como forma de representar conceptos lingüísticos, suponen la base de cualquier metodología difusa. Desde un punto de vista teórico, cualquier función definida sobre el universo de discurso de una variable que tome valores en el intervalo [0, 1] puede ser utilizada como una función de pertenencia [1]. Teniendo en cuenta el significado lingüístico asociado a estas funciones, las formas más utilizadas corresponden a la descripción difusa de valores concretos (funciones δ), de intervalos (funciones π) y de intervalos extremos del universo de discurso (funciones *s* y funciones *z*)[2].

Desde un punto de vista práctico, las funciones de pertenencia se suelen definir como funciones analíticas. Generalmente, la expresión matemática de estas funciones contiene una serie de parámetros que deben ser introducidos al definir la función. Por ejemplo, la definición de una función de pertenencia gaussiana requiere la introducción del valor del centro y la anchura de la función. Las funciones analíticas más utilizadas incluyen funciones lineales a tramos (triángulos, trapecios y rampas) y funciones no lineales (campanas y sigmoides) [3]. La Figura 1 muestra una representación gráfica de estas funciones analíticas.

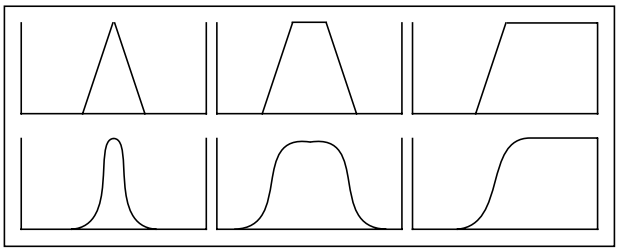


Figura 1: Algunas de las funciones analíticas más usadas.

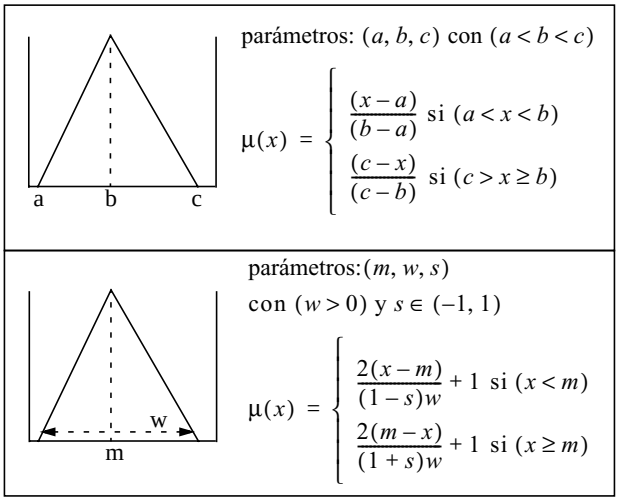


Figura 2: Diferentes expresiones para definir una función triangular.

A la hora de afrontar el diseño de un sistema difuso, no sólo es importante considerar la forma de las funciones de pertenencia a utilizar, sino también la expresión matemática más adecuada para estas funciones. Por ejemplo, una función triangular puede ser definida mediante tres parámetros: el ápice y los límites del triángulo (Fig. 2.a), o el ápice, anchura y sesgo (Fig. 2.b). En general, no se ha prestado demasiada atención a la selección de la expresión matemática más adecuada para las funciones de pertenencia. Otros aspectos, como la selección de la forma de las funciones o los operadores a utilizar han sido mucho más estudiados, ya que influyen directamente en el comportamiento del sistema. Por otro lado, la definición de las funciones de pertenencia en términos de un conjunto de parámetros o de otro no tiene influencia sobre el funcionamiento del sistema difuso.

Este trabajo ha sido financiado parcialmente por el proyecto CICYT TIC2001-1726.

Sin embargo, cuando se aplican técnicas de *soft computing* en el diseño de sistemas difusos (como la identificación, ajuste o simplificación automática de estos sistemas) la selección de los parámetros que definen las funciones cobra una gran importancia. Estas técnicas suelen modificar automáticamente el valor de estos parámetros, por lo que resulta muy importante utilizar expresiones paramétricas que permitan mantener la interpretabilidad lingüística del sistema bajo estos procesos automatizados.

La estructura de este trabajo es la siguiente. El apartado 2 describe las diferentes formas de definir paramétricamente un conjunto de funciones de pertenencia, así como la interpretación lingüística asociada y sus ventajas e inconvenientes. El apartado 3 describe cómo se han desarrollado estas formas de representación en XFL3, el lenguaje de especificación formal del entorno Xfuzzy 3. Por último, el apartado 4 muestra un ejemplo en el que se utilizan diferentes definiciones y se estudian los resultados que permiten obtener.

2 FORMAS DE DEFINICIÓN DE FUNCIONES DE PERTENENCIA

La primera cuestión a la hora de definir un conjunto de funciones de pertenencia (relacionadas con las etiquetas lingüísticas de una variable) es si utilizar una lista de parámetros para definir el conjunto de funciones completo, o si definir cada función de pertenencia por medio de sus propios parámetros. En el primer caso estaremos hablando de *familias de funciones de pertenencia*, mientras que el segundo caso se conoce como *funciones de pertenencia libres* (Fig. 3).

La definición de las funciones de pertenencia en forma de familia tiene varias ventajas. En primer lugar, requiere un número de parámetros menor que el que utilizan las funciones libres. Esto puede facilitar el ajuste de los sistemas y permitir el uso de algunos algoritmos de aprendizaje (como los algoritmos de enfriamiento simulado o los algoritmos genéticos) que no son apropiados cuando el número de parámetros a ajustar es alto. En segundo lugar, la utilización de una lista de parámetros comunes facilita la garantía del significado lingüístico de las funciones, ya que no pueden variar su orden, simetría o grado de solapamiento por construcción. Por último, las familias permiten definir funciones con un grado de solapamiento fijo. Esto resulta muy útil para poder aplicar mecanismos de inferencia simplificados (utilizando reglas activas, por ejemplo) lo que aumenta la velocidad de inferencia del sistema y permite la integración en circuitos difusos. El principal inconveniente de este tipo de definición es que representa restricciones muy fuertes a las variaciones de las funciones, lo que puede ser un obstáculo a la hora de optimizar el comportamiento del sistema. Esto es especialmente importante en el caso de variables de salida.

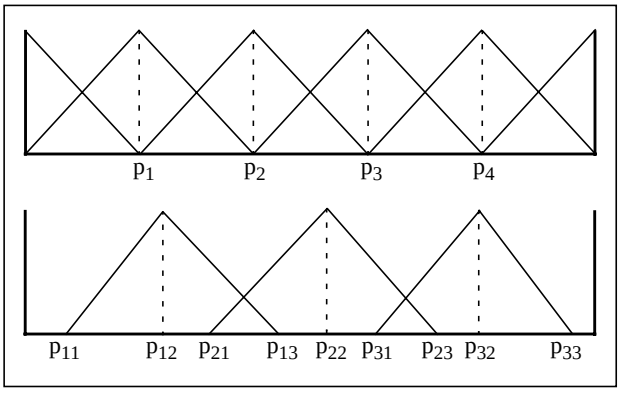


Figura 3: Differentes formas de definición: a) Familia de funciones; (b) Funciones de pertenencia libres.

Por su parte, la definición de funciones de pertenencia libres no impone ninguna restricción respecto al movimiento de las funciones en procesos de ajuste automático. Esto resulta muy útil en procesos de adquisición de conocimiento, ya que permite aprender la forma, posición y grado de solapamiento más adecuado para las funciones. Las funciones libres permiten obtener mejores resultados en los procesos de optimización, ya que utilizan un mayor número de grados de libertad. El principal inconveniente es la dificultad para mantener el significado lingüístico en procesos de optimización automáticos y que pueden generar representaciones incompatibles con esquemas de inferencia simplificados.

En general, las funciones de pertenencia libres resultan más apropiadas para describir variables de salida, mientras que las familias de funciones son más adecuadas para describir variables de entrada. En los casos en los que no existe mucha información sobre la forma más adecuada para representar una variable de entrada, una buena táctica suele ser representarlas inicialmente mediante funciones libres y realizar un proceso de ajuste, de forma que el resultado de esta optimización permita obtener conocimiento sobre la forma, simetría y grado de solapamiento adecuados. Esta información puede emplearse para describir la función por medio de una familia de funciones adecuada.

Un aspecto importante, tanto para la definición de funciones libres como para la de familias de funciones, es la selección de la descripción paramétrica más adecuada. Una descripción paramétrica adecuada es aquella con el mayor grado de libertad posible capaz de mantener el significado lingüístico de las funciones, así como las restricciones particulares que considere el diseñador. La Figura 4 muestra un ejemplo con un conjunto de 7 funciones triangulares. Si se describen por medio de funciones libres se necesitan 21 parámetros (3 por función) (Fig. 4b). Es posible mantener el orden y el grado de solapamiento 2 utilizando una familia de funciones triangulares con 5 parámetros (Fig. 4c). Si la variable descrita tiene un comportamiento simétrico, se

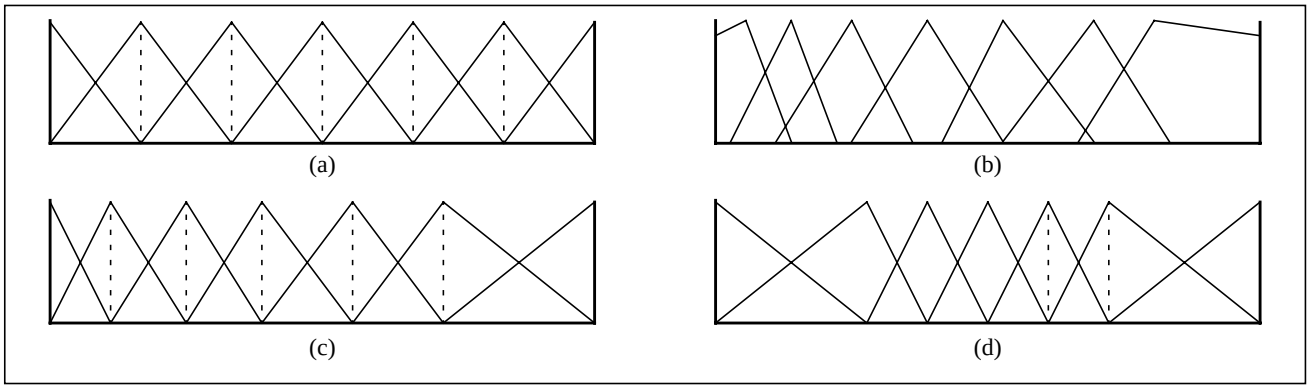


Figura 4: Ejemplo de la evolución de un conjunto de funciones: (a) estado inicial; (b) estado final con funciones libres; (c) estado final con una familia de funciones; (d) estado final con una familia de funciones simétricas.

puede utilizar una familia de funciones triangulares simétricas, utilizando tan solo 2 parámetros (Fig. 4d).

La capacidad de utilizar tanto funciones de pertenencia libres como familias de funciones es una característica muy deseable en los entornos CAD de diseño de sistemas difusos. Estos entornos deberían ofrecer las familias y funciones más usuales. Un conjunto típico de funciones libres contiene triángulos, trapezoides, rampas, campanas, sigmoides y singularidades. Un conjunto típico de familias contiene familias de triángulos, de trapecios, de campanas y de *splines* (para un grado de solapamiento 3), así como sus versiones simétricas. Otra característica deseable es la capacidad de crear nuevas funciones y familias e integrarlas en el flujo de diseño del entorno. El siguiente apartado muestra cómo se han considerado estas características en el entorno Xfuzzy 3.

3 LA DEFINICIÓN DE FUNCIONES DE PERTENENCIA EN XFL3

XFL3 es un lenguaje de descripción formal de sistemas difusos complejos [4], definido como lenguaje de especificación común de las herramientas integradas en el entorno de desarrollo Xfuzzy 3 [5]. Estas herramientas cubren las diferentes etapas del flujo de diseño de los sistemas difusos: descripción, identificación, ajuste, simplificación, verificación y síntesis. Las principales características de XFL3 son su alta expresividad, extensibilidad y modularidad. La modularidad se refiere a la capacidad del lenguaje para describir sistemas jerárquicos formados por diferentes módulos, tanto bases de reglas como bloques no difusos. En cuanto a la expresividad, el lenguaje permite describir reglas difusas por medio de cualquier combinación de conectivas difusas y modificadores lingüísticos. Por último, el lenguaje incorpora mecanismos que permiten extender el conjunto de funciones disponibles para representar los diferentes bloques del sistema: operadores difusos, modificadores lingüísticos, funciones de pertenencia libres, familias de funciones, métodos de concreción o bloques no difusos.

La descripción de un sistema por medio de XFL3 se divide en dos partes: la descripción lógica (incluida en ficheros de extensión “xfl”) y la descripción funcional (incluida en ficheros llamados *paquetes de funciones*), que describe matemáticamente las funciones utilizadas en la descripción lógica. La descripción lógica contiene la definición de tres tipos de objetos (conjunto de operadores, tipos de variables y bases de reglas), así como la descripción de la estructura modular del sistema. La definición de las funciones de pertenencia asignadas a las variables se incluye en los tipos de variables. La sintaxis de estos objetos en XFL3 es la siguiente:

```
type identificador [min,max; card] {
    fam_id[] paquete.familia(parámetros);
    mf_id paquete.función(parámetros);
    mf_id fam_id[índice];
.... }
```

Esta definición incluye el nombre del objeto, la descripción del universo de discurso (que incluye los límites del mismo y la cardinalidad), la lista de etiquetas lingüísticas (*mf_id*) y la definición de la función de pertenencia asociada. Esta función puede ser una función de pertenencia libre (seleccionada en un paquete de funciones) o un miembro de una familia de funciones, definida previamente mediante su selección en un paquete de funciones. La Figura 5 muestra un ejemplo de definición de un tipo de variable utilizando funciones libres o familias de funciones.

Los paquetes de funciones contienen la definición matemática de las funciones utilizadas en la descripción lógica del sistema. Estos paquetes se definen en archivos con extensión “.pkg” y pueden ser creados libremente por los diseñadores. De esta forma, el conjunto de funciones disponibles en el diseño de sistemas difusos puede ser ampliado. Un paquete contiene un conjunto de definiciones de funciones de seis tipos diferentes: funciones binarias (asociadas a las conectivas difusas), funciones unarias (asociadas a los modificadores lingüísticos), métodos de concreción, funcio-

```
type TInput1 [0,100] {
 very_small xfl.triangle(-25,0,25);
 small      xfl.triangle(0,25,50);
 medium     xfl.triangle(25,50,75);
 big        xfl.triangle(50,75,100);
 very_big   xfl.triangle(75,100,125);
}

type TInput2 [0,100] {
 family[]   xfl.triangular(25,50,75);
 very_small family[0];
 small      family[1];
 medium     family[2];
 big        family[3];
 very_big   family[4];
}
```

Figura 5: Ejemplo de definición de tipos de variable utilizando funciones libres y familias de funciones.

nes concretas (asociadas a bloques no difusos), funciones de pertenencia libres y familias de funciones de pertenencia. Estas definiciones se basan en una serie de bloques que definen los parámetros de la función, la descripción matemática en varios lenguajes de programación y algunas otras características.

La Figura 6 y la Figura 7 muestran un ejemplo de definición de una función de pertenencia libre y de una familia de funciones de pertenencia, respectivamente, en los que se refleja el contenido de los diferentes bloques de las definiciones. El bloque *parameters* permite definir el conjunto de parámetros de la función. El último identificador puede ir seguido de corchetes, indicando que se trata de una lista de parámetros. El bloque *requires* expresa los requisitos que deben cumplir los valores de estos parámetros para representar una definición válida. En este bloque se pueden utilizar las constantes *min* y *max* (universo de discurso) y la función *sorted()*. La definición de una familia de funciones debe contener el bloque *members* que indica el número de funciones que contiene la familia.

Los bloques *java*, *ansi_c* y *cplusplus* contienen la descripción matemática de la función en estos lenguajes de programación. Estos bloques se dividen en cinco subbloques: *equal*, *greatereq*, *smallereq*, *center* y *basis*. El subbloque *equal* define el grado de pertenencia del valor *x*. Los subbloques *greatereq* y *smallereq* definen el grado de pertenencia del valor *x* para las modificaciones lingüísticas *mayor_o_igual* y *menor_o_igual*. Los subbloques *center* y *basis* calculan el valor del centro y de la base de la función. En la definición de familias de funciones, la variable *i* indica el índice de la función a evaluar.

El bloque *derivative* define las derivadas de la función. Esta información es importante para poder utilizar las funciones en procesos de ajuste automático que usen algoritmos basados en el gradiente del sistema. Este bloque se divide en los cinco subbloques descritos anteriormente y su objetivo es almacenar en la variable *deriv[]* el valor de las derivadas respecto a cada parámetro.

La modificación automática de los parámetros puede generar situaciones en las que las restricciones de estos parámetros no se cumplan. El bloque *update* se utiliza para evitar estos problemas. El código de este bloque actualiza los valores de los parámetros (variable *pos[]*) a partir de los desplazamientos deseados (variable *desp[]*) teniendo en cuenta la selección de parámetros a ajustar (variable *adj[]*), asegurando que la descripción de la función sea consistente. La función *sortedUpdate()* puede utilizarse para asegurar una actualización que mantenga una condición de orden en los valores de los parámetros.

El contenido de estos bloques sirve de base para generar una clase Java que contiene toda la funcionalidad incluida en la definición. Esta clase puede ser instanciada en tiempo de ejecución, permitiendo el uso de nuevas funciones definidas por el diseñador. Este mecanismo dota al entorno Xfuzzy 3 de una enorme flexibilidad en su capacidad de diseño. Las herramientas integradas en el entorno han sido programadas para soportar este tipo de funciones extendidas por el usuario.

4 EJEMPLO DE APLICACIÓN

Consideremos como ejemplo el diseño de un sistema difuso que aproxime el comportamiento de la función mostrada en la Figura 8. Se trata de una función con dos variables de entrada que se utiliza comúnmente para evaluar aproximadores difusos [6]. Para representar el comportamiento de esta función se ha utilizado un conjunto de 441 patrones de entrada-salida. Inicialmente se ha considerado una descripción de las variables de entrada por medio de 7 funciones de pertenencia gaussianas distribuidas homogéneamente. La base de reglas considerada contiene todas las combinaciones posibles de estas funciones, lo que forma un total de 49 reglas. Cada una de estas reglas contiene una conclusión independiente descrita por medio de una función gaussiana. Inicialmente el comportamiento del sistema es plano, ya que las 49 funciones de pertenencia de salida son iguales. Para aproximar el comportamiento de este sistema a la función objetivo se ha utilizado la herramienta de aprendizaje automático integrada en Xfuzzy 3 [7], y se han seleccionado como ajustables todos los parámetros de las funciones de pertenencia (126). La Figura 9 representa la descripción de las funciones de pertenencia tras el proceso de aprendizaje. El comportamiento del sistema se aproxima a la función objetivo con un RMSE del 0.46%.

A partir del resultado anterior se puede extraer información para obtener una representación que pueda ser sintetizada mediante técnicas de inferencia simplificada. En particular,

```
mf triangle {
  parameter a, b, c;
  requires { a<b && b<c && b>=min && b<=max }
  java {
    equal {
      if(a<x && x<=b) return (x-a)/(b-a);
      if(b<x && x<c) return (c-x)/(c-b);
      return 0;
    }
    greatereq { ... }
    smallereq { ... }
    center { ... }
    basis { ... }
  }
  ansi_c { ... }
  cplusplus { ... }
  derivative {
    equal {
      if(a<x && x<b) {
       deriv[0] = (x-b)/((b-a)*(b-a));
       deriv[1] = (a-x)/((b-a)*(b-a));
      } else if(b<x && x<c) {
        deriv[1] = (c-x)/((c-b)*(c-b));
        deriv[2] = (x-b)/((c-b)*(c-b));
      } else if(x==a) {
        deriv[0] = 0.5/(a-b);
      } else if(x==b) {
        deriv[1] = 0.5/(a-b) + 0.5/(c-b);
      } else if(x==c) {
        deriv[2] = 0.5/(c-b);
      }
    }
    ....
  }
  update {
    pos = sortedUpdate(pos,desp,adj);
    if(pos[1]<min) pos[1]=min;
    if(pos[2]<=pos[1]) pos[2] = pos[1]+step;
    if(pos[1]>max) pos[1]=max;
    if(pos[0]>=pos[1]) pos[0] = pos[1]-step;
  }
}
```

Figura 6: Ejemplo de definición de una función de pertenencia libre: la función *triangle*.

```
family triangular {
 parameter p[];
 requires { p.length>0 && p[0]>min &&
            p[p.length-1]<max && sorted(p) }
 members { return p.length+2; }
 java {
  equal {
   double a = (i==0? min-1 : (i==1 ? min : p[i-2]));
   double b = (i==0? min : (i==p.length+1? max : p[i-1]));
   double c = (i==p.length? max :
               (i==p.length+1? max+1 : p[i]));
   return (a<x && x<=b? (x-a)/(b-a) :
           (b<x && x<c? (c-x)/(c-b) : 0));
  }
  greatereq { ... }
  smallereq { ... }
  center { ... }
  basis { ... }
 }
 ansi_c { ... }
 cplusplus { ... }
 derivative {
   equal {
     double a = ...;
     double b = ...;
     double c = ...;
     if(i>=2) {
       if(a<x && x<b) deriv[i-2] = (x-b)/((b-a)*(b-a));
       else if(x==a)  deriv[i-2] = 0.5/(a-b);
       else deriv[i-2] = 0;
     }
     if(i>=1 && i<=p.length) {
       if(a<x && x<b) deriv[i-1] = (a-x)/((b-a)*(b-a));
       else if(b<x && x<c) deriv[i-1] = (c-x)/((c-b)*(c-b));
       else if(x==b) deriv[i-1] =  0.5/(a-b) + 0.5/(c-b);
       else  deriv[i-1] = 0;
     }
     if(i<p.length) {
       if(b<x && x<c) deriv[i] = (x-b)/((c-b)*(c-b));
       else if(x==c) deriv[i] = 0.5/(c-b);
       else deriv[i] = 0;
     }
   }
   .....
 }
 update {
    pos = sortedUpdate(pos,desp,adj);
    if(pos[0]<min) {
       pos[0]=min;
       for(int i=1;i<p.length; i++) {
         if(pos[i]<=pos[i-1]) pos[i] = pos[i-1]+step;
         else break;
       }
    }
    if(pos[p.length-1]>max) {
       pos[p.length-1]=max;
       for(int i=p.length-2; i>=0; i--) {
         if(pos[i]>=pos[i+1]) pos[i] = pos[i+1]-step;
         else break;
       }
    }
  }
}
```

Figura 7: Ejemplo de definición de una familia de funciones de pertenencia; la familia *triangular.*

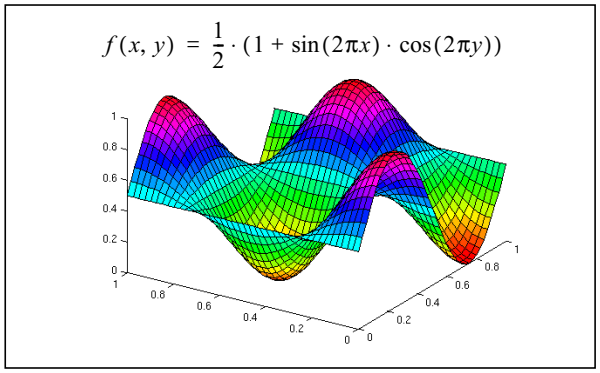


Figura 8: Función objetivo.

las variables de entrada muestran una descripción simétrica y el grado de solapamiento parece estar limitado a 3. Estas características pueden imponerse por construcción utilizando una descripción de las variables de entrada por medio de familias de B-splines simétricos. De esta forma, el número de parámetros de estas variables se reduce de 28 a 4, utilizando 2 parámetros por familia. La Figura 10 muestra el resultado de un proceso de aprendizaje utilizando esta nueva representación. En esta representación se puede apreciar cómo las funciones de pertenencia de la variable de salida tienden a agruparse, y pueden reducirse de 49 a 9 funciones por medio de un proceso de *clustering*. El grado de aproximación es ahora de un RMSE del 0.89%. Por tanto, el uso de familias de funciones de pertenencia ha empeorado ligeramente el grado de aproximación, pero permite reducir el número de reglas activas de 49 a 9, fijando un grado de solapamiento 3 por construcción. Esto permite generar representaciones con una velocidad de inferencia mayor y facilita la síntesis hardware del sistema difuso.

5 CONCLUSIÓN

Al introducir las técnicas de aprendizaje automático en el flujo de diseño de los sistemas difusos, la descripción paramétrica de las funciones de pertenencia adquiere una gran

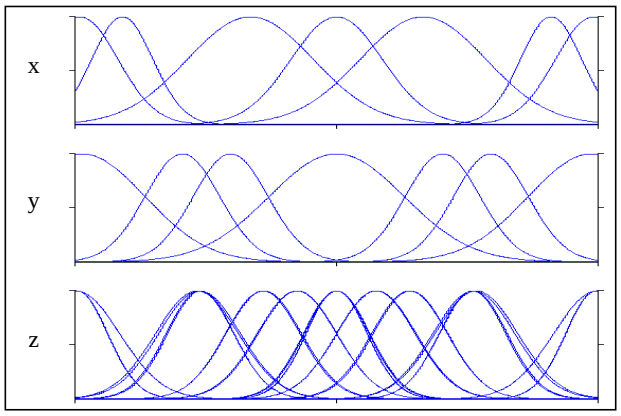


Figura 9: Resultados obtenidos utilizando funciones libres.

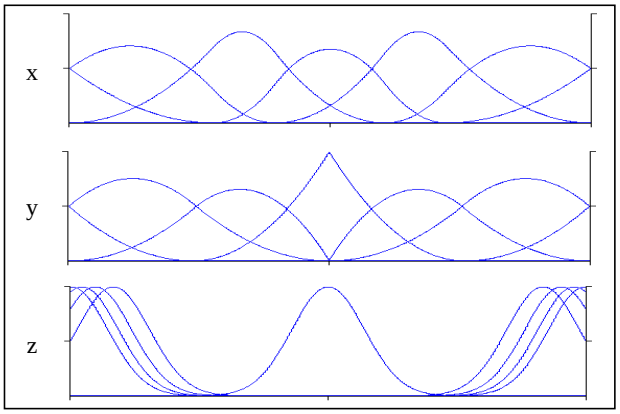


Figura 10: Resultados obtenidos utilizando familias de funciones.

importancia. Una descripción paramétrica adecuada permite mantener ciertas características deseables, como el orden, simetría o grado de pertenencia de las funciones, a lo largo de los procesos de modificación automáticos. La última versión de XFL3, el lenguaje de especificación de Xfuzzy 3, permite utilizar tanto funciones de pertenencia libres como familias de funciones definidas por el usuario, de manera que éste pueda introducir nuevas definiciones que representen las restricciones deseadas para cada sistema. Una información más detallada sobre XFL3 y Xfuzzy 3 puede encontrarse en la página oficial de Xfuzzy:

`http://www.imse.cnm.es/Xfuzzy/`

Referencias

[1] L.A. Zadeh, “Fuzzy sets”, Information and control, Vol.8, pp. 338-358, 1965.

[2] E. Cox, The Fuzzy Systems Handbook, AP Professional, 1999.

[3] T.J. Ross, Fuzzy Logic with Engineering Applications, McGraw-Hill inc., 1995.

[4] F.J. Moreno-Velo, S. Sánchez-Solano, A. Barriga, I. Baturone, D. R. López, “XFL3: A New Fuzzy System Specification Language”, Mathware & Soft Computing, Vol. 8, No. 3, pp. 239- 253, 2001.

[5] F.J. Moreno Velo, I. Baturone, S. Sánchez Solano, A. Barriga, “Rapid Design of Fuzzy Systems with Xfuzzy”, IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE 2003), pp. 342-347.

[6] R. Rovatti, R. Guerrieri, “Fuzzy sets of rules for system identification”, IEEE Trans. Fuzzy Systems, Vol 4, No. 2, pp.89-102, 1996.

[7] F.J. Moreno Velo, I. Baturone, R. Senhadji, S. Sánchez Solano, “Tuning Complex Fuzzy Systems by Supervised Learning Algorithms”, IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE 2003), pp. 226-231.

EXTRACCIÓN DE REGLAS DIFUSAS PARA DESCRIBIR LA DINÁMICA DE REDES NEURONALES RECURRENTES

Cuéllar M.P.
Dpto. Ciencias de Computación e I.A
ETS. Ing. Informática
Universidad de Granada
C/. Pdta Daniel Salcedo Aranda, s.n.
CP 18003. Granada
manupc@decsai.ugr.es

Delgado M.
Dpto. Ciencias de Computación e I.A
ETS. Ing. Informática
Universidad de Granada
C/. Pdta Daniel Salcedo Aranda, s.n.
CP 18003. Granada
mdelgado@ugr.es

Pegalajar M.C.
Dpto. Ciencias de Computación e I.A
ETS. Ing. Informática
Universidad de Granada
C/. Pdta Daniel Salcedo Aranda, s.n.
CP 18003. Granada
mcarmen@decsai.ugr.es

Resumen

Las redes neuronales se han mostrado como una buena herramienta para aproximar la solución de problemas complejos. Sin embargo, una vez que una red neuronal ha sido entrenada, es difícil interpretar el funcionamiento interno de la misma. Este problema aumenta de forma considerable en redes recurrentes. En este trabajo se presenta un método para mejorar la interpretabilidad de una red neuronal recurrente completa, de modo que el conocimiento que encapsula la estructura dinámica de la red pueda ser codificado en forma de reglas difusas, sin pérdida de precisión.

Palabras Clave: Redes Neuronales Recurrentes, Interpretación, Reglas Difusas.

1 INTRODUCCIÓN

Las redes neuronales artificiales se han utilizado para resolver multitud de problemas complejos, obteniendo un buen comportamiento con un alto grado de precisión. Sin embargo, una vez que se ha entrenado una red neuronal, ésta codifica el conocimiento dentro de su estructura interna, dificultando la interpretabilidad. Por esta razón, las redes neuronales han sido tradicionalmente calificadas como "cajas negras". Por otra parte, la teoría de los conjuntos difusos, introducida por Zadeh en 1965 [12], nos permite utilizar conceptos imprecisos mediante términos matemáticos bien definidos. Desde entonces, la lógica difusa ha sido aplicada a multitud de problemas, con muy buenos resultados. Los sistemas basados en reglas difusas son una buena herramienta para modelar conocimiento, debido a que poseen un alto grado de interpretabilidad.

Las redes neuronales pueden "aprender" un problema fácilmente dado un conjunto suficiente de ejemplos, ya que hay un gran número de algoritmos de entrenamiento adaptados para cada tipo de red. Sin embargo, es difícil construir un sistema basado en reglas difusas cuando no se posee conocimiento a priori sobre el problema. Además, una red neuronal puede generalizar un problema mejor que un sistema difuso clásico, ya que la red posee una visión general acerca del problema a tratar, mientras que el sistema de reglas difusas suele tener un conocimiento local del espacio de soluciones.

A partir de los 90 surgen numerosas propuestas para interpretar el conocimiento de una red neuronal FeedForward por medio de reglas difusas [1][9][10][11]. Para las redes neuronales recurrentes, la mayoría de los métodos que se han propuesto están basados en la identificación de autómatas finitos. Por ejemplo, C.L. Giles *et al.* extraen autómatas finitos determinísticos [2][3][5] (y también difusos [4]) a partir de redes recurrentes de segundo orden, entrenadas principalmente en problemas de inferencia gramatical. I. Visser *et al.* utilizan Modelos Ocultos de Markov para extraer autómatas finitos determinísticos de redes recurrentes de Elman [8]. La extracción de reglas difusas en redes recurrentes es un proceso mucho más complejo que en redes *feedforward*, ya que existe una dependencia temporal por el retraso que se produce en la recurrencia. Dicha dependencia temporal hace que una red neuronal recurrente pueda verse como un sistema dinámico, lo que dificulta enormemente el análisis directo del impacto de las entradas con respecto a las salidas. Por esta razón, la extracción de reglas en redes neuronales recurrentes puede descomponerse en dos etapas: en la primera, se realiza un análisis de la red neuronal como sistema dinámico, de forma que se obtengan las relaciones existentes entre el estado de la red y las conexiones de entrada, con respecto a las salidas. En la segunda etapa, deberá realizarse un análisis del sistema que modela la red, de forma que se obtenga conocimiento sobre el

problema concreto a tratar, una vez conocida la dinámica y evolución de la red recurrente asociada.

En este trabajo se propone un método para solucionar el problema correspondiente a la primera etapa. El objetivo es aumentar la interpretabilidad de una red neuronal recurrente completa [6][7], de modo que pueda identificarse un conjunto de reglas difusas que presenten el mismo funcionamiento que la estructura dinámica de la red. En la sección 2, se presenta el modelo de red neuronal recurrente completa. Seguidamente, en la sección 3 se expone el método que transforma una red neuronal recurrente completa (no interpretable) en una red recurrente equivalente, con un mayor grado de interpretabilidad. La sección 4 muestra cómo se obtiene el sistema de reglas difusas equivalente a la dinámica de la red. Por último, la sección 5 expone las conclusiones obtenidas.

2 MODELO DE RED NEURONAL

Una red neuronal recurrente completa consta de dos capas de neuronas; una de ellas es usada como entrada de datos, y otra como capa de neuronas ocultas y/o de salida. Además, la red guarda en una capa de neuronas de estado los valores que la capa de salida tenía en el instante anterior. Este esquema se ilustra en la figura 1.

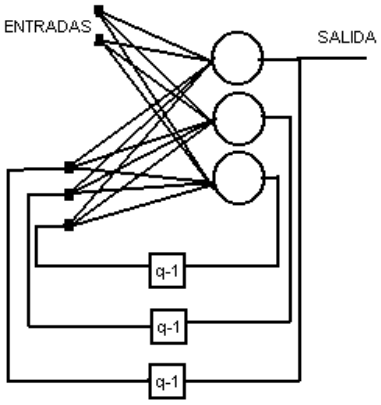


Figura 1. Red Recurrente Completa

La figura 1 muestra un ejemplo de la estructura de una red recurrente completa con dos entradas, una salida, y dos neuronas ocultas. Por tanto, la capa de neuronas de estado tendrá tres neuronas, y en el instante **t** contendrán los valores que las neuronas de salida y ocultas habían tenido en el instante **t-1**, respectivamente. La ecuación 1 muestra el comportamiento que rige la red. Sea $\mathbf{X_i(n+1)}$ el valor de la neurona oculta/de salida **i** en el instante **n+1**. El cálculo de este valor se realiza de la siguiente forma:

$$X_i(n+1)=F\left(\sum_{j=1}^{I} W_{i,j}u_j(n)+\sum_{j=1}^{H} W_{i,j+I}X_j(n)\right) \qquad (1)$$

$$F(x)=\frac{\tanh(x)+1}{2} \qquad (2)$$

- $\mathbf{W_{i,j}}$ Es el peso de la conexión entre el valor de entrada **j** y la neurona oculta/de salida **i**.
- $\mathbf{W_{i,j+I}}$ Es el peso de la conexión entre el valor del estado dado por la neurona oculta **j** y la neurona oculta **i**.
- **I** es el número de entradas.
- **H** es el número de neuronas existentes en la capa oculta/de salida.
- **F(·)** es la función de activación de la neurona oculta/de salida **i**

3 EXTRACCIÓN DE REGLAS DE LA DINÁMICA DE REDES RECURRENTES

En esta sección se presenta el método para transformar una red neuronal recurrente completa en otra red recurrente equivalente, pero cuyo grado de interpretabilidad sea superior al de la red original. Esto nos permite obtener un conjunto de reglas difusas de tipo Mamdani que muestran el funcionamiento dinámico de la red, independientemente de las entradas y las salidas al sistema, de forma que se pueda realizar un análisis posterior sobre la evolución de la dinámica de la red.

El método consta de 4 pasos: descomposición de entradas. difuminación de entradas, descomposición de salidas, y agregación de salidas. Una vez aplicados estos pasos, podrá identificarse fácilmente un conjunto de reglas difusas de la forma: *Si la entrada i es X, entonces el aporte de la entrada i para conseguir la salida j es Y*.

Podrían haberse utilizado métodos distintos al aquí propuesto, existentes en las referencias, para obtener un sistema de reglas difusas, pero obteniendo resultados y conclusiones distintos. Nuestro objetivo es obtener reglas que sean lo más interpretables posible, por lo que nuestro método devolverá reglas de tipo Mamdani. Por otra parte, al tratar de analizar la red como sistema dinámico, el conjunto de reglas resultantes deberá ser independiente de los datos del problema (entradas y salidas). No se ha encontrado ninguna referencia en la bibliografía donde se proponga un método con los mismos requisitos que los planteados.

Mostraremos la evolución de una red a lo largo de la aplicación del método con un ejemplo simple. Partiremos de la red recurrente completa que se muestra en la figura 2.

Las subsecciones 3.1, 3.2, 3.3 y 3.4 muestran cómo se aplican los diferentes pasos de método para conseguir la red recurrente completa interpretable.

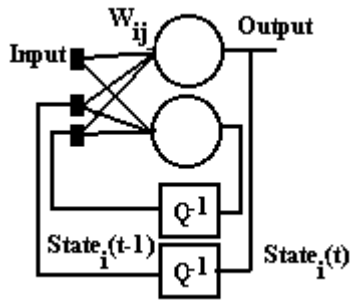


Figura 2. Ejemplo de red recurrente completa con 1 entrada, 1 salida y 1 neurona oculta

3.1. DIFUMINACIÓN DE ENTRADAS

La red transformada debe tener el mismo comportamiento que la inicial. Por tanto, la ecuación 1 debe cumplirse. La filosofía del método es simplificar la red de modo que ésta llegue a ser legible. El primer paso consiste en descomponer las neuronas de entrada de modo que éstas puedan ser analizadas de forma independiente al resto de conexiones. La descomposición de entradas puede resumirse en los siguientes pasos:

1. *Para cada neurona de entrada o de estado j*
    2. *Para cada neurona oculta/de salida i*
        3. *Crear una nueva neurona k en una capa intermedia, con función de activación A(x)=x.*
        4. *Eliminar conexión de neurona j a neurona i.*
        5. *Crear una nueva conexión desde la neurona j a la neurona k, con pero igual a W_{ij}.*
        6. *Crear una nueva conexión desde la neurona k hasta la neurona oculta/de salida i.*

Tras aplicar este procedimiento, se ha generado una capa intermedia de neuronas las cuales tienen una única conexión de entrada y una única conexión de salida. La figura 3 muestra el estado de la red de la figura 2 una vez que la descomposición de entradas ha sido aplicada.

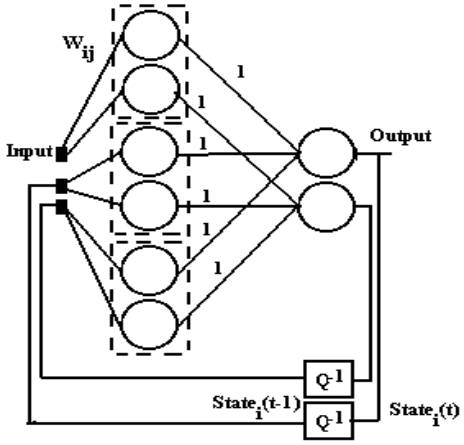


Figura 3. Descomposición de entradas

3.2. DIFUMINACIÓN DE SALIDAS

Este paso realiza dos importantes tareas: Eliminar los pesos de la red e implementar una forma de interpretar las entradas de la red. Estas acciones se llevan a cabo cambiando la función de activación de la capa intermedia, a la que llamaremos L(1), por la función de activación mostrada en la ecuación 3.

$$G_{ij}(x) = \frac{\tanh(W_{ij}x) + 1}{2} \qquad (3)$$

Para mantener la validez de la ecuación 1, una nueva capa, llamada L(2), debe ser introducida en la red. Las neuronas de esta nueva capa deberán tener función de activación $H_{ij}(x)$= atanh(2x-1). El paso 2 puede resumirse en el siguiente procedimiento:

1. *Crear una capa intermedia, L(2), entre las capas L(1) y la capa de neuronas ocultas/de salida*
2. *Para cada neurona de entrada o de estado j*
3. *Para cada neurona k de L(1) conectada a j*
    4. *Generar una nueva neurona p en L(2)*
    5. *Eliminar la conexión entre el nodo k y la neurona i de la capa oculta/de salida*
    6. *Añadir una nueva conexión desde k hacia p, con peso igual a 1.*
    7. *Añadir una nueva conexión desde p hasta i, con peso igual a 1.*
    8. *Asignar función de activación $H_{ij}(x)$ a la neurona p.*

Una vez aplicado este paso, la red neuronal de la figura 2 queda transformada en la siguiente red:

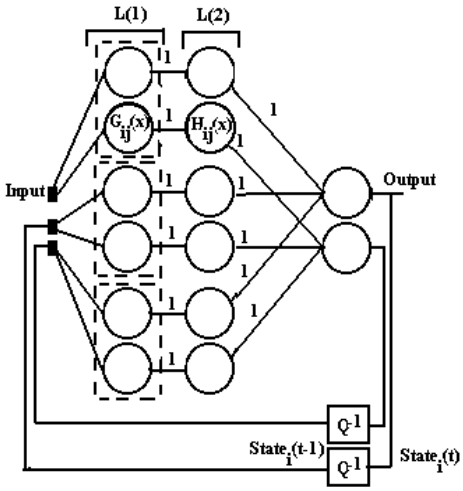


Figura 4. Difuminación de entradas

3.3. DESCOMPOSICIÓN DE SALIDAS

Este proceso es simétrico al paso 1. La descomposición de las salidas nos permite conocer la contribución de cada entrada respecto a cada salida. De nuevo, una nueva capa intermedia -a la que se denominará L(3)- debe ser creada para poder obtener este conocimiento. El paso 3 puede resumirse según el siguiente procedimiento:

1. *Crear una capa intermedia, L(3), entre las capas L(2) y la capa oculta/de salida.*
2. *Para cada neurona o en L(2)*
   3. *Para cada neurona i en la capa oculta/de salida, conectada a p*
   4. *Crear un nuevo nodo o en L(3), con función de activación F(x) (ver ecuación 2)*
   5. *Eliminar conexión entre las neuronas p e i*
   6. *Crear una conexión de p a o, con peso igual a 1.*
   7. *Crear una conexión de o a i, con peso igual a 1.*

Una vez aplicado el paso 3 a la red de la figura 2, la red neuronal recurrente resultante se muestra en la figura 5.

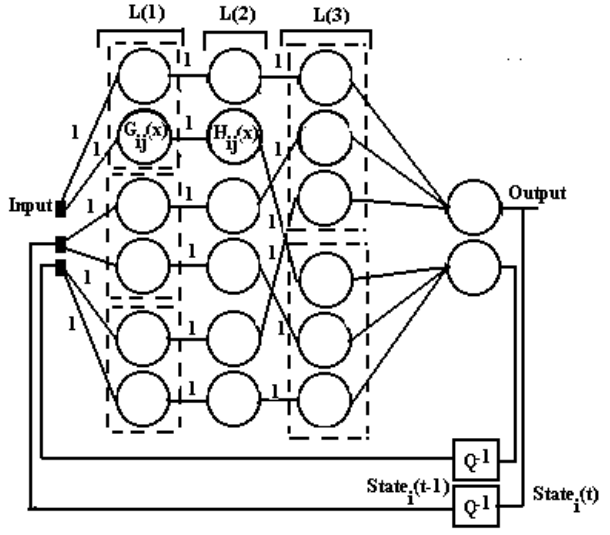


Figura 5. Ejemplo de descomposición de salidas

3.4. AGREGACIÓN DE SALIDAS

Es importante resaltar que, una vez aplicado el paso 3, la red resultante no tiene el mismo comportamiento que la red inicial. Para solucionar este problema, debemos realizar la agregación de salidas. Usando la propiedad de la tangente hiperbólica de una suma (ecuación 4), podemos agrupar las salidas de 3 neuronas a, b y c, -notadas Z_a, Z_b y Z_c- de la capa L(3), conectadas a la neurona k de la capa oculta/de salida, de la forma que se muestra en la ecuación 5.

$$\tanh(x+y)=\frac{\tanh(x)+\tanh(y)}{1+\tanh(x)\tanh(y)} \tag{4}$$

$$Q_k(Z_a,Z_b,Z_c)=Q_k\left(Z_a,\frac{Q_k(Z_b,Z_c)+1}{2}\right)=$$
$$=\frac{(2Z_a-1)+Q_k(Z_b,Z_c)}{1+(2Z_a-1)Q_k(Z_b,Z_c)} \tag{5}$$

Por tanto, la salida real de la neurona k de la capa oculta/de salida, D_k, se calculará de la siguiente forma:

$$D_k=\frac{Q(Z_1,Z_2,...,Z_{NC_k})+1}{2} \tag{6}$$

Donde NC_k es el número de neuronas de L(3) conectadas a la neurona k de la capa oculta/de salida. D_k será usada como función de activación de la neurona k. D_k realiza la tarea de combinar todo el aporte que realiza cada entrada individualmente para conseguir la salida de la neurona k, consiguiendo así la respuesta deseada para dicha neurona.

Una vez que una red neuronal recurrente completa ha sido transformada en otra red recurrente equivalente, pero con mayor grado de interpretabilidad, pasaremos a describir el método para identificar el sistema de reglas difusas que codifica la red.

4 IDENTIFICACIÓN DEL SISTEMA BASADO EN REGLAS DIFUSAS

Recordemos que el significado de cada regla venía dado en términos del aporte de una entrada j para conseguir la salida i. Así, la regla R_{ij}, la cual proporciona la contribución de la entrada j a la salida i, tendrá la siguiente estructura:

R_{ij}: *SI* $Input_j(t)$ *es* A_{ij} *ENTONCES la contribución de* $Input_j(t)$ *para obtener* $Output_i(t)$ *es* B_{ij}.

Sean $Input_j(t)$, $State_j(t)$, $State_j(t)$ y $Output_i(t)$ variables lingüísticas. En la red de la figura 5, $Input_j(t)$ se corresponde con la entrada j a la red, en el tiempo t. $State_i(t)$ se corresponde con la neurona oculta/de salida i en el tiempo t. $State_j(t-1)$ se corresponde con la neurona de estado j en el tiempo t. Por último, $Output_i(t)$ se corresponde con $State_i(t)$ si la neurona i es de salida. El siguiente paso es identificar los valores para A_{ij} y B_{ij}. Si consideramos el aporte que realiza la neurona de entrada j para obtener la salida i, únicamente existe un camino en la red de la figura 5 que une dichas neuronas. Como los nodos de la capa L(3) proporcionan la contribución para la salida i a la cual están conectados, por medio de su función de activación F(x), podemos decir que "*la contribución de* $Input_j(t)$ *para obtener* $Output_i(t)$ *es F*". En

este punto, podemos definir un valor lingüístico difuso F, con función de pertenencia $\mu_F(x)= F(x)$. F es un valor difuso para la variable lingüística $Output_i(t)$. Así, se ha conseguido interpretar el consecuente de una regla a partir de la estructura de la red de la figura 5. Una vez llegados a este punto, la pregunta es: ¿Qué valor de $Input_j(t)$ produce el grado de pertenencia adecuado del consecuente a F?. Si revisamos las ecuaciones 3 y 5, podemos observar que el valor de la función de activación $G_{ij}(x)$ de un nodo p de la capa L(1) es igual al valor de la función de activación F(x) en la capa L(3) para la neurona a la cual p está conectada. En otras palabras, el valor de la función de activación $G_{ij}(x)$ es igual al grado de pertenencia del consecuente de la regla R_{ij} a F. Así, podemos definir un valor lingüístico G_{ij}, cuyo grado de pertenencia sea $\mu_{Gij}(x)= G_{ij}(x)$. Como el grado de pertenencia del consecuente de una regla a F está determinado por el grado de pertenencia de $Input_j(t)$ a G_{ij}, podemos reescribir la regla R_{ij} de la siguiente forma:

R_{ij}: SI $Input_j(t)$ es G_{ij} ENTONCES la contribución de $Input_j(t)$ para obtener $Output_i(t)$ es F

De esta forma, podemos extraer un conjunto de reglas difusas con la estructura anterior, a partir de la red neuronal de la figura 5. Una vez hallado el conjunto de reglas, la salida del sistema debe ser la agregación de la contribución de cada regla para construir cada salida. Para agregar dos reglas, usaremos la T-Conorma mostrada en la ecuación 7 [13].

$$S(x,y)=\frac{x+y}{1+xy} \qquad (7)$$

Debemos notar la gran similitud entre la T-Conorma de la ecuación 7 y la propiedad de la tangente hiperbólica de una suma (ecuación 4). Debido a que la salida de la tangente hiperbólica se encuentra en el intervalo [-1,1], y la de una T-Conorma en el intervalo [0,1], haremos una transformación de los datos de modo que nos aseguremos que la tangente hiperbólica se encuentre en [0,1], para poder usar la ecuación 4 como agregación. La figura 6 muestra un ejemplo de la agregación de dos reglas:

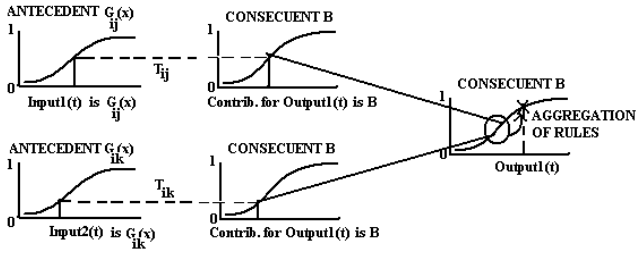


Figura 6. Ejemplo de agregación de dos reglas

Para el ejemplo de agregación de reglas de la figura 6, la contribución de Input1(t) e Input2(t) para obtener Output1(t) son T_{11} y T_{12} respectivamente. Como T_{11} y T_{12} pertenecen al intervalo [0,1], entonces

$$\exists\, O_{11}, O_{12}\in\Re: \tanh(O_{11})= T_{11}, \tanh(O_{12})= T_{12}$$

La agregación de las salidas es la siguiente (ver ecuaciones 4 y 7):

$$\frac{T_{11}+T_{12}}{1+T_{11}T_{12}}=\tanh(O_{11}+O_{12})$$

El valor $\tanh(O_{11}+O_{12})$ es la contribución de las entradas Input1(t) e Input2(t) para obtener Output1(t). Para hallar el valor adecuado de Output1(t), deberemos realizar la desdifuminación de la salida de la siguiente forma:

$$\text{Output1}(t)=2\cdot\text{atanh}(2\tanh(\frac{O_{11}+O_{12}}{2})-1)$$

En términos generales, la agregación de n reglas se llevará a cabo mediante la T-Conorma definida en la ecuación 7 de la siguiente forma:

$$S(T_{R_1},T_{R2},...T_{Rn})=\tanh(O_{R_1},O_{R2},...,O_{Rn})$$

La desdifuminación, para un total de n reglas, se realizará de la siguiente forma:

$$\text{Output}_i(t)=n\cdot\text{ath}(2\text{th}(\frac{O_{R_1}+O_{R_2}+...+O_{R_n}}{n})-1) \qquad (8)$$

El valor de $Output_i(t)$ es igual al dado por la red neuronal inicial (figura 2). Por tanto, hemos conseguido extraer un sistema de reglas difusas de la red neuronal, el cual es equivalente a la misma. No obstante, la interpretabilidad de las reglas puede mejorarse. Si observamos la figura 6, podemos ver que los conjuntos difusos mostrados son del tipo "*mayor que aproximadamente x*", cuando el valor del peso de la red, W_{ij}, es positivo. En cambio, si el valor del peso es negativo, los conjuntos difusos serán del tipo "*menor que aproximadamente x*". La figura 7 muestra un ejemplo de estos tipos de conjunto difuso.

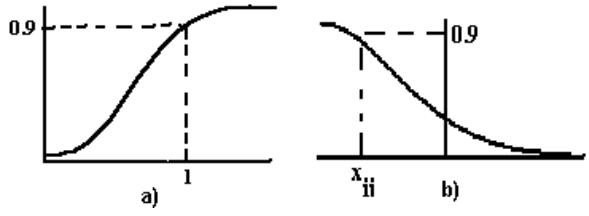


Figura 7. a) Ejemplo de valor difuso "mayor que aproximadamente 1". b) Ejemplo de valor difuso "menor que aproximadamente x_{ii}

Dada esta característica de las funciones de activación de las neuronas de las capas L(1) y L(3), podemos reescribir las reglas obtenidas a partir de la red, de la siguiente forma:

R_{ij}: *SI Input$_j$(t) es mayor/menor que aproximadamente$G^{-1}{}_{ij}$ ENTONCES la contribución de Input$_j$(t) para obtener Output$_i$(t) es mayor que aproximadamente 1.1*. Para terminar con el ejemplo que se ha estado ilustrando en los apartados 3 y 4, supondremos unos pesos W_{ij} para la red de la figura 2 iguales a los mostrados en la tabla 1.

Tabla 1:Pesos para la red ejemplo de la figura 2.

I\j	**1**	**2**
1	0.35	-0.2
2	-2.3	2.02
3	1.43	-3.33

Con este conjunto de pesos, las reglas obtenidas a partir de la red de la figura 2, una vez transformada en la red de la figura 7, son las siguientes:

R_{11}: *SI Input$_1$(t) es mayor que aproximadamente 3.13 ENTONCES la contribución de Input$_1$(t) para obtener Output$_1$(t) es mayor que aproximadamente 1.1*
R_{12}: *SI State$_1$(t-1) es menor que aproximadamente –0.47 ENTONCES la contribución de State$_1$(t-1) para obtener Output$_1$(t) es mayor que aproximadamente 1.1*
R_{13}: *SI State$_2$(t-1) es mayor que aproximadamente 0.77 ENTONCES la contribución de State$_2$(t-1) para obtener Output$_1$(t) es mayor que aproximadamente 1.1*
R_{21}: *SI Input$_1$(t) es menor que aproximadamente –5.5 ENTONCES la contribución de Input$_1$(t) para obtener State$_2$(t) es mayor que aproximadamente 1.1*
R_{22}: *SI State$_1$(t-1) es mayor que aproximadamente 0.54 ENTONCES la contribución de State$_1$(t-1) para obtener State$_2$(t) es mayor que aproximadamente 1.1*
R_{23}: *SI State$_2$(t-1) es menor que aproximadamente –0.33 ENTONCES la contribución de State$_2$(t-1) para obtener State$_2$(t) es mayor que aproximadamente 1.1*

5 CONCLUSIONES

En este trabajo se ha presentado un modelo para aumentar la interpretabilidad de una red recurrente completa, de forma que la extracción de un conjunto de reglas difusas para modelar la dinámica de la red se puede realizar de forma sencilla. Además, el comportamiento del sistema de reglas extraído es idéntico al de la red neuronal. La ventaja principal del método es que permite una interpretación, mediante reglas de tipo Mamdani, de la dinámica de una red neuronal recurrente, sin pérdida de precisión. Además, el sistema de reglas no necesita obtener información adicional del sistema modelado mediante la red neuronal. De hecho, el sistema no trata de modelar el sistema cuyo comportamiento simula la red neuronal, sino el funcionamiento de la red y su estructura interna. Es por ello que el método se convierte en una herramienta potente para interpretar el sistema dinámico modelado mediante redes neuronales recurrentes.

Referencias

[1] A.B. Tickle, R. Andrews, Mostefa Golea, J. Diederich. The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks. *IEEE Transactions on Neural Networks* 9(6), pág. 1057-1068, 1998.

[2] C.L. Giles, S. Lawrence, Ah Chung Tsoi. Noisy Time Series Prediction using a Recurrent Neural Network and grammatical Inference. *Machine Learning* vol. 44(1:2), pág. 161–183, 2001.

[3] C.W. Omlin, C.L. Giles. Extraction of Rules from Discrete-Time Recurrent Neural Networks. *Neural Networks* 9(1), pág. 41-52, 1996.

[4] C. W. Omlin, K.K. Thornber, C.L. Giles. Fuzzy Finite-State Automata can be deterministically encoded into Recurrent Neural Networks. *IEEE Transactions on Fuzzy Systems* vol. 6(1), pág. 76-89, 1998.

[5] C.W.Omlin, C.L. Giles. Stable Encoding of Large Finite-State Automata in Recurrent Neural Networks with Sigmoid Discriminants. *Neural Computation* 8(4), pág. 675, 1996.

[6] Cuéllar M.P., Delgado M., Pegalajar M. y Pérez-Pérez R. Predicción de Series Temporales mediante Redes Neuronales Recurrentes y Algoritmos Bioinspirados. *Proceedings of MAEB'04 Congress, Córdoba (Spain)*, pág. 593-600, 2004.

[7] Danilo P. Mandic, Jonathon A. Chambers. Recurrent Neural Networks for Prediction. *Wiley, John & Sons, Incorporated*, 2001.

[8] I.Visser, M.E.J. Raijmakers, P.C.M. Molenaar. Hidden Markov model interpretations of neural networks. *Behavioral and Brain Sciences* 23(4), pág.494, 2000.

[9] J.L. Castro, C.J. Mantas, J.M. Benítez. Interpretation of Artificial neural Networks by Means of Fuzzy Rules. *IEEE Trans. Neural Networks* vol. 13(1), pág. 101-116, 2002.

[10] J.M. Benítez, J.L. Castro, I. Requena. Are Artificial Neural Networks Black Boxes?. *IEEE Trans. Neural Networks* 8 (5), pág. 1156-1164, 1997.

[11] J.M. Benítez, A. Blanco, M. Delgado, I. Requena. Neural methods for Obtaining Fuzzy Rules. *Mathware & Soft Computing* 3, pág. 17-29, 1996.

[12] L.A. Zadeh. Fuzzy Sets. *Information and Control* vol. 8, pág. 338-353, 1965.

[13] O. Cordón, F. Herrera, A. Peregrín. Applicability of the Fuzzy operators in the Design of Fuzzy Logic Controllers. *Fuzzy Sets and Systems* 86, pág. 15-41, 1997.

IMPLEMENTACIÓN DE UN SISTEMA NEURO-FUZZY SOBRE UN SoPC

G. Bosque **I. del Campo** **J. Echanobe** **J.M Tarela**
Dpto. de Electricidad y Electrónica
UPV-EHU
webbopeg@lg.ehu.es ines@we.lc.ehu.es javi@we.lc.ehu.es tarela@we.lc.ehu.es

Resumen

Este artículo presenta la implementación heterogénea HW/SW sobre un dispositivo SoPC (System on Programable Chip) de un sistema Neuro-Fuzzy de tipo ANFIS, trabajando en tiempo real. Con objeto de obtener una implementación eficiente se proponen una serie de restricciones en el algoritmo de aprendizaje. Se realizan diversos experimentos sobre funciones de test para comprobar la capacidad de aproximación del algoritmo propuesto.

Palabras clave: Lógica Fuzzy, Redes Neuronales, Sistemas Neuro-Fuzzy, Sistema en un Chip Programable (SoPC), Implementación Hardware/Software.

1. Introducción

Desde sus inicios tanto los sistemas fuzzy [6],[17],[18] como las redes neuronales artificiales [14], han sido usadas para aproximar funciones arbitrariamente complejas [9]. Ambas técnicas presentan distintas ventajas e inconvenientes:

- Sistemas fuzzy: Tienen capacidad de manejar lenguaje impreciso pero sin posibilidad de entrenamiento.
- Redes neuronales artificiales: Existen algoritmos de entrenamiento pero es difícil extraer el conocimiento adquirido.

La unión de ambas técnicas da lugar a los sistemas Neuro-Fuzzy, éstos se estructuran en tres categorías: sistemas en los que los componentes interactúan, sistemas en los que una tecnología asiste al diseño de la otra y por último, sistemas en los que las tecnologías se funden para formar una nueva. La mas común, dentro de la segunda categoría, es aquella en la que la *red neuronal* ajusta los parámetros del sistema *fuzzy*. Esta aproximación es eficiente cuando se dispone de un conjunto suficientemente grande de datos de entrada/salida para la definición del sistema.

Una arquitectura neuro-fuzzy, que muestra una gran capacidad de aproximación universal es la denominada ANFIS (Adaptative Neural Fuzzy Inference System) [8],[15]. Esta solución se basa en la equivalencia entre un sistema fuzzy de tipo Sugeno y una red neuronal, lo que permite el entrenamiento de los parámetros del sistema. El sistema presenta una fácil separación de parámetros (antecedentes y consecuentes) de la red para la aplicación de algoritmos de aprendizaje híbridos. En concreto, se utiliza un algoritmo de mínimos cuadrados para el ajuste de consecuentes y un algoritmo de "backpropagation" para los antecedentes.

En términos generales, la implementación de sistemas neuro-fuzzy se puede efectuar tanto en plataformas HW como SW. Las implementaciones SW resultan más flexibles y con las velocidades de procesamiento actuales dan solución a gran parte de las aplicaciones. Sin embargo existen áreas especificas, con altas exigencias de velocidad, donde una solución HW se hace imprescindible. Recientemente han aparecido en el mercado dispositivos conocidos como SoPC (Sistemas en un Chip Programable) que integran en el mismo chip uno o varios procesadores, periféricos y lógica programable. Estas plataformas permiten conjugar la flexibilidad del SW con la velocidad del HW en un único chíp [7],[11].

En este trabajo presentamos la implementación sobre un SoPC de un sistema neuro-fuzzy de tipo ANFIS. Con el fin de hacer más eficiente la implementación, introducimos una serie de restricciones relativas a las funciones de pertenencia del sistema. Con objeto de verificar que el sistema modificado mantiene su capacidad de aproximación, se han realizado una serie de experimentos con funciones no lineales, utilizadas por diversos autores como funciones de test.

El presente trabajo está estructurado de la siguiente manera: la sección 2 describe el sistema ANFIS junto con las restricciones impuestas. La sección 3 describe el algoritmo híbrido de entrenamiento. La sección 4 presenta un estudio comparativo de la capacidad de aproximación del sistema propuesto con el de otros autores. La sección 5 propone una partición HW-SW del sistema y su implementación sobre un SoPC. Finalmente, la sección 6 expone las conclusiones del trabajo.

2. Descripción de un Sistema Neuro-Fuzzy

Las reglas de un sistema fuzzy de tipo Sugeno de orden uno, se muestran a continuación. Por simplicidad tomaremos dos variables de entrada ($x1$,$x2$) y una de salida y:

R1: si x_1 es $A1$ y x_2 es $B1$ entonces

$$f_1 = p_1 \cdot x_1 + q_1 \cdot x_2 + r_1$$

R2: si x_1 es $A1$ y x_2 es $B2$ entonces

$$f_2 = p_2 \cdot x_1 + q_2 \cdot x_2 + r_2$$

.....

Rnxm: si x_1 es An y x_2 es Bm entonces

$$f_{nxm} = p_{nxm} \cdot x_1 + q_{nxm} \cdot x_2 + r_{nxm}$$

La red neuronal equivalente a este sistema fuzzy, denominada ANFIS, se muestra en el ejemplo de la Fig. 1. Su estructura en capas presenta el siguiente significado:

Capa 1: calcula el valor de pertenencia de las entradas.

$$\mu_{A1}(x_1),\ \mu_{A2}(x_1),..,\ \mu_{An}(x_1),\ \mu_{B1}(x_2),\ \mu_{B2}(x_2),..,\ \mu_{Bm}(x_2) \quad (1)$$

Capa 2: calcula, para cada regla, el producto de los valores de pertenencia

$$w_j = \mu_{Ak}(x_1) \cdot \mu_{Bl}(x_2) \quad (2)$$

Capa 3: normaliza los productos

$$\overline{w}_j = \frac{w_j}{w_1 + w_2 + .. + w_{nxm}} \quad (3)$$

Capa 4: aporta la función consecuente

$$f_j = p_j \cdot x_1 + q_j \cdot x_2 + r_j \quad (4)$$

Capa 5: realiza el sumatorio de todas las salidas de los nodos de la capa 4

$$y = \overline{w}_1 \cdot f_1 + .. + \overline{w}_{nxm} \cdot f_{nxm} \quad (5)$$

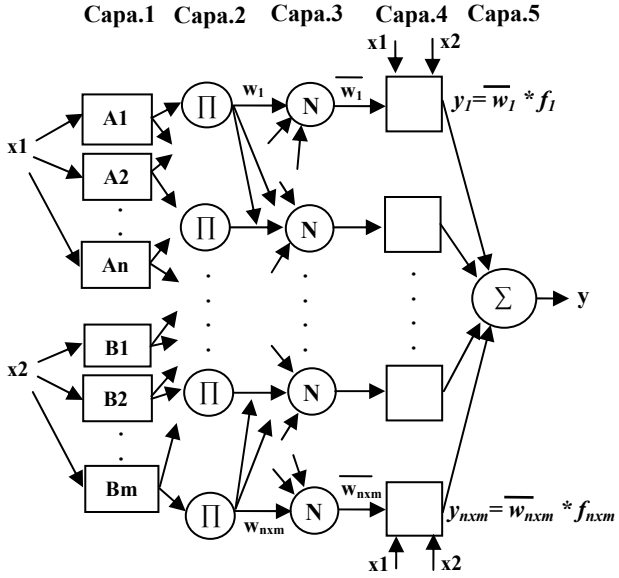


Fig. 1 : Ejemplo de red ANFIS para un sistema de dos entradas y una salida

Con objeto de obtener una implementación más eficiente de la red (1)-(5), introducimos una serie de restricciones:

1. Los antecedentes están solapados dos a dos

2. Dada una entrada, la suma de todas las funciones de pertenencia en los antecedentes, debe ser igual a uno para cada dimensión.

$$\sum \mu_{Ak}(x_1) = 1 \quad \sum \mu_{Bl}(x_2) = 1 \quad (6)$$

3. Las funciones Sugeno serán de orden cero, por lo que los consecuentes se reducen a una constante.

Para verificar las dos primeras restricciones escogemos funciones triangulares dispuestas como se muestra en la fig. 2.

Como consecuencia de las anteriores restricciones tenemos, en primer lugar, que el número de reglas activas para una entrada dada, es igual a 4 (2^n para n entradas).

$$w_i = \mu_{Ak}(x_1) \cdot \mu_{Bl}(x_2)$$
$$w_{i+1} = \mu_{Ak}(x_1) \cdot \mu_{Bl+1}(x_2)$$
$$w_j = \mu_{Ak+1}(x_1) \cdot \mu_{Bl}(x_2)$$
$$w_{j+1} = \mu_{Ak+1}(x_1) \cdot \mu_{Bl+1}(x_2) \quad (7)$$

Los subíndices de las variables w se obtienen según las siguientes expresiones :

i = *(k-1)*numero_antecedentes_de_x_1 + l*

$i+1$ = *(k-1)*numero_antecedentes_de_x_1 + (l+1)*

j = *((k-1)-1)*numero_antecedentes_de_x_1 + l*

$j+1$ = *((k-1)-1)* numero_antecedentes_de_x_1 + (l+1)* (8)

Donde *numero_antecedentes_de_x_1* expresa el máximo número de antecedentes A_k.

En la fig 2 estas reglas son las 4 que intervienen en la zona sombreada, donde *Ak* es *A2*, *Ak+1* es *A3*, *Bl* es *B3* y *Bl+1* es *B4* . Por otra parte el denominador en (3) se hace uno según (6), por lo que se elimina la división en la capa 3. La salida de la capa 5 viene dada entonces por:

$$y = \sum_i w_i \cdot r_i + \sum_j w_j \cdot r_j \quad (9)$$

Donde j se refiere a las cuatro reglas activas.

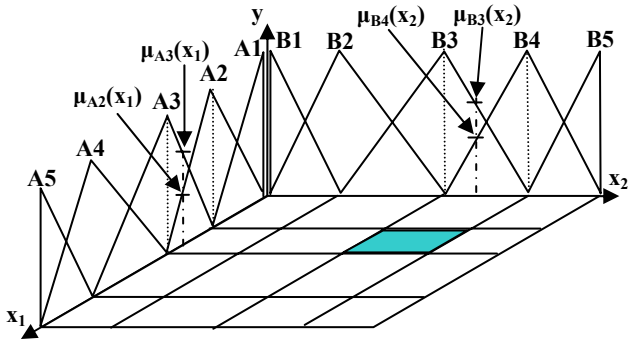


Fig. 2 : Funciones de pertenencia – reglas activas

Al algoritmo propuesto con las restricciones descritas lo denominaremos AHRT (Algoritmo Híbrido de Reglas Triangulares).

Como conclusión diremos que obtenemos una simplificación del sistema, lo que redunda en una implementación más simple sobre el SoPC con el consiguiente aumento de la velocidad de ejecución y de un menor requerimiento de recursos en el SW y/o HW.

3. Algoritmo Híbrido de Entrenamiento

Según el esquema de red AHRT, podemos utilizar algoritmos de aprendizaje para modificar los parámetros del sistema neuro-fuzzy. En concreto, el algoritmo híbrido planteado, calcula los parámetros consecuentes por medio del método de mínimos cuadrados y los parámetros antecedentes por medio del método del gradiente descendente (backpropagation) [8],[10],[16].

Una red AHRT se expresa matricialmente de la siguiente forma :

$$\boldsymbol{y} = \boldsymbol{A} \cdot \boldsymbol{C} \qquad (10)$$

La matriz salida $\boldsymbol{y}$ es igual al producto de dos matrices, la matriz $\boldsymbol{A}$ representa la aportación de los parámetros antecedentes y la matriz $\boldsymbol{C}$ representa la aportación de los parámetros consecuentes. Los parámetros antecedentes están implícitos en los valores w_j (2).

El algoritmo híbrido de entrenamiento realiza la optimización de los distintos parámetros que conforman la red AHRT [8]. Para realizar la optimización, se recurre a una función de error siendo ésta la función a optimizar. El método de mínimos cuadrados minimiza la suma de los cuadrados de los errores y es eficaz cuando se dispone de una función combinación lineal de funciones como la obtenida en la red AHRT. La optimización basada en la derivada, trabaja con el gradiente de una función de error, buscando la dirección de un mínimo (local). [8],[16]. El uso de algoritmos de gradiente descendente y mínimos cuadrados, constituye el denominado entrenamiento híbrido.

3.1 Estimador de Mínimos Cuadrados

Para el entrenamiento de la red partimos de un conjunto de valores $(x_i : y_i)$, siendo y_i el valor objetivo. La aproximación obtenida será mejor cuanto mayor sea el número de valores de entrenamiento. Deberá encontrarse la matriz $\hat{\boldsymbol{C}}$, tal que se minimice el error obtenido entre el valor objetivo y_i y el valor calculado $y'_i = \boldsymbol{A} \cdot \hat{\boldsymbol{C}}$:

$$\boldsymbol{e_i} = \boldsymbol{y_i} - \boldsymbol{A} \cdot \hat{\boldsymbol{C}} \qquad (11)$$

Más concretamente, la función a minimizar es el error cuadrático medio que para el conjunto de valores de entrenamiento se alcanza con un vector $\hat{\boldsymbol{C}}$ que cumpla la ecuación (5):

$$\hat{\boldsymbol{C}} = (\boldsymbol{A}^{\boldsymbol{T}} \cdot \boldsymbol{A})^{-1} \cdot \boldsymbol{A}^{\boldsymbol{T}} \cdot \boldsymbol{y} \qquad (12)$$

3.2 Método del Gradiente Descendente

El método del Gradiente Descendente minimiza una función de error *E(a)* definida sobre un espacio de entradas n_dimensional $\boldsymbol{a} = [a_1, a_2, ..., a_n]^T$, buscando un (posible) punto que minimice el error. Este vector $\boldsymbol{a}$, en nuestro caso, se corresponde con los parámetros de las funciones de pertenencia triangulares. En general, la función *E(a)*, puede ser no lineal con respecto al parámetro ajustable $\boldsymbol{a}$. Debido a la complejidad del error, se recurre a un *algoritmo iterativo* para explorar el espacio de entradas eficientemente. En los métodos de *iteración descendente*, el punto siguiente $\boldsymbol{a}_{k+1}$ se determina por un paso del punto actual $\boldsymbol{a}_k$ en la dirección de un vector $\boldsymbol{g}$:

$$\boldsymbol{a}_{k+1} = \boldsymbol{a}_k + \eta \cdot \boldsymbol{g} \qquad \text{(k=1, 2, 3, ...)} \quad (13)$$

Donde η regula el tamaño del paso.

El vector $\boldsymbol{g}$ es el gradiente de la función diferenciable $E : R^n \rightarrow R$ en $\boldsymbol{a}$, es decir, es el vector de la primera derivada de E:

$$\boldsymbol{g}(\boldsymbol{a}) = \nabla \boldsymbol{E}(\boldsymbol{a}) = \left[\frac{\partial E}{\partial a_1}, \frac{\partial E}{\partial a_2}, ..., \frac{\partial E}{\partial a_n} \right]^T \qquad (14)$$

Idealmente se desea encontrar un valor $\boldsymbol{a}_{k+1}$ tal que satisfaga que el gradiente sea 0. Esta es una condición *necesaria* porque el gradiente es cero en algún *punto estacionario*, llamado máximo, mínimo o punto de paso (silla). En la práctica, sin embargo, es difícil resolver esta ecuación analítica. Para minimizar la función objetivo, los procedimientos descendentes son repetidos hasta conseguir cumplir los criterios de paro [8].

4. Evaluación con Funciones de Test

Por medio de estructuras similares a la de la fig.1 determinaremos la capacidad de aproximación del sistema AHRT propuesto, utilizando funciones de test. En concreto utilizamos algunas de las empleadas en el trabajo de Lee-Ouyang [9]. En dicho trabajo se evalúa un sistema neuro-fuzzy similar al AHRT, obteniéndose resultados superiores a otros métodos existentes [10],[16]. Las funciones de test son las siguientes :

a. $y = \sin(x_1 \pi) \cdot \sin(x_2 \pi) \qquad x_1 \in [-1,1]\ x_2 \in [0,1]$

b. $y = x_2 \sin(x_1) + x_1 \cos(x_2) \qquad 0 \le x_1, x_2 \le \pi$

Para cada función de test evaluaremos la respuesta del AHRT en función del número de reglas y el número de iteraciones de aprendizaje. La bondad de la aproximación vendrá dada por la función de error SSE (Square Stimator Error):

$$SSE = \frac{1}{2 \cdot p} \sum_{i=1}^{p} e_i^2 \qquad (15)$$

Siendo p el número de puntos de entrenamiento. Los resultados se presentan en tablas comparativas donde además se incluye el error generalizado SSE_G que se obtiene para puntos distintos a los del aprendizaje. Para cada función de test se indican también los errores SSE y SSE_G obtenidos por el método de Lee-Ouyang [9].

3.1 Experimento Nº 1 : $y = \sin(x_1 \cdot \pi) \cdot \sin(x_2 \cdot \pi)$

La función objetivo se representa en la figura 3 y en la Tabla I se reflejan ensayos con diferente número de reglas. En primer lugar, se observa que con 9 reglas (3x3) el error SSE obtenido por nuestro algoritmo es 0.0528 para 11 iteraciones. Este valor está lejos del valor 0.0023 obtenido por Lee-Ouyang para el mismo número de reglas e iteraciones. Sin embargo, si aumentamos a 16 reglas (4x4), se obtiene un valor mejor, 0.0014, para igual número de iteraciones. Más aún, con solo 3 iteraciones, se alcanza el mismo error que el comparado. En cuanto a la capacidad de generalización tenemos que se obtienen resultados comparables utilizando 16 reglas y de nuevo resultados peores si las reglas se reducen a 9. La Fig. 4 refleja la superficie aproximada obtenida en la iteración número 11 con 4x4 reglas.

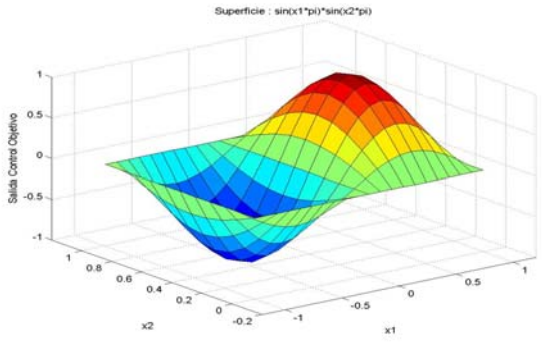


Fig. 3: Función objetivo Experimento Nº1

Tabla I : Experimento Nº 1

	Nº de Reglas	Iteraciones	SSE	SSE_G
Lee-Ouyang	10	11	0.0023	0.0016
ENSAYO Nº 1 - 3x3 Reglas				
AHRT	9	11	0.0528	0.1258
ENSAYO Nº 2 – 4x4 Reglas				
AHRT	16	11	0.0014	0.00184
AHRT	16	3	0.002	0.002

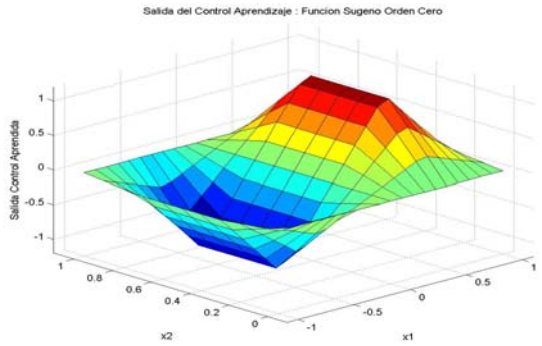


Fig. 4 : Función aproximada 4x4 - reglas Experimento Nº1 después de 11 iteraciones

3.2 Experimento Nº 2 : $y = x_2 \sin(x_1) + x_1 \cos(x_2)$

En este experimento la función objetivo es la de la figura 5. En la Tabla II se reflejan los ensayos con diferente número de reglas. En el primer ensayo, se observa que con 16 reglas (4x4) el error SSE obtenido por nuestro algoritmo es 0.0034 para 6 iteraciones. Este valor es comparable al valor 0.0031 obtenido por Lee-Ouyang para el mismo número de reglas e iteraciones. Si en este mismo ensayo, realizamos 16 iteraciones, el error obtenido es igual al comparado. Aumentando el número de reglas a 25 (5x5), se obtiene un valor mejor, 0.00125, a igual número de iteraciones. Observamos que, con solo 2 iteraciones, el error obtenido, 0.0014, se reduce a menos de la mitad del error comparado. En cuanto a la capacidad de generalización podemos decir que se obtienen resultados comparables utilizando 16 reglas y resultados mejores ampliando a 25 el número de reglas. La Fig. 6 refleja la superficie aproximada en la sexta iteración con 5x5 reglas.

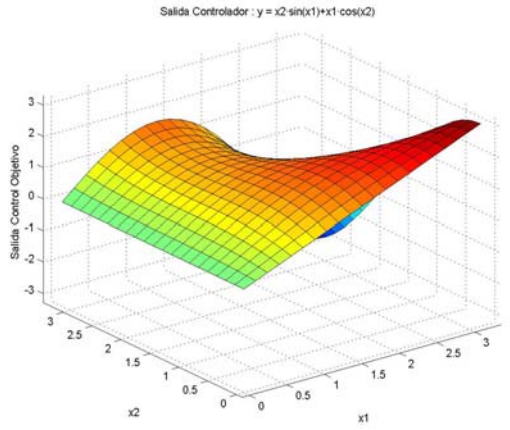


Fig. 5 : Función objetivo Experimento Nº2

Tabla II : Experimento Nº 2

	Nº de Reglas	Iteraciones	SSE	SSE_G
Lee-Ouyang	16	6	0.0031	0.0023
ENSAYO Nº 1 – 4x4 Reglas				
AHRT	16	6	0.0034	0.002836
AHRT	16	16	0.0031	0.00239
ENSAYO Nº 2 – 5x5 Reglas				
AHRT	25	6	0.00125	0.001234
AHRT	25	2	0.0014	0.00138

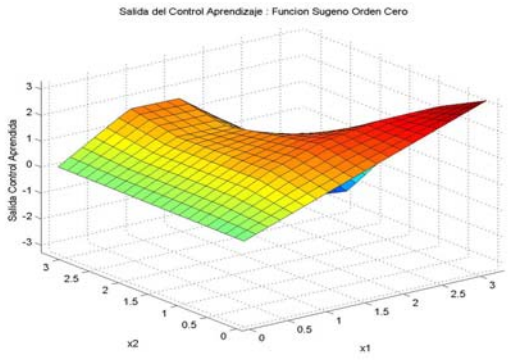


Fig. 6 : Función aproximada 5x5 reglas Experimento Nº2

Como conclusión de los dos experimentos realizados, podemos decir que la capacidad de aproximación de nuestro algoritmo es comparable a la obtenida por Lee-Ouyang. Sin embargo, se constata que es necesario utilizar un número de reglas ligeramente mayor para obtener valores iguales o mejores. La explicación a este hecho radica en la menor flexibilidad de las funciones de pertenencia triangulares frente a las gausianas utilizadas en el algoritmo de Lee-Ouyang.

5. Implementación Hw/Sw en un SoPC

La implementación del algoritmo AHRT sobre un SoPC requiere realizar una partición en bloques HW/SW. Se opta por implementar la red neuronal (1)-(5) sobre el HW y el algoritmo de aprendizaje (10)-(13) sobre el SW. Esta decisión está basada en la diferenciación de los dos tipos de procesos que intervienen en la red neuro-fuzzy:

1. Proceso de aprendizaje. Realiza un cálculo masivo durante un serie de iteraciones, ajustando los parámetros de la red. Otra razón que lleva a realizar el proceso de aprendizaje en SW es la precisión requerida por este proceso (tasas de aprendizaje y derivadas), puede necesitar un ancho de palabra superior al utilizado en el proceso de red.
2. Proceso de cálculo de la red (fase feedforward). Requiere de rápida computación y ejecución en tiempo real. El HW permite el paralelismo propio de las redes neuronales.

El SoPC seleccionado dispone de un microprocesador ARM [2],[12] y un bloque de lógica programable (PLD). Sobre el ARM se ha implementado el programa que realiza el proceso de aprendizaje, este programa se ha escrito en C [4][13]. La red se ha implementado sobre el PLD utilizando VHDL[3]. La comunicación entre el ARM y el PLD se realiza por medio del bus AMBA-AHB[2]. La fig 9 muestra la disposición de los bloques.

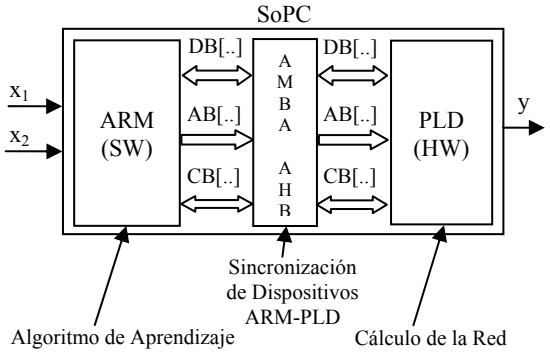


Fig. 9 : Esquema de bloques del SoPC y partición del sistema

La facilidad de cálculo introducida por el algoritmo AHRT se muestra en la fig.10 donde puede verse la construcción de las funciones de pertenencia $\mu_{Ai}(x)$ y de $\mu_{Ai-1}(x)$ de la variable x. El PLD solo necesita la pendiente m_i y el parámetro a_i (primer vértice del triángulo antecedente). Nótese que $\mu_{Ai-1}(x)=1-\mu_{Ai}(x)$. En una implementación digital esta operación consiste en el complemento lógico.

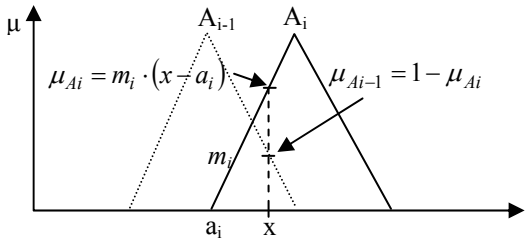


Fig. 10 : Valor calculado de la función de pertenencia

El bloque PLD se estructura según los tres bloques siguientes :

1. AHB_slave: Es la máquina de estados que diferencia las fases de Direccionamiento, Datos, Ráfagas de Datos (Burst) y Error, entrega al bloque REGISTROS direcciones y datos y recibe de él los datos calculados por el bloque ALU.
2. REGISTROS: Realiza el encaminamiento de los datos entre el bloque AHB_slave y ALU y entrega señales de habilitación de las fases de cálculo.
3. ALU: Realiza los cálculos de la red ANFIS. El bloque REGISTROS va entregando secuencialmente los datos necesarios para el cálculo de la red. Simplificando su representación, estos son : x_1, x_2, a_1, a_2, m_1, m_2, r_1, r_2, r_3 y r_4. El resultado de las operaciones es entregado al bloque REGISTROS.

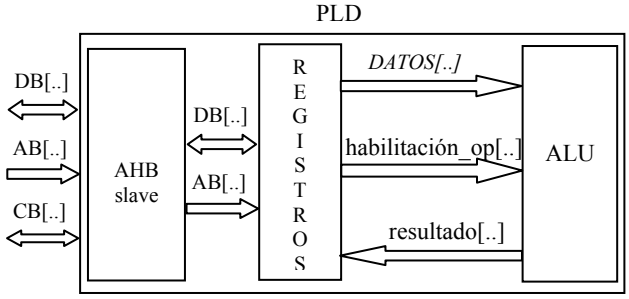


Fig. 11 : Bloques Hardware implementados en el PLD

Para realizar la implementación escogemos la familia Excalibur_ARM (APEX 20KE) de Altera [1] [5]. El dispositivo sobre el que se va a realizar la implementación va a depender de la precisión requerida por el ensayo. Podemos trabajar con representaciones numéricas de 8-16-32 bits.

a. Representación en 8 bits: el dispositivo es el EPXA1F484C1 [1], este incorpora un procesador ARM922T de 32 bits de arquitectura Harvard, pudiendo operar hasta 200Mhz. Incorpora un PLD de 4160 elementos lógicos, el bloque PLD opera a frecuencia mitad (100MHz). Dispone de memoria de simple puerta SRAM de 32KB y de 16KB de SRAM de doble puerta. La tarjeta de desarrollo sobre la que se realiza la implementación es la EPXA1.
La implementación ocupa 4000 elementos lógicos.

b. Representación en 16-32 bits: el dispositivo es el EPXA4F672C1 [1]. Idéntica arquitectura de procesador que el anterior dispositivo. Incorpora un PLD de 16750 elementos lógicos. Dispone de memoria de simple puerta SRAM de 128KB y de 64KB de SRAM de doble puerta..
La implementación ocupa aproximadamente 13900 elementos lógicos en 32 bits y 7700 en 16 bits.

Los ensayos se han realizado con una frecuencia de reloj de PLD de 50MHz y una representación de datos en 16 bits, necesitándose cinco accesos de escritura y uno de lectura al bloque ALU. El bloque ALU está estructurado en tres unidades de cálculo que realizan diferentes fases de operación de la red ANFIS. Cada unidad de cálculo, si es necesario, puede introducir ciclos de espera, interviniendo sobre la señal HREADY del ARM. Cada unidad introduce los ciclos de espera adecuados a la frecuencia de trabajo.

a. La primera unidad invierte 3ns para el cálculo, lo que no requiere de ciclos de espera.
b. La segunda unidad invierte 26ns para el cálculo, requiriendo un ciclo de espera.
c. La tercera unidad invierte 28.5ns para el cálculo, requiriendo un ciclo de espera.

Conclusiones

En este trabajo se propone la implementación heterogénea HW/SW de un sistema neuro-fuzzy sobre un SoPC trabajando en tiempo real. El sistema elegido es una red de tipo ANFIS sobre la que se han impuesto una serie de restricciones para hacer más eficiente dicha implementación. La eficacia del sistema se ha verificado analizando la capacidad de aproximación de funciones no lineales y comparando los resultados obtenidos con métodos propuestos por otros autores.

La partición HW/SW se ha hecho de forma que la red se implementa en la parte HW y el algoritmo de entrenamiento híbrido en la parte SW. La red neuronal (fase feedforward) requiere de rápida computación y ejecución en tiempo real. Por este motivo se ha hecho un diseño HW de esta parte del sistema. La arquitectura desarrollada refleja el paralelismo propio de las redes neuronales. Por otra parte, el algoritmo de aprendizaje, implementado en SW, involucra un cálculo masivo durante una serie de iteraciones, ajustando los parámetros de la red. Este ajuste requiere alta precisión en las tasas de aprendizaje y derivadas, por lo que necesita un ancho de palabra superior al utilizado en el bloque HW.

Agradecimientos

Este trabajo ha sido parcialmente financiado por fondos CICYT y FEDER (TIC2002-02458), y la Universidad del País Vasco (UPV224.310-E-14029/2001).

Bibliografía y Referencias

[1] Altera. Excalibur Device Overview. May. 2002, Ver.2.0.
[2] ARM. AMBA Specification. Rev. 2.0.
[3] Brown S. Vranesic Z. Fundamentals of Digital Logic With VHDL Designs. Ed. McGraw-Hill. 2000.
[4] Ceballos F.J. Curso de Programación C/C++. Ed. Rama. 1995
[5] Excalibur Devices. Hardware Reference Manual. Nov. 2002, Ver. 3.1.
[6] George J.K, Folger T.A. Fuzzy Sets, Uncertainty, And Information. ED. Prentice-Hall International. 1998.
[7] Hardware Implementation of Intelligent System. Ed. Teodorescu H.N, Jan L.C, Kandel A. 2001.
[8] Jang J.S.R , Sun C.T., Mizutani E. Neuro-Fuzzy and Soft Computing. Ed. Prentice Hall 1997.
[9] Lee S.J, Ouyang Ch.S. A Neuro-Fuzzy System Modeling With Self-Constructing Rule Generation and Hybrid SVD-Based Learning. IEEE Transactions on Fuzzy Systems Vol.11, Nº 3. pp. 341-353. June 2003.
[10] Lin Y.,Cunningham III G.A., Coggeshal S.V. Using Fuzzy Partitions to Create Fuzzy Systems from Input-Output Data and Set de Initial Weights in a Fuzzy Neural Network. IEEE Trans. Fuzzy Syst. Vol.6, pp. 12-32, Feb. 1998.
[11] Reyneri L.M. Implementation Issues of Neuro-Fuzzy Hardware : Going Toward HW/SW Codesign. IEEE Transactions On Neural Networks, Vol. 14, Nº 1, pp 176-194. January 2003.
[12] Seal D. ARM : Architecture Reference Manual. Ed. Addison-Wesley. 2001.
[13] Schildt H. ANSI C : C a su Alcance. Ed. Osborne McGraw-Hill
[14] Tsoukalas L.H. Uhrig R.E. Fuzzy and Neural Approaches in Engineering. Ed. Jhon Wisley & Sons, Inc. 1997.
[15] Wesley J. Fuzzy and Neural Approaches in Engineering. Ed. John Wiley & Sons, 1997. New York.
[16] Wong C.C., Chen C.C. A Hybrid Clustering an Gradient Descent Approach for Fuzzy Modeling. IEEE Trans. Syst. Man. Cybern. B, Vol.29, pp. 686-693, Dec. 1999
[17] Zadeh L.A. Fuzzy Sets. Information and Control, 8, pp. 338-353. 1965.
[18] Zimmerman H.J. Fuzzy Set Theory and its Applications. Second edition. Ed. Kluwer Academic Publishers. 1993.

UNA ESTRATEGIA DIFUSA PARA LA APLICACIÓN DE OPERADORES EN UN ALGORITMO EVOLUTIVO

Antonio J. Rivera
Dpto. Informática
Universidad de Jaén
arivera@ujaen.es

Julio Ortega
Dpto. Arquitectura y Tecnología de Computadores
Universidad de Granada
julio@atc.ugr.es

Ignacio Rojas
Dpto. Arquitectura y Tecnología de Computadores
Universidad de Granada
ignacio@atc.ugr.es

M. José del Jesus
Dpto. Informática
Universidad de Jaén
mjjesus@ujaen.es

Resumen

Este trabajo presenta un nuevo algoritmo bioinspirado para el diseño de redes neuronales, concretamente Redes de Funciones de Base Radial (RBFNs). El método de aprendizaje es un algoritmo coevolutivo en el que cada individuo representa una neurona y la población al completo la RBFN. Destaca en el mismo la incorporación de Sistema Basado en Reglas Difusas (SBRD) que representa conocimiento experto sobre los operadores a aplicar a una neurona en base a sus características. El algoritmo se ha aplicado a la aproximación de funciones obteniendo buenos resultados.

Palabras Clave: RBFNs, Coevolución, SBRDs, Aproximación de funciones.

1. INTRODUCCION

Una red neuronal de funciones de base radial (RBFNs) [2] implementa una función $f(x)$ de R^n en R:

$$f(x) = \sum_{i=1}^{m} w_i \phi_i(x) \qquad (1)$$

donde $x \in R^n$, y existen m funciones de base radial (RBFs) Una Función de Base Radial (RBF), ϕ_i, se caracteriza por un centro $c_i \in R^n$, con respecto al cual tiene una salida simétrica y un factor de escalado de radio $d_i \in R$ lo que suele expresarse como $\phi_i(x) = \phi(||x - c_i|| / d_i)$, donde $\| \|$ suele ser la norma euclídea. De entre las posibles elecciones para ϕ_i [17], en este trabajo se ha elegido la función gaussiana $\phi(r)=exp(-r^2)$. Este tipo de funciones empezó a utilizarse como neuronas de una red neuronal durante los años 80 [11][13].

El principal objetivo de cualquier algoritmo de diseño de estas redes es determinar la posición de los centros c_i, los radios d_i y los pesos w_i para alcanzar los mejores resultados. Para conseguir este objetivo en este trabajo describimos un método de aprendizaje bioinspirado que optimiza tres objetivos:

- La aportación de una neurona (RBF) a la salida de la red neuronal completa.
- El error cometido por cada RBF en su zona de actuación.
- El solapamiento entre las RBFs que componen la red.

El método evolutivo utiliza una estrategia difusa que representa el conocimiento experto sobre la decisión de eliminar o ajustar una neurona en el proceso de aprendizaje. En la siguiente subsección describimos cuestiones generales sobre el uso de la computación evolutiva en el diseño de redes neuronales.

1.1. DISEÑO Y OPTIMIZACIÓN DE RBFNS

Dentro de la computación evolutiva, el diseño de RBFNs se puede clasificar dentro de las siguientes tendencias [22]:

1.1.1. Evolución clásica de redes neuronales

En los algoritmos de optimización de este grupo un individuo de la población representa una red neuronal que compite según una función de adaptación que se corresponde con la eficiencia de cada red. Dentro de este procedimiento general se pueden clasificar los trabajos de [16][7].

1.1.2. Población de individuos que coevolucionan

En los algoritmos de optimización de este grupo cada elemento de la población es una RBF de la red, en vez de una red completa, evolucionando las neuronas en base a su función de adaptación [20]. De esta manera las neuronas de una población trabajan juntas, cooperando para determinar una red neuronal que sea más eficiente, aunque deben competir por sobrevivir. En este tipo de algoritmos es necesario definir dos aspectos:

1. La *asignación de crédito* [19], que conlleva determinar una medida de la eficiencia para cada elemento de la población con respecto a la eficiencia de toda la población.

2. El mantenimiento de una población de neuronas que desarrollen diferentes partes del espacio de soluciones. Una solución a este problema es el uso de una técnica de nichos (*niching*) [1][18].

En la sección 2 de este trabajo se describe nuestro algoritmo enmarcado bajo este segundo enfoque. En la sección 3 se presentan los resultados que muestran la eficacia del procedimiento y en la sección 4 se describen las conclusiones.

2. DESCRIPCIÓN DEL ALGORITMO

El algoritmo propuesto presenta un nuevo método bioinspirado en el que una población formada por un conjunto de RBFs evoluciona para obtener una RBFN que constituye una solución al problema planteado. Dentro del modelo bioinspirado, los individuos de la población irán evolucionando y los peor adaptados se modificarán para intentar sobrevivir o se eliminarán.

Por su funcionamiento, el algoritmo propuesto implementa conceptos de cooperación y competición entre los elementos de una población. Estos conceptos, unidos a las aportaciones que posteriormente se describen diferencian nuestro algoritmo del trabajo [20], pionero en el desarrollo coevolutivo de RBFs. En nuestro desarrollo las RBFs cooperan para construir una RBFN que consiga unos resultados eficientes y además compiten por su subsistencia, desapareciendo si no consiguen mantener un buen valor en los factores que las caracterizan: su aportación a la red, solapamiento entre neuronas y el error dentro de su radio. Para determinar las peores RBFs, éstas se ordenarán utilizando conceptos de las técnicas multiobjetivo [3]. En [20] las RBFs evolucionan optimizando un único objetivo en el que el factor más importante es el peso de la neurona.

Otra de las principales aportaciones de este algoritmo es la estrategia de aplicación de los operadores a las RBFs ya que está regida por un sistema difuso, concretamente un Sistema Basado en Reglas Difusas [9]. Este sistema difuso tendrá como entradas los factores que caracterizan una RBF (su aportación a la red, error dentro de su radio y el solapamiento entre neuronas) y su salida determinará la probabilidad de aplicación de cada operador sobre la RBF seleccionada. El uso de este sistema confiere al algoritmo una mayor adaptabilidad [8] en la evolución general que éste sigue.

Con estos pasos se pretende que el algoritmo alcance una solución que proporcione por si sola un buen error de aproximación. Para mejorar esta solución y conseguir una predicción de muy buena calidad se aplica un algoritmo de minimización del error clásico como el de Levenberg-Marquardt [10]. El algoritmo es el resultado de la tesis doctoral [15] y en este trabajo se presentan nuevos datos obtenidos en la aproximación de funciones.

2.1. DESCRIPCIÓN DETALLADA

Los principales pasos del algoritmo se muestran en la Figura 1 y se detallan a continuación:

```
1. Inicializar RBFN
2. Entrenar RBFN
3. Evaluar RBFs
4. Ordenar y seleccionar peores RBFs
5. Aplicar operadores a peores RBFs
6. Añadir las RBFs eliminadas
7. Entrenar RBFN
8. Si la condición de fin no se
   verifica ir al paso 3 si no
   terminar
9. Aplicar Levenberg-Marquardt
```

Figura 1: Algoritmo empleado para el diseño de RBFNs

```
1. Inicializar RBFN
```

En el primer paso, el algoritmo construye una RBFN. Esto se realiza inicializando una población de RBFs, donde los centros se asignan aleatoriamente, respetando una distancia mínima con respecto a los centros de las funciones base ya establecidas. El cálculo de los radios, se determina en torno a la media de la distancia de una RBF a su RBF más cercana.

```
2. y  7. Entrenar RBFN
```

Para determinar los pesos de la población de RBFs se usa el algoritmo LMS [21]. Este algoritmo se aplica cíclicamente hasta que no mejora el valor del error durante tras dos iteraciones consecutivas. El parámetro α se fija al valor típico 0.3.

```
3. Evaluar RBFs
```

Para evaluar cada RBF se consideran tres parámetros que caracterizan el trabajo de una función base en la red:

- a_i: aportación de una función base a la salida de la red.
- e_i: error que se produce dentro del radio de una RBF.
- s_i: solapamiento de una función base con sus vecinas.

Para determinar la aportación de la RBF ϕ_i, $a_{i,,}$ se considerará su peso w_i y el número de puntos dentro de su radio pe_i, de este modo se define:

$$rac_i = |w_i| \cdot (pe_i / medpe) \quad (2)$$

donde *medpe* es la media de los valores pe_i. Con esto el valor de a_i se calcula como:

$$a_i = rac_i / da \quad (3)$$

donde *da* es un factor que se utiliza para desplazar convenientemente los valores a_i al rango de entrada del SBRD que más tarde se utilizará para decidir la aplicación de los operadores. Esta operación es necesario realizarla

ya que el valor de los valores rac_i depende de los valores del ejemplo que se estén considerando.

Para ofrecer una medida del error dentro del radio de la RBF ϕ_i, e_i, se tendrá el cuenta la raiz cuadrada del error cuadrático medio normalizada $nrmse_i$, dentro de su radio y la desviación del error, también dentro de su radio $ndesv_i$, de esta manera se define:

$$rerr_i = nrmse_i \cdot (ndesv_i / medndesv) \quad (4)$$

donde *mnedesv* es la media de los valores $ndesv_i$. Por los motivos anteriormente citados e_i se calcula como:

$$e_i = rerr_i / de \quad (5)$$

El solapamiento de una RBF ϕ_i con el resto de RBFs se cuantifica por una función s_i, que se define como:

$$si = \sum_{j}^{m} s_{ij} \quad (6)$$

donde m es el número de RBFs y s_{ij}, mide el solapamiento entre las RBFs i y j, (ϕ_i y ϕ_j):

$$s_{ij} = \begin{cases} 1 - (\| \phi_i, \phi_j \| / d_i) & si \| \phi_i, \phi_j \| < d_i \\ 0 & en\ otro\ caso \end{cases} \quad (7)$$

En este caso y por la forma de calcular los valores s_i no hace falta desplazarlos ya que se definen en el rango de entrada del sistema de lógica difusa.

4. Ordenar y seleccionar peores RBFs

Las RBFs se van a ordenar, mediante una técnica multiobjetivo [3] (que considerará para cada RBF ϕ_i los valores en los distintos objetivos, a_i, e_i, s_i), siendo en general peores RBFs aquellas que tienen sus tres objetivos dominados por los de otras RBFs. Si no existe una RBF peor en los tres objetivos se buscarán las RBFs que sean peores en dos objetivos, si existe más de una se elegirá aleatoriamente la peor RBF entre ellas. Si tampoco existen RBFs peores en dos objetivos, se busca la peor RBF en cada objetivo y de entre estas cogemos al azar la peor RBF. Siguiendo las bases de funcionamiento del algoritmo y a partir de la heurística se seleccionan la mitad de las RBFs peores, para aplicar los operadores.

5. Aplicar operadores a peores RBFs

Los dos operadores propuestos trabajo son:

1. Eliminación de una RBF (OP1): Se selecciona una RBF y se elimina.
2. Ajuste de una RBF (OP2): El centro c_i y el radio r_i de una RBF ϕ_i se ajustan en función de un pequeño porcentaje aleatorio del radio d_i y según un algoritmo LMS para el ajuste centros y radios.

Al igual que en [22] no se cree conveniente utilizar un operador de cruce.

Para elegir el operador a aplicar a una determinada RBF ϕ_i, se utiliza un conjunto de reglas difusas tipo Mandani [9], donde se tienen en cuenta para cada ϕ_i los objetivos definidos anteriormente: a_i, e_i, s_i. Para esto se definen tres variables lingüísticas, una para cada objetivo (*va* para *a*, *ve* para *e*, y *vs* para *s*) con 3 etiquetas lingüísticas para cada variable. Las funciones de pertenencia para las variables lingüísticas se representan en la Fig. 2 y son fruto del conocimiento experto en diseño de RBFNs. Las variables consecuentes de las reglas son *vop1* y *vop2*, tienen la función de pertenencia expresada en la parte inferior de la Figura 2 y representan la probabilidad de aplicar un operador u otro.

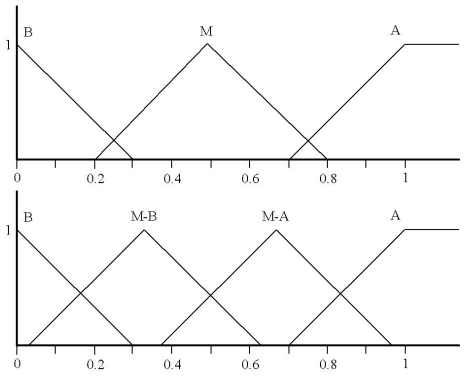


Figura 2: Funciones de pertenencia, superior para *vp*, *vs* y *ve*, inferior para *vop1* y *vop2*.

El conjunto de reglas difusas, que refleja el conocimiento experto en optimización de RBFNs, se muestra en la Tabla 1. Estas reglas contienen información experta del tipo: si disminuye la aportación de una RBF, aumentar la probabilidad de borrarla; si aumenta el solapamiento de una función base, aumentar la probabilidad de borrarla, etc.

El grado de disparo de cada regla se calcula con la t-norma mínimo y la salida de la misma con el operador de implicación mínimo. Para obtener un valor nítido a partir del conjunto difuso de salida, obtenido con la t-conorma máximo, se aplica el método del centro de área o centro de gravedad [6]. A partir de este sistema difuso se obtienen dos valores nítidos, a los cuales se les asocia con la probabilidad de aplicar cada uno de los dos operadores.

Tabla 1: Reglas difusas utilizadas

Antecedentes			*Consecuentes*	
va	*ve*	*vs*	*vop1*	*vop2*
B			M-A	M-A
M			M-B	A
A			B	A
	B		B	A
	M		M-B	A
	A		M-A	M-A
		B	B	A
		M	M-B	A
		A	M-A	M-A

```
6. Añadir las RBFs eliminadas
```

Este proceso se irá repitiendo iterativamente para cada una de las nuevas RBFs que haya que introducir en la RBFN. Como novedad se introducirán las funciones base en las zonas de mayor error, en lugar de en los puntos de mayor error, tal y como se hace tradicionalmente en este tipo de algoritmos.

```
7. Aplicar Levenberg-Marquardt
```

Para terminar se aplicará el algoritmo de minimización de Levenberg-Marquardt [10] con valores estándares para su funcionamiento.

3. RESULTADOS EXPERIMENTALES

El algoritmo propuesto se ha aplicado a la aproximación de funciones unidimensionales y bidimensionales. Las características de la experimentación son las siguientes:

- La condición de parada es que el bucle principal cumpla 300 iteraciones. Tras estas 300 iteraciones nos quedaremos con la RBFN que mejor error cuadrático medio (NRMSE) haya aportado. Si en las últimas 50 iteraciones se produce una mejora en error de la RBFN se suman, a las iteraciones actuales, 50 iteraciones más.
- El algoritmo en total se ha ejecutado un total de 30 veces para obtener un mejor, un peor y un valor medio de este NRMSE. En algunas ocasiones y para poder realizar comparaciones con otros trabajos se ha utilizado el MSE o error cuadrático medio.
- Como conjunto de entrenamiento para las funciones unidimensionales se han tomado 100 puntos equidistribuidos en el intervalo de definición de las funciones. El conjunto de test está formado por 1000 puntos igualmente equidistribuidos.
- Para la función bidimensional el conjunto de entrenamiento está formado por 400 puntos distribuidos en una rejilla de 20 x 20 celdas, escogiendo un punto de cada celda aleatoriamente. El conjunto de test está formado por 961 puntos, escogidos de igual manera de una rejilla de 31 x 31.

A continuación se muestran los resultados obtenidos para cada función y las comparaciones con otros métodos, en concreto con los siguientes:

1. *Nie y Lee* [12] que utiliza un sistema difuso basado en reglas.
2. *Dickerson y Kosco* [5] que utiliza un sistema híbrido neuro-difuso.
3. *Cherkassky* [4] que utiliza mapas autoorganizativos y perceptrones multicapa.
4. *Pomares* [14] basado en la generación de reglas difusas.
5. *González* [7] que es un algoritmo de diseño de RBFNs, basado en la evolución de una población donde los individuos son RBFNs, y que además realiza minimización con Levenberg-Marquardt. El uso de este tipo de población y otras técnicas hace que este algoritmo sea más costoso en tiempo y recursos, que el aquí presentado.

A continuación se describen las funciones utilizadas en la experimentación:

Función *nie*

Usada en [12] para probar sus algoritmos, se define:

$$nie(x) = 3e^{x^2}\,\mathrm{sen}(\pi x) \qquad x \in [-1,1] \qquad (8)$$

Los resultados obtenidos para esta función por nuestra propuesta se muestran en la Tabla 2 y la comparación con los resultados de otros algoritmos en la Tabla 3.

Tabla 2: Resultados para la función *nie*

Num. RBFs	*NRMSE Entrenamiento*			
	Mejor	*Peor*	*Medio*	*Des. Tip.*
3	0.0200	0.1330	0.1288	0.0155
4	0.0059	0.0096	0.0081	0.0006
5	0.0041	0.0081	0.0058	0.0006
6	0.0001	0.0053	0.0014	0.0016
7	0.0001	0.0046	0.0010	0.0009
Num. RBFs	*NRMSE Test*			
	Mejor	*Peor*	*Medio*	*Des. Tip.*
3	0.0195	0.1353	0.1280	0.0158
4	0.0059	0.0093	0.0079	0.0006
5	0.0042	0.0086	0.0061	0.0007
6	0.0001	0.0050	0.0014	0.0016
7	0.0001	0.0041	0.0010	0.0009

Tabla 3: Comparativa con otros algoritmos (función *nie*)

Algoritmo	*RBFs / Dif*	*MSE* (Entrenamiento)
Nie y Lee [12] *(Datos con varios parámetros)*	30	0.0111
		0.0082
		0.0078
		0.0068
Pomares [14]	6	0.0113
	7	0.0092
	10	0.0020
González [7]	4	5.73E-5
	5	2.87E-5
	6	1.74E-5
Algoritmo Propuesto	4	6.2E-5
	5	3.3E-5
	6	4E-6

Función *dick*

Esta función fue usada en [5] y se define como:

$$dick(x)=3x(x-1)(x-1.9)(x+0.7)(x+1.8) \qquad x\in[-2.1,2.1] \qquad (9)$$

Nuestros resultados para esta función se muestran en la Tabla 4 y la comparativa con los resultados de otros algoritmos en la Tabla 5.

Tabla 4: Resultados para la función *dick*

Num. RBFs	*NRMSE Entrenamiento*			
	Mejor	*Peor*	*Medio*	*Des. Tip.*
3	0.2819	0.5212	0.3130	0.0614
4	0.1007	0.2402	0.1853	0.0301
5	0.0029	0.0818	0.0802	0.0110
6	0.0009	0.0814	0.0231	0.0342
7	0.0002	0.0025	0.0014	0.0003
Num. RBFs	*NRMSE Test*			
	Mejor	*Peor*	*Medio*	*Des. Tip.*
3	0.3246	0.5403	0.3533	0.0576
4	0.1071	0.2925	0.2066	0.0447
5	0.0029	0.0871	0.0854	0.0117
6	0.0012	0.0870	0.0250	0.0364
7	0.0003	0.0030	0.0017	0.0003

Tabla 5: Comparativa con otros algoritmos (función *dick*)

Algoritmo	*RBFs / Dif*	*MSE* (Entrenamiento)
Dickerson y Kosko [5] *(Datos con varios parámetros)*	6	28.25
		10.53
		7.92
		3.07
Pomares [14]	5	5.01
	6	1.35
	7	0.46
González [7]	5	0.30
	6	7.4E-4
	7	7.2E-4
Algoritmo Propuesto	5	0.29
	6	0.07
	7	9.8E-5

Función f_7

Esta función fue usada en [4] y se define como:

$$f_7(x_1,x_2) = 1.9\left[1.35 + e^{x_1} sen(13(x_1 - 0.6)^2) e^{-x_2} sen(7x_2)\right]$$
$$x_1, x_2 \in [0,1]$$

Los resultados obtenidos por nuestro algoritmo para esta función se muestran en la Tabla 6. La comparación con otros algoritmos se muestra en la Tabla 7.

Los resultados obtenidos por el algoritmo demuestran que éste se comporta correctamente ya que:

- Se ha ido reduciendo el error conforme se han introducido nuevas RBFs.
- La desviación típica es baja lo que indica que el algoritmo es robusto.
- Existe una gran similitud entre los datos en el conjunto de entrenamiento y en el de test, lo que supone que el algoritmo ha generalizado adecuadamente.

Tabla 6: Resultados para la función f_7

Num. RBFs	*NRMSE Entrenamiento*			
	Mejor	*Peor*	*Medio*	*Des. Tip.*
7	0.1161	0.1901	0.1633	0.0196
10	0.0557	0.1221	0.0687	0.0201
13	0.0244	0.0444	0.0342	0.0042
16	0.0175	0.0321	0.0235	0.0037
19	0.0139	0.0240	0.0173	0.0026
22	0.0113	0.0208	0.0140	0.0027
Num. RBFs	*NRMSE Test*			
	Mejor	*Peor*	*Medio*	*Des. Tip.*
7	0.1130	0.1917	0.1621	0.0200
10	0.0578	0.1230	0.0700	0.0196
13	0.0292	0.0460	0.0372	0.0035
16	0.0200	0.0359	0.0277	0.0036
19	0.0167	0.0298	0.0215	0.0031
22	0.0155	0.0308	0.0221	0.0037

Tabla 7: Comparativa con otros algoritmos (función f_7)

Algoritmo	*RBFs / Dif*	*NRMSE* (Test)
Cherkassky	40	0.034
Pomares [14]	4 × 4 (PT)	0.161
	5 × 5 (PT)	0.109
	7 × 6 (PT)	0.059
González [7]	7	0.1426
	13	0.0590
	19	0.0324
Algoritmo Propuesto	7	0.1621
	13	0.0372
	19	0.0215

Si analizamos los resultados obtenidos y su comparación con otros algoritmos se observa que en general el algoritmo de González [7] y el aquí presentado destacan con cierta claridad sobre el resto de los algoritmos. En cuanto a los resultados de estos dos algoritmos no se puede afirmar que haya uno que destaque sobre el otro, ya que dependiendo de la función evaluada y del número de RBFs obtiene uno u otro, mejores resultados.

Sin embargo y si atendemos a la complejidad de los algoritmos, el algoritmo aquí presentado trabaja con una población de RBFs por lo que es poco complejo y costoso en cuanto a todo tipo de recursos, mientras que el de González trabaja con una población de RBFNs, lo que evidentemente lo hace más complejo y costoso. Esto hace que en este punto el algoritmo presentado supere al de González.

4. CONCLUSIONES

Este trabajo presenta un algoritmo bioinspirado para diseñar redes de funciones de base radial. En él se mantiene una población de RBFs que coevoluciona (cooperando y compitiendo) teniendo en cuenta una serie

de parámetros que caracterizan el trabajo de un individuo o RBF. Por otro lado, el procedimiento tiene carácter adaptativo y dinámico, gracias al uso de una técnica multiobjetivo para la ordenación y selección de RBFs y a la utilización de un SBRD que interviene en la elección del operador a aplicar.

Los resultados conseguidos mejoran nuestras propuestas anteriores y las de otros autores en el campo de la aproximación de funciones. Es especialmente destacable la simplicidad del algoritmo y el bajo coste de ejecución frente a otros algoritmos evolutivos de optimización de RBFNs. Por último resaltar que en la bibliografía existen pocos trabajos que, como el presentado, diseñen redes neuronales dentro de un marco coevolutivo.

Dentro de los futuros trabajos se estudiarán las relaciones entre los distintos parámetros, ampliar las características multiobjetivo del algoritmo, así como la posibilidad de adaptar el algoritmo a otras problemas como la clasificación de patrones.

Agradecimientos: Este trabajo se enmarca dentro de los proyectos TIC2000-1348, DPI2001-3219 y TIC2002-04036-C05-04 del Ministerio de Ciencia y Tecnología.

Referencias

[1] Beasly, D.; Bull, D. R.; Martin R.R.: "A sequential niche technique for multimodal function optimization." Evolutionary Computation, 1. Págs. 101-125. 1993

[2] Broomhead, D; Lowe, D. "Multivariable functional interpolation and adaptive networks". Complex System. 2. 1988. pp 321-355.

[3] Coello, C.C.:"An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the art and future trends". Congress on Evolutionary Computation, CEC'99, pp.3-13, 1999.

[4] Cherkassky, V.; Gehring, D.; Mulier, F."Comparison of adaptive methods for function estimation from samples", IEEE Trans. Neural Networks, vol.7, no.4, pp.969-984, July 1996.

[5] Dickerson, J.; Kosko, B. (1996). "Fuzzy Function Approximation with Ellipsoidal Rules", IEEE Trans. Syst. Man and Cyber.- Part B, vol.26, no.4, pp.542-560.

[6] Driankov, D.; Hellendoorn H.; Reinfrank, M.: *An introduction to fuzzy control*. Springer-Verlag, 1993.

[7] González, J.; Rojas, I.; Ortega, J.; Pomares, H.; Fernández, F.; Diaz, A. "Multiobjective evolutionary optimization of the size, shape and position parameters of radial basis function networks for function approximation", IEEE Transactions on Neural Networks, 2003, vol.14, n.6, pp.1478-1495.

[8] Herrera, F.; Lozano M.: "Adaptative Genetic Operators Based on Coevolution with Fuzzy Behaviors". IEEE Transactions on Evolutionary Computation. Vol 5 N. 2. 149-165. 2001

[9] Mandani, E.; Assilian, S. "An experitment in linguistic synthesys with a fuzzy logic controller". Int. J. Man-Machine Stud., v. 7, n. 1, pp. 1-13, 1975

[10] Marquardt, D. W "An algorithm for least-squares estimation of non-linear inequalities," SIAM J. Appl. Math., vol. 11, pp. 431-441, 1963.

[11] Moody, J; Darken, C.: "Fast learning networks of locally-tuned processing units," Neural Computation, vol. 3 n. 4 pp.579-588, 1.991

[12] Nie, J.H.; Lee, T.H.:"Rule based modeling: Fast construction and optimal manipulation". IEEE Trans. Syst. Man Cyber., Part A, 26(6) 1996

[13] Platt, J: "A resource-allocating network for function interpolation", Neural Computation 3, 213-225. 1.991

[14] Pomares, H.; Rojas, I.; Ortega, J.; González, J.; Prieto, A.: "A systematic approach to Self-Generating Fuzzy Rule-Table for Function Approximation". IEEE Trans. Syst., Man, and Cyber., Part B, 30(3):431-447. 2000

[15] Rivera, A. "Diseño y Optimización de Redes de Funciones de Base Radial Mediante Técnicas Bioinspiradas". Ph.D. dissertation, Univ. Granada, Spain, Sept. 2003

[16] Rivas, V.;Castillo, P.A.;Merelo-Guervós JJ. "Evolved rbf networks for time-series forecasting and function approximation." J. Merelo Guervós; et al. editor. LNCS 2439. Springer-Verlag, pp. 505 Sept. 2002

[17] Rojas,I.; Valenzuela,O.; Prieto,A.: "Statistical Analysis of the Main Parameters in the Definition of Radial Basic Function Networks", Lecture Notes in Computer Science, Vol.1240, pp. 882-891, Springer-Verlag, June 1997.

[18] Sareni, B.; Krähenbühl, L.:"Fitness sharing and niching methods revisited". IEEE Trans. on Evolutionary Computation, Vol.2, No.3, pp.97-106. September, 1998.

[19] Smalz, R.; Conrad, M.:"Combining evolution with credit apportionment: a new learning algorithm for neural nets". Neural Netwroks, Vol.7, No.2, pp.341-351, 1994.

[20] Whitehead, B.A.; Choate, T.D.:"Cooperative-competitive genetic evolution of Radial Basis Function centers and widths for time series prediction". IEEE Trans. on Neural Networks, Vol.7, No.4, pp.869-880. July, 1996.

[21] Widrow, B.; Lehr, M.A."30 Years of adaptative neural networks: perceptron, madaline and backpropagation". Proc. of the IEEE, v. 78 n. 9, sept. 90.

[22] Yao, X "Evolutionary artificial neural networks" Proceedings of the IEEE, 87 (9): pp. 1423-1447, September 1999

Equivalencia funcional entre General Regression Neural Network y Sistemas de Inferencia Difusa

José Luis Aznarte
jlaznarte@decsai.ugr.es
Dpto. Ciencias de la Computación e I.A.
Universidad de Granada
18071 Granada – Spain

José Manuel Benítez
J.M.Benitez@decsai.ugr.es
Dpto. Ciencias de la Computación e I.A.
Universidad de Granada
18071 Granada – Spain

Juan Luis Castro
castro@decsai.ugr.es
Dpto. Ciencias de la Computación e I.A.
Universidad de Granada
18071 Granada – Spain

Abstract

En este documento demostraremos que el modelo *General Regression Neural Network* (GRNN) es funcionalmente equivalente a un Sistema de Inferencia Difusa del tipo propuesto por Takagi, Sugeno y Kang (SID TSK), bajo restricciones leves. Este resultado permite dotar de interpretabilidad a un sistema GRNN previamente entrenado y también proporciona un nuevo método para ajustar los parámetros del SID. Además, la relación de equivalencia resulta en una GRNN que emplea una nueva familia de núcleos de Parzen.

Palabras Clave: General Regression Neural Newtork, Sistemas Basados en Reglas difusas TSK, relación de equivalencia funcional

1. Introducción

Este trabajo demuestra la equivalencia funcional entre la *General Regression Neural Network* (GRNN) y los Sistemas de Inferencia Difusa de tipo TSK. Estos modelos surgen de la aplicación de varios paradigmas de la Computación Flexible (razonamiento probabilístico, redes neuronales y razonamiento difuso), y ambos han sido aplicados con éxito en multitud de problemas. Mostraremos cómo, bajo restricciones leves, estos dos modelos son equivalentes, lo cual redunda en una mejor comprensión y uso de cada modelo. Además, esta relación de equivalencia permite mejoras en ambos modelos, como es la interpretabilidad de la GRNN y un nuevo método rápido para la construcción de bases de reglas TSK.

Es mucho el esfuerzo dedicado a establecer relaciones entre representaciones del conocimiento basadas en reglas y sistemas neuronales como los perceptrones multicapa. En [4] y [5] se presentó una semántica de redes neuronales como reglas lógicas bivaluadas que podían ser usadas en un proceso de inferencia. Thrun, en [12], asignó *intervalos de validez* al rango de activación de las unidades de una red MLP, obteniendo así reglas *SI-ENTONCES* de dichos intervalos. Otra contribución interesante es la debida a Benítez *et al.* en [2], en la que se desarrolló un método formal para la extracción de reglas desde redes neuronales entrenadas. Dicho método empleaba una nueva lógica conectiva de gran potencia, el operador *o-interactivo*.

En cuanto a las redes neuronales basadas en núcleos, Jang en [6] trabajó en una aproximación a la equivalencia entre sistemas TSK y Redes de Función de Base Radial. Comentarios posteriores sobre este trabajo [1] probaron que había ciertos aspectos por resolver en la propuesta de Jang y presentaron una modificación a la misma.

2. General Regression Neural Networks

El modelo básico GRNN fue publicado en 1991 por Donald F. Specht [11] como una extensión de un modelo anterior, la Probabilistic Neural Network (PNN), y reinventado posteriormente por Schiøler y Hartmann [7]. En realidad, la GRNN es el nombre que Specht dio a la regresión basada en núcleos de Nadayara-Watson [9, 13].

La regresión de una variable dependiente $\mathbf{Y}$ sobre una variable independiente $\mathbf{X}$ es el cálculo del valor más probable de $\mathbf{Y}$. En general, $\mathbf{Y}$ es un vector, pero para simplificar, todos los Y serán considerados escalares en adelante.

Sea $\mathbf{x}$ un valor de la variable aleatoria $\mathbf{X}$. Con la función continua de densidad de probabilidad (pdf, de *probability density function*) $f(\mathbf{X}, Y)$ de $\mathbf{X}$ e Y la media condicional $E[Y|\mathbf{x}]$ y la regresión

$\hat{y}(\mathbf{x})$ para un dado $\mathbf{x}$ pueden ser calculadas mediante

$$\hat{y}(\mathbf{x}) \equiv E[Y|\mathbf{x}] = \frac{\int_{-\infty}^{+\infty} Y \cdot f(\mathbf{x}, Y)\mathrm{d}Y}{\int_{-\infty}^{+\infty} f(\mathbf{x}, Y)\mathrm{d}Y}. \quad (1)$$

Si la pdf no es conocida, debe ser aproximada mediante ejemplos. Un ejemplo se define como una tupla $(\mathbf{x}, y)$. Para lograr una aproximación no paramétrica de $f(\mathbf{X}, Y)$ Specht emplea la clase de aproximadores consistentes propuesta por Parzen [10] y Cacoullos [3].

Parzen mostró cómo, en el caso univariante, puede construirse una familia de aproximadores de la pdf $f(x)$. Propuso el aproximador

$$f_n(x) = \frac{1}{n}\sum_{j=1}^{n} K(x, z_j) \quad (2)$$

donde K es una función núcleo definida por

$$K(X, Z) = \frac{1}{\sigma^d}\, h\left[\frac{\delta(X, Z)}{\sigma}\right] \quad (3)$$

siendo σ un parámetro de suavizado llamado ancho de núcleo, d la dimensión de $\mathbf{X}$, δ una medida de distancia y h una función que satisface las siguientes condiciones:

$$i. \quad \sup_{-\infty<x<+\infty} |h(x)| < \infty \quad (4)$$

$$ii. \quad \int_{-\infty}^{+\infty} |h(x)| \cdot \mathrm{d}x < \infty \quad (5)$$

$$iii. \quad \lim_{x\to\infty} |x \cdot h(x)| = 0 \quad (6)$$

$$iv. \quad \int_{-\infty}^{+\infty} h(x) \cdot \mathrm{d}x = 1 \quad (7)$$

Parzen mostró que el aproximador así construido es consistente en media cuadrática, es decir, que

$$E|f_n(X) - f(X)|^2 \to 0 \text{ cuando } n \to +\infty. \quad (8)$$

Según [10], el aproximador de probabilidad tiene, en el caso univariante, la siguiente forma:

$$\hat{f}(X, Y) = \frac{1}{n\sigma^2}\sum_{i=1}^{n} h\left(\frac{X - X_i}{\sigma}\right) \cdot h\left(\frac{Y - Y_i}{\sigma}\right). \quad (9)$$

Para calcular la regresión univariante $\hat{y}(x)$ de Y para un x dado, hay que elegir una función núcleo apropiada, insertar la ecuación (9) en la ecuación (1) y calcular las integrales.

Cacoullos [3] extendió el resultado de Parzen para considerar también el caso multivariante, y obtuvo la versión multivariante de la ecuación (9):

$$\hat{y}(\mathbf{x}) = \frac{\sum_{i=1}^{n} y_i \cdot h\left(\frac{\delta(\mathbf{x}, \mathbf{x}_i)}{\sigma}\right)}{\sum_{i=1}^{n} h\left(\frac{\delta(\mathbf{x}, \mathbf{x}_i)}{\sigma}\right)} \quad (10)$$

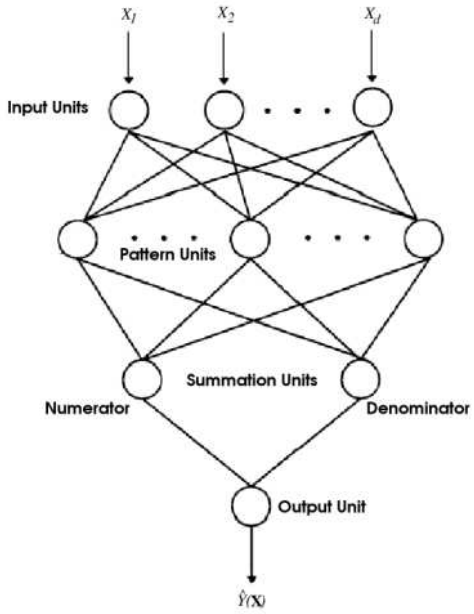


Figura 1: Arquitectura básica de la GRNN.

Specht construyó la GRNN (cuya estructura es la mostrada en la figura 1) usando la métrica euclidiana

$$\delta(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T(\mathbf{x}_1 - \mathbf{x}_2)} \quad (11)$$

y el núcleo gaussiano, es decir,

$$h(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right). \quad (12)$$

Hay muchas otras funciones de base radial (piramidal, cuadrada etc. cf. [8, p. 168]) que pueden ser usadas aquí en lugar de la gaussiana. La función de base radial suele ser escogida atendiendo a criterios computacionales.

3. Reglas difusas *Si-Entonces* y sistemas de inferencia difusa

Los sistemas de inferencia difusa (SID) o sistemas basados en reglas difusas (SBRD) son un conocido marco computacional basado en los conceptos de la Teoría de Conjuntos Difusos, en las reglas difusas de tipo *Si-Entonces* y en el razonamiento difuso. En este trabajo nos centraremos en los SBRD de tipo TSK, que son aquellos cuyas reglas tienen un consecuente que toma la forma de una función lineal de las entradas.

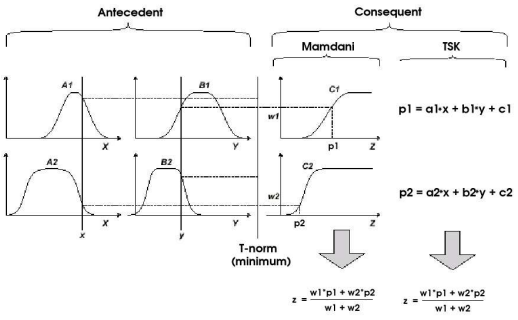


Figura 2: Esquemas de razonamiento difuso.

3.1. Sistema de inferencia difusa TSK

Una regla general de un SID TSK tiene la forma siguiente:

$$\begin{aligned} \textit{Regla } i: \text{SI } x_1 \text{ ES } A_1^i \text{ Y } x_2 \text{ ES } A_2^i \text{ Y} \\ \ldots \text{ Y } x_d \text{ ES } A_d^i \\ \text{ENTONCES } y_i = p_i(\mathbf{x}) = a_1^i x_1 + a_2^i x_2 + \\ + \ldots + a_d^i x_d + a_{d+1}^i \end{aligned}$$

donde x_i son variables de entrada; y_i es una salida; A_j^i son conjuntos difusos para las variables de entrada; y p_i es la función lineal de salida.

El mecanismo de razonamiento difuso está ilustrado en la figura 2, bajo el rótulo "TSK". La *intensidad de disparo* de la regla i-ésima se obtiene como la t-norma (usualmente el operador producto o mín) de los valores de pertenencia de la premisa:

$$\omega_i(\mathbf{x}) = \prod_{j=1}^{d} \mu_{A_j^i}(x_j) \tag{13}$$

o bien

$$\omega_i(\mathbf{x}) = \min_j \left\{ \mu_{A_j^i}(x_j) \right\}. \tag{14}$$

La salida total puede calcularse como la suma ponderada de la salida de cada regla,

$$y(\mathbf{x}) = \sum_{i=1}^{R} \omega_i p_i(\mathbf{x}), \tag{15}$$

o, más comúnmente, como la media ponderada (como se muestra en la figura 2)

$$y(\mathbf{x}) = \frac{\sum_{i=1}^{R} \omega_i p_i(\mathbf{x})}{\sum_{i=1}^{R} \omega_i}. \tag{16}$$

4. Equivalencia funcional entre General Regression Neural Networks y Sistemas de Inferencia Difusa

Como hemos visto, los modelos GRNN y SID tienen fundamentaciones matemáticas bastante distintas. Sin embargo, vamos a demostrar cómo son, esencialmente, formas distintas de realizar un mismo cálculo. Haremos uso del siguiente resultado:

Lema. *La funcion nucleo de Parzen de tipo gaussiano es equivalente a la intensidad de disparo de una regla difusa que emplee funciones de pertenencia gaussianas.*

Demostracion. En el caso univariante, la prueba es trivial porque en ese caso la función núcleo es

$$K(X, Z) = \frac{1}{\sigma(2\pi)^{1/2}} \exp\left\{ -\frac{1}{2} \left(\frac{X - Z}{\sigma} \right)^2 \right\}, \tag{17}$$

que es exactamente la expresión de una función de pertenencia gaussiana con parámetros Z y σ aplicada sobre X:

$$K_\sigma(X, Z) = \mu_{A\{Z,\sigma\}}(X) = \omega(X). \tag{18}$$

En el caso multivariante, siendo $\delta(\cdot, \cdot)$ la métrica euclidiana, tenemos

$$\begin{aligned} &\exp\left\{ -\frac{\delta(\mathbf{X}, \mathbf{Z})^2}{2\sigma} \right\} = \\ &= \exp\left\{ -\frac{\sqrt{(\mathbf{X} - \mathbf{Z})^T (\mathbf{X} - \mathbf{Z})}^2}{2\sigma^2} \right\} \\ &= \exp\left\{ -\frac{(\mathbf{X} - \mathbf{Z})^T (\mathbf{X} - \mathbf{Z})}{2\sigma^2} \right\} \\ &= \exp\left\{ -\frac{(X_1 - Z_1)^2 + \ldots + (X_d - Z_d)^2}{2\sigma^2} \right\} \\ &= \exp\left\{ -\frac{(X_1 - Z_1)^2}{2\sigma^2} - \ldots - \frac{(X_d - Z_d)^2}{2\sigma^2} \right\} \\ &= \prod_{j=1}^{d} \exp\left\{ -\frac{(X_j - Z_j)^2}{2\sigma^2} \right\}, \end{aligned}$$

y, por tanto,

$$K_\sigma(\mathbf{X}, \mathbf{Z}) = \prod_{j=1}^{d} \mu_{A_j^{\{Z,\sigma\}}}(X_j). \tag{19}$$

Según la ecuación (13),

$$K(\mathbf{X}, \mathbf{Z}) = \omega(\mathbf{X}). \tag{20}$$

□

Usando este lema podemos demostrar fácilmente el siguiente

Teorema (Equivalencia funcional). *El modelo GRNN es funcionalmente equivalente a un modelo SID TSK cuyas reglas tengan consecuentes constantes.*

Demostracion. Podemos construir un SID TSK equivalente a una GRNN dada. El sistema tendrá las siguientes características:

- Habrá tantas reglas difusas como unidades tenga la GRNN en su capa de patrones (*pattern layer*).
- Cada regla difusa estará relacionada con una unidad de patrón de la GRNN, y tendrá la siguiente forma:

  SI x_1 ES A_1^i Y x_2 ES A_2^i Y $\ldots$ Y x_d ES A_d^i
  ENTONCES $y_i = p_i(\mathbf{x}) = Y_i$

  donde:

  - el operador t-norma (Y) será la operación producto,
  - las funciones de pertenencia difusa de las etiquetas A_j^i serán funciones gaussianas con parámetros $\{z_j^i, \sigma\}$, siendo z_j^i las componentes individuales de $\mathbf{z}^i$ (el dato almacenado en la unidad de patrón) y σ el ancho de núcleo.
  - La función del consecuente, $p_i(\mathbf{x})$ es igual al valor de salida del dato correspondiente a la i-ésima unidad de patrón.

Si se dan estas condiciones, de las ecuaciones (10) y (16) se desprende que el SID TSK así obtenido realizará el mismo cálculo que la GRNN correspondiente.

Inversamente, si lo que tenemos es un SID TSK con t-normas multiplicativas, funciones de pertenencia gausianas y consecuentes constantes, la GRNN equivalente se construye fácilmente considerando el antecedente de cada regla como una función núcleo de Parzen (según el lema anterior). □

Analizaremos ahora algunas consecuencias de este resultado.

4.1. La GRNN como un sistema de inferencia difuso TSK

A partir del teorema anterior, es posible ver la GRNN como un SID TSK y aplicar todo el conocimiento disponible sobre este modelo a aquél. En particular, esto implica que la GRNN satisface las propiedades asociadas con los SIDs, como, por ejemplo, que son aproximadores universales. Es posible también aplicar algoritmos de entrenamiento diseñados para modelos SID para ajustar los parámetros de la GRNN, algo que será de especial utilidad cuando se usen múltiples σ.

Uno de los principales inconvenientes del modelo GRNN es que padece de problemas de dimensionalidad. Cuando el tamaño del conjunto de entrenamiento aumenta, la red crece hasta tamaños inmanejables. Esta es la razón por la que Specht [11] sugirió un algoritmo de construcción/entrenamiento basado en técnicas de agrupamiento. Si se construye la red usando dicho método, contendrá un número reducido de unidades de patrón, pero su estructura de cálculo será la misma. Por tanto seguirá siendo posible obtener el sistema difuso de inferencia equivalente de la forma explicada anteriormente.

4.1.1. Un ejemplo

Para ilustrar la equivalencia propuesta, consideremos el siguiente ejemplo. Sea una GRNN con dos entradas y cuatro unidades de patrón, cuyos datos de origen son los puntos

$$\mathbf{x}^1 = (0{,}13, 0{,}15;\ 0{,}4)$$
$$\mathbf{x}^2 = (0{,}18, -0{,}19;\ 0{,}9)$$
$$\mathbf{x}^3 = (-0{,}41, -0{,}30;\ 1{,}1)$$
$$\mathbf{x}^4 = (-0{,}15, 0{,}14\ 0{,}5)$$

La figura 3 (a) muestra los núcleos de Parzen que la GRNN emplearía para cada uno de estos datos. La figura 3 (b), por otro lado, muestra el estimador basado en núcleos de Parzen que se obtendría de los cuatro puntos. Este modelo GRNN es equivalente a un sistema de inferencia difuso cuyas reglas son las siguientes:

$$\textit{Regla } 1: \text{SI } x_1 \text{ ES } A_1^1 \text{ Y } x_2 \text{ ES } A_2^1 \text{ ENTONCES } y_1 = 0{,}4$$
$$\textit{Regla } 2: \text{SI } x_1 \text{ ES } A_1^2 \text{ Y } x_2 \text{ ES } A_2^2 \text{ ENTONCES } y_2 = 0{,}9$$
$$\textit{Regla } 3: \text{SI } x_1 \text{ ES } A_1^3 \text{ Y } x_2 \text{ ES } A_2^3 \text{ ENTONCES } y_3 = 1{,}1$$
$$\textit{Regla } 4: \text{SI } x_1 \text{ ES } A_1^4 \text{ Y } x_2 \text{ ES } A_2^4 \text{ ENTONCES } y_4 = 0{,}5.$$

La forma de cada etiqueta A_i^j es la que aparece en la figura 4, y su expresión es

$$A_i^j(x) = \frac{1}{\sigma\sqrt{2\pi}}\ \exp\left\{-\frac{1}{2}\left(\frac{x - x_i^j}{\sigma}\right)^2\right\}.$$

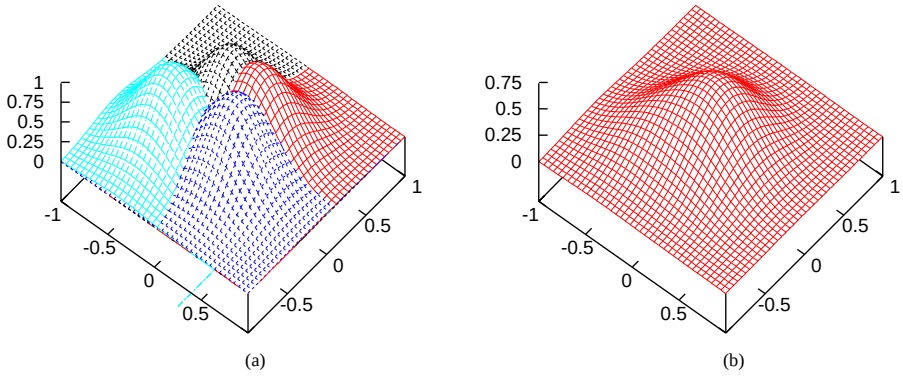


Figura 3: GRNN con cuatro datos: (a) núcleos de Parzen para cada dato; (b) aproximador de Parzen obtenido.

Como ya fue advertido en el caso de la equivalencia entre RBFN y SID [1], el SID que se obtiene tiene la particularidad de que todas las funciones de pertenencia tienen la misma varianza. Esta es una restricción poco usual en estos modelos, pero no ha de afectar excesivamente a su eficacia.

Las reglas obtenidas pueden ser interpretadas en términos de conocimiento lingüístico. Por ejemplo, si tomamos la regla 4, su significado sería *"si* x_1 *es aproximadamente* $-0{,}15$ *y* x_2 *es aproximadamente* $0{,}14$ *entonces la salida sera* $0{,}5$*"*. La salida final del sistema sería una media ponderada de las salidas de las regla, en la que la ponderación vendría dada por el grado de pertenencia de las variables de entrada en cada regla.

4.2. El sistema de inferencia difuso TSK como una GRNN

Como hemos apuntado ya, si un SID TSK (con consecuente constante) emplea funciones de pertenencia gaussianas, es equivalente a una GRNN. Esto puede verse claramente con el ejemplo de la sección 4.2.1 siguiente. Pero ¿qué ocurre si tenemos un SID que emplee otro tipo de funciones de pertenencia no gaussianas? ¿Sería posible construir una GRNN equivalente? Ilustraremos este punto con un ejemplo sencillo.

Sea un SID con dos entradas y funciones de pertenencia triangulares. Para cada regla, la agregación de los grados de pertenencia de cada variable resulta en una superficie tridimensional que satisface todas las condiciones (expresadas más arriba) necesarias para que una función sea núcleo de Parzen (aunque una constante de normalización podría ser necesaria para verificar la propiedad (7)). De aquí se desprende que la condición de funciones de pertenencia gaussianas puede ser relajada.

4.2.1. Un ejemplo empleando funciones de pertenencia gaussianas

Con fines ilustrativos, tomemos un sistema de inferencia difusa con t-normas multiplicativas cuya base de reglas está formada por las siguientes:

$$\begin{aligned}
&\textit{Regla } 1: \text{SI } x_1 \text{ ES } A_1^1(x) \text{ Y } x_2 \text{ ES } A_2^1(x)\\
&\qquad\qquad \text{ENTONCES } y_1 = p_1(x_1, x_2)\\
&\textit{Regla } 2: \text{SI } x_1 \text{ ES } A_1^2(x) \text{ Y } x_2 \text{ ES } A_2^2(x)\\
&\qquad\qquad \text{ENTONCES } y_2 = p_2(x_1, x_2)\\
&\textit{Regla } 3: \text{SI } x_1 \text{ ES } A_1^3(x) \text{ Y } x_2 \text{ ES } A_2^3(x)\\
&\qquad\qquad \text{ENTONCES } y_2 = p_2(x_1, x_2)\\
&\textit{Regla } 4: \text{SI } x_1 \text{ ES } A_1^4(x) \text{ Y } x_2 \text{ ES } A_2^4(x)\\
&\qquad\qquad \text{ENTONCES } y_2 = p_2(x_1, x_2)
\end{aligned}$$

en las que los A_j^i son funciones de pertenencia gaussianas. Si queremos construir una GRNN equivalente, sabemos que tendrá cuatro unidades de patrón, cuyas funciones de base radial serán de la forma:

$$f_i(\mathbf{x}) = \prod_{j=1}^{2} A_j^i(x_j) = \prod_{j=1}^{2} e^{\left(-\frac{\|x_j - x_j^i\|^2}{2\sigma^2}\right)}. \quad (21)$$

Si el consecuente de las reglas difusas es una constante $c_i = y_i$, tenemos la implicación complementaria a la descrita en la sección 4.1, y podemos, por tanto, establecer que una GRNN con un tipo especial de núcleos de Parzen *compuestos* es funcionalmente equivalente a un sistema de inferencia difusa con este tipo de consecuentes.

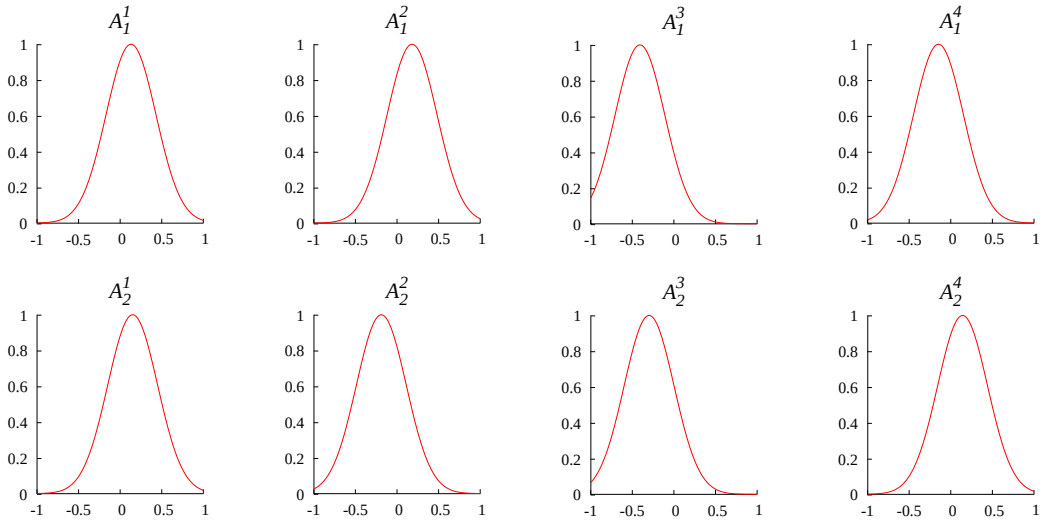


Figura 4: Funciones de pertenencia difusa del SID equivalente a la GRNN.

5. Conclusiones y notas finales

En este documento, hemos explorado las conexiones entre dos componentes de la Computación Flexible, la GRNN y el SID TSK. Hemos probado que, bajo restricciones leves, son equivalentes.

Como consecuencia inmediata de este hecho, tenemos que las propiedades de cada modelo son válidas para el otro, y que los métodos pueden ser intercambiados. Por ejemplo, cabe pensar en un nuevo método para entrenar la GRNN y en un nuevo método de construcción de un sistema TSK. Además, es posible idear nuevos modelos híbridos a partir del resultado expuesto, de forma que aúnen las ventajas de ambos modelos originales y minimicen sus inconvenientes. Estos métodos y modelos, entre otras posibilidades que la equivalencia ofrece, serán estudiados en el futuro.

Referencias

[1] H.C. Andersen, A. Lotfi, and L.C. Westphal. Comments on "functional equivalence between radial basis function networks and fuzzy inference systems". *IEEE Transactions on Neural Networks*, 1998.

[2] J.M. Benítez, J.L. Castro, and I. Requena. Are artificial neural networks black boxes? *IEEE Transactions on Neural Networks*, 1997.

[3] T. Cacoullos. Estimation of a multivariate density. *Annals of the Institute of Statistical Mathematics*, 1966.

[4] S.L. Gallant. Connectionist expert systems. *Communications of the ACM*, 1988.

[5] M.J. Healy and T.P.Caudell. Acquiring rule sets as a product of learning in a logical neural architecture. *IEEE Transactions on Neural Networks*, 1997.

[6] J.-S.R. Jang and C.-T. Sun. Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Transactions on Neural Networks*, 1993.

[7] H. Schiøler and U. Hartmann. Mapping neural networks derived from the parzen window estimator. *Neural Networks*, 1992.

[8] Timothy Masters. *Advanced Algorithms for neural networks: a C++ sourcebook*. Wiley and Sons, 1995.

[9] E.A. Nadayara. On estimating regression. *Theory Probability Applications*, 1964.

[10] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 1962.

[11] D.F. Specht. A general regression neural network. *IEEE Transactions on Neural Networks*, 1991.

[12] S. Thrun. Extracting rules from artificial neural networks with distributed representations. *Advances in Neural Information processing Systems*, 1995.

[13] G.S. Watson. Smooth regression analysis. *Sankhya, Series A*, 1964.

INICIALIZACIÓN DE CENTROS DE RBFs UTILIZANDO CLUSTERING PARA PROBLEMAS DE APROXIMACIÓN FUNCIONAL

A. Guillén, I. Rojas, J. González, L. J. Herrera, H. Pomares, A. Prieto
Dpto. Arquitectura y
Tecnología de Computadores
Universidad de Granada
aguillen@atc.ugr.es

Resumen

Dentro del diseño de una red de RBFs para problemas de aproximación funcional, la inicialización de los centros de las RBFs ha sido resuelta mediante algoritmos de clustering que no se ajustaban a las características del problema de aproximación funcional. A continuación describimos un algoritmo que tiene en cuenta la función objetivo que queremos aproximar y que nos proporcionará unos resultados muy por encima de los algoritmos usados hasta ahora.

Palabras Clave: clustering, Fuzzy, k-means, CFA, RBF, migración.

1 INTRODUCCIÓN

Cuando queremos obtener una función de base radial (RBF) para resolver un problema de aproximación funcional, el primer paso debe ser inicializar los centros de nuestras RBF. Para esta tarea se utilizan frecuentemente algoritmos de clustering [1, 2, 8], que dadas las características de estos, no proporcionan unas soluciones lo suficientemente buenas puesto que necesitan ser adaptados al problema de aproximación funcional. Los principales problemas de los algoritmos de clustering en relación a nuestro problema son:

1. La variable de salida sólo puede tomar valores en un intervalo de etiquetas válidas definido a priori, mientras que en los problemas de aproximación funcional la salida puede tomar cualquier valor dentro de un intervalo de numeros reales.

2. Como consecuencia del punto anterior, un error en una clasificación de dos puntos cercanos, en un algoritmo de clasificación puede originar un aumento del error muy grande ya que no se define una distancia entre clases. En los problemas de aproximación funcional, si el error es un número cercano al valor real de la salida, el error de aproximación aumenta en una proporción menor.

3. En nuestro problema, deseamos interpolar una serie de valores que no aparecían en el conjunto de entrenamiento, cosa que en los algoritmos de clustering no se contempla exactamente ya que las salidas del sistema están restringidas al conjunto de clases aprendido.

A continuación, se describe un nuevo algoritmo de clustering que tiene en consideración las características del problema que estamos intentando resolver, adaptándose a ellas y mediante el empleo de lógica difusa, nos proporcionará unas soluciones mucho mejores de las que nos proporcionan los algoritmos de clustering tradicionales.

2 DESCRIPCIÓN DEL PROBLEMA: APROXIMACIÓN DE FUNCIONES

El problema de estimar una función desconocida F a partir de muestras del tipo $\left\{(\vec{x}^k; y^k); k = 1, ..., n\right\}$ con $y^k = F(\vec{x}^k) \in \Re$ y $\vec{x}^k \in \Re^d$ (es decir, aproximación de funciones a partir de un conjunto finito de vectores de E/S), es aprender una relación funcional desconocida entre los vectores de entrada $\vec{x}^k$ y sus salidas asociadas y^k utilizando un conjunto de ejemplos de entrenamiento. Cuando ya tengamos esta relación funcional, seremos capaces de obtener la salida a partir de entradas que no estaban dentro de las que se nos proporcionó en el conjunto de entrenamiento.

Una red de funciones de base radiales $\mathcal{F}$ con estructura fija para aproximar una función desconocida F de d entradas y una salida a partir de un conjunto de muestras del tipo $\left\{(\vec{x}^k; y^k); k = 1, ..., n\right\}$ con $y^k = F(\vec{x}^k) \in \Re$ y $\vec{x}^k \in \Re^d$, tiene una serie de parámetros, los cuales de-

beremos optimizar:

$$\mathcal{F}\left(\vec{x}^k; C, R, \Omega\right) = \sum_{j=1}^{m} \phi(\vec{x}^k; \vec{c}^j, r^j) \cdot \Omega_j \qquad (1)$$

donde $C = \{\vec{c}^1, ..., \vec{c}^m\}$ es el conjunto de parámetros del sistema que definen la ubicación en el espacio de entrada de las funciones base de la red, $R = \{r^1, ..., r^m\}$ es el conjunto de parámetros que define el radio de cada función base, $\phi(\vec{x}^k; \vec{c}^j, r^j)$ representa a una función base radial y $\Omega = \{\Omega_1, ..., \Omega_m\}$ es el conjunto de pesos de la red.

En este artículo vamos a describir un algoritmo que nos proporcionará la posición de los centros de las funciones de forma que podamos calcular el resto de los parámetros en función de estos siguiendo el procedimiento que se muestra a continuación:

1. Inicializar los centros $\vec{c}^j$ de las funciones base
2. Inicializar los radios r^j de las funciones base
3. Calcular de forma óptima el valor de los pesos ω_j
4. Minimizar el error de aproximación de la red inicial

3 ALGORITMO DE CLUSTERING ANTERIOR: CFA

Este algoritmo, descrito en [5], tiene en cuenta la variabilidad de la salida de la función que queremos aproximar.

Para llevar a cabo esta tarea, el algoritmo CFA incorpora un conjunto $O = \{o^1, ..., o^m\}$ que representa una estimación de la salida esperada para los puntos que pertenezcan a un mismo *cluster*. El valor de o^j se obtiene como una media ponderada de las salidas de los ejemplos de entrenamiento pertenecientes al *cluster* C_j.

La función de distorsión a minimizar en se define como:

$$\varepsilon = \frac{\sum_{j=1}^{m} \sum_{\vec{x}^k \in C_j} \|\vec{x}^k - \vec{c}^j\|^2 \omega_{kj}}{\sum_{j=1}^{m} \sum_{\vec{x}^k \in C_j} \omega_{kj}} \qquad (2)$$

donde ω_{kj} pondera la influencia de cada ejemplo de entrenamiento $\vec{x}^k$ en la posición final del prototipo j-ésimo. Matemáticamente, ω_{kj} se define como:

$$\omega_{kj} = \frac{|F(\vec{x}^k) - o^j|}{\max_{i=1}^{n}\{F(\vec{x}^i)\} - \min_{i=1}^{n}\{F(\vec{x}^i)\}} + \vartheta_{\min}, \quad \vartheta_{\min} > 0 \qquad (3)$$

El primer sumando de esta expresión calcula una distancia normalizada (en el intervalo [0,1]) entre $F(\vec{x}^k)$ y o^j, mientras que el segundo sumando es un umbral de mínima contribución (cuando no exista controversia ω_{kj} tomará el valor $\vartheta_{\min}$). A medida que el valor de $\vartheta_{\min}$ decrece, el algoritmo CFA fuerza a los prototipos a concentrarse en las zonas del espacio de entrada en las que la función objetivo sea más variable.

El esquema general que sigue el algoritmo CFA es el siguiente:

$error_1 = \infty$
Repetir
  Calcular particion de los vectores
  Actualizar prototipos y sus salidas estimadas
  Migrar prototipos
  Calcular $error_i$
Hasta(abs($error_{i-1}$-$error_i$<$umbral$)

3.1 ACTUALIZACIÓN DE LOS PROTOTIPOS Y SUS SALIDAS ESPERADAS

Una vez que todos los vectores de entrada han sido asignados a algún *cluster* C_j, se deben actualizar los prototipos y la estimación de sus salidas. Esta actualización se lleva a cabo mediante el siguiente proceso iterativo:

$$\vec{c}^j = \frac{\sum_{\vec{x}^k \in C_j} \vec{x}^k \omega_{kj}}{\sum_{\vec{x}^k \in C_j} \omega_{kj}} \quad o^j = \frac{\sum_{\vec{x}^k \in C_j} F(\vec{x}^k) \omega_{kj}}{\sum_{\vec{x}^k \in C_j} \omega_{kj}} \qquad (4)$$

3.2 MIGRACIÓN DE PROTOTIPOS

Debido a que se realiza un proceso iterativo, podemos llegar a un mínimo local y quedarnos estancados en esa zona del espacio solución, para solucionar este problema, se incorpora en el CFA un paso de migración de forma que los prototipos con poca utilidad vayan a zonas donde puedan ser más útiles. Se define la utilidad como:

$$u_j = \frac{\varepsilon_j}{\bar{\varepsilon}} \quad \forall j = 1, ..., m \quad \bar{\varepsilon} = \frac{1}{m} \sum_{j=1}^{m} \varepsilon_j = \frac{\varepsilon}{m} \qquad (5)$$

$$\varepsilon_j = \frac{\sum\limits_{\vec{x}^k \in C_j} \|\vec{x}^k - \vec{c}^j\|^2 \omega_{kj}}{\sum\limits_{j=1}^{m} \sum\limits_{\vec{x}^k \in C_j} \omega_{kj}} \quad (6)$$

La solución óptima la obtendremos cuando cada centro contribuya en igual medida a la distorsión total [4].

4 ALGORITMO FCFA

Este algoritmo se basa en dos algoritmos utilizados anteriormente en problemas de clustering: Fuzzy k-means[3] y CFA[5]. El algoritmo de base es el Fuzzy k-mean al que se le han adaptado algunos elementos presentes en el CFA de modo que vamos a tener en cuenta la variabilidad de la salida de la función que queremos aproximar.

El algoritmo FCFA, al igual que el Fuzzy c-means, se basa sobre una partición fuzzy del conjunto de vectores de entrada, al contrario de lo que se hacia en el algoritmo CFA donde teníamos una partición de voronoi, donde cada vector solo puede pertenecer a un cluster. Dentro de este conjunto de vectores de entrada, vamos a considerar tanto las coordenadas de los vectores como sus salidas concatenando las salidas de los vectores de entrada con la matriz de sus coordenadas de los vectores de entrada con sus salidas correspondientes, como ya se hiciera en [7].

Sea $U = [u_{i,k}]$ la matriz que contiene los valores de pertenencia de un vector a un cluster y V la matriz que contiene las coordenadas de los centros de los clusters. La convergencia hacia una solución se realizará mediante un proceso iterativo en el cual, vamos calculando la matriz U y los prototipos partiendo de una incialización aleatoria de cualquiera de ellos dos ($V_{t-1} \rightarrow U_t \rightarrow V_t$ o bien $U_{t-1} \rightarrow V_t \rightarrow U_t$). El criterio de parada dependiendo de el tipo de iteración será:$\|V_{t-1} - V_t\| < \varepsilon$ o $\|U_{t-1} - U_t\| < \varepsilon$.

El cálculo de los elementos de las matrices U y V se obtiene mediante un proceso iterativo igual que el que se utiliza en el Fuzzy C-means [3]:

$$u_{ik} = \left(\sum_{j=1}^{c} \left(\frac{D_{ikA}}{D_{jkA}} \right)^{\frac{2}{m-1}} \right)^{-1} \quad (7)$$

$$v_i = \frac{\sum\limits_{k=1}^{n} u_{ik}^m x_k}{\sum\limits_{k=1}^{n} u_{ik}^m} \quad (8)$$

Es mas conveniente realizar las iteraciones y establecer el criterio de parada utilizando V ya que podremos converger mas fácilmente (el numero de parámetros que varían en cada iteración es sensiblemente menor que en U) y es mas eficiente además de necesitar menos memoria.

El esquema que sigue el algoritmo FCFA es el siguiente:

Inicializar una matriz U_1 de forma aleatoria
Calcular los $centros_1$ a partir de U_1
Repetir
    Calcular w
    Calcular distancia entre $centros_i$ y los [X,Y]
    Calcular nueva U_i
    Calcular nuevos $centros_i$ a partir de U_i
    Migrar
Hasta(abs($centros_{i-1}$-$centros_i$¡$umbral$)

Los elementos nuevos con respecto al algoritmo CFA y Fuzzy sobre el que nos hemos basado son los siguientes: crear una partición fuzzy de los datos de entrada, modificación del calculo de la distancia para el calculo de los centros en cada iteración y la introducción del paso de migración.

4.1 CALCULO DE LA DISTANCIA

La modificación en el calculo de la distancia se ha trabajado en [6], aunque no se han tenido en cuenta las características concretas de nuestro problema. En el algoritmo Fuzzy tradicional, la distancia entre el centro de un cluster y un vector de entrada viene definida como la distancia euclídea:

$$d_{k,j} = \|\vec{x}^k - \vec{c}^j\|^2 \quad (9)$$

En esta distancia solo se tiene en cuenta las posiciones en el espacio de los dos puntos, de modo que la matriz de pertenencia U tendrá unos valores menores para los vectores de entrada que se encuentre muy alejados de un centro. Debido a las características de nuestro problema, esto no tiene por que ser siempre cierto, ya que dado un prototipo, este puede contener a muchos vectores de entrada con la misma salida, aunque estos se encuentren muy alejados del prototipo. Para tener esto en cuenta, hemos utilizado el parámetro w (formula 3) utilizado en el algoritmo CFA, el cual, mide la diferencia entre la salida esperada de un centro de un cluster y la salida de un vector de entrada. Gracias a este parámetro podremos modificar la distancia euclídea de forma que se premie que la diferencia entre la salida de un centro de un cluster y un vector de entrada sea pequeña. Conforme w se hace mas pequeño, la distancia disminuye haciendo que ese vec-

tor pertenezca en mayor medida a ese prototipo. El calculo de la distancia se realiza según la siguiente formula:

$$d_{k,j} = \|\vec{x}^k - \vec{c}^j\|^2 \cdot w(k,j) \tag{10}$$

4.2 MIGRACIÓN

La segunda modificación que se ha realizado sobre el algoritmo Fuzzy es la incorporación de un paso de migración tras el calculo de los nuevos centros de los clusters mediante el proceso iterativo.

La migración se lleva a cabo de una forma similar a como lo hace el CFA, aunque también hay algunos cambios al respecto. El esquema que sigue la migración es el siguiente:

1. Calcular la distorsión actual de la partición y la utilidad de cada prototipo.
2. **Mientras** (haya algún prototipo con utilidad $<$ 1 no procesado)
   (a) Elegir a un prototipo con utilidad $>$ 1 (Selección de Baker)
   (b) Elegir al prototipo con menor utilidad no procesado hasta ahora
   (c) Calculamos el menor paralelepípedo que contenga a los vectores de entrada pertenecientes al centro con utilidad $>$ 1 seleccionado
   (d) Colocamos los centros a igual distancia desde el centro de la diagonal y reparticionamos utilizando fuzzy-cmeans
   (e) Calculamos la nueva distorsión
   (f) Si la distorsión anterior es menor, deshacemos el cambio

En el paso 2.c tenemos que establecer un umbral para decidir que vectores de entrada pertenecen realmente a ese cluster, en nuestro caso utilizamos la media de los valores de la matriz U para ese centro en concreto.

La principal modificación con respecto al CFA reside en el cálculo de la distorsión de los prototipos. En el CFA se consideran siempre a todos los prototipos para ser destino u origen de la migración, en nuestro caso no, haremos una selección de prototipos que no tienen por que ser considerados en el proceso de migración no afectando tampoco al calculo de la distorsión.

El cálculo de la distorsión, según la cual efectuaremos la migración o no, se realiza de la siguiente forma:

- Acumular el error entre la salida del prototipo y la de los vectores de entrada pertenecientes a este cluster (establecemos un umbral de pertenencia igual a la media de U)
- Contar el numero de prototipos que pertenecen a este cluster
- Seleccionar a los prototipos que tengan un error por encima de la media de error por prototipo y seleccionar a los prototipos a los que les pertenezcan un numero de vectores de entrada por debajo de la media de vectores por prototipo
- Asignarle a estos clusters la distorsión: error $\times$ numero de vectores

Al hacer esta selección estamos excluyendo a los centros que poseen un número muy grande de vectores de entrada y tienen un error pequeño respecto a estos.

5 RESULTADOS EXPERIMENTALES

Para compara los resultados de los distintos algoritmos se usará el error cuadrático medio normalizado (NRMSE) y se define como:

$$\mathbf{NRMSE} = \sqrt{\frac{\sum_{k=1}^{n} \left(F(\vec{x}^k) - \mathcal{F}\left(\vec{x}^k; C, R, \Omega\right)\right)^2}{\sum_{k=1}^{n} \left(F(\vec{x}^k) - \bar{F}\right)^2}} \tag{11}$$

donde $\bar{F}$ es la media de las salidas producidas por la función objetivo para todos los vectores de entrada.

Para el calculo de los radios de las RBF hemos utilizado la técnica de los k-vecinos con k=1.

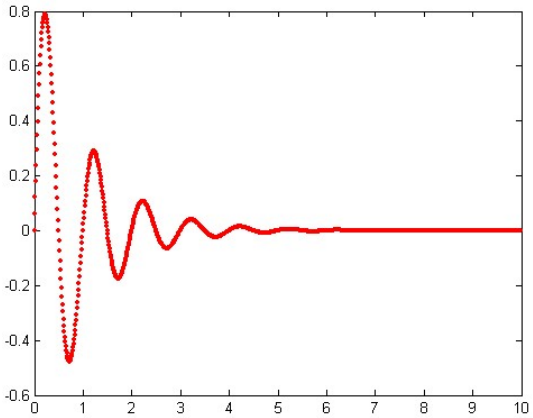


Figura 1: Función objetivo f_1 para ser aproximada

En la tabla 1 podemos ver los resultados obtenidos por el algoritmo CFA en comparación con el FCFA, así

como los resultados obtenidos aplicando el algoritmo Fuzzy C-means, para aproximar la función f_1(Figura 1).

Tabla 1: Error de aproximación para la función f_1.

Clusters	**Fuzzy C-Means**	**CFA**	**FCFA**
5	0.988(0.001)	0.928(0.001)	0.906(0.001)
6	0.985(0.002)	0.813(0.128)	0.696(0.000)
7	0.983(0.002)	0.781(0.052)	0.359(0.005)
8	0.980(0.001)	0.537(0.007)	0.351(0.007)
9	0.978(0.002)	0.514(0.070)	0.343(0.003)
10	0.976(0.003)	0.509(0.113)	0.339(0.000)

Una vez obtenidos estos resultados de los algoritmos, se les ha aplicado un método de búsqueda local $Levenberg-Marquardt$ para realizar una explotación del espacio solución, en la tabla 2 podemos ver los resultados obtenidos, apreciando una sensible mejora con respecto a los primeros.

Tabla 2: Error de aproximación tras el procedimiento de minimización local para la función f_1.

Clusters	**Fuzzy C-Means**	**CFA**	**FCFA**
5	0.812(0.205)	0.081(0.028)	0.103(0.069)
6	0.783(0.341)	0.090(0.008)	0.103(0.060)
7	0.284(0.115)	0.081(0.016)	0.069(0.017)
8	0.150(0.163)	0.053(0.028)	0.048(0.031)
9	0.300(0.160)	0.056(0.027)	0.027(0.024)
10	0.285(0.334)	0.047 (0.015)	0.011(0.015)

A la vista de estos resultados podemos apreciar como, además de situar los prototipos de los clusters con un error menor en una primera etapa, los sitúa de tal forma que al aplicar la minimización local también obtenemos mejores resultados que los otros algoritmos.

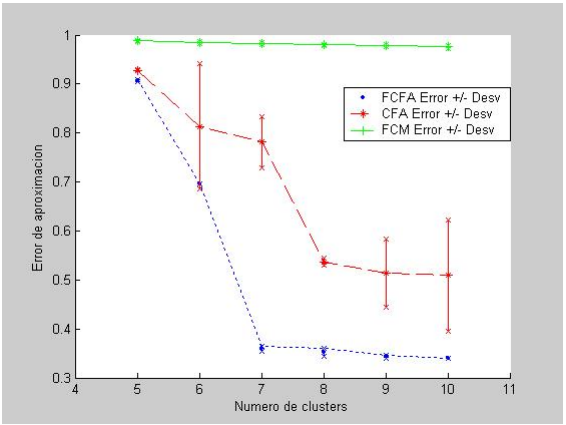


Figura 2: Error y desviación típica de los algoritmos CFA y FCFA para aproximar la función f_1

Como podemos ver en la figura 3, los resultados que obtenemos son muy buenos debido a que los prototipos se colocan desde el primer momento en zonas del espacio con una variabilidad alta en la salida. También podemos apreciar el valor de la salida esperada para cada prototipo viendo como se corresponde en gran medida con los vectores de entrada que están cerca de él y que por tanto le pertenecen en mayor medida.

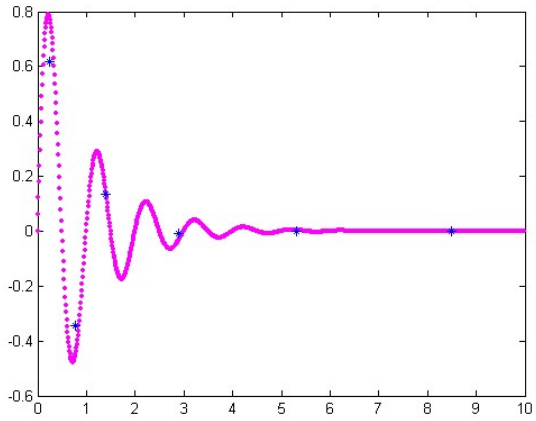


Figura 3: Situación de los prototipos para aproximar la función f_1

Para dos dimensiones, hemos realizado el mismo experimento utilizando la función que se muestra en la figura 4. Podemos ver como los resultados del FCFA superan a los del CFA, tanto en una primera solución (tabla 3) como tras la optimización mediante la búsqueda local (tabla 4).

Tabla 3: Error de aproximación para la función f_2.

Clusters	**Fuzzy C-Means**	**CFA**	**FCFA**
4	0.790(0.034)	0.806(0.026)	0.441(0.092)
5	0.710(0.021)	0.777(0.044)	0.327(0.038)
6	0.698(0.045)	0.682(0.053)	0.369(0.081)

Tabla 4: Error de aproximación para la función f_2 tras optimización.

Clusters	**Fuzzy C-Means**	**CFA**	**FCFA**
4	0.343(0.015)	0.021(0.019)	0.025(0.021)
5	0.188(0.024)	0.031(0.029)	0.005(0.003)
6	0.022(0.043)	0.029(0.052)	0.003(0.003)

6 CONCLUSIONES

En este árticulo hemos visto como podemos utilizar RBFs para resolver problemas de aproximación funcional. Dentro del diseo de las redes de RBFs, el paso

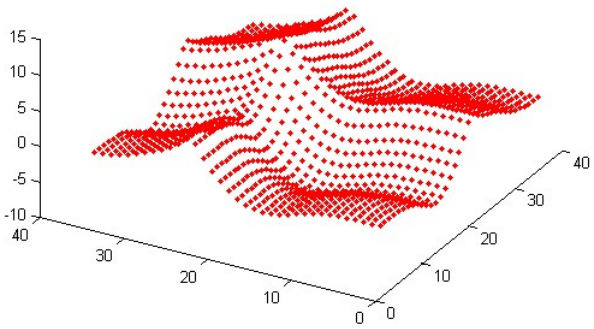


Figura 4: Función objetivo f_2

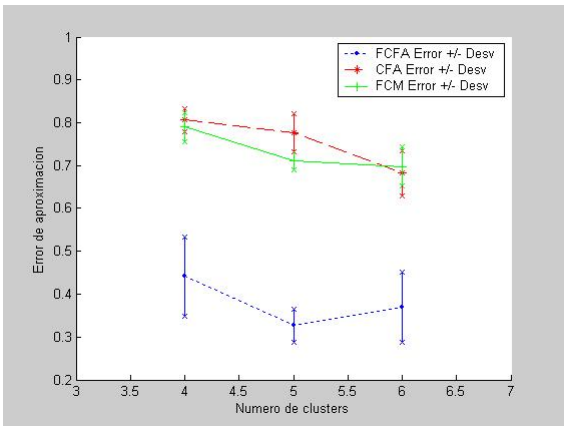


Figura 5: Error y desviación típica de los algoritmos CFA y FCFA para aproximar la función f_2

de inicialización de los centros se realiza utilizando algoritmos de clustering, pero estos, no se adaptan a las caracterśticas del problema haciendo que los resultados al aproximar las funciones, no sean lo suficientemente buenos.

En este contexto, se diseño el algoritmo CFA, que si tenia en cuenta que estamos intentando aproximar una funciń. El CFA tenia un gran defecto, la partición de los datos se hacia de tal forma que no podíamos establecer un grado de pertenencia con los vectores de entrada, además de que el proceso de migración tenia algunos aspectos susceptibles de ser mejorados. Nuestro nuevo algoritmo, el FCFA, resuelve tanto el problema de particionar el conjunto de vectores de entrada como el de la migración, obteniendo unos resultados mucho mejores que su predecesor, el CFA.

Del análisis de los resultados podemos deducimos las siguientes conclusiones:

1. La realización de un paso previo para inicializar los centros de las RBF es crucial para obtener buenos resultados a la hora de aproximar una función mediante RBFs.

2. Al comparar nuestro algoritmo con su predecesor, el CFA, hemos apreciado como efectivamente el FCFA nos proporciona mejores resultados mediante la utilización de la lógica difusa y de la migración.

3. La aplicacin de una búsqueda local tras la ejecución del algoritmo mejora significativamente los resultados.

Agradecimientos

Este trabajo ha sido parcialmente financiado gracias al proyecto español CICYT DPI2001-3219.

Referencias

[1] A. Baraldi and P. Blonda. A Survey of Fuzzy Clustering Algorithms for Pattern Recognition – Part I. *IEEETransSMC, Part B*, 29(6):786–801, Diciembre 1999.

[2] A. Baraldi and P. Blonda. A Survey of Fuzzy Clustering Algorithms for Pattern Recognition – Part II. *IEEETransSMC, Part B*, 29(6):786–801, Diciembre 1999.

[3] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, Nueva York, 1981.

[4] A. Gersho. Asymptotically Optimal Block Quantization. *IEEE Transanctions on Information Theory*, 25(4):373–380, Julio 1979.

[5] J. Gonzlez, I. Rojas, H. Pomares, J. Ortega, and A. Prieto. A new Clustering Technique for Function Aproximation. *IEEE Transactions on Neural Networks*, 13(1):132–142, Enero 2002.

[6] R. J. Hathaway, J. C. Bezdek, and Y. Hu. Generalized Fuzzy c-Means Clustering Strategies using Lp Norm Distances. *IEEE Transactions on Fuzzy Systems*, 8(5):576–582, Octubre 2000.

[7] W. Pedrycz. Conditional Fuzzy Clustering in the Design of Radial Basis Function Neural Networks. *IEEE Transactions on Neural Networks*, 9(4):601–612, Julio 1998.

[8] Z. Uykan, C. Gzelis, M. E. Celebei, and H. N. Koivo. Analysis of Input-Output Clustering for Determining Centers of RBFN. *IEEE Transactions on Neural Networks*, 11(4):851–858, Julio 2000.

EVALUATION OF FUZZY TIME SERIES BY NN WITH SCL TECHNIQUES

Dušan Marček
University of Zilina,, Faculty of Management Science and Informatics
Moyzesova 20, 01026 Zilina, Slovakia
marcek@fria.utc.sk

Summary

Traditionally, fundamental Box-Jenkins analysis [2] and ARCH-GARCH techniques have been the mainstream methodologies used to develop time series models in finance. A dynamical process with linguistic values of the actual inflation observations is studied. To generate fuzzy rules, neural networks with SCL-based product-space clustering is used. Finally, we present our preliminary results and some further experiments that we have performed.

Keywords: Neural Network, Classical and Fuzzy Time Series, SCL Algorithm.

1 INTRODUCTION

Fuzzy economic time series are difficult to estimate,. because the relationships among their variables may be complicated. In particular, the value of each variable may depend on the values taken by many others in several previous time periods. Very often it is difficult to express exactly these dependencies, or there is not known hypothesis for that. Very frequently, in such cases more sofisticated approaches are considered. These approaches are based on the human experience knowledge and consist of series linguistic expressions each of which takes the form of an ´ if ... then ...´ fuzzy rule, and they are well known under the common name fuzzy controllers. But also, an expert is usually unable linguistically describe the behaviour of economic processes in particular situations. Hence, most recent researches in the fuzzy controllers design for deriving of linguistically interpreted fuzzy rules have been centred on developing automatic methods to build these fuzzy rules using a set of numerical input-output data. For applying of these methods it is supposed that a database describing previous input-output behaviour of a system is available [13]. Majority of these models and data-driven techniques rely on the use Takagi-Sugeno type controllers and fuzzy/non-fuzzy neural networks [5], [7-9], [18], [23], clustering/fuzzy-clustering and genetic algorithm approaches [3], [4], [6], [10], [11], [22], [24].

The goal of this paper is to illustrate that two distinct areas, i.e. fuzzy sets theory and computational networks, may be used to economic time series modelling. This approach goes under the term ´soft computing´ as a synergy of methodologies useful to solve problems using some form of intelligence that differ from traditional computing. Statistical and fuzzy time series models are presented in Section 2. Direct determination of the fuzzy rules based on the SCL (Supervised Competitive Learning) technique is described in Section 3. Concluding remarks are offered in Section 4. We assume that readers are acquainted with fundamental principles and methods of econometrics, statistical and fuzzy logic inference.

2 TIME SERIES AND FUZZY TIME SERIES MODELLING

To illustrate modelling approaches used in conventional and fuzzy time series, we consider the time readings of the inflation (the 514 of monthly inflation observations in the U. S. from February 1956 to November 1998) was published at http://neatideas.com/data/inflatdata.htm. Fig. 1 illustrates the time plot of this time series. This time series shows no apparent trend or periodic structure. To build a forecast model the sample period for analysis y_1, ..., y_{344} was defined, i.e. the period over which the forecasting model can be developed and estimated, and the ex post forecast period (validation data set), y_{345}, ..., y_{514} as the time period from the first observation after

the end of the sample period to the most recent observation.

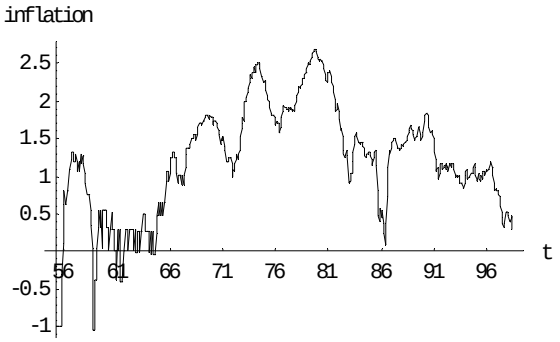


Figure 1: Natural logarithm of monthly inflation from February 1956 to November 1998.

Input selection and data preprocessing are of crucial importance to the development of time series models. Potencial inputs (independent variables) were chosen based on traditional statistical tools. These include the autocorrelation function (ACF), the partial autocorrelation function (PACF) and the Akaike Information Criterion (AIC) [2]. Following this, at the starting point, we have formulated a model relating the value y_t of the series at time t that depends only on its previous value y_{t-1} and on the random disturbance ε_t, i.e.

$$y_t = \xi + \phi_1 y_{t-1} + \varepsilon_t \quad (1)$$

where the variable y_t (in our case the first difference of inflation rate) is explaned by only its previous value, Using Levinson-Durbin algorithm [2], [5] the model (1) is statistically fitted as

$$\hat{y}_t = -0{,}1248 y_{t-1} \quad (2)$$

In contrast to the conventional time series, the observations of fuzzy time series are fuzzy sets (the observations of conventional time series are real numbers), the universes of discourse for the fuzzy sets are subsets of $\Re^1$, where $\Re^1$ is the set of real numbers, either naturally or artificially defined. Song and Chisson [6] give a thorough treatment of these models. Let X_t, (t = ..., 1, 2, ...), a subset of $\Re^1$, be the universe of discourse on which fuzzy sets y_t^i, (i = 1, 2, ...) are defined and Y_t is the collection of y_t^i, (i = 1, 2, ...). Then Y_t, (t = ..., 1, 2, ...) is called a fuzzy time series on X_t, (t = ..., 1, 2, ...).

The fuzzy time series modelling procedure consists of an implementation of several steps, usually as follows:

1. Define the input-output variables and the universes of discourse.
2. Define (collect) linguistic values and fuzzy sets on the universes of discourse.
3. Define (find) fuzzy relations (fuzzy rules).
4. Apply the input to the model and compute the output.
5. Defuzzify the output of the model.

Firstly, in the fuzzification process, we specified input and output variables. The input variable x_{t-1} is the lagged first difference of inflation values $\{y_t\}$ and is calculated as $x_{t-1} = y_{t-1} - y_{t-2}$, t = 3, 4, ... The output variable x_t is the first difference of inflation values $\{y_t\}$ and is calculated as $x_t = y_t - y_{t-1}$, t = 2, 3, ... The variable ranges are as follows:

$-0{,}75 \le x_t,\ x_{t-1} \le 0{,}75$

These ranges define the universe of discourse within which the data of x_{t-1} and x_t are, and on which the fuzzy sets have to be, specified. The universes of discourse were partioned into the seven intervals.

Next, we specified the fuzzy-set values of the input and output fuzzy variables. The fuzzy sets numerically represented linguistic terms. Each fuzzy variable assumed seven fuzzy-set values as follows: NL: Negative Large, NM: Negative Medium, NS: Negative Small, Z: Zero, PS: Positive Small, PM: Positive Medium, PL: Positive Large.

Fuzzy sets contain elements with degrees of membership. Fuzzy membership functions can have different shapes. The triangular membership functions were chosen. Fig. 2 shows membership function graphs of the fuzzy sets above.

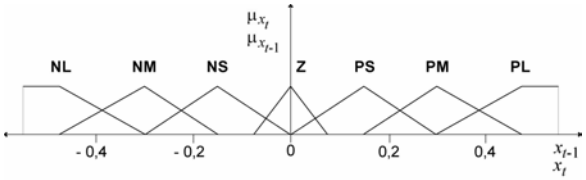


Figure 2: The membership functions of fuzzy variables x_{t-1} and x_t.

The input and output spaces were partioned into the seven disjoint fuzzy sets. From membership function graphs μ_{t-1}, μ_t in Fig. 2 is shown that the seven intervals [-0,75; -0,375], [-0,375; -0,225], [-0,225 -0,075], [-0,075; 0,075], [0,075; 0,225], [0,225; 0,375], [0,375; 0,75] correspond respectively to NL, NM, NS, Z, PS, PM, PL.

Next, we specified the fuzzy rule base or the bank of fuzzy relations. The appendix describes the neural network (see Fig. 3) which uses the SCL to derive fuzzy rules from data. As shown in Fig. 4(b) the bank contains the 5 fuzzy rules. For example the fuzzy rule of the 34th block corresponds to the following fuzzy relation

$$\text{IF } x_{t-1}^i = \text{PM THEN } x_t^j = \text{PS} \quad (3)$$

Finally, we determined the output action given the input conditions. We used the Mamdani´s implication [4]. Each

fuzzy rule produces the output fuzzy set clipped at the degree of membership determined by the input condition and the fuzzy rule. When the input value, say $x_{t-1}^{i} = x_{344}^{i}$, is applied to the time series model

$$x_t^j = x_{t-1}^i \circ R_{ij}(t, t-1), \quad (4)$$

where $x_t^j \in X_t$, $x_{t-1}^i \in X_{t-1}$, $i \in I$, $j \in J$, I and J are indices sets for X_t and X_{t-1} respectively, " ∘ " is the sign for the *max-min* composition, $R_{ij}(t, t-1)$ is the fuzzy relation among the observations at t and t -1 times, the output value $x_t^j = x_{345}^j$ can be calculated. Since the Eq. (4) is equivalent to the linguistic conditional statement

$$\text{"if } x_{t-1}^i \text{ then } x_t^j \text{"}, \quad (5)$$

we have $R_{ij}(t, t-1) = x_{t-1}^i \times x_t^j$, where "×" is the Cartesian product.

It is possible to compute the output fuzzy value x_t^j by the following simple procedure consisting of three steps:

- Compute the membership function values $\mu_{NL}(x_{t-1})$, $\mu_{NM}(x_{t-1})$, ..., $\mu_{PL}(x_{t-1})$ for the input x_{t-1} using the membership functions pictured in Fig. 2.
- Substitute the computed membership function values in fuzzy relations (4), (5).
- Apply the *max-min* composition to obtain the resulting value x_t^j of fuzzy relations.

Following the above principles, we have obtained the predicted fuzzy value for the inflation $x_t = x_{345}^j$ = 0,74933.

To obtain a simple numerical value in the output universe of discourse, a conversion of the fuzzy output is needed. This step is called defuzzification. The simplest defuzzification scheme seeks for the value $\hat{x}_t$ that is of middle membership in the output fuzzy set. Hence, this defuzzification method is called the Middle of Maxima, abbreviated MOM. Following this method, we have obtained the predicted value for the $\hat{x}_{345}$ = - 0,15 . The remaining forecasts for ex post forecast period t = 346, 347, ... may be generated similarly.

3 GENERATING FUZZY RULES WITH PRODUCT-SPACE CLUSTERING

The neural network pictured in Fig. 3 was used to generate structured knowledge of the form „if A, then B" from a set of numerical input-output data. In Section 2 we defined cell edges with the seven intervals of the fuzzy-set values in Fig. 2. The interval - 0,75 ≤ x_t, x_{t-1} ≤ 0,75 was partitioned into seven nonuniform subintervals that represented the seven fuzzy-set values NL, NM, NS, Z, PS, PM, and PL assumed by fuzzy variables x_{t-1} and x_t. The Cartesian product of these subsets defines $7 \times 7 = 49$ fuzzy cells in the input-output product space $\Re^2$.

We can represent all possible fuzzy rules as 7-by-7 linguistic matrix (see Fig. 4). The idea is to categorise a given set or distribution of input vectors $\mathbf{x}_t = (x_{t-1}, x_t)$, t = 1, 2, ..., 344 into $7 \times 7 = 49$ classes, and then represent any vector just by the class into which it falls.

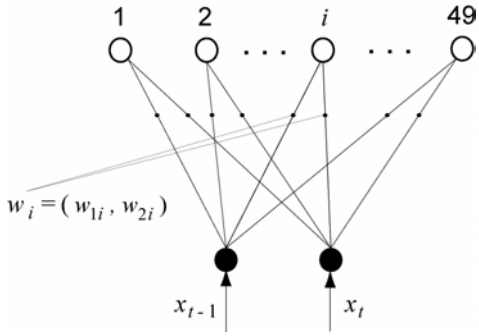


Figure 3: The topology of the network for fuzzy rules generating by SCL-based product-space clustering.

For each random input sample $\mathbf{x}_t = (x_{t-1}, x_t)$, the wining vector $\mathbf{w}_{i'} = (w_{1i'}, w_{2i'})$ was updated by the SCL algorithm according to

$$\left.\begin{aligned} \tilde{w}_{1i'} &\leftarrow \tilde{w}_{1i} + \eta\,(\tilde{x}_{1t} - \tilde{w}_{1i}) \\ \tilde{w}_{2i'} &\leftarrow \tilde{w}_{2i} + \eta\,(\tilde{x}_{2t} - \tilde{w}_{2i}) \end{aligned}\right\} \quad if\ i = i'$$

$$\left.\begin{aligned} \tilde{w}_{1i'} &\leftarrow \tilde{w}_{1i} - \eta\,(\tilde{x}_{1t} - \tilde{w}_{1i}) \\ \tilde{w}_{2i'} &\leftarrow \tilde{w}_{2i} - \eta\,(\tilde{x}_{2t} - \tilde{w}_{2i}) \end{aligned}\right\} \quad if\ i \neq i'$$

where i' is the winning unit defined $\left\|\tilde{\mathbf{w}}_{i'} - \tilde{\mathbf{x}}_t\right\| \leq \left\|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_t\right\|$ for all i, and where $\tilde{\mathbf{w}}_i$ and $\tilde{\mathbf{x}}_t$ is a normalized version of $\mathbf{w}_i$ and $\mathbf{x}_t$ respectively, η is the learning rate.

Supervised Competitive learning (SCL)-based product-space clustering classified each of the 344 input-output data vectors into 9 of the 49 cells as shown in Fig. 4(a). Fig. 4(b) shows the fuzzy rule bank. We added a rule to the rule bank if the count of input-output vectors in particular cells was larger than the value 0,05N, where N = 344 is number of data pairs (x_{t-1}, x_t), t = 1, 2, ..., N in the input and output series. For example the most frequent rule represents the cell 34. From most to least important (frequent) the fuzzy rules are (PM; PS), (PS; PL), (NL; NS), (PS; PL), and (PS; PS).

4 CONCLUSION

In this paper, we have presented an application of the fuzzy time series model to forecast an autoregressive

process. A formal framework for the definition of fuzzy

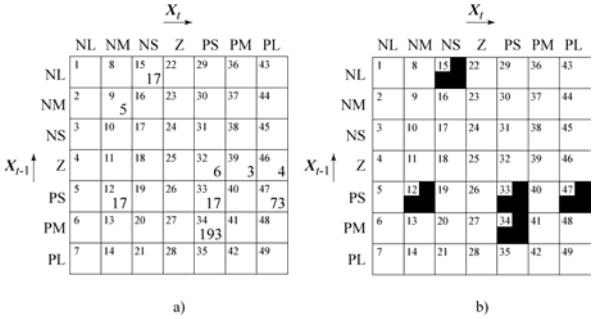


Figure 4: Distribution of input-output data (x_{t-1}, x_t) in the input-output product space $X_{t-1} \times X_t$ (a). Bank of fuzzy rules of the time series modelling system (b).

rules has been presented. This framework is based on the simple competitive learning of networks. The neural network with the SCL clustering technique was used to determine the fuzzy relation (fuzzy rules) of first-order fuzzy time series models directly from data. The proposed method is also suitable for parameter estimations of econometric models, in applications of deterministic non-linear dynamics and chaos theory in contemporary economics and finance.

As a final point, let us examine what has been gained by use of a fuzzy time series model over an ordinary AR(1) model for the output x_{345}. For this purpose, we have computed prediction limits on the one-step-ahead forecast from the AR(1) model, and fuzzy time series model. The 95 percent interval around the actual inflation value based on the statistical theory is $(-0{,}0442; 0{,}05043)$, An intuitive method for constructing confidence intervals for fuzzy time series model is simply the defuzzification method First of Maxima and First of Minima to obtain prediction limits on the one-step-ahead forecast. In our example, the „confidence“ interval for fuzzy time series value $\hat{x}_{345}$ = 0,00312 is (-0,30256 to 0,3088). These intervals are very large. The actual value for the AR(1) model fell not within the forecast interval, and moreover, its sign is opposite to the forecast value sign.

The method may be of real usefulness in practical applications where usually the expert can not explain linguistically, what control actions the process takes or there is no knowledge of the process.

Because the results were based of chosen inflation rates and data set, they were difficult to generalize in others situations. Yet, the results certainly provide a rational way for improvement of forecasting ability in chaotic economic systems.

Acknowledgement

This work is supported by Slovak grant foundation under the grant No. 1/9183/02.

References

[1] Box, G. E. P., and G. M. Jenkins: Time Series Analysis, Forecasting and Control, Holden-Day, San Francisco, CA, 1970

[2] Brockwell, P.J. and Davis, R.A.: Time Series: Theory and Methods. Springer-Verlag, New York, 1987

[3] Kosko, B.: Neural networks and fuzzy systems - a dynamical systems approach to machine intelligence. Prentice-Hall Int., Inc. 1992

[4] Mamdani, E., H.: Application of a fuzzy logic to approximate reasoning using linguistic synthesis. IEEE Trans. Comput. 26 (1977), pp. 1182-1191

[5] Morettin, P., A.: The Levinson algorithm and its applications in time series analysis. International Statistical Review, 52, 1984, pp. 83-92

[6] Song, Q., Chisson, B., S.: Fuzzy time series and its models. Fuzzy Sets and Systems, 54, 1993, pp. 269-277

[7] Sugeno, M., and Yasukawa, T.: A fuzzy-logic-based approach to quantitative modelling. IEEE Trans. Fuzzy System. Vol. 1, No. 1, Feb. 1993, pp. 7-31

INDUCCIÓN DIRECTA DE MODELOS DIFUSOS TEMPORALES

Juan Moreno-Garcia
Universidad de Castilla-La Mancha
Departamento de Informática
Avda Carlos III s/n, Toledo
Juan.Moreno@uclm.es

Carlos González, Ramón Manjavacas
Universidad de Castilla-La Mancha
Departamento de Informática
Paseo de la Universidad nº4, Ciudad Real
{Carlos.Gonzalez,Ramon.Manjavacas}@uclm.es

Resumen

Los modelos difusos temporales (MDTs) se utilizan para representar de una manera lingüística los sistemas dinámicos (SDs). Están compuestos por reglas difusas temporales (RDTs) que están ordenadas. Cada RDT representa una zona temporal. Los MDTs se obtienen a partir de un modelo difuso inducido por otro algoritmo de inducción. Primeramente, se utiliza un algoritmo de inducción para obtener un modelo difuso, y a continuación, se transforma el modelo difuso obtenido en un MDT utilizando un algoritmo de transformación [6]. El objetivo de este trabajo es presentar un algoritmo directo de inducción de MDTs, es decir, un método que no necesita un MDT como entrada.

Palabras Clave: Inducción de modelos, modelado temporal, modelado lingüístico.

1 INTRODUCCIÓN

Un SD puede ser definido como un sistema que cambia en el tiempo. En un SD existen un conjunto de relaciones entre las variables de entrada y salida que representan las modificaciones de las variables de salida cuando las variables de entrada son modificadas. En los SDs, también se necesita conocer la evolución temporal de estas variables porque los valores de las variables en un tiempo dependen de los valores de las variables en los instantes anteriores. Los SDs con magnitudes físicas continuas verifican que la evolución de las variables es continua en el tiempo, esto es, en el instante $t+1$ el valor de la variable v_{t+1} es muy parecido al valor de la variable v_t en el instante t. Esta hipótesis se supone cuando se definen los MDTs.

Los SDs se modelan utilizando diferentes técnicas tradicionales [8, 5]. Se modelan para explicar y predecir su comportamiento, y para analizar sus propiedades dinámicas. Se pueden modelar utilizando de la lógica difusa [7, 1, 2, 3]. Los MDTs se utilizan para representar la evolución del SD. Esta estructura simula el comportamiento de un sistema en función de la entrada. Informalmente, los MDTs son una secuencia ordenada de reglas difusas, donde cada regla representa una zona temporal.

Una diferencia entre nuestro método y otros es que nosotros modelamos los SDs mediante sistemas ordenados de reglas, de esta forma, cada regla representa instantes consecutivos con la misma etiqueta de salida. Utilizando este método es posible detectar que variables cambian de una regla a otra, es decir, de un instante o otro. Otra diferencia es que utilizamos etiquetas lingüísticas [11] a lo largo de todo el proceso de inducción, así, los modelos obtenidos son más descriptivos. La última diferencia es que nuestro método agrupa utilizando la variable de salida tomando los ejemplos consecutivos que tienen la misma salida. En [6] se presenta un algoritmo para transformar un modelo difuso en un MDT. Este método es interesante cuando se dispone de un modelo difuso que es una buena aproximación al sistema que representa. El objetivo de este trabajo es presentar un método de inducción de MDT que no necesite otro modelo difuso como entrada.

En la sección 2 se presentan algunas consideraciones previas. Después, se expone el método de inducción de los MDTs. En la sección 4 se utiliza el algoritmo en un ejemplo, y se finaliza con las conclusiones.

2 CONSIDERACIONES PREVIAS

Antes de estudiar en detalle como se obtiene un MDT, se mostrará la estructura de nuestro método. Se utiliza el algoritmo 2 que tiene dos entradas: un conjunto de ejemplos E y $m+1$ conjuntos ordenados de etiquetas

lingüísticas $SA_1 \dots SA_m, SC$.

En las subsecciones 2.1 y 2.2 se ofrece una descripción detallada de las dos entradas del algoritmo. La sección 2.3 define los MDTs.

2.1 CONJUNTO DE EJEMPLOS E

Una serie de tiempo es una secuencia de datos que es ordenada en el tiempo [9]. Un SD se puede representar por medio de una serie temporal. Cada ejemplo de la serie contiene los valores de las variables en un instante. Esta sección muestra formalmente la estructura de la serie de tiempo que se representa mediante un conjunto de ejemplos E.

Cada ejemplo de E tiene un conjunto de m variables reales de entrada definidas en el dominio $X = X_1 \times \dots \times X_m \subset R^m$, y una variable real de salida $S \subset R$. $E = \{e_1 \dots e_n\}$ siendo $e_i = (x_1^i \dots x_m^i, s^i, t_i)$, donde $x_j^i \in X_j$, $s^i \in S$ y t_i es el tiempo en que ocurre e_i.

2.2 CTOS ORDENADOS DE ETIQUETAS

Los MDTs trabajan con variables lingüísticas que pueden tomar valores de un conjunto ordenado de etiquetas lingüísticas con algunas restricciones sobre su dominio. Estas variables se denominan *variables lingüísticas continuas* (desde ahora variables). Las etiquetas lingüísticas (desde ahora etiquetas) de las variables se definen antes de obtener el MDT.

Se define un conjunto ordenado de etiquetas para cada variable de entrada X_j. Su estructura es $SA_j = \{SA_j^1 \dots SA_j^{i_j}\}$. Para la variable de salida S se define un conjunto ordenado de etiquetas $SC = \{SC^1 \dots SC^{i_y}\}$. En estos conjuntos el superíndice i es la posición de la etiqueta, $i_j = |SA_j|$ e $i_y = |SC|$.

El orden de las etiquetas en los conjuntos de etiquetas se basa en el método de defuzzification llamado *media de máximos* (MOM) [4]. El MOM de una etiqueta SA_j^a lo representamos como $MOM(SA_j^a)$. Una etiqueta SA_j^a es menor que SA_j^b si ambas están definidas para X_j y el MOM de SA_j^a es menor que el MOM de SA_j^b. De igual forma se puede establecer cuando una etiqueta es menor o igual que otra.

X_j es una variable lingüística continua si tiene asociada una semántica a cada SA_j^i definida sobre ella que verifica: (1) Cada etiqueta SA_j^i se define para un dominio ordenado; (2) Las etiquetas definidas sobre X_j están ordenadas mediante el MOM; (3) La suma del grado de pertenencia de un valor x a todas las etiquetas definidas en SA_j debe ser 1 ($\sum_{SA_j^i \in SA_j} \mu_{SA_j^i}(x) = 1$); (4) Cada x_j en el dominio de X_j tiene 1 o 2 etiquetas con un grado de pertenencia mayor que 0. Así, una de estas expresiones se verifica: ($\mu_{SA_j^i}(x_j) = 1$) ó ($\mu_{SA_j^i}(x_j) = a$ y $\mu_{SA_j^{i+1}}(x_j) = 1 - a$). Si el grado de pertenencia de x_j a SA_j^i se incrementa o decrementa entonces el grado de pertenencia de x_j a SA_j^{i+1} se decrementará o incrementará en el mismo grado.

2.3 MODELOS DIFUSOS TEMPORALES

Los SDs estudiados tienen una variable de salida. Nuestro método de inducción agrupa utilizando la variable de salida, así, todos los ejemplos consecutivos con la misma etiqueta de salida (la etiqueta con más grado de pertenencia al valor de salida s^i del ejemplos e_i) se representan con la misma regla.

Un MDT está compuesto por **Reglas Difusas Temporales** ($RDTs$) que tienen dos componentes: un antecedente y una etiqueta de salida. El antecedente es una conjunción de disyunciones de expresiones X_j *is* SA_j^i. El antecedente tiene asociado un conjunto de ejemplos consecutivos $E_{p,u}$. Un **conjunto de ejemplos consecutivos** $E_{p,u}$ es un conjunto de ejemplos que verifica: (1) $p \leq u$ y $E_{p,u} \subset E$; (2) $\forall k \in [p,u], \exists e_k \in E_{p,u}$; (3) $\forall e_a, e_b \in E_{p,u}, a \neq b$, donde E es el conjunto de ejemplos; p, u, k, a y b son los índices a los ejemplos de E.

Se utilizan los intervalos lingüísticos para representar las disyunciones, y el conjunto de m intervalos lingüísticos para representar las conjunciones de disyunciones de una RDT. **Un intervalo lingüístico de longitud** c es una secuencia de c etiquetas consecutivas definidas en SA_j que comienzan en la etiqueta p. Se representa como $LI_{j,p}^c = \{SA_j^p \dots SA_j^{p+(c-1)}\}$, donde $SA_j = \{SA_j^1 \dots SA_j^{i_j}\}$ es un conjunto ordenado de i_j etiquetas en X_j, p es la posición en SA_j de la primera etiqueta del intervalo lingüístico y c es el número de etiquetas del intervalo. Por ejemplo, el intervalo $LI_{j,3}^4$ representa la disyunción **(**$(x_j$ **is** $FN)$ **OR** $(x_j$ **is** $NR)$ **OR** $(x_j$ **is** $FP)$ **OR** $(x_j$ **is** $P)$**)** si se utiliza el conjunto de la figura 3.

Un **conjunto ordenado de** m **intervalos** SLI_m es una secuencia de m intervalos definida sobre m variables. Se representa como $SLI_m = \{LI_{1,p_1}^{c_1} \dots LI_{m,p_m}^{c_m}\}$ donde p_j es la posición en SA_j de la primera etiqueta del intervalo j y c_j es el número de etiquetas del intervalo j. Los SLI_m se utilizan para representar lingüísticamente el rango de valores de m variables de entrada. Por ejemplo, si para cada variable de entrada se utiliza un conjunto ordenado de 7 etiquetas representadas en la figura 3, entonces, $SLI_3 = \{LI_{1,5}^3, LI_{2,1}^2, LI_{3,2}^4\} = \{\{FP, P, VP\}, \{VN, N\}, \{N, FN, NR, FP\}\}$ representa la conjunción de disyunciones **(**$(x_1$ **is** $FP)$ **OR** $(x_1$ **is** $P)$ **OR** $(x_1$ **is** $VP)$**) AND (**$(x_2$ **is** $VN)$ **OR** $(x_2$ **is** $FN)$**) AND (**$(x_3$ **is** $N)$

OR (x_3 **is** FN) **OR** (x_3 **is** NR) **OR** (x_3 **is** FP)).

Así, una **regla difusa temporal** $R_{p,u}$ es una regla difusa que verifica: (1) Su antecedente es un SLI_m; (2) Su consecuente es una etiqueta $SC^w \in SC$; (3) Su SLI_m tiene asociado un conjunto de ejemplos consecutivos $E_{p,u}$. $E_{p,u}$ indica los instantes que una RDT representa. Luego una RDT $R_{p,u}$ es una regla difusa que representa los instantes del p al u en un SD. Se pueden comparar dos RDTs, así, $R_{k,l}$ está incluida en $R_{i,j}$ si el índice i es menor o igual que el índice k y el índice j es mayor o igual que l. Una regla $R_{i,j}$ es menor que $R_{k,l}$ si $R_{k,l}$ no está incluida en $R_{i,j}$ y $i < k$. De la misma forma se define cuando una RDT es igual a otra RDT, o cuando una RDT es mayor que otra. Utilizando estos conceptos se pueden ordenar las RDTs. Como puede verse, el orden del cada regla viene dado por el conjunto de ejemplos asociado a cada SLI_m.

Para finalizar esta sección, el algoritmo 1 ofrece el método de inferencia de los MDTs. Para comprender su comportamiento se estudiará la función de pertenencia de $LI^c_{j,p}$. La ecuación 1 define la función de pertenencia de un $LI^c_{j,p}$.

$$\mu_{LI^c_{j,p}}(a_j) = \sum_{SA^z_j \epsilon LI^c_{j,p}} \mu_{SA^z_j}(a_j) \qquad (1)$$

donde a_j es un valor discreto definido en el dominio de la variable continua X_j, $z \in [p..c-1]$.

La ecuación 2 define la función de pertenencia de un SLI_m.

$$\mu_{SLI_m}(e_i) = *(\mu_{LI^{c_j}_{j,p_j}}(x_j)) \qquad (2)$$

donde e_i es un ejemplo del conjunto E, $j\epsilon[1..m]$ y * es una t-norma.

Algoritmo 1 Método de Inferencia

```
j ← 1
for i = 1 to |E| do
  if μ_{SLI_m^j}(e_i) > μ_{SLI_m^{j+1}}(e_i) then
    s ← Cons(R_j)
  else
    s ← Cons(R_{j+1})
    j ← j + 1
  end if
end for
```

El algoritmo de inferencia necesita como entrada un conjunto de ejemplos. El ejemplo y regla actual se indican mediante los índices i y j respectivamente. El bucle *for* se utiliza para recorrer el conjunto de ejemplos E. Si el grado de pertenencia de e_i al SLI_m de R_j (SLI^j_m) es mayor que el grado de pertenencia de e_i al SLI_m de R_{j+1} (SLI^{j+1}_m) la salida es la etiqueta de salida de R_j ($s \leftarrow Cons(R_j)$), en otro caso, la salida es la etiqueta de salida de R_{j+1} ($s \leftarrow Cons(R_{j+1})$) y la nueva regla actual es R_{j+1} ($j \leftarrow j+1$). Se necesita un proceso de defuzzification [4, 10] para obtener un valor discreto de salida.

3 ALGORITMO DE INDUCCIÓN

Esta sección está dividida en dos subsecciones. La primera explica dos conceptos necesarios para comprender el algoritmo de inducción, y la segunda muestra el funcionamiento del mismo.

3.1 ALGUNAS DEFINICIONES

Primero se verá cómo obtener el SLI_m de una RDT que representa un conjunto de ejemplos consecutivos. Para hacer esto, se simplifica el SLI_m con todas las posibles etiquetas para cada uno de los m intervalos (todas las etiquetas de los conjuntos $SA_1 \dots SA_m$ respectivamente). El método de simplificación consiste en que se cambian la primera y última etiquetas para cada intervalo del SLI_m, luego el número de etiquetas se reduce.

La primera y última etiqueta (SA^f_j y SA^l_j) de cada intervalo $LI^{c_j}_{j,p_j}$ son la primera y última que verifican las ecuaciones 3 y 4 respectivamente.

$$\exists e_i \epsilon E_{p,u} / \mu_{SA^f_j}(x^i_j) > 0 \qquad (3)$$

$$\exists e_i \epsilon E_{p,u} / \mu_{SA^l_j}(x^i_j) > 0 \qquad (4)$$

donde x^i_j es el valor real en la posición j del ejemplo e_i.

Así, las nuevas primera y última etiquetas son la primera SA^f_j y última SA^l_j en $LI^{c_j}_{j,p_j}$ tal que uno o más ejemplos en E_{SLI_i} tienen un grado de pertenencia a SA^f_j y SA^l_j mayor que cero respectivamente. El intervalo simplificado $LI^{l-f+1}_{j,f}$ tiene $l-f+1$ etiquetas.

El segundo concepto es la definición de unión total de RDTs. Primeramente, se mostrarán las definiciones de *unión de intervalos* y *unión de conjuntos ordenados de intervalos* (definiciones 3.1 y 3.2 respectivamente), y finalmente, se estudiará la definición de unión total de RDTs.

Definición 3.1 *Sean n $LI^{c_{j_1}}_{j,p_{j_1}} \dots LI^{c_{j_n}}_{j,p_{j_n}}$ definidos sobre la variable X_j, su unión es el intervalo $LI^{u-p+1}_{j,p}$ donde: (1) La primera etiqueta SA^p_j de $LI^{u-p+1}_{j,p}$ es la menor de $LI^{c_{j_1}}_{j,p_{j_1}} \dots LI^{c_{j_n}}_{j,p_{j_n}}$; (2) La última etiqueta SA^u_j de $LI^{u-p+1}_{j,p}$ es la mayor de $LI^{c_{j_1}}_{j,p_{j_1}} \dots LI^{c_{j_n}}_{j,p_{j_n}}$; (3) El intervalo está compuesto por cada etiqueta desde SA^p_j a SA^u_j, esto es, $LI^{u-p+1}_{j,p} = \{SA^p_j \dots SA^u_j\}$.*

Definición 3.2 *Sean n* $SLI_m^1 \dots SLI_m^n$*, su unión se define como otro* SLI_m*, donde cada intervalo* $LI_{j,p_j}^{c_j}$ *de* SLI_m *se obtiene realizando la unión de los n intervalos* $LI_{j,p_{j_1}}^{c_{j_1}} \dots LI_{j,p_{j_n}}^{c_{j_n}}$ *de* $SLI_m^1 \dots SLI_m^n$ *(definición 3.1).*

Definición 3.3 *Sean n RDTs* $R_{p_1,u_1} \dots R_{p_n,u_n}$*, su unión se define como la regla* $R_{p,u}$ *donde: (1) Su* SLI_m *se obtiene realizando la unión de los n* $SLI_m^1 \dots SLI_m^n$ *de* $R_{p_1,u_1} \dots R_{p_n,u_n}$ *(definición 3.2); (2) El conjunto de ejemplos asociado es igual a* $E_{SLI_m} = \bigcup E_{SLI_m^j}$ *donde* $j \in [1..n]$*; (3) La etiqueta de salida se calcula utilizando la ecuación 5.*

3.2 INDUCCIÓN DE MDTs

El algoritmo 2, llamado Algoritmo Directo de Inducción de MDTs, se utiliza para inducir los MDTs. Sus entradas son: un conjunto de ejemplos E; los conjuntos $SA_1 \dots SA_m$ para las m variables de entrada; y el conjunto SC para la variable de salida.

La idea del algoritmo consiste en recorrer los ejemplos de E, para cada ejemplo se calcula la RDT que lo representa y si la RDT obtenida tiene la misma etiqueta de salida que la RDT que representa sus ejemplos anteriores entonces se realiza la unión total de las dos RDTs, en otro caso, se añade la RDT que representa los ejemplos anteriores al MDT.

En este algoritmo, cada RDT se representa como una tupla con tres componentes: $R_i = \{SLI_m^i, E^i, SC^a\}$, donde SLI_m^i representa el antecedente de R_i, E^i es el conjunto de ejemplos consecutivos asociado al SLI_m^i, y finalmente, $SC^a \in SC$ es la etiqueta de salida.

Algoritmo 2 Algoritmo de inducción directa de MDT

$MDT \leftarrow \theta$
$E^{cur} \leftarrow \{e_1\}$
$SLI_m^{cur} \leftarrow Simplify(SA_1 \dots SA_m)$
$SE^{cur} \leftarrow CalculateConsequent(E^{cur})$
for $i = 2$ to $|E|$ **do**
  $E^i \leftarrow \{e_i\}$
  $SLI_m^i \leftarrow Simplify(SA_1 \dots SA_m)$
  $SE^i \leftarrow CalculateConsequent(E^i)$
  if $SE^{cur} = SE^i$ **then**
    $R_{cur} \leftarrow TotalUnion(R_{cur}, R_i)$
    $E^{cur} \leftarrow E^{cur} \bigcup E^i$
  else
    $MDT \leftarrow MDT + R_{cur}$
    $R_{cur} \leftarrow R_i$
  end if
end for
$MDT \leftarrow MDT + R_{cur}$

A continuación, se estudia el comportamiento del algoritmo 2. En dicho algoritmo, MDT es el modelo difuso temporal. R_{cur} es una RDT, denominada *regla actual*, que representa los ejemplos anteriores a la RDT que representa el ejemplo e_i (R_i). Sus tres componentes son: SLI_m^{cur}, E^{cur} y SE^{cur}. R_i es la RDT que representa el ejemplo número i en E, sus tres componentes son SLI_m^i, E^i y SE^i.

Primeramente, el MDT se inicializa a θ ($MDT \leftarrow \theta$) y se crea la primera regla actual R_{cur} que representa a e_1, proceso que se realiza en tres acciones: (1) E^{cur} es inicializado a $\{e_1\}$ ($E^{cur} \leftarrow \{e_1\}$); (2) Cada intervalo $LI_{j,p_j}^{c_j}$ del SLI_m^{cur} de R_{cur} se inicializa a la simplificación del conjunto ordenado de etiquetas SA_j utilizando como conjunto de ejemplos asociado $E^{cur} = \{e_1\}$ ($SLI_m^{cur} \leftarrow Simplify(SA_1 \dots SA_m)$); (3) Se calcula la etiqueta de salida de la primera regla actual R_{cur} utilizando la ecuación 5 ($SE^{cur} \leftarrow CalculateConsequent(E^{cur})$):

$$max_{SC^w} \mu_{SC^w}(v) \tag{5}$$

donde w=1..i_y y $v = \frac{\sum_{i=1}^{|E^{cur}|} s^i}{|E^{cur}|}$, siendo $|E^{cur}|$ el número de ejemplos en E^{cur} y s^i el valor de salida de cada e_i que pertenece a E^{cur}.

En resumen, la etiqueta seleccionada para la salida de la regla es la que tiene el máximo grado de pertenencia al valor medio de los valores de salida de los ejemplos de E^{cur}. Se pueden ver otras posibilidades en [4, 10].

Posteriormente, el bucle **for** se utiliza para examinar los ejemplos del e_2 al último ejemplo e_n de E. Para cada ejemplo e_i se calcula la regla R_i donde: SLI_m^i se asigna a la simplificación de $SA_1 \dots SA_m$ utilizando $E^i = \{e_i\}$ como conjunto de ejemplos asociado ($SLI_m^i \leftarrow Simplify(SA_1 \dots SA_m)$); y la etiqueta de salida SE^i se calcula utilizando la ecuación E^i ($SE^i \leftarrow CalculateConsequent(E^i)$).

Después, si la etiqueta de salida de R_{cur} (SE^{cur}) es igual a la etiqueta de salida de R_i (SE^i), significa que las dos reglas (que representan conjuntos de ejemplos consetivos en el tiempo) tienen la misma salida, así, se asigna R_{cur} a la unión total de las reglas R_{cur} y R_i (definición 3.3), porque representan la misma salida ($R_{cur} \leftarrow TotalUnion(R_{cur}, R_i)$); y E^{cur} se asigna a la unión de E^{cur} y E^i ($E^{cur} \leftarrow E^{cur} \bigcup E^i$). En otro caso, R_{cur} se añade a MDT ($MDT \leftarrow MDT + R_{cur}$) y R_{cur} se asigna a R_i ($R_{cur} \leftarrow R_i$). Cuando el bucle **for** termina, la última regla actual R_{cur} se añade a MDT ($MDT \leftarrow MDT + R_{cur}$).

4 DESCRICIPCIÓN DEL MÉTODO

Esta sección muestra el comportamiento del método. Para ello se obtiene un modelo del endeudamiento de

Andalucía desde 1987 a 2000. La variable de salida es el endeudamiento D/PIB, y las variables de entrada son los ingresos no financieros INF y el tipo de interés T. Estos datos de entrada al algoritmo han sido obtenidos de varias fuentes fiables como son el Banco Nacional de España, el Ministerio de Hacienda, y el Instituto Nacional de Estadística.



Figura 1: Conjunto de etiquetas de la variable INF

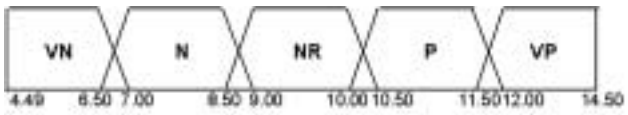


Figura 2: Conjunto de etiquetas de la variable T



Figura 3: Conjunto de etiquetas de la variable D/PIB

Se utiliza un conjunto ordenado de 9 etiquetas para la variable INF (figura 1), un conjunto ordenado de 5 etiquetas para la variable T (figura 2), y finalmente, un conjunto ordenado de 7 etiquetas para la variable D/PIB (figura 3). En estas figuras Z es cero, VN is Muy Negativo, N es Negativo, FN es Poco Negativo, NR es Normal, FP is Poco Positivo, P es Positivo, VP es Muy Positivo y O es Uno.

La tabla 1 muestra el MDT obtenido. Las figuras 4, 5 y 6 muestra la zona temporal de cada intervalo y etiqueta de salida para las variables de entrada y salida respectivamente. Los recuadros con R_n representan la zona temporal de los intervalos y etiquetas de salida, donde n es el número de regla siendo $R_1, R_2 \dots R_7$ las reglas $R_{1,3}, R_{4,4} \dots R_{10,11}$ respectivamente. Como se puede ver, los intervalos y etiquetas de salida para INF, T y D/PIB son "continuos".

Finalmente, se ha realizado un proceso de inferencia utilizando el algoritmo 1, el conjunto de ejemplos E se utiliza como entrada. El error obtenido es 0.2439. La figura 7 muestra gráficamente la comparación entre la salida real (línea negra) y la salida obtenida (línea gris). La línea obtenida es muy parecida a la real.

5 CONCLUSIONES

Se ha presentado un método para obtener directamente MDTs. Los MDTs son un nuevo método para

Tabla 1: MDT Obtenido.

R.	INF	T	PIB	Ti.
$R_{1,3}$	$\{Z \dots N\}$	$\{P, VP\}$	VN	$\{e_1 \dots e_3\}$
$R_{4,4}$	$\{VN\}$	$\{VP\}$	N	$\{e_4\}$
$R_{5,5}$	$\{Z\}$	$\{VP\}$	FN	$\{e_5\}$
$R_{6,6}$	$\{NR\}$	$\{VP\}$	NR	$\{e_6\}$
$R_{7,7}$	$\{FP\}$	$\{NR, P\}$	FP	$\{e_7\}$
$R_{8,9}$	$\{NR, FP\}$	$\{NR, P\}$	P	$\{e_8, e_9\}$
$R_{10,11}$	$\{P \dots O\}$	$\{VN, N\}$	VP	$\{e_{10} \dots e_{14}\}$

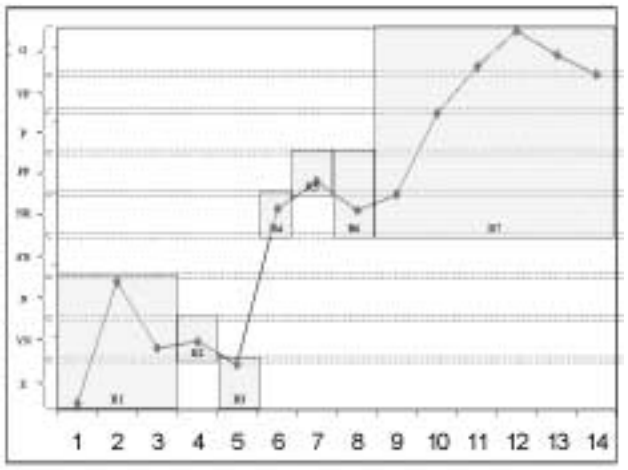


Figura 4: Variable INF

representar lingüísticamente los SDs. Así, el modelo describe cualitativamente el rango de valores utilizando los intervalos lingüísticos. Se define el conjunto ordenado de etiquetas para cada variable antes de la utilización del algoritmo, como resultado de esto, el modelo obtenido es más comprensible porque el número de etiquetas y el dominio de cada etiqueta se define fuera del algoritmo, por ejemplo, por un profesional en el área de aplicación del algoritmo. Una buena área de aplicación de nuestro método es el deporte. Los MDTs pueden detectar las fases del movimiento mediante un conjunto de RDTs, y explicar cómo cada variable evoluciona a lo largo del tiempo.

La evolución del tiempo se representa en el orden de las RDTs. Una RDT representa instantes consecutivos con la misma etiqueta de salida. Los MDTs reflejan las variables de entrada que cambian de un instante a los instantes siguientes. Nuestro método no tiene ningún problema cuando se obtienen diferentes valores de salida con los mismos valores de entrada debido al orden de las reglas. Así, es posible encontrar dos o más RDTs con el mismo antecedente y diferente consecuente.

Normalmente, los algoritmos de inducción agrupan mediante las variables de entrada. Nuestro método

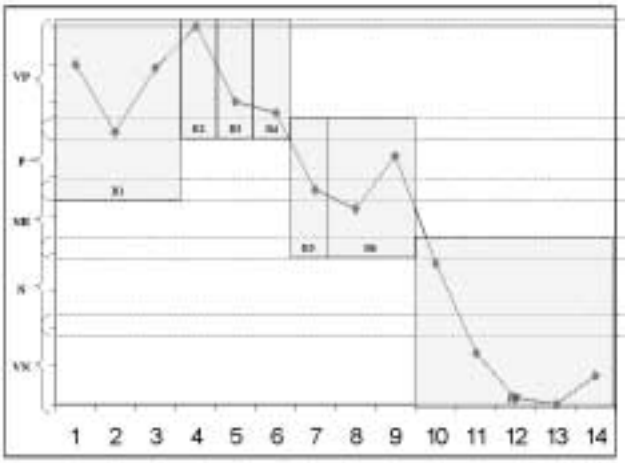


Figura 5: Variable T

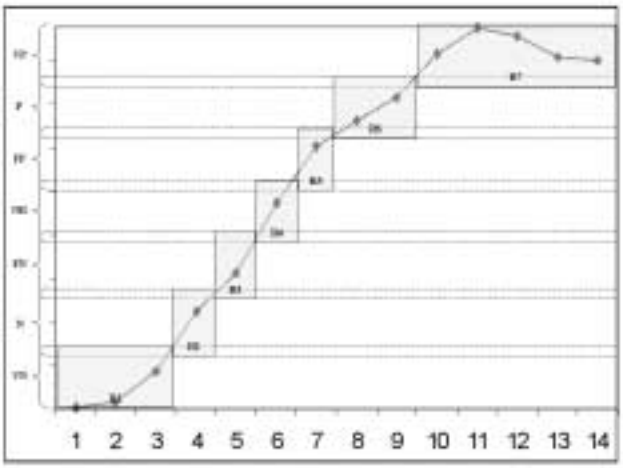


Figura 6: Variable D/PIB

agrupa utilizando la variable de salida, esto es una forma diferente de trabajar si lo comparamos con otros algoritmos.

Los resultados obtenidos confirman que el error obtenido con nuestro método es pequeño, así, los MDTs son una buena aproximación al SD que representan.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Ciencia y Tecnología y la Junta de Comunidades de Castilla-La Mancha mediante los proyectos DIMOCLUST TIC2003-08807-C02-02 y PREDACOM PBC-03-004.

Referencias

[1] J.L. Castro, J.J. Castro-Schez, J.M. Zurita. Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems. *Fuzzy Sets and Systems* vol. 101, pag. 331-342, 1999.

[2] M.Delgado and A. Gonzalez. An inductive learning procedure to identify fuzzy system. *Fuzzy Sets and Systems*, vol. 55, pag. 121-132, 1993.

[3] A. Gonzalez. A learning methodology in uncertain and impresise enviroments. *International Journal of Intelligence Systems*, vol 10, pag. 357-371, 1995.

[4] W. V. Leekwijck, E. E. Kerre. Defuzzification: criteria and classification. *Fuzzy Sets and Systems*, vol. 108 n.2, pag. 159-178, 1999.

[5] D.G. Luenberger. Introduction to Dynamic Systems: Theory, Models and Applications. *John Wiley and sons*. 1979.

[6] J. Moreno-Garcia, L. Jimenez, J.J. Castro-Schez, L. Rodriguez. A linguistic modelling approach for dynamic systems by using Temporal Fuzzy Models. *Subm. to Int. Journal of Approximate Reasoning*, 2004.

[7] J. Moreno-Garcia, L. Jimenez, J.J. Castro-Schez, L. Rodriguez. A direct linguistic induction method for systems. *Fuzzy Sets and Systems*. Por aparecer. 2004.

[8] K. Ogata. Ingenieria de Control Moderna. *Prentice-Hall HispanoAmericana SA*. 1998.

[9] S.M. Paudit, S.M. Wu. Time Series and Systems Analysis, with applications. New York, Wiley, 1983.

[10] K. Tanaka. An introduction to fuzzy logic for practical aplications. *Springer*. 1998.

[11] L. Zadeh. The concept of a linguistic variable and its applications to Approximate reasoning. Part I,II,III. *Information Science*. 1975.

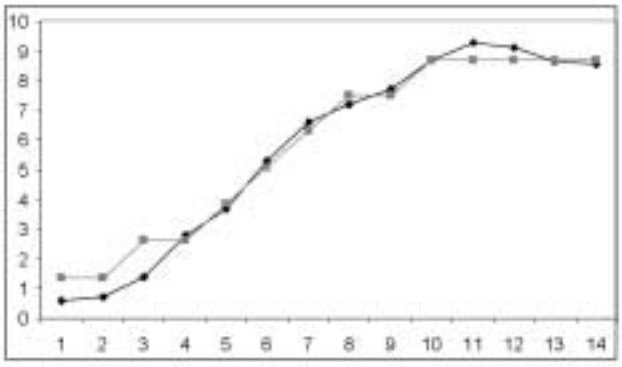


Figura 7: Proceso de inferencia

CONVERSIÓN DE MODELOS DIFUSOS TEMPORALES A CADENAS DIFUSAS TEMPORALES

Juan Moreno-Garcia U. de Castilla-La Mancha Departamento de Informática Avda Carlos III s/n, Toledo Juan.Moreno@uclm.es

Luis Jiménez, José Jesús Castro-Schez U. de Castilla-La Mancha Departamento de Informática P. de la Universidad nº 4, Ciudad Real {Luis.Jimenez,JoseJesus.Castro}@uclm.es

Luis Rodríguez U. de Castilla-La Mancha Departamento de Informatica Pza. Manuel Meca, Almadén Luis.Rodriguez@uclm.es

Resumen

Los modelos difusos temporales (MDTs) y las cadenas difusas temporales (CDTs) se utilizan para modelar sistemas dinámicos(SDs). Las reglas de los MDTs (reglas difusas temporales, RDTs) están ordenadas. Las CDTs se basan en dos conceptos diferentes: el método tradicional de representación de SDs llamado vector de estados [6], y las variables lingüísticas [8]. Este trabajo presenta un algoritmo para convertir los MDTs en CDTs. Esto es interesante cuando se tiene un MDT obtenido utilizando [3], ya que el MDT mantiene las propiedades de un modelo difuso de entrada, luego la CDT obtenida mantendrá también estas propiedades.

Palabras Clave: Modelado temporal, modelos difusos, sistemas dinámicos.

1 INTRODUCCIÓN

Un objetivo de nuestra investigación es definir estructuras que representen la evolución en el tiempo de un SD de forma lingüística [6, 2]. Utilizamos para ello los MDTs y las CDTs que se basan en la lógica difusa [7] y las variables lingüísticas [8]. Los MDTs [3] simulan el comportamiento del sistema en función de las entradas. Las RDTs de MDTs están ordenadas. Los MDTs son una secuencia de RDTs donde cada RDT modela una zona temporal. Las CDTs [4, 5] están basadas en la misma idea que los MDTs, el orden de sus componentes. Las CDTs se basan en dos conceptos, el método tradicional de representación de los SDs denominado vector de estado, y las variables lingüísticas de la lógica difusa [7]. Así, las CDTs son cualitativas y representan la zona temporal utilizando los estados y transiciones lingüísticas entre dos estados lingüísticos. En esta introducción se mostrarán algunos conceptos necesarios para comprender los MDTs y las CDTs, y una breve introducción a ambos.

1.1 ALGUNOS CONCEPTOS PREVIOS

Sea un SD con m variables reales de entrada $X_1 \dots X_m$ y una variable real de salida S. El comportamiento del sistema se obtiene mediante un conjunto de ejemplos $E = \{e_1 \dots e_n\}$ siendo $e_i = (x_1^i \dots x_m^i, s^i, t_i)$ donde $x_j^i \in X_j$, $s^i \in S$ y t_i es el tiempo en el que ocurre el ejemplo i. Se define un **conjunto ordenado de etiquetas lingüísticas** SA_j para cada una de las variables de entrada X_j (figura 1). Su estructura es $SA_j = \{SA_j^1 \dots SA_j^{i_j}\}$, donde i es la posición de SA_j^i en SA_j e i_j es el número de etiquetas en SA_j ($i_j = |SA_j|$). Se define un **conjunto ordenado de etiquetas lingüísticas** SC para la variable de salida S. Su estructura es $SC = \{SC^1 \dots SC^{i_y}\}$ con $i_y = |SC|$. Estos conjuntos se definen a priori.



Figura 1: Conjunto ordenado de etiquetas SA_j

Los MDTs y las CDTs trabajan con variables lingüísticas. Éstas tienen definidas un conjunto ordenado de etiquetas sobre su dominio y se denominan **variables lingüísticas continuas** (desde ahora *variables*). X_j es una variable que tiene asociada una semántica a cada SA_j^i definida sobre ella que verifica: (1) Cada SA_j^i se define sobre un dominio ordenado. (2) La etiqueta definida en X_j está ordenado por el medio de máximos (MOM) [1]. (3) La suma del grado de pertenencia de un valor x a todas las etiquetas definidas en una variable continua debe ser 1 ($\sum_{SA_j^i \in SA_j} \mu_{SA_j^i}(x) = 1$). (4) Cada x_j en X_j verifica: ($\mu_{SA_j^i}(x_j) = 1$) ó ($\mu_{SA_j^i}(x_j) = a$ y $\mu_{SA_j^{i+1}}(x_j) = 1-a$).

Estas variables toman un **intervalo lingüístico** como valor. Un intervalo lingüístico (desde ahora intervalo) $LI^c_{j,p} = \{SA^p_j \dots SA^{p+(c-1)}_j\}$ para una variable X_j es un subconjunto ordenado de etiquetas de SA_j que empieza en la etiqueta p y tiene c etiquetas. Su función de pertenencia es la suma del grado de pertenencia a_j a cada etiqueta del intervalo (ecuación 1 con $z \in [p..c-1]$).

$$\mu_{LI^c_{j,p}}(a_j) = \sum_{SA^z_j \epsilon LI^c_{j,p}} \mu_{SA^z_j}(a_j) \quad (1)$$

Un **conjunto ordenado de m intervalos** es una secuencia de m intervalos definidos sobre m variables para cada una de las variables de entrada, y se representan como $SLI_m = \{LI^{c_1}_{1,p_1}, \dots LI^{c_m}_{m,p_m}\}$. La función de pertenencia de un SLI_m se calcula aplicando una t-norma entre el grado de pertenencia de los intervalos del SLI_m (ecuación 2).

$$\mu_{SLI_m}(e_i) = *(\mu_{LI^{c_j}_{j,p_j}}(x_j)) \quad (2)$$

donde $e_i = (x^i_1 \dots x^i_m, s^i, t_i)$ es un ejemplo que pertenece a E, $j\epsilon[1..m]$ y * es una t-norma.

1.2 INTRODUCCIÓN A LOS MDTs

Un **modelo difuso temporal** M_{Tord} es un modelo compuesto de RDTs que están ordenadas. Una **RDT** $R_{p,u}$ es una regla difusa que verifica que: (1) Su antecedente es un SLI_m. (2) Su consecuente es una etiqueta $SC^w \in SC$. (3) Tiene asociado a su SLI_m un conjunto de ejemplos consecutivos $E_{p,u}$. Un **conjunto de ejemplos consecutivos** $E_{p,u}$ es un conjunto de ejemplos que verifica: (1) $p \leq u$ (2) $E_{p,u} \subset E$ (3) $\forall k \in [p,u], \exists e_k \in E_{p,u}$ (4) $\forall e_a, e_b \in E_{p,u}, a \neq b$. $E_{p,u}$ tiene $u - p + 1$ ejemplos que son consecutivos en el tiempo y se utiliza para ordenar las reglas del MDT.

Algoritmo 1 Método de inferencia de MDTs

```
j ← 1
for i = 1 to |E| do
  if μ_{SLI^j_m}(e_i) > μ_{SLI^{j+1}_m}(e_i) then
    s ← Cons(R_j)
  else
    s ← Cons(R_{j+1})
    j ← j + 1
  end if
end for
```

El algoritmo 1 muestra el método de inferencia de los MDTs. Su entrada es un conjunto de ejemplos E que representa un SD. Los índices i y j apuntan al ejemplo y regla actual respectivamente. El bucle **for** se utiliza para examinar los ejemplos de E. Si el grado de pertenencia de e_i al SLI_m de R_j (SLI^j_m) es mayor que el grado de pertenencia de e_i al SLI_m de R_{j+1} (SLI^{j+1}_m) entonces la salida es la etiqueta de salida de R_j, en otro caso, la salida es la etiqueta de salida de la regla R_{j+1} y la nueva regla actual es R_{j+1}. Un proceso de defuzzificación [7] obtendría una salida discreta.

1.3 INTRODUCCIÓN A LAS CDTs

Informalmente, una CDT es una secuencia de estados y transiciones lingüísticas. Un estado lingüístico se utiliza para representar el SD en instantes consecutivos. Entre dos estados consecutivos se establece una transición lingüística que indica la condición necesaria para pasar al siguiente estado lingüístico. El cambio de estado se describe de manera lingüística.

Un **estado lingüístico** i (desde ahora estado) se define como una tupla $est_i =< A^i_m, SE_i >$ donde A^i_m es el SLI_m del estado i correspondiente a las m variables de entrada del SD, y SE_i es la salida del estado i correspondiente a la variable de salida del SD. Una **transición lingüística** i (desde ahora transición) es una tupla $trans_i =< T^i_m, ST_i >$ donde T^i_m es el SLI_m de la transición i correspondiente a las m variables de entrada del SD, y ST_i es la etiqueta de salida de la transición i correspondiente a la variable de salida del SD. Una **CDT** es una tupla $CHAIN =< EST, TRANS >$ donde $EST = \{est_1 \dots est_{ns}\}$ es un conjunto ordenado de ns estados, y $TRANS = \{trans_1 \dots trans_{ns-1}\}$ es un conjunto ordenado de $ns - 1$ transiciones. La transición i refleja las condiciones para cambiar de est_i a est_{i+1}.

Algoritmo 2 Método de inferencia de las CDTs

```
act ← 1
for i = 1 to |E| do
  if μ_{A^act_m}(e_i) > μ_{T^act_m}(e_i) then
    s ← SE_act
  else
    s ← ST_act
    act ← act + 1
  end if
end for
```

Para reproducir el comportamiento del SD se ofrece el método de inferencia de las CDTs (algoritmo 2). En este algoritmo, *act* indica el estado y la transición actual, E el conjunto de ejemplos, e_i el ejemplo i de E y s la salida. El algoritmo usa un conjunto de ejemplos E como entrada y está basado en la definición del estado est_{act} denominado *estado actual*. est_{act} indica el estado en el que está el SD, y permite calcular la salida en ese instante. El método de inferen-

cia comienza seleccionando est_1 como el primer estado actual. Luego se calculan $\mu_{A_m^{act}}(e_i)$ del estado est_{act} y $\mu_{T_m^{act}}(e_i)$ de la transición $trans_{act}$. Si $\mu_{A_m^{act}}(e_i)$ es mayor que $\mu_{T_m^{act}}(e_i)$ entonces la salida obtenida s es SE_{act} (correspondiente a est_{act}) y no hay cambio de estado. En otro caso, si $\mu_{A_m^{act}}(e_i)$ es menor o igual que $\mu_{T_m^{act}}(e_i)$, entonces hay un cambio de estado: la salida obtenida s es ST_{act} (correspondiente a $trans_{act}$) y el nuevo estado actual es el siguiente de la CDT (est_{act+1}). Este proceso se repite para todos los ejemplos de E.

2 CONVERSIÓN DE MDTs A CDTs

Los MDTs pueden ser convertidos a CDTs, para ello se usa el algoritmo 3. Primero, se mostrará un problema que ocurre entre dos RDTs consecutivas de un MDT.

Una CDT representa los cambios de estado a lo largo del tiempo para las variables de entrada y de salida. Luego para transformar un MDT a CDT se han de buscar los cambios en los SLI_m y en la etiqueta del consecuente de las RDTs del MDT. En algunas RDTs consecutivas no se reflejan cambios en sus SLI_m. Por ejemplo, en $R_{4,5}$ y $R_{6,7}$ de la tabla 1 el SLI_3 de $R_{6,7}$ está incluido en el SLI_3 de $R_{4,5}$ (definición 2.1), es decir, para los intervalos de las variables x, y y z se verifica que $\{VP\} \subset \{VP\}$, $\{N\} \subset \{VN, N\}$ y $\{P\} \subset \{NR, P\}$ respectivamente. En general, la RDT siguiente tiene su SLI_m contenido en el SLI_m de su RDT anterior. Este problema ocasiona que no se puedan detectar los cambios de estado de una RDT a la siguiente, para solucionarlo en el algoritmo se realiza la unión total de las RDTs consecutivas (definición 2.5) que no reflejen cambios en sus SLI_m.

Tabla 1: RDTs

$R_{p,u}$	**x**	**y**	**z**	**s**
$R_{4,5}$	VP	VN a N	NR a P	NR
$R_{6,7}$	VP	N	P	P

Ahora, se muestran algunas definiciones necesarias para comprender el comportamiento del algoritmo.

Definición 2.1 *Sean SLI_m^1 y SLI_m^2 dos conjuntos ordenados de m intervalos y jϵ[1..m], entonces $SLI_m^1 \subset SLI_m^2$ si $\forall LI_{j,p_1}^{c_1} \in SLI_m^1, \forall LI_{j,p_2}^{c_2} \in SLI_m^2$ $LI_{j,p_1}^{c_1} \subset LI_{j,p_2}^{c_2}$*

Definición 2.2 *Sea $SA_j = \{SA_j^1 \dots SA_j^{i_j}\}$ un conjunto ordenado de etiquetas y dos etiquetas SA_j^p y SA_j^u de SA_j, la etiqueta $SA_j^{\frac{p+u}{2}}$ se denomina etiqueta central. El exponente $\frac{p+u}{2}$ es redondeado por exceso si $SA_j^p < SA_j^u$, y por defecto si $SA_j^p > SA_j^u$.*

Se utiliza el concepto de etiqueta central para calcular la salida asociada a cada transición entre dos estados i y $i+1$. Así, cuando SE_i y SE_{i+1} son consecutivas la etiqueta de salida ST_i de la transición i es SE_{i+1}, en otro caso, ST_i será la etiqueta intermedia entre SE_i y SE_{i+1}. La etiqueta central entre SE_i y SE_{i+1} se representará como $CL(SE_i, SE_{i+1})$ en el algoritmo 3. Es importante destacar que no se obtiene el mismo resultado al calcular la etiqueta central de SA_j^p y SA_j^u que la etiqueta central de SA_j^u y SA_j^p. Por ejemplo, si $SA_j^p = SA_j^4$ y $SA_j^u = SA_j^5$ se tiene que $SA_j^{\frac{p+u}{2}} = SA_j^{\frac{4+5}{2}} = SA_j^{4.5} = SA_j^5$, puesto que $SA_j^4 < SA_j^5$ el redondeo es por exceso. Si $SA_j^p = SA_j^5$ y $SA_j^u = SA_j^4$ se tiene que $SA_j^{\frac{5+4}{2}} = SA_j^4$, puesto que $SA_j^5 > SA_j^4$ el redondeo es por defecto.

Definición 2.3 *Sean $LI_{j,p1_j}^{c1_j}$ y $LI_{j,p2_j}^{c2_j}$ dos intervalos con sus etiquetas definidas sobre un conjunto SA_j. Se define la diferencia entre el intervalo $LI_{j,p2_j}^{c2_j}$ respecto del intervalo $LI_{j,p1_j}^{c1_j}$ como el conjunto de etiquetas: $LI_{j,p2_j}^{c2_j} - LI_{j,p1_j}^{c1_j} = \{SA_j^i \in LI_{j,p2_j}^{c2_j} \setminus SA_j^i \notin LI_{j,p1_j}^{c1_j}\}$*

Definición 2.4 *Sean $SLI_m^1 = \{LI_{1,p1_1}^{c1_1} \dots LI_{m,p1_m}^{c1_m}\}$ y $SLI_m^2 = \{LI_{1,p2_1}^{c2_1} \dots LI_{m,p2_m}^{c2_m}\}$, su diferencia es el conjunto ordenado de m conjuntos de etiquetas formado por: $SLI_m^2 - SLI_m^1 = \{LI_{1,p2_1}^{c2_1} - LI_{1,p1_1}^{c1_1} \dots LI_{m,p2_m}^{c2_m} - LI_{m,p1_m}^{c1_m}\}$*

Como puede verse, la diferencia entre SLI_m^2 y SLI_m^1 es un conjunto de m conjuntos de etiquetas, donde cada conjunto se obtiene como la diferencia de los intervalos $LI_{j,p2_j}^{c2_j}$ y $LI_{j,p1_j}^{c1_j}$ de SLI_m^2 y SLI_m^1 respectivamente. Si todos los conjuntos que pertenecen al conjunto obtenido es vacío la diferencia es el conjunto vacío, y se representa como θ.

Definición 2.5 *Sean n RDTs $R_{p_1,u_1} \dots R_{p_n,u_n}$, su unión total se define como la regla $R_{p,u}$ donde: (1) Su SLI_m se obtiene mediante la unión de $SLI_m^1 \dots SLI_m^n$. (2) El conjunto de ejemplos asociado es igual a $E_{SLI_m} = \bigcup E_{SLI_m^j}$ donde j $\in$ [1..n]. (3) La etiqueta del consecuente se calcula utilizando la ecuación 3.*

$$max_{SC^w} \mu_{SC^w}(v) \tag{3}$$

donde w=1..i_y y $v = \frac{\sum_{i=1}^{|E_{SLI_m}|} s^i}{|E_{SLI_m}|}$ siendo E_{SLI_m} el conjunto de ejemplos de SLI_m y s^i el valor de salida del ejemplo e_i que pertenece a E_{SLI_m}.

Finalmente, se muestra el funcionamiento del algoritmo de conversión (algoritmo 3). R_{aux} es una RDT y N_{nodo} es el número del nodo actual. Las dos primeras sentencias del algoritmo inicializan R_{aux} y N_{nodo} a los valores R_1 y 1 respectivamente. Después, el bucle for recorre las reglas del modelo M_{Tord} utilizando el índice i. Ya en el bucle, si la diferencia entre el

Algoritmo 3 Conversión de un MDT a CDT

$R_{aux} \leftarrow R_1$
$N_{nodo} \leftarrow 1$
for $i = 1$ to $|M_{Tord}| - 1$ **do**
  if $(SLI_m^{i+1} - SLI_m^{aux}) = \theta$ **then**
    $R_{aux} \leftarrow TotalUnion(R_{aux}, R_{i+1})$
  else
    $A_m^{N_{nodo}} \leftarrow SLI_m^{aux}$
    $SE_{N_{nodo}} \leftarrow Cons(R_{aux})$
    if $N_{nodo} > 1$ **then**
      $T_m^{N_{nodo}-1} \leftarrow SLI_m^{aux}$
      $ST_{N_{nodo}-1} \leftarrow CL(SE_{N_{nodo}-1}, SE_{N_{nodo}})$
    end if
    $R_{aux} \leftarrow R_{i+1}$
    $N_{nodo} \leftarrow N_{nodo} + 1$
  end if
end for
$A_m^{N_{nodo}} \leftarrow SLI_m^{aux}$
$SE_{N_{nodo}} \leftarrow Cons(R_{aux})$
$T_m^{N_{nodo}-1} \leftarrow SLI_m^{aux}$
$ST_{N_{nodo}-1} \leftarrow CL(SE_{N_{nodo}-1}, SE_{N_{nodo}})$

SLI_m^{i+1} de R_{i+1} y el SLI_m^{aux} de R_{aux} es θ (el cambio de estado no puede ser detectado) entonces R_{aux} se asigna a la unión total de las reglas R_{aux} y R_{i+1} (definición 2.5). Si la diferencia no es θ, el cambio de estado puede ser detectado, luego se calculan el nuevo estado $est_{N_{nodo}} =< A_m^{N_{nodo}}, SE_{N_{nodo}} >$ y la transición anterior al mismo $trans_{N_{nodo}-1} =< T_m^{N_{nodo}-1}, ST_{N_{nodo}-1} >$. Finalmente, R_{aux} se asigna a R_{i+1} y N_{nodo} se incrementa en 1. Las siguientes acciones se realizan para calcular $est_{N_{nodo}}$: (1) Asignar $A_m^{N_{nodo}}$ al SLI_m^{aux} de R_{aux}; (2)Asignar $SE_{N_{nodo}}$ al consecuente de R_{aux}.

Estas acciones se realizan para crear la transición $trans_{N_{nodo}-1}$: (1) Asignar $T_m^{N_{nodo}-1}$ a SLI_m^{aux} de R_{aux}; (2) Asignar $ST_{N_{nodo}}$ a la etiqueta central entre $SE_{N_{nodo}-1}$ y $SE_{N_{nodo}}$.

Finalmente, fuera del bucle, se añaden la última transición y estado a la CDT.

Queremos destacar las siguientes acciones del algoritmo:

1. La diferencia de los conjuntos ordenadas SLI_m^{aux} se utilizan para detectar si hay cambio. Si la diferencia es θ significa que el SLI_m de la siguiente regla está contenido en el SLI_m de la regla anterior, es decir, el cambio de estado no puede ser detectado.

2. La regla R_{aux} se utiliza para crear el nuevo nodo $est_{N_{nodo}}$, esto es, los componentes de $est_{N_{nodo}}$ ($A_m^{N_{nodo}}$ y $SE_{N_{nodo}}$) se asignan al SLI_m y a la etiqueta de salida de la regla R_{aux} respectivamente.

3. SLI_m^{aux} de R_{aux} y la etiqueta central entre los dos estados se utilizan para calcular la transición. La etiqueta central es útil cuando la evolución de los valores de salida es más rápida que la evolución de los valores de entrada y el cambio de estado no se puede detectar. La etiqueta central como salida permite que la salida obtenida con la CDT sea más suave es este cambio de estado.

3 FUNCIONAMIENTO DEL ALGORITMO

El algoritmo 3 se utiliza para obtener una CDT. Este algoritmo necesita un conjunto ordenado de etiquetas para cada una de las variables y un MDT M_{Tord} como entrada. En este caso, la figura 1 y la tabla 2 muestran el conjunto ordenado de 7 etiquetas para cada variable y M_{Tord} (obtenido con el método de inducción en [3]) respectivamente. Las reglas de M_{Tord} están numeradas como $R_1 \dots R_8$. Antes de la primera iteración del bucle las variables R_{aux} y N_{nodo} se inicializan a R_1 y 1 respectivamente.

Tabla 2: MDT M_{Tord}

$R_{p,u}$	**x**	**y**	**s**
$R_1 = R_{1,2}$	VN	VP	P
$R_2 = R_{3,9}$	$N\ a\ P$	$FP\ a\ VP$	VP
$R_3 = R_{10,11}$	VP	FP	P
$R_4 = R_{12,12}$	VP	NR	FP
$R_5 = R_{13,14}$	VP	$FN\ a\ NR$	NR
$R_6 = R_{15,15}$	VP	FN	FN
$R_7 = R_{16,17}$	VP	$N\ a\ FN$	N
$R_8 = R_{18,25}$	$N\ a\ VP$	$VN\ a\ N$	VN

Tabla 3: Diferencia entre SLI_2 de R_2 y R_{aux}

SLI_2	**x**	**y**
R_2	$\{N\ a\ P\}$	$\{FP\ a\ VP\}$
R_{aux}	$\{VN\}$	$\{VP\}$
$Diferencia$	$\{N\ a\ P\}$	$\{FP\ a\ P\}$

En la primera iteración del bucle se calcula la diferencia entre los SLI_2 de R_{i+1} y de R_{aux}, como $i = 1$, se calcula la diferencia entre los SLI_2 de R_2 y R_{aux} (tabla 3). La primera columna representa el nombre de la regla, la segunda y tercera son los intervalos de las variables x e y respectivamente. La diferencia de ambos conjuntos ordenados de intervalos se muestra en la tercera fila. La diferencia de los intervalos de la variable x es $\{N\ a\ P\}$ ya que $\{N\ a\ P\} - \{VN\} = \{N\ a\ P\}$. La diferencia de los intervalos de la variable y es $\{FP, P\}$

ya que $\{FP, P, VP\} - \{VP\} = \{FP, P\}$. Así, se obtiene $\{\{N\ a\ P\}, \{FP, P\}\}$ como la diferencia de los dos SLI_2. Ya que la diferencia no es θ, se ejecutan las sentencias asociadas a la sentencia **else**. El estado $est_1 =< A_2^1, SE_1 >$ donde $A_2^1 = SLI_2^{aux} = \{\{VN\}, \{VP\}\}$ y $SE_1 = Cons(R_{aux}) = \{P\}$.

El siguiente **if** se utiliza para obtener la transición anterior al estado insertado (est_1). est_1 es el primer estado, luego estas sentencias no se ejecutan (el primer estado no tiene una transición anterior). ($N_{nodo} > 1$) chequea esto, como $N_{nodo} = 1$, la transición no se calcula. Finalmente, la última sentencia **else** asigna las variables R_{aux} y N_{nodo} a R_2 y 2 respectivamente.

Tabla 4: Diferencia entre los SLI_2 de R_3 y R_{aux}

SLI_2	**x**	**y**
R_3	$\{VP\}$	$\{FP\}$
R_{aux}	$\{N\ a\ P\}$	$\{FP\ a\ VP\}$
$Diferencia$	$\{VP\}$	θ

En la segunda iteración, se calcula la diferencia del SLI_2 de R_3 y R_{aux}. La tabla 4 muestra la diferencia de ambos SLI_2. La diferencia de los intervalos de las variables x e y es $\{VP\} - \{N\ a\ P\} = \{VP\}$ y $\{FP\} - \{FP, P, VP\} = \theta$ respectivamente. Así, se obtiene $\{\{VP\}, \theta\}$ como diferencia de los dos SLI_2, luego se ejecutan las sentencias asociadas a la sentencia **else** ($\{\{VP\}, \theta\} \neq \theta$). El estado $est_2 =< A_2^2, SE_2 >$ donde $A_2^2 = SLI_2^{aux} = \{\{N\ a\ P\}, \{FP, P, VP\}\}$ y $SE_2 = Cons(R_{aux}) = \{VP\}$. La sentencia **if** asigna la transición a: $T_2^1 = SLI_2^{aux} = \{\{N\ a\ P\}, \{FP, P, VP\}\}$ y $ST_1 = CL(SE_1, SE_2) = \{VP\}$. Esta sentencia **if** se verifica siempre para el resto de iteraciones ya que la condición ($N_{nodo} > 1$) se cumple y N_{nodo} siempre se incrementa. Finalmente, la última sentencia **else** asigna R_{aux} y N_{nodo} a R_3 y 3 respectivamente.

Tabla 5: Diferencia entre SLI_2 de R_4 y R_{aux}

SLI_2	**x**	**y**
R_4	$\{VP\}$	$\{NR\}$
R_{aux}	$\{VP\}$	$\{FP\}$
$Diferencia$	θ	$\{NR\}$

Se calcula la diferencia entre los SLI_2 de R_4 y R_{aux} para $i = 3$ (tabla 5). La diferencia de los intervalos de las variables x e y es $\{VP\} - \{VP\} = \theta$ y $\{NR\} - \{FP\} = \{NR\}$ respectivamente. Así, la diferencia de ambos SLI_2 es $\{\theta, \{NR\}\} \neq \theta$, es decir, se ejecutan las sentencias asociadas al **else**. El nuevo estado $est_3 =< A_2^3, SE_3 >$ donde $A_2^3 = SLI_2^{aux} = \{\{VP\}, \{FP\}\}$ y $SE_3 = Cons(R_{aux}) = \{P\}$. La sentencia **if** asigna la transición a: $T_2^2 = SLI_2^{aux} = \{\{VP\}, \{FP\}\}$ y $ST_2 = CL(SE_2, SE_3) = \{P\}$. Finalmente, la última sentencia **else** asigna R_{aux} y N_{nodo} a R_4 y 4 respectivamente.

Tabla 6: Diferencia entre SLI_2 de R_5 y R_{aux}

SLI_2	**x**	**y**
R_5	$\{VP\}$	$\{FN, NR\}$
R_{aux}	$\{VP\}$	$\{NR\}$
$Diferencia$	θ	$\{FN\}$

Se calcula la diferencia entre el SLI_2 de R_5 y R_{aux} para $i = 4$ (tabla 6). La diferencia de los intervalos de las variables x e y es $\{VP\} - \{VP\} = \theta$ y $\{FN, NR\} - \{FN\} = \{FN\}$ respectivamente. Así, la diferencia de ambos SLI_2 es $\{\theta, \{FN\}\}$, es decir, se ejecutan las sentencias asociadas al **else**. El nuevo estado $est_4 =< A_2^4, SE_4 >$ donde $A_2^4 = SLI_2^{aux} = \{\{VP\}, \{NR\}\}$ y $SE_4 = Cons(R_{aux}) = \{FP\}$. La sentencia **if** asigna la transición a: $T_2^3 = SLI_2^{aux} = \{\{VP\}, \{NR\}\}$ y $ST_3 = CL(SE_3, SE_4) = \{FP\}$. Finalmente, la última sentencia **else** asigna R_{aux} y N_{nodo} a R_5 y 5 respectivamente.

Tabla 7: Diferencia entre SLI_2 de R_6 y R_{aux}

SLI_2	**x**	**y**
R_6	$\{VP\}$	$\{FN\}$
R_{aux}	$\{VP\}$	$\{FN, NR\}$
$Diferencia$	θ	θ

Se calcula la diferencia entre el SLI_2 de R_6 y R_{aux} para $i = 5$ (tabla 7). La diferencia de los intervalos de las variables x e y es $\{VP\} - \{VP\} = \theta$ y $\{FN\} - \{FN, NR\} = \theta$ respectivamente. Así, la diferencia de ambos SLI_2 es $\{\theta, \theta\}$. Esto significa que se ejecutan las sentencias asociadas al **if**, es decir, se realiza la unión total de las reglas R_{aux} y R_6 (tabla 8).

Tabla 8: Unión total de R_{aux} y R_6

R_i	**x**	**y**	**s**	$E^i_{SLI_m}$
R_{aux}	$\{VP\}$	$\{FN, NR\}$	NR	$\{e_{13}, e_{14}\}$
R_6	$\{VP\}$	$\{FN\}$	FN	$\{e_{15}\}$
$\cup$	$\{VP\}$	$\{FN, NR\}$	NR	$\{e_{13} \ldots e_{15}\}$

Se calcula la diferencia entre SLI_2 de R_7 y R_{aux} para $i = 6$ (tabla 9). La diferencia de los intervalos de las variables x e y es $\{VP\} - \{VP\} = \theta$ y $\{N, FN\} - \{FN, NR\} = \{N\}$ respectivamente. Así, la diferencia de ambos SLI_2 es $\{\theta, \{N\}\}$, es decir, se ejecutan las sentencias asociadas a la sentencia **else**. El nuevo estado $est_5 =< A_2^5, SE_5 >$ donde $A_2^5 = SLI_2^{aux} = \{\{VP\}, \{FN, NR\}\}$ y $SE_5 = Cons(R_{aux}) = \{NR\}$. La sentencia **if** asigna la transición a: $T_2^4 = SLI_2^{aux} = \{\{VP\}, \{FN, NR\}\}$ y

Tabla 9: Diferencia entre SLI_2 de R_7 y R_{aux}

SLI_2	**x**	**y**
R_7	$\{VP\}$	$\{N, FN\}$
R_{aux}	$\{VP\}$	$\{FN, NR\}$
$Diferencia$	θ	$\{N\}$

$ST_4 = CL(SE_4, SE_5) = \{NR\}$. Finalmente, la última sentencia **else** asigna R_{aux} a R_7 y N_{nodo} a 6.

Se calcula para la diferencia entre el SLI_2 de R_8 y R_{aux} para la última iteración del bucle ($i = 7$). La diferencia de los intervalos de las variables x e y es $\{N\ a\ VP\}$ - $\{VP\}$ = $\{N\ a\ P\}$ y $\{VN, N\} - \{N, FN\} = \{VN\}$ respectivamente. Así, la diferencia de ambos SLI_2 es $\{\{N\ a\ P\}, \{VN\}\}$, es decir, se ejecutan las sentencias asociadas a la sentencia **else**. El nuevo estado $est_6 =< A_2^6, SE_6 >$ donde $A_2^6 = SLI_2^{aux} = \{\{VP\}, \{N, FN\}\}$ y $SE_6 = Cons(R_{aux}) = \{N\}$. La sentencia **if** asigna la transición a: $T_2^5 = SLI_2^{aux} = \{\{VP\}, \{N, FN\}\}$ y $ST_5 = CL(SE_5, SE_6) = \{FN\}$. Finalmente, la última sentencia **else** asigna R_{aux} a R_8 y N_{nodo} a 7.

Fuera del bucle se añaden la última transición y estado. Así, $est_7 =< A_2^7, SE_7 >$ donde $A_2^7 = SLI_2^{aux} = \{\{N\ a\ VP\}, \{VN, N\}\}$ y $SE_7 = Cons(R_{aux}) = \{VN\}$; y $trans_6 = < T_2^6, ST_6 >$ donde $T_2^6 = SLI_2^{aux} = \{\{N\ a\ VP\}, \{VN, N\}\}$ y $ST_6 = CL(SE_6, SE_7) = \{VN\}$.

La representación gráfica de la CDT obtenida se muestra en la figura 2. Algunos intervalos en las transiciones tienen el valor $NULL$ para indicar que su valor está incluído en el intervalo del estado previo, es decir, no hay cambios para esas variables. El valor $NULL$ se utiliza para representar gráficamente las variables que cambian.

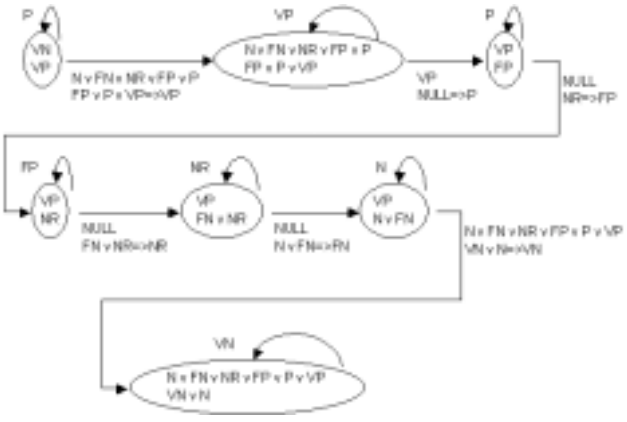


Figura 2: CDT obtenida

4 CONCLUSIONES

En este trabajo revisamos los conceptos de MDT [3] y CDT [4, 5] utilizados para representar SDs. Se presenta un algoritmo para obtener una CDT a partir de un MDT, y se muestra su comportamiento mediante un caso práctico de conversión de un MDT a una CDT. Como primera conclusión, se obtiene que las CDT pueden ser obtenidas mediante dos métodos: (a) *Un método directo* [4]; (b) *Un método indirecto*: primeramente, se obtiene un MDT [3], y finalmente, su utiliza el algoritmo presentado para obtener la CDT desde el MDT. Como segunda conclusión, las dos CDTs obtenidas por ambos métodos son equivalente, aunque esto habría que demostrarlo formalmente. Como trabajos futuros proponemos demostrar que las CDTs obtenidas con el método directo son equivalentes a las CDTs obtenidas con el método indirecto. También deberíamos estudiar el comportamiento de los MDTs y CDTs con lo mismos datos de entrada (pensamos que serán muy parecidos).

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Ciencia y Tecnología (proyecto DIMOCLUST TIC2003-08807-C02-02) y la Junta de Comunidades de Castilla-La Mancha (proyecto PREDACOM PBC-03-004.

Referencias

[1] W. V. Leekwijck, E. E. Kerre. Defuzzification: criteria and classification. *Fuzzy Sets and Systems, vol 108 n.2, pag. 159-178*, 1999.

[2] D.G. Luenberger. Introduction to dynamic systems: theory, models and applications. *John Wiley and sons*, 1979.

[3] J. Moreno, L. Jiménez, J.J. Castro-Schez, J.A. Olivas. Temporal Fuzzy Model for Dynamic Systems. In *IC-AI Proceedings*, April 2002.

[4] J. Moreno, L. Jimenez, J.J. Castro-Schez, L. Rodriguez. Induction of Temporal Fuzzy Chains. In *ICEIS Proceedings*, Angers, France, April 2003.

[5] J. Moreno, L. Jimenez, J.J. Castro-Schez, L. Rodriguez. Modelling with Temporal Fuzzy Chains. In *EUSFLAT Proceedings*, Germany, 2003.

[6] K. Ogata. Ingenieria de control moderna. *Prentice-Hall HispanoAmericana SA*, 1998.

[7] K. Tanaka. An introduction to fuzzy logic for practical aplications. *Springer*, 1998.

[8] L. Zadeh. The concept of a linguistic variable and its applications to Approximate reasoning. Part I, II, III. *Information Science*, 1975.

PROCESAMIENTO SERIE-PARALELO DE REDES DE RESTRICCIONES TEMPORALES BORROSAS DISYUNTIVAS

Alfonso Bosch, Francisco Guil
Dpto. Lenguajes y Computación
Universidad de Almería
04120 – La Cañada (Almería)
{abosch,fguil}@ual.es

Roque Marín
Dpto. Ing. de la Inf. y las Comunicaciones
Universidad de Murcia
30071 – Espinardo (Murcia)
roque@dif.um.es

Resumen

El modelo de Redes de Restricciones Temporales Borrosas Disyuntivas (RRTBD) permite la representación y el razonamiento con restricciones complejas, tanto cualitativas como cuantitativas. Debido a su expresividad, su tratamiento es de complejidad exponencial. Sin embargo, existe un subconjunto de estas redes que, debido a sus características topológicas, resultan tratables y se denominan RRTBD Serie-Paralelo.

Palabras Clave: Problemas de Satisfacción de Restricciones Temporales Borrosas. Redes Serie-Paralelo.

1 INTRODUCCIÓN

Las Redes de Restricciones Temporales Borrosas Disyuntivas (RRTBD) [1] representan el modelo más general de Redes de Restricciones Temporales Borrosas [10]. Permiten expresar restricciones temporales disyuntivas mediante un conjunto finito de distribuciones de posibilidad, normalizadas y convexas. Las restricciones temporales borrosas permiten combinar información precisa e imprecisa, la cual puede ser cualitativa y cuantitativa. Este modelo permite tener grados intermedios de consistencia y cuantificar la posibilidad y necesidad de una relación entre eventos o consulta.

Este modelo generaliza el marco TSCP (Temporal Constraint Satisfaction Problem) propuesto en [8] y permite expresar restricciones tales como "El tratamiento con Aceite de verano tiene que realizarse una semana antes o dos semanas después del tratamiento con Bifentrin". Este tipo de expresiones se encuentran sobre todo en tareas de planificación, aunque también pueden encontrarse en aplicaciones de control, tareas de diagnóstico y bases de datos temporales.

El principal inconveniente de este modelo es que su resolución es un problema NP-complejo [8] y, por lo tanto, computacionalmente intratable. Existen en la literatura distintas técnicas para abordar este tipo de problemas sin necesidad de utilizar retroceso para determinar consistencia u obtener soluciones [2,4]. Destacamos las técnicas que exploran las características topológicas de las redes, como por ejemplo, el algoritmo de descomposición en árbol y el algoritmo SP (Series-Parallel). En [5] se presenta una comparativa de ambas técnicas en la que se obtuvieron resultados que indicaban la superioridad del esquema serie-paralelo.

En este trabajo se propone la modificación del algoritmo SP para determinar la consistencia de una RRTBD Serie-Paralelo, teniendo en cuenta que el algoritmo, al que denominamos SP^+, presenta una complejidad lineal.

La sección 2 presenta las bases del modelo RRTBD. La sección 3 describe el subconjunto tratable denominado Redes Serie-Paralelo. En la sección 4 se presentan las bases del algoritmo SP^+ y, por último, la sección 5 recoge las conclusiones y trabajos futuros.

2 EL MODELO RRTBD

Una red L^d de restricciones temporales borrosas disyuntivas (RRTBD) está formada por un conjunto finito de n+1 variables temporales $X_0,...,X_n$ (X_0 origen de tiempos), cuyo dominio es un conjunto de instantes precisos equidistantes τ, y un conjunto finito de restricciones binarias disyuntivas L_{ij}^d entre dichas variables.

Una restricción binaria disyuntiva L_{ij}^d sobre las variables temporales X_i e X_j se define mediante un conjunto finito de distribuciones de posibilidad, $\{\pi_{ij}^1, \pi_{ij}^2, ..., \pi_{ij}^k\}$, definidas sobre un conjunto I isomorfo de los enteros, normalizadas y convexas. Para $s \in I, \pi_m(s) \in [0,1]$ representa la posibilidad de que una cantidad de tiempo m pueda ser, precisamente, s unidades temporales.

Una asignación de valores a las variables X_i, X_j / Xi=a; X_j=b, a, b ∈ τ, satisface la restricción L_{ij}^d sí y sólo si satisface al menos una de las restricciones individuales de las que consta:

$$\exists \pi_{ij}^p \in L_{ij}^d / \pi_{ij}^p(b-a) > 0 \quad (1)$$

El máximo grado de posibilidad con que la asignación X_i = a, X_j = b satisface L_{ij}^d es:

$$\sigma_{ij}^{\max}(a,b) = \max_{1 \le p \le k} \pi_{ij}^p(b-a) \quad (2)$$

Es necesario definir un máximo grado de posibilidad porque la distribución de posibilidad π_{ij}^p no tiene por qué ser única, ya que a pesar de estar en forma canónica, puede haber solapamiento en los conjuntos borrosos asociados a las distribuciones de posibilidad de L_{ij}^d.

Una restricción L_{ij}^d entre las variables X_i, X_j define también una restricción simétrica L_{ji}^d, y la ausencia de restricción entre dos variables se define como la restricción universal π_U. Las restricciones L_{ii}^d tienen que ser iguales al elemento cero, π_0, definido como $\pi_0(0) = 1$, y $\pi_0(m) = 0 \ \forall m \neq 0$ (no tiene sentido que una variable sea distinta de sí misma), ya que en ese caso el problema sería inconsistente.

Una RRTBD puede representarse con un grafo dirigido, donde cada nodo corresponde a una variable y cada arco a una restricción entre las variables conectadas, omitiendo las restricciones simétricas y universales. El conjunto de todas las posibles soluciones de una RRTDB L^d se define como el subconjunto borroso de τ^n asociado con la distribución de posibilidad dada por:

$$\pi_S(t_1,...,t_n) = \min_{\substack{0 \le i \le n \\ 0 \le j \le n}} (\sigma_{ij}^{\max}(t_i, t_j) \quad (3)$$

Una n-tupla de instantes precisos T = (t_1, ... t_n) ∈ τn es una solución σ-posible si π_S(T) = σ. Diremos que una RRTBD L^d es consistente si tiene al menos una solución 1-posible, y diremos que es inconsistente en caso contrario.

Dada una RRTDB L^d, existen múltiples redes que generan el mismo conjunto de soluciones, es decir, son equivalentes a L^d. La red equivalente a L^d cuyas restricciones son mínimas se denomina red mínima M^d asociada a L^d. Si M^d tiene una restricción vacía, L^d es inconsistente. El cálculo de la red mínima requiere resolver todas las posibles combinaciones generadas seleccionando una distribución de posibilidad de cada restricción disyuntiva, y su complejidad es exponencial [12]. Si hay un máximo de p distribuciones de posibilidad en cada restricción, y la red tiene q restricciones disyuntivas y n variables, la red mínima M^d se calcula con una complejidad O($p^q n^3$), donde n^3 es el coste de resolver cada caso no disyuntivo. Esto nos obliga a buscar enfoques para intentar reducir el coste del cálculo de la red mínima y/o la detección de la inconsistencia del problema.

Cuando es inevitable usar el retroceso para resolver una RRTBD, es conveniente partir de una red equivalente pero más reducida que la original. Las técnicas de preproceso de las redes eliminan información inútil, obteniendo una red equivalente a la red original, además de determinar correctamente, pero no completamente, su inconsistencia [3]. Pero antes de tener que recurrir a las técnicas de preproceso, resulta interesante explorar las características topológicas de las redes de restricciones con el objetivo de seleccionar métodos efectivos para resolverlas. De entre todas las técnicas que se han propuesto en la literatura, destacamos por su eficiencia computacional las redes serie-paralelo, una subclase tratable de las RRTBD.

3 REDES SERIE-PARALELO

Una red es serie-paralelo [11] respecto a un par de nodos i,j si puede reducirse al arco (i,j) aplicando repetidamente esta operación de reducción:

- seleccionar un nodo de grado 2 o inferior
- quitarlo de la red
- conectar sus vecinos (si no estaban ya conectados).

Si una red es serie-paralelo respecto a cualquier par de nodos, es una red serie-paralelo. Cuando una red es serie-paralelo, al aplicarle el algoritmo de camino-consistencia PC-1 [11] se obtiene la red mínima.

La Figura 1 muestra una red que es serie-paralelo, ya que puede reducirse a cualquiera de sus arcos aplicando el proceso de reducción:

1. Se elimina X_2.
2. Se elimina X_1, conectando X_0 con X_3.
3. Se elimina X_3, conectando X_0 con X_4.

Sin embargo, la Figura 2 presenta una red que no es serie-paralelo, ya que no existe ninguna secuencia de reducción admisible para ningún arco. Se puede ver que, tras eliminar X_0, no queda ningún otro nodo con grado menor o igual que 2.

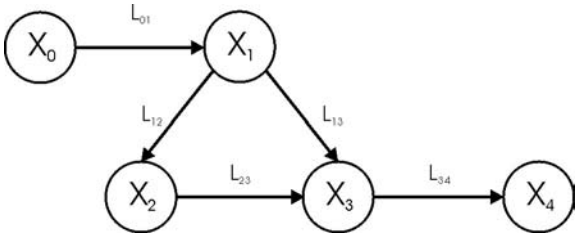


Figura 1. Ejemplo de red serie-paralelo.

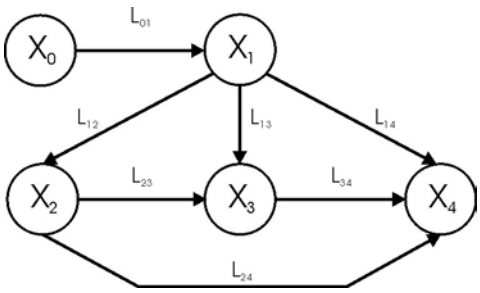


Figura 2. Ejemplo de red que no es serie-paralelo.

Existe un algoritmo eficiente que permite comprobar si una red es serie-paralelo con una complejidad O(n) [13]. Este algoritmo se basa en la propiedad de que una red es serie-paralelo o, lo que es equivalente, es un árbol-2 parcial, si no contiene ningún grafo homomorfo a K_4, el grafo completo de 4 nodos. En la Figura 2 podemos observar que, tras eliminar el nodo X_0 queda, precisamente, un grafo homomorfo a K_4.

4 EL ALGORITMO SP$^+$

Al determinar si una red es serie-paralelo con el algoritmo SP, trabajamos solamente con la estructura de la red, y no usamos la información de las restricciones. Con una modificación del algoritmo para manejar la información de las restricciones, podemos obtener información útil sobre la red al tiempo que determinamos si es o no serie-paralelo.

Nuestra propuesta inicial consiste en incorporar una operación de relajación de restricciones cada vez que se elimina un nodo o variable de grado 2. Esta operación actualiza una restricción eliminando los valores no permitidos por un camino de longitud dos, haciendo que la restricción sea camino-consistente respecto a dicho camino. La operación es la siguiente:

$$L_{ij} \leftarrow L_{ij} \dot{\cap} L_{ik} \dot{\oplus} L_{kj} \qquad (4)$$

En esta operación, k es el nodo de grado 2 a eliminar en un paso del algoritmo, mientras que i y j son los nodos vecinos del eliminado. Si la restricción entre i y j no existía, se crea una nueva restricción inducida. Si en el proceso encontramos una restricción vacía, la red es inconsistente, como demostraremos a continuación.

El algoritmo modificado es el siguiente:

```
Algoritmo SP+
begin
  for each i=0..n Calcular-grado (i)
  ColaNodos = {nodos con grado 1 y 2}
  While (ColaNodos <> ∅ and |V| > 2)
    begin
       nodo=Extraer(ColaNodos)
       if Grado(nodo) = 2 then
          L_ij ← L_ij ∩̇ L_ik ⊕̇ L_kj
       if L_ij = ∅ then exit

       V ← V - {nodo}
       if Grado(nodo) = 1 then
         Grado(Vecino(nodo)) --
       if Grado(Vecino(nodo)) = 2 then
         Introducir(ColaNodos, Vecino(nodo))
       else if Conexo(Vecinos(nodo)) then
              Grado(Vecinos(nodo)) --
       if Grado(Vecinos(nodo)) = 2 then
          Introducir(ColaNodos, Vecinos(nodo))
       else E ← E + {ArcoVecinos(nodo)}
    end
  if (ColaNodos = ∅ and |V| > 2) then
    exit ("La red no es serie-paralelo")
end
```

Otra de las modificaciones del algoritmo es que continúa hasta obtener un arco, que corresponde a una restricción entre dos variables, en lugar de parar cuando ya sabemos que no puede haber un grafo homomorfo con K_4. Esta modificación busca obtener una restricción entre las dos variables que sea camino-consistente con toda la red, hecho que también demostraremos en esta sección. Antes de pasar a enumerar y demostrar los resultados, vamos a ver un ejemplo de funcionamiento del algoritmo.

Ejemplo. Sea la RRTBD que se muestra en la Figura 3. El proceso de ejecución del algoritmo seguiría los pasos que se muestran a continuación:

1. Se elimina X_2. Al eliminar X_2, de grado 2, actualizamos el arco (X_1, X_3) utilizando la operación de relajación expresada en la ecuación (4), $L_{13} \leftarrow L_{13} \dot{\cap} L_{12} \dot{\oplus} L_{23}$, quedando L_{13} = {(54, 55, 70, 72),(89, 90, 105, 106)}.

2. Se elimina X_1. Se crea el arco (X_0, X_3), con $L_{03} \leftarrow L_{03} \dot{\cap} L_{01} \dot{\oplus} L_{13}$, o lo que es lo mismo, L_{03} = {(58, 60, 77, 80), (78, 80, 150, 152)}.

3. Se elimina X_3, creándose el arco (X_0, X_4), y realizando la operación de relajación

$L_{04} \leftarrow L_{04} \dot{\cap} L_{01} \dot{\oplus} L_{14}$, o sea, L_{04} = (62, 65, 160, 163), arco camino-consistente con respecto a la red.

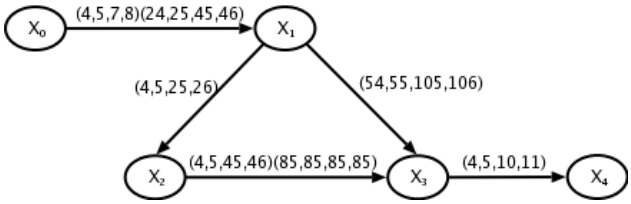


Figura 3. Ejemplo de RRTBD.

A continuación vamos a formalizar las propiedades del algoritmo.

Lema 1.- Si el algoritmo de reducción serie-paralelo se detiene por encontrar una restricción vacía, la red es inconsistente.

Demostración.- El algoritmo parte de la red original L^d, y la modifica con esta operación: componer dos restricciones disyuntivas L^d_{ik} y L^d_{kj}, y asignar a los elementos x de L^d_{ij} el mínimo (intersección) entre el grado de posibilidad $\max_{1\le p\le k} \pi^p_{ij}(x) > 0$ y el máximo para valores u,v / $x = u + v$ del mínimo entre los grados de posibilidad $\max_{1\le p\le k} \pi^p_{ik}(u) > 0$, $\max_{1\le p\le k} \pi^p_{kj}(v) > 0$ (composición). Es decir, eliminamos los valores x que no pertenecen al conjunto de soluciones, ya que para todas las tuplas T en las que aparezca x, $\pi_S(T)$ =0. Y si pertenecen al conjunto de soluciones, les asignamos una posibilidad que nunca será inferior al grado de posibilidad de una solución en la que participen. Como el conjunto de soluciones de la red resultante H^d contiene las mismas soluciones que L^d, y el grado de posibilidad de las soluciones no se altera, se cumple la definición de equivalencia entre ambas redes [6].

El algoritmo se detiene si encuentra una restricción vacía, $\pi_\varnothing$, al calcular una de las restricciones L^d_{ij}. Si una red RRTBD contiene una restricción vacía, es inconsistente, ya que no podremos encontrar un par de valores que satisfagan dicha restricción. Como la red calculada es equivalente al problema original, y las redes equivalentes tienen el mismo conjunto de soluciones, el problema original es inconsistente.

Lema 2.- Si el algoritmo de reducción serie-paralelo termina sin encontrar una restricción vacía y decide que la red es serie-paralelo, entonces la red es consistente.

Demostración.- En cada paso, el algoritmo de reducción serie-paralelo elimina una variable de la red, incorporando la información del camino de longitud dos que pasa por dicho nodo en la red reducida. En la red reducida no hay valores de la restricción L^d_{ij} prohibidos por restricciones con las variables previamente eliminadas.

Por ello, si llegamos a una red reducida con una sola restricción entre dos variables, no hay ninguna restricción vacía entre los nodos restantes. Además, la red es serie-paralelo, por lo que la red camino-consistente equivale a la red mínima.

Cada restricción calculada en la red reducida es camino-consistente respecto al camino que pasa por la variable eliminada. Por lo tanto, la restricción final es camino-consistente respecto a la red completa, luego es posible obtener una red equivalente camino-consistente.

Vamos a suponer que el algoritmo termina correctamente pero la red es inconsistente. En este caso, hay un camino C_{ij} entre dos variables X_i, X_j que prohíbe todos los valores permitidos por la restricción L_{ij}. Al reducir la red con el algoritmo, pueden darse dos situaciones que afecten a las variables de este camino:

a) Eliminar una variable ajena al camino, incorporando su información
b) Eliminar una variable del camino, conectando directamente dos variables del camino y calculando una restricción entre ellas

En ambos casos, se mantiene la información del camino C_{ij}, ya que prohibirá los mismos valores que el camino original, o incluso será más restrictivo. Llegará un momento en que tendremos sólo 3 nodos del camino C_{ij}, y pueden darse dos situaciones:

a) Eliminación de una variable ajena al camino. Puede darse sucesivamente.
b) Eliminación de una variable del camino

En el caso b), al calcular la restricción correspondiente, como el camino definía un bucle inconsistente, obtendremos una restricción vacía, por lo que el algoritmo se detendrá antes de decidir que la red es serie-paralelo, contradiciendo la hipótesis de partida.

Lema 3.- Si el algoritmo de reducción serie-paralelo termina sin encontrar una restricción vacía y decide que la red es serie-paralelo, entonces el arco final obtenido es la restricción mínima entre las dos variables a las que ha sido reducida la red.

Demostración.- La restricción final L_{ij} es camino-consistente con todas las restricciones de la red. Como la

red es serie-paralelo, si una restricción es camino-consistente respecto a la red, esta restricción es mínima.

El Lema 3 nos ofrece un método alternativo para calcular una restricción mínima en una red serie-paralelo. Basta con ejecutar el algoritmo sin incluir en la cola de nodos las dos variables cuya relación mínima deseamos calcular. Si la red es serie-paralelo, lo es respecto a cualquier par de variables, por lo que esto no impedirá que el algoritmo llegue al final.

Existe otro método propuesto por Dechter y col. [7] que consiste en ejecutar el algoritmo DPC, de camino-consistencia direccional, sobre la red tomando las variables en orden inverso al de su eliminación al comprobar si la red es serie-paralelo. Al ejecutar este algoritmo en el orden citado, se obtiene la restricción entre las dos primeras variables según dicho orden, así como una red que no necesita retroceso.

Para ver la utilidad de nuestra propuesta, vamos a comparar las complejidades de ambas aproximaciones, en función del número de operaciones de relajación que ejecutan. La complejidad de DPC es $O(nW^*(d)^2)$, siendo $W^*(d)$ el número máximo de padres de un nodo en el grafo inducido, que siempre será inferior a n. El grafo inducido es el grafo obtenido al aplicar DPC, que contendrá más restricciones que el original. En cambio, la complejidad del algoritmo de reducción serie-paralelo es de O(n), ya que el número máximo de pasos es igual al número de nodos, y en cada paso se ejecuta como máximo una operación de relajación.

Además, la red obtenida al aplicar el algoritmo de reducción serie-paralelo también es descomponible, si asignamos valores a las variables en el orden inverso al obtenido al reducir la red y comprobar que es serie-paralelo.

Lema 4.- Si una red es serie-paralelo, la red equivalente obtenida con las restricciones calculadas al aplicar el algoritmo SP^+ es descomponible en el orden inverso de eliminación de las variables al reducir la red.

Demostración.- En el caso de los nodos de grado uno, resulta evidente que siempre podemos encontrar un valor de la variable eliminada que sea compatible con la variable que se quedó en la red. Para el caso de los nodos de grado dos, tenemos que ver si es posible encontrar siempre un valor de la variable *k* que sea compatible con valores asignados previamente a las variables *x,j*.

Si partimos de la situación inicial, tenemos una restricción entre dos variables *i, j* que es camino-consistente respecto a un camino de longitud dos que pasa por un nodo *k*. Por la definición de camino-consistencia, una vez asignados valores *u, v* a las variables *i,j* que satisfagan la restricción L_{ij}, existe un valor *x* para la variable *k* que satisface las restricciones L_{ik} y L_{kj}.

Cuando incorporamos una nueva variable, estará en la misma situación que la inicial, formando un camino de longitud dos entre dos variables que tienen asignados valores que satisfacen una restricción camino-consistente respecto a este nuevo camino, luego podremos encontrar un valor para la nueva variable que satisfaga ambas restricciones. Debido a la topología de la red y el orden de asignación de las variables, sólo es necesario verificar este tipo de condiciones, por lo que la red es descomponible.

5 CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo hemos propuesto el algoritmo SP^+, una variante del algoritmo para determinar si una red es serie-paralelo que maneja la información de los arcos que representan las restricciones de un problema. El algoritmo presenta una complejidad lineal sobre el número de variables, y permite decidir la consistencia de una red de restricciones temporales borrosas disyuntivas, si esta es serie-paralelo. También puede detectar en algunos casos la inconsistencia aunque la red no sea serie paralelo. Además, calcula la restricción mínima entre dos variables de la red, y genera una red descomponible que no necesita retroceso en el orden inverso al obtenido al ejecutar el algoritmo.

La ventaja frente a otras aproximaciones de manejo de redes serie-paralelo propuestas en la literatura radica en la complejidad lineal de la propuesta, ya que la alternativa DPC presenta una complejidad que de hecho es cúbica. Esto hace que las redes disyuntivas serie-paralelo sean tratables con este enfoque.

Como trabajos futuros, podemos citar en primer lugar la implementación y evaluación de esta técnica para ver su aplicabilidad. También nos planteamos aprovechar la información obtenida durante la ejecución de SP+ para poder obtener una red camino-consistente completa, que equivale a la red mínima si la red es serie-paralelo, sin necesidad de ejecutar el algoritmo PC-2 de Mackworth [9], que presenta una complejidad elevada.

Otra línea de trabajo sería la búsqueda de otras subclases tratables dentro del modelo RRTBD en función de la topología de las redes, así como buscar métodos de descomposición de las redes que permitan aprovechar la tratabilidad de fragmentos que correspondan a diferentes subclases.

Referencias

[1] A. Bosch, M. Torres, I. Navarrete and R. Marín. Tree Decomposition of Disjunctive Fuzzy Temporal Constraint Networks. In *Proc. of International ICSC Congress on Computational Intelligence: Methods and Applications (CIMA'2001): Second International ICSC Symposium on Fuzzy Logic and Applications (FLA'2001)*, paper #1714-066, 7 pages, 2001.

[2] A. Bosch, M. Torres, and R. Marín. Reasoning with Disjunctive Fuzzy Temporal Constraint Networks. In *Proc. of 9th Int. Symposium on Temporal Representation and Reasoning (TIME'02)*. IEEE Computer Society Press, pages 36-43, 2002.

[3] A. Bosch, M. Torres y R. Marín. Evaluación de Algoritmos de Preproceso para Redes de Restricciones Temporales Borrosas Disyuntivas. En *Actas del XI Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF'02)*, pages 133-139, 2002.

[4] A. Bosch, F. Guil, C. Martinez, and R. Marín. Solving Fuzzy Temporal Problems without Backtracking. *Eurasia-ITC'02*, pages 91-103, 2002.

[5] A. Bosch, F. Guil, C. Martinez, and R. Marín. Series-Parallel and Tree-Decomposition Approaches for Fuzzy Constraint Networks. In *Proc. of the 8th Ibero-American Conference on Artificial Intelligence (IBERAMIA'02).* Lecture Notes in Artificial Intelligence, 2527:275-284, 2002.

[6] A. Bosch. *Modelos y Técnicas para la Gestión de Redes de Restricciones Temporales Borrosas. Tesis Doctoral. Universidad de Murcia. 2003.*

[7] R. Dechter and J. Pearl. Network-based heuristics for constraint-satisfaction problems. *Artificial Intelligence, 34:1-38, 1987.*

[8] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence, 49:61-95, 1991.*

[9] A. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1):99-118, 1977.

[10] R. Marin, S. Barro, A. Bosch, and J. Mira. Modeling time representation from a fuzzy perspective. Cybernetic & Systems, 25(2):207-215, 1994.

[11] U. Montanari. Networks of constraints: fundamental properties and applications to picture processing. *Information Science*, 7:95-132, 1974.

[12] I. Navarrete, R. Marín and M. Balsa. Redes de restricciones temporales disyuntivas borrosas, *En Actas de ESTYLF'95*, pp. 57-63, 1995.

[13] J.A. Wald and C.J. Colbourn. Steiner trees, partial 2-trees, and minimum IFI networks, *Networks 13:159-167*, 1983.

CONTROLADORES BORROSOS TEMPORALES DIFUMINADOS EVOLUTIVOS, APLICADOS AL ENCAMINAMIENTO ADAPTATIVO EN REDES DE COMUNICACIONES REALES

Manuel A. Gadeo Martos
Departamento de Electrónica
Universidad de Jaén
C/ Alfonso X, el Sabio nº. 28
23700 Linares (Jaén)
e-mail: gadeo@ujaen.es

Juan R. Velasco Pérez
Departamento de Automática
Universidad de Alcalá de Henares
Campus Universitario N-II, km. 31.5
28871 Alcalá de Henares (Madrid)
e-mail: juanra@aut.uah.es

Resumen

En este documento se realiza un análisis comparativo de las estrategias de razonamiento seguidas en controladores lógico borrosos (FLCs) y controladores borrosos temporales difuminados (FTFLCs). Partiendo de la problemática asociada a los FLCs se presentan los FTFLCs como solución, y como estrategia evolutiva para la adquisición de conocimiento se propone la utilización de algoritmos genéticos (AAGG) sobre FLCs y AAGG "dirigidos" sobre FTFLCs ($AAGGd_{FTFLCs}$). Para mostrar la utilidad de estas técnicas se han aplicado a la mejora del encaminamiento adaptativo en redes de comunicaciones reales, en las que se producen situaciones no ideales como pérdida de paquetes y caída de enlaces.

Palabras Clave: Controlador lógico borroso, controlador borroso temporal difuminado, algoritmo genético, redes de comunicaciones, encaminamiento adaptativo.

1 INTRODUCCIÓN

Muchas aplicaciones de los sistemas basados en conocimiento [10], necesitan manejar hechos que ocurren y varían con el tiempo. Por esta razón han surgido varios modelos para la representación y procesamiento del conocimiento temporal [1,2,3].

En los FTFLCs se generaliza el modelo propuesto en [2], introduciendo el concepto de difuminación temporal [8], que recoge una percepción no lineal del tiempo, que dota de mayor precisión fiabilidad y certeza a las observaciones y acciones cercanas en el tiempo. Efecto que se consigue gracias a la "difuminación" de los conjuntos borrosos (CCBB) temporales. En el motor de inferencia esta "difuminación" se traduce en una deformación de los CCBB temporales mediante unas transformaciones matemáticas, con el fin de generar los CCBB temporales transformados [8].

El documento se organiza de la siguiente manera: en la Sección 2 se muestran las similitudes y diferencias existentes en las estrategias de razonamiento seguidas en FLCs [10] y FTFLCs. En la Sección 3 se analiza la problemática asociada al control con FLCs, incluyéndose también una justificación teórica de las mejoras introducidas por los FTFLCs. En la sección 4 se describe el método de aplicación combinada de $AAGG_{FLCs}$ y $AAGGd_{FTFLCs}$, para conseguir una mejora en el aprendizaje. En la sección 5 se muestra los problemas asociados al encaminamiento en las redes de comunicaciones. En la sección 6 se aplica $AAGG_{FLCs}$ y $AAGGd_{FTFLCs}$ para optimizar el encaminamiento en redes de comunicaciones. Los resultados experimentales obtenidos son mostrados en la subsección 6.5. En la sección 7 se generalizan los resultados obtenidos.

2 ESTRATEGIAS DE RAZONAMIENTO SEGUIDAS EN FLCs Y FTFLCs

Los FTFLCs se pueden ver como una variante de los FLCs, por lo tanto sus estrategias de razonamiento tendrá una tendrán una parte común y una parte diferenciada.
a) La parte común se concreta en que en ambos controladores la salida del motor de inferencias se obtiene defuzificando el total de CCBB generados por las reglas disparadas, presentes en ese instante.
b) La parte específica a los FTFLCs se concreta en tres etapas [7,8]:

1. Realización de una transformación no lineal del tiempo, que posibilita la concentración de la región temporal próxima y la expansión de la región temporal lejana.
2. Generación de los CCBB temporales transformados.
3. Generación de las Componentes Contributivas.
La salida del motor de inferencia para cada variable de operación, en un instante cualquiera, (valor tomado por la variable de operación) se obtiene defuzificando el conjunto de sus Componentes Contributivas (CCBB truncados) existentes en ese instante [7,8].

3 JUSTIFICACIÓN DE LA UTILIDAD AÑADIDA POR LOS FTFLCs

3.1. PROBLEMÁTICA ASOCIADA A FLCs

En sistemas en los que la modificación de las variables de contexto se propaga con relativa facilidad en el tiempo [5], será necesario ajustar adecuadamente el valor dado a las variables de operación, con el objetivo de que al finalizar el intervalo temporal de propagación, de cada acción, se consiga corregir el desajuste de las variables de contexto.

Esta dificultad para asignar un valor adecuado a las variables de operación crecerá:
a) En aquellos sistemas, situados en un entorno ruidoso, en los que el ruido externo (acciones externas) modifica el valor tomado por las variables de operación. Disminuyendo así la probabilidad de una actuación correcta de las reglas [6].
b) En aquellos sistemas, en los que sobre las variables de contexto actúa un ruido externo, que impide conocer con precisión el valor tomado por éstas. Disminuyendo así la probabilidad de elegir, correctamente, el valor a dar a las variables de operación [7].

3.2. SOLUCIONES PROPUESTAS EN FTFLCs

Para salvar este problema, una buena solución puede consistir en introducir, en la lista de reglas de la base de conocimiento (BC), una serie de reglas con idéntico antecedente a aquella que se quiere complementar, presentando en los CCBB del consecuente valores que provocan la modificación de los estados del sistema en el sentido adecuado. Incluyéndose ahora el consecuente temporal, en el que sus CCBB tomarán valores que retrasan adecuadamente la aplicación de la acción de apoyo. Esta solución corrige en parte los desajustes, provocando un aumento en la velocidad de consecución de los objetivos marcados para las variables de contexto, mejorando así la "bondad" de la BC [7].

Para salvar la influencia del ruido, una buena solución puede consistir en aumentar el intervalo de actuación temporal de la acción correctora, sin aumentar su influencia global sobre el sistema. Efecto que se consigue deformando los CCBB temporales, aumentando su base y disminuyendo su altura, siendo su área igual a la del CB original. Deformación, que para compensar la disminución en la probabilidad de ubicar correctamente la acción correctora, deberá crecer a medida que aumenta el tiempo que separa la observación del sistema y la acción programada para su control [7].

4 PROCESO DE APRENDIZAJE. APLICACIÓN DE AAGG SOBRE FLCs Y DE AAGG "DIRIGIDOS" SOBRE FTFLCs

El procedimiento propuesto se compone de dos pasos:
1) Aplicar los $AAGG_{FLCs}$, sobre una población inicial compuesta por 20 BBC elegidas al azar, con el objetivo de obtener una buena BC no óptima [9].
2) Aplicar los $AAGGd_{FTFLCs}$ [9], partiendo de una población inicial compuesta por 20 BBC (derivadas de la obtenida en el paso anterior), para obtener una mejor BC.

Técnica a la que se ha llamado AAGG "dirigidos" sobre FTFLCs. Técnica que tiene el objetivo de mejorar la velocidad de aprendizaje, así como obtener mejoras sobre la aplicación separada de $AAGG_{FLCs}$ y $AAGG_{FTFLCs}$. Está técnica ha sido descrita detalladamente en [7] y [9], siendo el objetivo del presente trabajo su aplicación al encaminamiento en redes de comunicaciones, a fin de conseguir una mejora en su calidad de servicio.

5 EL PROBLEMA DEL ENCAMINAMIENTO ADAPTATIVO EN REDES DE COMUNICACIONES

Uno de los problemas principales en la redes de comunicación consiste en definir una política de encaminamiento eficiente. La política más simple hace uso de un algoritmo de encaminamiento de "camino más corto" [4], que consiste en enviar paquetes a través de los caminos compuestos por un menor número de saltos. Esta política no es optima para todas las condiciones de tráfico en la red, debido a que se generan rutas preferidas que se congestionarán rápidamente. En está situación se formarán largas colas de paquetes, y como consecuencia un aumento en el tiempo de encaminamiento.

Muchas de las redes y protocolos asociados (por ejemplo: Arpanet, y su protocolo de encaminamiento OSPF2),

emplean encaminamiento adaptativo distribuido, basado en algoritmos tradicionales como los de Dijkstra o el de Ford y Fulkerson. Estos algoritmos se ejecutan periódicamente, tomando la información sobre las condiciones del tráfico y del estado de los enlaces, que llega, procedente de los nodos adyacentes o del resto de nodo de la red. Teniendo en cuenta que el intercambio de información de estatus de red consume ancho de banda, el valor del periodo de actualización debe ser suficientemente grande como para no cargar en exceso la red. En estos algoritmos, las rutas que comúnmente soportan una carga fuerte son evitadas para encaminar, así como las rutas que soportan una carga débil serán seleccionadas ahora para encaminar. Lo que causa una oscilación no esperada en el proceso de encaminamiento. La naturaleza distribuida del sistema a controlar introduce dificultades adicionales al proceso de encaminamiento, achacables a que las medidas de estatus sufrirán un retraso (debido a su propagación en enlaces), y además éstas no estarán disponibles continuamente, siendo necesario realizar un muestreo a intervalos finitos [11].

6 AAGG SOBRE FLCs Y AAGG "DIRIGIDOS" SOBRE FTFLCs, APLICADOS AL ENCAMINAMIENTO ADAPTATIVO DISTRIBUIDO

6.1. DESCRIPCIÓN DE LA RED Y DEL TRÁFICO OFRECIDO

Para comprobar la viabilidad y conveniencia de la aplicación de los controladores borrosos temporales difuminados, aplicados al encaminamiento adaptativo distribuido, se ha simulado el comportamiento de la red que se muestra el la figura 1.

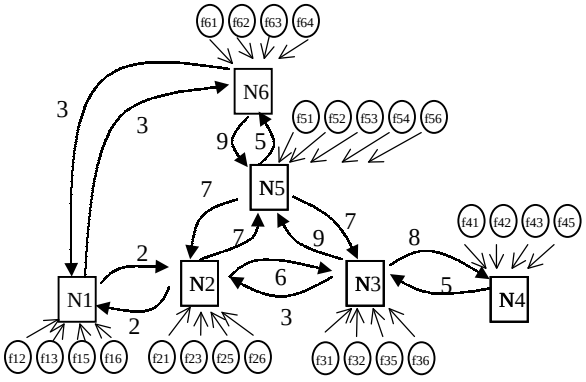


Figura 1. Modelo de red.

En este modelo se puede observar 6 nodos con 26 fuentes generadoras de paquetes (f_{ij} , "i" es el nodo origen y "j" es el nodo destino de los paquetes), siendo el intervalo de simulación de 10 segundos. Todos los enlaces presentan un ancho de banda de 10 kbps, y cada enlace se ha modelado como un sistema de cola M/M/1, siendo el número situado sobre cada enlace, la métrica inicial.

En cada fuente de paquetes, el tráfico ofrecido se modela mediante dos procesos estocásticos:
1. El Tiempo entre llegada de paquetes, que presenta distribución exponencial, con valor medio "τ", en este caso τ=0,1 s.
2. El Tiempo de servicio demandado por cada paquete, que presenta una distribución exponencial, con valor medio "s". En el ejemplo que se presenta "s" varía desde 0,02 hasta 0,09 s., y por tanto, el tamaño de paquete variará desde 200 hasta 900 bits.

6.2. EVALUACIÓN DE LAS PRESTACIONES DE UNA RED DE COMUNICACIONES

La política de encaminamiento debe orientarse a alcanzar la "calidad de servicio" deseada. En una red de conmutación de paquetes los requerimientos de "calidad de servicio" vienen dados, entre otro parámetros, por: el retardo medio sufrido por los paquetes, la varianza en el retardo (jitter), la tasa de pérdidas, el ancho de banda. En este caso sólo se tendrá en cuenta el retardo medio, el jitter y la tasa de pérdidas, para obtener así una evaluación de las prestaciones de la red. Para evaluar las prestaciones de la red se ha usado la siguiente expresión:

$$E = 0{,}7 \cdot P + 0{,}3 \cdot (1 - (0{,}8 \cdot Rn + 0{,}2 \cdot Vn))$$

E: función de evaluación. P: (paquetes llegados con éxito / total paquetes) (%). Rn: retardo medio normalizado. Vn: varianza del retardo normalizada. El peso dado a cada parámetro, lo ha sido en función de las prioridades establecidas a la hora de plasmar el concepto de calidad de servicio.

Para obtener una correcta evaluación de las prestaciones de la red, se simula el comportamiento de la red con:
a) Ocho niveles de carga de tráfico, iguales en todas las fuentes (ρ): 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9.
b) Treinta semillas. Usadas para generar dos secuencias de números pseudoaleatorias, a fin de modelar el tráfico.

En el experimento propuesto, para cada nivel de tráfico se ha realizado treinta simulaciones (una por cada semilla). Para cada nivel de carga la evaluación global es el valor medio de las treinta evaluaciones individuales.

6.3. COMPARACIÓN DE ESTRATEGIAS DE ENCAMINAMIENTO

Para realizar la comparación de las prestaciones de la red, se han simulado cuatro estrategias de encaminamiento:

a) Encaminamiento estático de camino más corto (SPR). Los paquetes son encaminados a lo largo de rutas fijadas, para cada sesión de tráfico. Rutas que son calculadas al comienzo de la sesión aplicando el algoritmo de Dijkstra. Se trata de un método no adaptativo.

b) Encaminamiento adaptativo de camino más corto (APR). En cada nodo se ejecuta el algoritmo de Dijkstra cada T segundos, tomando como parámetros de entrada el retardo medio de los paquetes, así como la varianza del retardo, en cada enlace adyacente, obtenidos durante el intervalo de medida (ver figura 2). Para obtener la métrica del enlace se usa la siguiente función:

$C = 0{,}8 \cdot Rn + 0{,}2 \cdot Vn$

C: métrica del enlace que se aplicará durante el intervalo de actuación de la misma. Rn: retardo medio de paquetes normalizado. Vn: Varianza del retardo normalizada.

c) Control borroso evolutivo (FLC). La métrica del enlace es la variable de operación del motor de inferencia. Las variables de contexto son el retardo medio de los paquetes y la varianza del retardo, en el enlace considerado, calculadas sobre el intervalo de medida. En cada nodo se calcula la métrica de los enlaces adyacentes, cada T s., que va a ser una constante durante el intervalo de actuación de la misma. A partir de la tabla de métricas, cada nodo ejecuta el algoritmo de Dijkstra y rellena su tabla de encaminamiento, con las rutas de camino más corto, calculadas cada T segundos.

d) Control borroso temporal difuminado evolutivo (FTFLC). La métrica del enlace es la variable de operación del motor de inferencia temporal difuminado. Las variables de contexto son el retardo medio de los paquetes y la varianza del retardo, en el enlace considerado, calculadas sobre el intervalo de medida. Aquí todos y cada uno de los nodos calcula las métricas, de todos los enlaces de la red y aplica el algoritmo de Dijkstra, en cada instante de simulación. Rellenando así las tablas de encaminamiento con las rutas de camino más corto obtenidas. De lo dicho se deduce que la métrica de cualquier enlace puede cambiar en cualquier instante de tiempo y por tanto el encaminamiento.

Al finalizar el intervalo de medida, cada nodo genera un paquete de control unicast que envía a los demás nodos de la red, conteniendo información de estado de los enlaces adyacentes, consistente en:
a) La métrica (encaminamiento APR y FLC)
b) La media y varianza del retardo (encaminamiento FTFLC).

Después del "intervalo de transmisión" los paquetes de control deberán haber llegado a sus nodos destinos, actualizando éstos su información de estado, que estará vigentes durante el "intervalo de actuación" (método de encaminamiento de estado de enlaces).

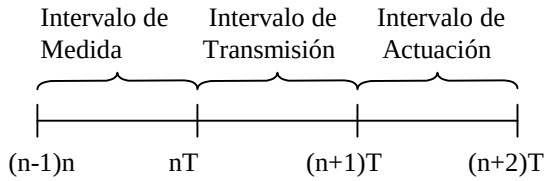


Figura 2. Intervalos de muestreo asociados al encaminamiento.

6.4. APRENDIZAJE EN EL PROCESO DE ENCAMINAMIENTO USANDO AA.GG. "DIRIGIDOS" SOBRE FTFLCS

En este caso el objetivo del proceso de aprendizaje es obtener una buena BC, con la que se pueda optimizar el proceso de encaminamiento, para cualquier nivel de carga en tráfico ofrecido. La BC buscada se obtendrá mediante la aplicación de $AAGG_{FLCs}$ y $AAGGd_{FTFLCs}$ a redes dotadas con encaminadores FLC y FTFLC respectivamente. Para ello en cada evaluación de las prestaciones de la red (evaluación de la bondad de la BC obtenida) se realizarán cinco simulaciones de red. Cada simulación de la red se ejecutará con un nivel de carga de red (ρ) distinto (ρ: 0.2, 0.35, 0.5, 0.65, 0.8). La evaluación final de cada BC se obtendrá como la media aritmética de los cinco experimentos.

6.5. RESULTADOS EXPERIMENTALES

Para la obtención de resultados, el sistema experimental utilizado ha sido la simulación de la red comentada. Sistema que presenta una persistencia temporal razonablemente alta, en los efectos provocados por las acciones de control sobre las variables de contexto [5].

Así, en primer lugar se ha realizado el proceso de aprendizaje genético, dividido en dos partes:
a) Los $AAGG_{FLCs}$ son aplicados para obtener una buena BC_{FLCs} sobre una red con encaminadores FLCs.
b) Los $AAGGd_{FTFLCs}$ son aplicados para obtener una buena BC_{FTFLCs} sobre una red con encaminadores FTFLCs.
Como resultado de este proceso se ha obtenido una BC, cuyos parámetros de difuminación toman valores que dan soporte al concepto de difuminación temporal [8].

En segundo lugar, se ha realizado una simulación de la red para cada una de las cuatro estrategias de encaminamiento propuestas, para así evaluar el comportamiento (prestaciones) de la red con diversos valores del intervalo de muestreo T.

Este bloque de 4 simulaciones, se ha realizado tres veces, cada unas de ellas con circunstancias distintas, a saber:
a) En condiciones de idealidad, no habiendo ni errores en paquetes, ni pérdida de paquetes en enlaces, ni caídas de enlaces. (comportamiento ideal).
b) No existiendo ni errores, ni pérdida de paquetes, pero si caída de enlaces. Concretamente se ha simulado la caída del enlace que une el nodo 1 con el 2, entre los instantes 3,020 s. y 7,204 s. (comportamiento real 1).
c) No produciéndose caída de ningún enlace, pero si errores y/o pérdidas en paquetes de control, en diversos enlaces de la red. Concretamente se ha simulado la pérdida de un 10 % de los paquetes de control distribuidos aleatoriamente, según una distribución uniforme. (comportamiento real 2).

La simulación de red en condiciones de comportamiento:
a) Real 1, conlleva la adición de un ruido sobre la variable operación.
b) Real 2, conlleva la adición de un ruido sobre las variables de contexto.

En las tablas adjuntas se presentan los resultados obtenidos, concretamente para un experimento (con T=0,5 s.) significativo de los 20 realizados (tomando diversos valores de T). Cada tabla muestra una evaluación comparativa, del comportamiento de la red, para diferentes niveles de carga, y diferentes estrategias de encaminamiento. Resaltándose para cada nivel de carga la "mejora", en prestaciones (en %) , aportada por la estrategia de control temporal difuminado sobre la de control borroso clásico ($(E_{FTFLC}-E_{FLC})/E_{FLC}$).

Carga	SPR	APR	FLC	FTFLC	Mejora (%)
0,2	0,895	0,648	0,919	0,929	1,088
0,3	0,800	0,590	0,870	0,879	1,034
0,4	0,753	0,591	0,819	0,838	2,320
0,5	0,720	0,585	0,781	0,802	2,689
0,6	0,693	0,590	0,752	0,774	2,926
0,7	0,672	0,598	0,729	0,751	3,018
0,8	0,655	0,597	0,714	0,735	2,941
0,9	0,641	0,597	0,699	0,721	3,147

Tabla 1. Evaluación comparativa del comportamiento de la red ideal.

Carga	SPR	APR	FLC	FTFLC	Mejora (%)
0,2	0,852	0,620	0,848	0,866	2,123
0,3	0,757	0,560	0,764	0,801	4,843
0,4	0,694	0,554	0,721	0,756	4,854
0,5	0,643	0,541	0,687	0,717	4,367
0,6	0,608	0,533	0,661	0,689	4,236
0,7	0,581	0,523	0,642	0,666	3,738
0,8	0,559	0,514	0,621	0,645	3,865
0,9	0,534	0,515	0,603	0,629	4,312

Tabla 2. Evaluación comparativa del comportamiento de la red real (con comportamiento 1),

Carga	SPR	APR	FLC	FTFLC	Mejora (%)
0,2	0,882	0,634	0,911	0,924	1,427
0,3	0,784	0,574	0,843	0,862	2,254
0,4	0,737	0,569	0,796	0,821	3,141
0,5	0,703	0,569	0,756	0,786	3,968
0,6	0,676	0,569	0,733	0,759	3,547
0,7	0,656	0,574	0,714	0,739	3,501
0,8	0,640	0,568	0,696	0,721	3,592
0,9	0,626	0,575	0,681	0,704	3,377

Tabla 3. Evaluación comparativa del comportamiento de la red real (con comportamiento 2).

En primer lugar, del análisis de los resultados experimentales mostrados, para cualquier nivel de carga de tráfico, tanto para redes con comportamiento ideal, como para aquellas que presenta un comportamiento real, se observa que:
a) Hay una mejora en las prestaciones de la red, obtenidas al aplicar encaminadores FLC, comparado con el uso de encaminadores SPR. Esta mejora es debida a que los encaminadores FLC, al comienzo de cada intervalo de actuación de la métrica, para cada sesión de tráfico, pueden modificar las rutas, pudiendo evitar así que los enlaces alcancen el estado de congestión.
b) Hay una mejora en las prestaciones de la red, obtenidas al aplicar encaminadores FTFLC, comparado con el uso de encaminadores FLC. Esta mejora es debida a que en los encaminadores FLC, durante cada intervalo de actuación de la métrica, y para cada sesión de tráfico, el encaminamiento permanece invariable. En cambio, en los encaminadores FTFLC, en cada intervalo de actuación, y para cada sesión de tráfico, pueden existir varias rutas. Con este método es posible evitar la congestión de los enlaces, en cualquier instante (no necesariamente en el instante de actualización o muestreo), pudiendo el nodo reencaminar los paquetes a través de otro enlace, menos congestionado, sin tener que esperar.
c) Hay una mejora en las prestaciones de la red, obtenidas al aplicar encaminadores SPR, comparado con el uso de encaminadores APR (en este caso, con tamaño de paquete unicast de control = 200 b.). Los encaminadores APR presentan una mayor adaptabilidad a la congestión que los encaminadores SPR, pero en este caso el efecto de una carga excesiva de paquetes de control, disminuye las prestaciones de la red.

Además se puede observar, para cualquier nivel de carga de tráfico, que la mejora en prestaciones atribuida a la utilización de nodos de encaminamiento FTFLC frente a los FLC, es mayor en las redes con comportamiento real (con caída de enlaces y pérdida de paquetes de control), que en las redes con comportamiento ideal.

1) En el caso de caída de un enlace, esta mejora extra es debida a que en las redes con encaminamiento FTFLC, una buena BC, generará, a lo largo del intervalo actuación, las métricas necesarias como para que las rutas

se modifiquen en algún(os) instante(s) intermedio(s) de dicho intervalo. Y por tanto sólo se perderán los paquetes ofrecidos al enlace desde que se cae éste hasta que se produce un cambio en la tabla de encaminamiento, que desvíe los paquetes. En cambio en los sistemas FLC se perderán todos los paquetes que son ofrecidos al enlace, desde su caída hasta que llega el siguiente instante de actualización, al finalizar el intervalo de actuación de la métrica.

2) En el caso de pérdida o error de un paquete de control (durante intervalo de transmisión), la información contenida en él, que se refiere al "estado" de los enlaces con origen en el nodo que genera el paquete (calculada durante el intervalo de medición), no llega a su nodo destino, y como resultado no se actualizará, en ellos, la parte de la tabla de métricas, relativa a dichos enlaces.

Si el encaminamiento del sistema es FLC, el efecto de la no actualización de sólo una parte de la tabla de métrica (correspondiente a todos los enlaces salientes de un nodo) en uno o varios nodos de la red, se puede traducir en el aumento de la congestión de uno o varios enlaces (si ya se han cargado en el intervalo de actuación anterior) o en la acentuación de la descarga de uno o varios enlaces (que puede provocar el aumento de la carga de otros enlaces ya cargados en el intervalo de actuación anterior).

En las redes con encaminamiento FTFLC, una buena BC, generará, a lo largo del intervalo actuación, las métricas necesarias como para que se modifiquen las rutas en algún(os) instante(s) intermedio(s) de dicho intervalo. De manera, que aunque un paquete de control no informe a su nodo destino de los nuevos "estados" de uno o varios enlaces, y por tanto se prorroguen las métricas aplicadas en el anterior intervalo de actuación, se pueda producir (también en este nuevo intervalo de actuación) un cambio en las rutas, que eviten los efectos nocivos comentados en el párrafo anterior. A ello se debe la mejora extra obtenida con el empleo de encaminamiento FTFLC frente a FLC.

7 CONCLUSIONES

Del análisis de los resultados experimentales se deduce que:

a) El uso de sistemas de encaminamiento FTFLC, consigue un aumento en las prestaciones de una red de comunicaciones, comparado con la utilización de sistemas de encaminamiento FLC. Esta ganancia o mejora se acentúa en condiciones reales, es decir, redes susceptibles a caídas de enlaces, y errores o pérdidas en paquetes, que introducen ruido en el sistema.

b) El uso de sistemas de encaminamiento FLC consigue un aumento en las prestaciones de una red de comunicaciones, comparado con la utilización de otros métodos de camino más corto.

c) El uso de $AAGGd_{FTFLCs}$ frente $AAGG_{FLCs.}$, aplicado al encaminamiento en redes de comunicaciones, produce una mejora en la "bondad" final de la BC obtenida, lo que se traduce en una mejora en las prestaciones de la red.

Referencias

[1] S. Barro, R. Marin, J. Mira, and A.R. Patón, "A model and a language for the fuzzy representation and the handling of time", Fuzzy Sets Systems, 61, 153-175 (1994)

[2] Carse B., Fogarty T.C., Munro A. "Artificial evolution of fuzzy rules base which represent time: a review of the temporal fuzzy classifier system". International journal of intelligent systems. 1998. Vol 13. Pag. 905-927.

[3] D. Dubois and H. Prade, "Processing fuzzy temporal Knowledge", IEEE Trans. Systems, Man Cybernet., 19, 729-744 (1989).

[4] Floyd, R.W., "Algorithm 97 (Shortest Path)". Communications of the ACM, vol 5(6). 1962.

[5] Gadeo M.A., Magdalena L., "Ámbito de aplicación de los controladores lógico borrosos (FLC), controladores temporales borrosos (TFLC) y controladores borrosos temporales difuminados (FTFLC)", Estylf, 2000, pp 309-314.

[6] Gadeo M.A., Magdalena L., "Estrategias evolutivas para la adquisición de conocimiento en controladores borrosos temporales difuminados, en entornos ruidosos", Caepia, 2001, pp 1177-1186.

[7] Gadeo M.A., Velasco J.R., " Evolutionary strategies to achieve knowledge in faded temporal fuzzy logic controllers in systems with noisy sensors". *Ipmu 2002*. Pag. 853-858.

[8] Gadeo M.A., Velasco J.R., "Faded Temporal Fuzzy Logic Controller (FTFLC) and Temporal Fuzzy Logic Controller (TFLC). Comparative study". *Ipmu, 2000*. Vol. II. pp. 732-737.

[9] Gadeo M.A., Velasco J.R., "Optimization of evolutionary strategies to achieve knowledge in faded temporal fuzzy logic controllers". *Eusflat 2001*. Pag. 156-159.

[10] Pedrycz W. "Fuzzy control and fuzzy systems". Research Studies Press Ltd., second, extended edition, 1993.

[11] Stalling W., "Data and Computer Communications". Macmillan, New York, 1994

ImmoSoftWeb: Aplicación de Gestión Inmobiliaria Vía Web Basada en Tecnologías Difusas

Carlos D. Barranco
carlosd@decsai.ugr.es

Jesús Campaña
jesuscg@fedro.ugr.es

Juan M. Medina
medina@decsai.ugr.es

Dept. de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada
Daniel Saucedo Aranda, s/n 18071 Granada (España)

Resumen

El presente artículo analiza la aplicación ImmoSoftWeb (ISW), una aplicación web que aprovecha las ventajas que aporta la aplicación de la teoría de Conjuntos Difusos al campo de la gestión inmobiliaria. ISW está desarrollada sobre una FRDB (Fuzzy Relational DataBase), empleando inicialmente un prototipo de servidor llamado FSQL, el cual proporciona a la aplicación capacidades para el manejo de datos y consultas difusas. Sin embargo, ISW es independiente de la FRDB subyacente ya que emplea una capa de abstracción basada en XML. Adicionalmente, en el presente artículo se analiza la forma en la que ciertos atributos de inmuebles pueden ser representados de forma difusa y como una consulta difusa puede ser empleada para expresar los requisitos de los clientes del sector.

Palabras Clave: Conjuntos Difusos, Imprecisión, Base de Datos, Aplicación, Gestión Inmobiliaria.

1. Introducción

En la actualidad, podemos encontrar cierta cantidad de propuestas de modelos de bases de datos relacionales difusas (FRDB) así como algunas implementaciones basadas en éstas [1, 2, 3, 4, 5, 6]. Se muestra, por tanto, una importante necesidad de buscar nuevos campos de aplicación para resolver problemas cotidianos utilizando dichos modelos. ISW [7] es una aplicación de gestión inmobiliaria que utiliza las ventajas aportadas por trabajos e investigaciones realizadas en el campo de las FRDB.

Como objeto de nuestra investigación elegimos el área de la gestión inmobiliaria debido a que los atributos de los inmuebles se prestan fácilmente a un tratamiento difuso, por el alto grado de imprecisión en sus valores.

ISW nos permite almacenar ciertos atributos de inmuebles conteniendo información imprecisa, permitiéndonos además la definición de consultas flexibles que nos conduzcan a obtener, desde la FRDB, no solo los inmuebles que se ajusten perfectamente a los requisitos, si no aquellos inmuebles que se asemejen a los que constituyen el objeto de nuestra búsqueda.

Nuestra aplicación ha sido implementada sobre un servidor de SQL difuso [8, 9, 10, 11] basado en el modelo GEFRED (a GEneralized model for Fuzzy RElational Databases)[4]. La FRDB empleada proporciona un lenguaje de consulta, Fuzzy SQL (FSQL) [10], que permite la interacción entre la aplicación y la FRDB. En el desarrollo de ISW se han empleado diversas tecnologías como JAVA, páginas JSP, XML y tecnologías adyacentes, como veremos más adelante.

La organización del presente artículo es la siguiente: En la sección 2 se muestra la manera en que la aplicación representa y maneja información difusa, los atributos de los inmuebles que serán tratados de forma difusa y las condiciones que podrán aplicar los usuarios de la aplicación. La sección 3 muestra la arquitectura de la aplicación. En la sección 4 se expone la forma en que la aplicación procesa las consultas flexibles, especificadas por los usuarios, para generar como salida el conjunto de inmuebles que se ajustan a éstas. Finalmente, la sección 5 expone las conclusiones finales y futuras líneas de investigación.

2. Manejo de Datos Difusos por la Aplicación

Los elementos empleados para el manejo de datos difusos pueden adoptar diferentes representaciones. En nuestro caso utilizaremos la representación definida en [5], que describe los siguientes tipos de datos:

Tabla 1: Tipos asociados a atributos de inmuebles.

Tipo Difuso	*Atributos*
Tipo 1	Habitaciones, Plantas
Tipo 2	Precio, Superficie, Antigüedad
Tipo 3	Clase, Orientación, Vistas, Iluminación, Conservación
Tipo Geográfico	Ubicación

1. *Datos precisos:* Estos utilizan la representación de datos proporcionada por el RDBMS anfitrión, y soportan consultas difusas sobre ellos. Estos datos son conocidos como "Datos de *Tipo 1*".

2. *Datos imprecisos sobre un dominio ordenado:* Este tipo de datos tienen asociada una función de pertenencia. En este caso representaremos dicha función de pertenencia haciendo uso de una distribución de posibilidad trapezoidal, que nos permite representar datos imprecisos como son los intervalos, con o sin margen, valores aproximados y valores precisos, obviamente. Por simplicidad llamaremos a estos datos "Datos de *Tipo 2*".

3. *Datos con analogía sobre un dominio discreto:* Este tipo de datos imprecisos se basan en dominios discretos en los que existen definidas relaciones de proximidad. Este grupo puede presentar datos como son escalares simples o distribuciones de posibilidad en dominios discretos. Denominaremos a este tipo de datos "Datos de *Tipo 3*".

Utilizaremos sobre los anteriores tipos de datos los siguientes operadores relacionales difusos:

1. *"Igual que" difuso (FEQ)*: Basado en medidas de posibilidad cuando es aplicado sobre datos de *Tipo 1* y *Tipo 2*, y basado en relaciones de proximidad si es utilizado sobre datos de *Tipo 3*.

2. *"Menor que" difuso (FLT), "Mayor que" difuso (FGT), "Mayor o igual que" difuso (FGEQ) y "Menor que o igual" difuso (FLEQ)*: Estos operadores están basados en extensiones difusas de los operadores relaciones clásicos, siendo solo aplicables sobre datos de *Tipo 1* y *Tipo 2*.

2.1. Atributos Susceptibles de Tratamiento Difuso

Consideraremos algunos de los atributos de inmuebles que pueden ser modelados utilizando los tipos difusos descritos. Estos son los presentes en la tabla 1.

Tabla 2: Relación de proximidad definida sobre el dominio *Clase*.

Piso	*Casa*	*Duplex*	*Ático*	**Clase**
0.75	0.3	0.2	0.75	*Apartamento*
	0.3	0.3	0.75	*Piso*
		0.75	0.1	*Casa*
			0.1	*Duplex*

Los atributos de *Tipo 1* sólo almacenan datos precisos. Por ejemplo, el atributo *Habitaciones* almacena el número de habitaciones del inmueble: 2, 5, ...

A los atributos de *Tipo 2* se les permite almacenar datos imprecisos, utilizando una representación trapezoidal de los mismos. Por ejemplo, el atributo *Precio* almacena el rango de precios en los que está el inmueble: "entre 1000 y 1500" o "exactamente 1250".

Todos los atributos de *Tipo 3* poseen asociados un dominio y una relación de proximidad. Este dominio, por ejemplo, para el atributo *Clase* lo conforman los escalares: "Apartamento", "Piso", "Casa", "Duplex" y "Ático". La relación de proximidad asociada para dicho atributo se muestra en la tabla 2.

Los atributos de *Tipo Geográfico* han sido diseñados para almacenar la ubicación del inmueble como un par de coordenadas (x, y) (una pareja de datos de *Tipo 1*), sobre la que se podrán imponer condiciones geográficas flexibles, como expondremos más adelante.

2.2. Representación de Condiciones Flexibles en el Interfaz de Usuario

A continuación, detallaremos las posibilidades que la interfaz de usuario ofrece para definir condiciones flexibles sobre los atributos de inmuebles.

2.2.1. Condiciones Numéricas Flexibles.

Las Condiciones Numéricas Flexibles son condiciones parametrizables por el usuario que pueden ser aplicadas sobre los atributos de *Tipo 1* o *Tipo 2* de los inmuebles presentes en la base de datos. La interfaz de usuario permite utilizar las siguientes:

1. *Condición "Entre flexible":* Es la variante flexible de la condición "entre" clásica, añadiendo un margen al intervalo. Será definida determinando un límite inferior (l) y superior (u) y un margen (m), y se aplicará utilizando la condición difusa $a\ FEQ\ [l-m, l, u, u+m]$, donde a, en futuras referencias, es el atributo sobre el que la condición flexible es aplicada, y $[\alpha, \beta, \gamma, \delta]$ representa una distribución trapezoidal de posibilidad.

Tabla 3: Cualificadores para condiciones de proximidad sobre atributos de *Tipo 3*.

Cualificador	*Umbral*	*Cualificador*	*Umbral*
Únicamente	1	Muy	0.9
Bastante	0.75	Medianamente	0.5
Poco	0.25	Cualquiera	0

2. *Condición "Aproximado a":* Este tipo de condición flexible aporta un mayor grado de libertad a la condición clásica "igual a". La condición se construye utilizando un valor central (v), al cual deseamos aproximarnos, junto con un margen (m), para especificar la flexibilidad de la aproximación. La presente condición flexible será aplicada como la condición difusa $a\ FEQ\ [v-m, v, v, v+m]$.

3. *Condición "Mayor que flexible":* Esta clase de condición flexible relaja la condición clásica "mayor que". La condición será definida con un límite inferior (l) y un margen (m), y será aplicada utilizando la condición difusa $a\ FGT\ [l-m, l, l, l]$.

4. *Condición "Menor que flexible":* Este tipo de condición es la versión flexible de la condición clásica "menor que". Debe de ser definida a través de un límite superior (u) y un margen (m). La condición flexible será aplicada usando su equivalente difuso $a\ FLT\ [u, u, u, u+m]$.

2.2.2. Cualificadores para Condiciones de Proximidad.

Este grupo de cualificadores está diseñado para ser empleado en condiciones de proximidad sobre atributos de *Tipo 3*. Utilizaremos estos cualificadores para modelar conceptos semánticos usados por el ser humano para especificar hasta que punto deben de parecerse los valores obtenidos en la consulta con el valor seleccionado por el usuario sobre un atributo.

Los cualificadores para condiciones de proximidad funcionan estableciendo un umbral, un nivel mínimo, para el grado de proximidad entre el escalar especificado en la consulta del usuario y el escalar almacenado en el atributo sobre el que es aplicada dicha condición.

Estas condiciones de proximidad son especificadas como condiciones difusas de igualdad del tipo $a\ FEQ\ v\ THOLD\ t$, donde a es el atributo afectado por la condición, v es el escalar seleccionado por el usuario y t es el valor de umbral establecido por el cualificador elegido por el usuario.

Los cualificadores disponibles y sus valores de umbral asociados se muestran en la tabla 3.

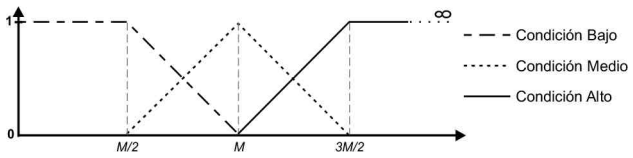


Figura 1: División del dominio subyacente por las condiciones adaptativas.

2.2.3. Condiciones Adaptables al Contexto.

Las condiciones adaptables al contexto son un grupo especial de condiciones numéricas flexibles, disponibles para su uso por parte de los usuarios, con una particularidad: son sensibles al contexto.

Utilizaremos este tipo de condiciones para realizar consultas sobre la base de datos de inmuebles utilizando conceptos humanos como "barato", "razonable" o "caro" (si nos referimos al precio de un inmueble) , o conceptos como "pequeña", "mediana" o "grande" (si nos referimos al tamaño de un inmueble).

Decimos que estas condiciones son adaptables al contexto debido a que estos conceptos humanos que modelan no son matemáticamente constantes, si no que su valor depende del contexto en el que se aplican. Analicemos esto utilizando un ejemplo: Una "casa unifamiliar *barata* en las afueras" no tiene un precio igual que un "piso *barato* en el centro de la ciudad". Está claro que ambos son "*baratos*", pero no son iguales numéricamente, sus rangos de precios son diferentes. Como podemos observar, debemos tener en consideración el contexto en el cual el inmueble está envuelto (p.ej. "un piso en el centro de la ciudad" o "una casa en las afueras") para obtener una representación numérica del rango de precios "*barato*".

Independientemente del atributo que consideremos, dispondremos de los siguientes rangos conceptuales:

1. *Bajo:* Incluye conceptos específicos como "barato" (precio) o "pequeño" (tamaño).

2. *Medio:* Incluye conceptos específicos como "módico" (precio) o "mediano" (tamaño).

3. *Alto:* Incluye conceptos específicos como "caro" (precio) o "grande" (tamaño).

Estos tres rangos conceptuales nos permiten dividir el dominio subyacente de los atributos numéricos (*Tipos 1 y 2*) en tres regiones difusas. Los límites de las regiones se calcularán en base al valor medio (M) del atributo, calculado sobre el "contexto" de la condición.

Para aplicar este tipo de condiciones, seguimos los siguientes pasos para obtener una representación trape-

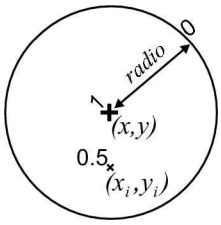


Figura 2: Área definida en una condición geográfica y su función de pertenencia.

zoidal numérica, basada en el valor medio del atributo en el contexto, de este tipo de condiciones:

1. *Obtención del contexto:* Aplicamos todas las condiciones definidas en la consulta especificada por el usuario exceptuando las condiciones adaptativas incluidas en ésta. Las tuplas obtenidas de la operación anterior serán consideradas el *contexto* impuesto por la consulta del usuario.

2. *Obtención del valor medio:* Calcularemos el valor medio, de los valores presentes en las tuplas pertenecientes al *contexto*, para aquellos atributos afectados por condiciones adaptativas.

3. *Creación de las representaciones trapezoidales de las condiciones adaptativas:* Crearemos representaciones trapezoidales concretas de todas las condiciones adaptativas presentes en la consulta, usando para ello el valor medio (M) de cada atributo sobre el que la condición adaptativa es aplicada, como muestra la figura 1.

4. *Obtención de los resultados de la consulta:* Una vez que están definidas todas las condiciones adaptativas como trapecios concretos, aplicaremos todas las condiciones presentes en la consulta para obtener el resultado final de ésta.

2.2.4. Condiciones Geográficas Difusas.

Hemos creado este tipo de condiciones para poder establecer condiciones flexibles sobre la ubicación del inmueble dentro de su localidad. Para establecer dicho tipo de condición el usuario especificará los parámetros x, y y $radio$. A partir de estos tres valores se define un área circular ubicada en el punto (x, y) y con radio de longitud $radio$, como muestra la figura 2.

Utilizaremos la fórmula 1 para calcular el grado de pertenencia para cada inmueble situado dentro del área especificada por la condición, donde (x, y) son las coordenadas del centro del área definida en la condición, (x_i, y_i) son las coordenadas de la ubicación del inmueble evaluado, $dist(a, b)$ representa la distancia Euclídea entre los puntos a y b, y $radio$ es el radio del área geográfica definida en la condición.

$$CDEG((x,y),(x_i,y_i)) = \frac{dist((x,y),(x_i,y_i))}{radio} \quad (1)$$

2.2.5. Grado de Ajuste del Inmueble a la Consulta.

El grado de ajuste del inmueble a la consulta es el valor que nos indica en qué medida se adecúa el inmueble a las condiciones establecidas en dicha consulta. Este grado se calcula utilizando una media ponderada de los grados de pertenencia de la tupla que representa el inmueble, para cada una de las condiciones presentes en la consulta. El cálculo de los pesos empleados en la media ponderada, se basan en un orden de prioridad para las condiciones establecido por el usuario.

Los pesos empleados en la media ponderada serán determinados utilizando las fórmulas (2) y (3), donde t es el número de condiciones definidas en la consulta, n el número de condiciones incluidas en el orden de prioridad, m el número de condiciones sin un orden de prioridad establecido, $w_p(i)$ el peso asociado a la condición con un nivel de prioridad i (siendo $i = 1$ el mayor nivel de prioridad, y $i = n$ el menor nivel) y w_u el peso asociado a las condiciones sin una especificación de prioridad establecida por el usuario.

$$w_p(i) = \begin{cases} \frac{1}{2^i} & i \neq t \\ \frac{1}{2^{i-1}} & i = t \end{cases} \quad i = 1...n \quad (2)$$

$$w_u = \frac{1}{m2^n} \quad (3)$$

3. Arquitectura de la Aplicación

Como modelo de diseño de la aplicación hemos elegido una arquitectura basada en capas con el propósito de facilitar el mantenimiento y la actualización de los componentes que la conforman.

3.1. Componentes

La aplicación se divide en tres capas que interaccionan entre sí, como se muestra en la figura 3.

La capa de presentación se encarga de generar la interfaz de usuario, con el objeto de proporcionar una interacción sencilla y cómoda con la aplicación. La capa de lógica implementa la funcionalidad y comportamiento esperado de la aplicación. Finalmente, la capa de acceso a datos permite a la lógica de la aplicación el acceso a los datos almacenados en la FRDB.

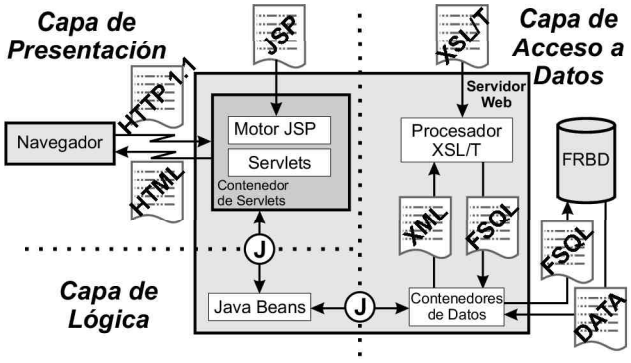


Figura 3: Arquitectura general de la aplicación.

3.2. Independencia de la Base de Datos Difusa

Las consultas de los usuarios son expresadas usando XML. Éstas, mediante el uso de hojas de estilo XSLT, son transformadas en consultas difusas expresadas en el lenguaje de consulta propio de la FRDB empleada. En el caso de que se produzcan cambios en dicho lenguaje, no es necesario modificar completamente la aplicación, solo será necesario realizar cambios en la hoja de estilo XSLT empleada, con el propósito de transformar las consultas expresadas en XML al nuevo lenguaje de consulta de la FRDB utilizada.

El anterior mecanismo proporciona independencia del lenguaje de consulta de la FRDB, lo cual es importante debido a que estamos trabajando sobre prototipos que evolucionan constantemente, lo que implicará, casi con total seguridad, cambios en su lenguaje de consulta.

4. Procesamiento de Consultas Flexibles

El objetivo principal de la aplicación, es comunicar al usuario de ésta con la FRDB empleando consultas difusas que contengan condiciones imprecisas establecidas de una manera cómoda y sencilla por el usuario de la aplicación. Pasaremos a describir el proceso que tiene lugar desde que el usuario define una consulta, empleando la interfaz, hasta que ésta es ejecutada sobre la FRDB y los resultados son devueltos al usuario.

En la figura 4 podemos observar la ruta seguida por los datos y las diferentes transformaciones que ocurren en cada etapa, para convertir una consulta flexible realizada por el usuario en FSQL, el lenguaje de consulta difusa de la FRDB subyacente.

1. La primera etapa abarca el proceso de comunicación entre el navegador web y el servidor. El usuario establece los parámetros de la consulta, utilizando un formulario similar al mostrado en la figura 5, y los envía. En el lado del servidor una página JSP recibe los datos de la consulta y los almacena en un *Java Bean*, preparado para la próxima etapa. El navegador y el servidor se comunican utilizando el protocolo HTTP evitando cualquier incompatibilidad entre plataformas.

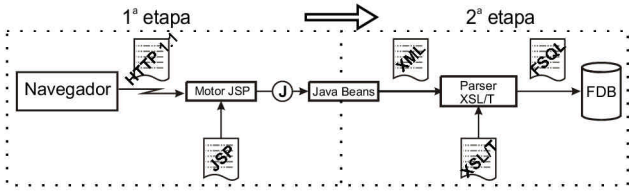


Figura 4: Del usuario a la FRDB.

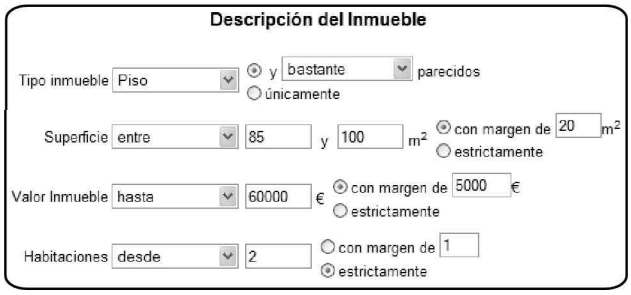


Figura 5: Interfaz web para la definición de consultas.

2. La segunda etapa consiste en la comunicación entre el servidor y la FRDB. En esta etapa entran en juego los mecanismos que nos permitirán independizarnos del lenguaje de consulta, y para ello utilizaremos, como hemos citado anteriormente, XML y XSLT. El *Java Bean*, que representa la definición de la consulta flexible realizada por el usuario, generará un documento XML describiendo dicha consulta. Este documento XML deberá ser transformado con el propósito de obtener sentencias FSQL, que puedan ser ejecutadas en la FRDB, para obtener los datos solicitados por el usuario. La transformación será realizada por medio de una hoja de estilo XSLT. Finalmente, ya obtenidas las sentencias FSQL la FRDB las ejecutará obteniendo los resultados a la consulta que ha planteado el usuario.

   Una vez obtenidos los resultados de la consulta planteada, seguimos la ruta de vuelta para llevar estos al usuario, como se muestra en la figura 6.

3. En la tercera etapa obtendremos los resultados de la FRDB via JDBC. Utilizando esos resultados el servidor de datos genera un documento XML, conteniendo los resultados de la consulta que seran mostrados al usuario.

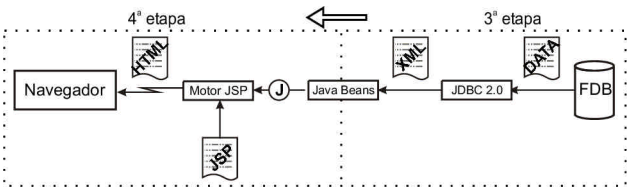


Figura 6: Desde la FRDB al usuario.

Resultados :: Comprar

ID	Ajuste ▼▲	Tipo ▼▲	Precio € ▼▲	Superficie m^2 ▼▲
51	96%	Atico	109.000 €	109.0 ±9% m^2
24	85%	Casa	97.000 €	120.0 - 150.0 ±9% m^2
4	80%	Bajo	90.000 €	120.0 - 125.0 ±9% m^2
9	78%	Apartamento	125.000 €	70.0 - 80.0 ±9% m^2
67	65%	Piso	80.000 €	140.0 ±10% m^2
60	64%	Piso	80.000 €	80.0 ±10% m^2
41	58%	Atico	70.000 €	80.0 ±10% m^2
48	39%	Apartamento	59.000 €	58.0 ±8% m^2

Figura 7: Interfaz mostrando los resultados.

4. En la cuarta etapa, el servidor de lógica carga los resultados desde el documento XML en un *Java Bean*. Este objeto contiene los datos necesarios para generar una respuesta al usuario, y los métodos para acceder a ellos. Por medio de una página JSP formatearemos los resultados, almacenados en el *Java Bean*, en formato HTML. Una vez generado el documento HTML, éste es enviado al navegador del usuario, el cual mostrará los resultados como se muestra en la figura 7.

5. Conclusiones Finales y Trabajos Futuros

En el presente artículo hemos mostrado los fundamentos básicos de la aplicación ISW, cuyo desarrollo ha permitido la consecución de los siguientes objetivos:

1. Mostrar que empleando una aproximación difusa se pueden resolver problemas reales, como hemos hecho con el problema de gestión inmobiliaria, utilizando la capacidad de manejo de datos difusos proporcionada por el servidor FSQL, y evidenciar que un enfoque difuso proporciona soluciones a problemas que las aproximaciones clásicas no permiten resolver.
2. Obtener una arquitectura de aplicación que proporcione una total independencia del lenguaje de consulta de la FRDB subyacente.

Los trabajos futuros estarán orientados a hacer un uso práctico de nuevas características, basadas en Soft Computing, proporcionadas por servidores FRDB venideros, como son Data Mining y Knowledge Discovery, aplicados al área de gestión inmobiliaria.

Agradecimientos

Este trabajo ha sido parcialmente soportado por los proyectos de investigación del Ministerio de Ciencia y Tecnología español, con referencias TIC2002-00480 y TIC2003-08687-C02-02.

Referencias

[1] H. Prade, C. Testemale, "Generalizing Database Relational Algebra for the Treatment of Incomplete or Uncertain Information and Vague Queries". Information Sciences 34(115–143) 1984.

[2] M. Zemankova-Leech, A. Kandel, "Fuzzy Relational Databases – A Key to Expert Systems". Cologne, Germany, TUV Rheinland 1984.

[3] M. Umano, "Freedom-O: A Fuzzy Database System". Fuzzy Information and Decision Processes. Gupta-Sanchez edit. North-Holand 1982.

[4] J.M. Medina, O. Pons, M.A. Vila, "GEFRED. A Generalized Model of Fuzzy Relational DataBases". Information Sciences, 76(1–2), 87–109 (1994).

[5] J.M. Medina, M.A. Vila, J.C. Cubero, O. Pons "Towards the Implementation of a Generalized Fuzzy Relational Database Model". Fuzzy Sets and Systems, 75(273–289)1995.

[6] S. Fukami, M. Umano, M. Muzimoto, H. Tanaka, "Fuzzy Database Retrieval and Manipulation Language", IEICE Technical Reports, Vol. 78, N. 233, pp. 65–72, AL-78-85 (Automata and Language) 1979.

[7] http://frontdb.ugr.es/ImmoSoftWeb

[8] J. Galindo, J.M. Medina, O. Pons, J.C. Cubero, "A Server for Fuzzy SQL Queries", in "Flexible Query Answering Systems", eds. T. Andreasen, H. Christiansen and H.L. Larsen, Lecture Notes in Artificial Intelligence (LNAI) 1495, pp. 164–174. Ed. Springer, 1998.

[9] J. Galindo, "Tratamiento de la Imprecisión en Bases de Datos Relacionales: Extensión del Modelo y Adaptación de los SGBD Actuales". Tesis Doctoral, Univ. de Granada (España), Marzo 1999.

[10] http://www.lcc.uma.es/~ppgg/FSQL.html

[11] I. Blanco, J.M. Martin-Bautista, O. Pons, M.A. Vila, "A Tuple-Oriented Algorithm for Deduction in a Fuzzy Relational Database". Int. Journal of Uncertainty, Fuzzyness and Knowledge-Based Systems vol. 11, pp. 47-66. 2003.

Agente vendedor difuso para entornos de negociacion automatica en comercio electronico

Ramón Manjavacas
Escuela Superior de Informática
Departamento de Informática
Universidad de Castilla-La Mancha
13071 Ciudad Real, España
Ramon.Manjavacas@uclm.es

José Jesús Castro-Schez
Escuela Superior de Informática
Departamento de Informática
Universidad de Castilla-La Mancha
13071 Ciudad Real, España
JoseJesus.Castro@uclm.es

Juan Moreno-Garcia
E.U.I.T. Industrial
Departamento de Informática
Universidad de Castilla-La Mancha
45071 Toledo, España
Juan.Moreno@uclm.es

Resumen

La realización de negocios dentro del campo del comercio electrónico involucra la formalización de modelos simples de interacción. El comercio electrónico implica la consumación de negociaciones entre vendedores y clientes, y de acuerdos fruto de estas últimas. En este artículo, se propone el diseño de un agente vendedor para entornos competitivos de negociación automática, en los cuales, los agentes vendedores y compradores negocian sobre distintos aspectos de productos o servicios. Dentro de un proceso de negociación el agente vendedor tendrá como objetivo la propuesta de ofertas que mejoren las ofrecidas por sus contrincantes persiguiendo maximizar sus beneficios. El diseño está fundamentado en la aplicación de técnicas basadas en lógica difusa y algoritmos genéticos.

Palabras Clave: comercio electrónico, negociación automática, agentes software, lógica difusa, algoritmos genéticos.

1 INTRODUCCION

La aparición de tecnologías, como Internet, ha flexibilizado la realización de negocios a lo ancho de todo el mundo. Este hecho ha provocado la aparición de una infraestructura para este nuevo paradigma de negocio, conocido como *comercio electronico* (e-commerce). Los procesos y tecnologías del comercio electrónico han introducido nuevas formas de llevar a cabo los negocios. Según [11] "El comercio electrónico está basado en compartir información entre organizaciones utilizando una gran variedad de diferentes tecnologías electrónicas, para llevar a cabo negocios... Esto también incluye procedimientos, políticas y estrategias para soportar la incorporación de toda esta información dentro del entorno de negocio".

Dependiendo de las partes implicadas en la negociación se distinguen los siguientes tipos de comercio electrónico [7] B2B, B2C, C2C y C2B, de los cuales, los dos primeros son los más importantes.

En relación a los tipos de comercio electrónico B2B y B2C se han propuesto algunos modelos (CBB - Consumer Buying Behavior, BBT - Business-to-Business Transaction,...) que especifican las distintas tareas y fases que llevan a cabo compradores y vendedores en sus procesos de negocio. En [8], se exponen las diferentes fases de un proceso de negocio tanto para compradores como para vendedores. En el lado del comprador las fases más comunes son: identificación de necesidades, corredor de producto, corredor de mercado, negociación, compra-entrega y servicio-evaluación. Del mismo modo las fases para un vendedor son: planificación de envío de ofertas, identificación de posibles clientes, negociación, cumplimentación de pedidos y soporte a clientes.

Esta visión ha abierto nuevas líneas de investigación en las cuales, se trabajan en sistemas autónomos (en mayor o menor medida) basados en *agentes software inteligentes* que intentan automatizar la mayor parte de estas tareas y fases. De ello surge la concepción del comercio electrónico llevado a cabo por agentes [1] [7] [10]. Un agente inteligente es un programa software que puede desarrollar una serie de tareas por él mismo con el fin de conseguir unos objetivos particulares [9]. Por ello, un agente inteligente debe reunir las características de: autonomía, reactividad y proactividad. Dentro del marco del comercio electrónico, los agentes se configuran como el mecanismo empleado para modelizar y simular el comportamiento de compradores y vendedores que intervienen en los procesos de negocio. Uno de los aspectos más relevantes del comercio electrónico es que la fase de negociación se convierta

en un proceso automático con el objetivo de liberar al usuario.

La negociación es el proceso por el cual dos o más partes intentan alcanzar un acuerdo o compromiso mediante argumentación [12]; no es un fin en sí mismo, sino un procedimiento para conseguir unos objetivos y satisfacer unos intereses [3]. En general, una de las propiedades más resaltable de la negociación en el comercio electrónico es el número de participantes. Se pueden diferenciar tres tipos de escenarios atendiendo al número de participantes involucrados: *uno a uno* (la negociación se compone de un comprador y un vendedor que intercambian ofertas); *uno a muchos* (un único comprador negocia con varios vendedores con el objetivo de obtener el máximo beneficio del contexto competitivo existente o al revés); y *muchos a muchos* (varios compradores negocian con varios vendedores, donde los vendedores ofrecen sus servicios o productos junto con sus condiciones y los compradores exponen sus necesidades).

En este artículo se realiza una investigación encaminada a automatizar el trabajo del vendedor, centrándose la propuesta en el diseño de un agente que aborda la automatización de la fase de negociación de un vendedor dentro de un marco de negociación competitiva en contextos B2C o B2B. En ellos, se llevarán a cabo negociaciones sobre distintos aspectos de productos o servicios entre muchos vendedores y un comprador.

El artículo está organizado como sigue. En la siguiente sección se presentará el marco donde interactuará el agente propuesto. Seguidamente se presentarán los fundamentos teóricos necesarios para la comprensión del agente propuesto. En la sección 4 se describe la estructura de dicho agente. Para finalizar, se exponen algunas conclusiones y trabajos futuros.

2 ENTORNO DE APLICACION

En este trabajo nos centraremos en contextos competitivos B2C o B2B de comercio electrónico, donde las negociaciones se ajustan a escenarios del tipo uno a muchos y los agentes vendedores y compradores manejarán ofertas definidas sobre aspectos negociables de un producto o servicio. Dichos aspectos podrán ser de muy diversa naturaleza y por ello, los tipos de datos manejados serán muy diversos. Los más comunes bajo nuestro criterio, son los siguientes: ordinales, nominales, boleanos, numéricos y graduados. Este hecho se tendrá presente en este trabajo.

Para que sea posible la negociación debe existir una *ZOna de Posible Acuerdo* (ZOPA). Esta zona está acotada inferiormente por el valor de reserva del vendedor y superiormente por el del comprador. El valor de reserva es el límite que no se puede rebasar si se pretende alcanzar un acuerdo. Desde el punto de vista del vendedor es esencial el conocimiento de los siguientes valores previos a un proceso de negociación: valores de reserva que identifican los *puntos de abandono* (PA); valores iniciales que indican posibles *puntos de entrada* (PE); y valores óptimos que denotan los *puntos optimos de venta* (PO) [3].

En este contexto, un agente comprador intentará obtener el máximo beneficio en la adquisición de un producto o servicio negociando con varios agentes vendedores que lo proporcionen. Para ello, dentro del proceso de negociación el comprador irá recibiendo ofertas de los distintos vendedores, las analizará y volverá a instar a los vendedores a que mejoren sus ofertas, teniendo en cuenta las ofertas propuestas por sus contrincantes y la ZOPA, que estará definida por sus valores óptimos y la mejor oferta recibida por parte de los vendedores.

El agente vendedor propuesto en este trabajo estará orientado a su aplicación a contextos de negociación de las características descritas anteriormente. Básicamente, este agente una vez iniciado el proceso de negociación con algún agente comprador, intentará mejorar su oferta empleando para ello la información proporcionada por el comprador sobre ofertas igualmente beneficiosas propuestas por otros vendedores. Este agente vendedor debe generar una oferta que se destaque de algún modo de dichas ofertas sin alejarse de su valor óptimo ni excediendo su punto de abandono.

En el diseño de este agente se asume por una parte que la terminología empleada por los agentes es común y por otra parte, que todo ese conocimiento previo requerido por el agente vendedor sobre los aspectos de los productos o servicios negociables, de las posibles dependencias entre ellos, etc., se conoce. La adquisición de toda esa información no es objetivo de este trabajo.

3 FUNDAMENTOS TEORICOS

Esta sección se centra en la exposición de los distintos fundamentos teóricos en los que se basa la estrategia de negociación definida para el agente vendedor, lo cual, es objetivo de nuestra investigación.

Inicialmente se parte del hecho de que el agente vendedor cuenta con una descripción detallada del producto o servicio que va a negociar con el agente comprador. Dicha descripción consistirá básicamente en un conjunto de características que el producto o servicio presenta ($\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$) junto con los

valores que cada una de ellas pueda tomar ($\mathcal{C}_v = \{Cv_1, Cv_2, \ldots, Cv_n\}$). Además también se cuenta con información referente a la prioridad asignada a cada una de la variables ($\mathcal{P} = \{p_{v_1}, p_{v_2}, \ldots, p_{v_n}\}$), la cual, será empleada en la fase de generación de ofertas dentro del propio proceso de negociación. Toda esta información puede adquirirse de formas muy diversas, ejemplos de ello se exponen en [5] [13].

El conocimiento de las variables que pueden intervenir en el proceso de negociación nos permite establecer las dimensiones de nuestro espacio de negociación. Por tanto, la negociación es en sí misma un proceso de búsqueda de un punto en dicho espacio, que satisfaga las necesidades de compradores y vendedores (Fig.1).

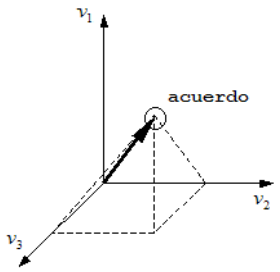


Figura 1: Espacio de negociación para 3 variables.

La definición del espacio de negociación nos permitirá ubicar de inicio ofertas que indiquen situaciones de abandono, de entrada y óptimas para el agente vendedor ante un proceso de negociación. De forma similar, ocurrirá con las distintas ofertas que el agente vaya generando durante dicho proceso. Mediante esta configuración, se podrán aplicar técnicas que nos permitan comparar entre las ofertas generadas y las de inicio.

3.1 EVALUACIÓN DE LA SIMILITUD ENTRE OFERTAS

El hecho de que los valores de las variables de cada una de las ofertas pueda ser de distinta naturaleza, exige el empleo de un formalismo que permita representar tanto la información precisa como imprecisa. Para tal fin, se propone utilizar la función trapezoidal de la lógica difusa, de manera similar, a la aplicación que se describe en [5]. De esta forma, se puede estudiar la similitud entre dos ofertas $\mathcal{O}_x$ y $\mathcal{O}_y$ según una variable v_i, como la medida de distancia, fruto del cálculo del área comprendida entre los dos valores difusos que toman dichas ofertas para la variable seleccionada. Esta medida se define formalmente en [4].

La medida anteriormente comentada presenta un problema. Esta considera que dos ofertas $\mathcal{O}_x$ y $\mathcal{O}_y$ son iguales aún cuando la intersección entre sus áreas no sea nula. Por ello, se empleará la distancia propuesta en [6] que resuelve este problema (1).

$$d(\mathcal{O}_x, \mathcal{O}_y, \vartheta) = \begin{cases} \frac{(a'-b)+(b'-a)}{2}, & \text{si } b \leq a', \\ \frac{b'-a}{2}, & \text{si } a' < b \\ & \text{y } b < b' \\ 0, & \text{si } b = b' \end{cases} \quad (1)$$

Esta medida de distancia se corresponde con el área definida por aquel conjunto de valores A cubiertos por la oferta más cercana al valor ideal $\mathcal{O}_x$, que no son cubiertos por los otros, B de la oferta más lejana $\mathcal{O}_y$ (Fig.2).

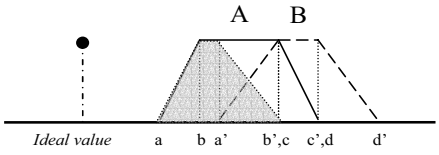


Figura 2: Distancia empleada

Esta medida puede ser normalizada como se define en (2), donde $d(\mathcal{O}_x, \mathcal{O}_y, \vartheta)$ es la medida comentada anteriormente y $d(min, max, \vartheta)$ es la separación existente entre el máximo y mínimo valor para la variable ϑ.

$$d_N(\mathcal{O}_x, \mathcal{O}_y, \vartheta) = \frac{d(\mathcal{O}_x, \mathcal{O}_y, \vartheta)}{d(min, max, \vartheta)} \quad (2)$$

Finalmente, la distancia existente entre dos ofertas $\mathcal{O}_x$ y $\mathcal{O}_y$ se define como la suma de las distancias normalizadas existentes para cada una de las variables (v_i) que entran en juego en la negociación (n), calculada según (3). Por tanto, la similitud existente entre dos ofertas será mayor cuanto menor sea la distancia.

$$\mathcal{D}(\mathcal{O}_x, \mathcal{O}_y) = \frac{\sum_{i=1}^{n} d_N(\mathcal{O}_x, \mathcal{O}_y, v_i)}{n} \quad (3)$$

3.2 GENERACIÓN AUTOMÁTICA DE OFERTAS

El proceso de negociación se puede contemplar como un proceso de búsqueda, como se presentó anteriormente. A partir de este punto, la estrategia de negociación mediante la cual el agente vendedor irá generando ofertas, puede fundamentarse en la aplicación de técnicas de exploración y optimización de soluciones sobre espacios de búsqueda. Entre las

distintas posibilidades para tal fin, en este trabajo se ha decidido emplear "algoritmos genéticos" [2].

Los algoritmos genéticos son técnicas basadas en la evolución, y en particular en los conceptos de variación y selección natural. El esquema básico de este tipo de algoritmos es el siguiente:

1. Generar una poblacion inicial aleatoria de n-elementos (cromosomas)
2. Evaluar la aptitud de cada cromosoma de la poblacion
3. Crear la siguiente generacion siguiendo los siguientes pasos:
   (a) Seleccionar dos cromosomas padres atendiendo a un criterio determinado
   (b) Cruzar los cromosomas padres para generar nuevos hijos
   (c) Mutar los cromosomas atendiendo a un criterio determinado
4. Sustituir la generacion actual por la generada y volver al paso 2. Este proceso flnalizara cuando se satisfaga la condicion de terminacion.

El algoritmo propuesto no se ajusta exactamente en el sentido definido en [2], debido a las características particulares del dominio de aplicación. A continuación se describen los aspectos más resaltables del mismo.

El primer punto a destacar es la forma de representación de la información, es decir, la estructura de los cromosomas. Cada cromosoma deberá representar una oferta existente en el proceso de negociación. Una oferta estará definida por un número determinado de variables ($\mathcal{V}$) y cada variable presentará un número determinado de conjuntos difusos en los que se ha dividido el dominio de la variable ($\mathcal{C}_v$). Atendiendo a esta definición, la estructura de cada cromosoma consistirá en una cadena de elementos, donde éstos representarán a cada uno de los conjuntos difusos existentes (Fig.3). La longitud de esta cadena $\mathcal{L}_{cro\ \ oso\ \ a}$ dependerá del número de conjuntos difusos ($|\mathcal{C}_v|$) en los que se haya dividido el dominio para cada una de las variables existentes en las ofertas (4).

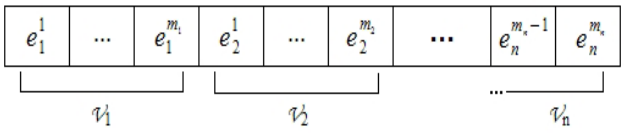


Figura 3: Cromosoma empleado

$$\mathcal{L}_{cro\ \ oso\ \ a} = \sum_{i=1}^{n} |\mathcal{C}_{vi}| \tag{4}$$

El alfabeto para cada uno de los elementos del cromosoma es $\mathcal{A} = \{0, 1, \sharp\}$. El significado de cada elemento del alfabeto es: el símbolo 1 es utilizado para indicar la presencia de una etiqueta de una variable, el símbolo 0 la ausencia de la misma y el símbolo $\sharp$ para reflejar el caso en el que una variable no entra en juego en un proceso de negociación. En este caso, todas las etiquetas de una variable presentarán este símbolo.

El agente contará inicialmente con una población de $\mathcal{N}$ cromosomas, consistente en ofertas que representen distintos puntos de entrada desde los cuales podría iniciarse la negociación. Esta población se tomará como base al inicio de una negociación.

La operación de cruzamiento consistirá en la selección aleatoria tanto de los cromosomas padre como de la variable a partir de la cual se aplicará tal acción. Esta operación se llevará a cabo a nivel de variable para evitar que surgan cromosomas hijos, que presenten dos etiquetas seleccionadas para una misma variable.

La operación de mutación se aplicará de forma aleatoria sobre un cromosoma hijo, cuando se supere un cierto valor umbral ρ previamente fijado. La operación consistirá en la selección aletoria de una de las variables que entre en juego en la negociación y seleccionar una etiqueta de dicha variable según la táctica empleada por el agente. Ejemplos de estas tácticas podrían ser: conservadora (se marcaría una etiqueta contigua a la actual), agresiva (seleccionar la opuesta a la actual) y de sondeo (cualquier etiqueta podría ser seleccionada).

Al aplicar estas operaciones pueden aparecer efectos secundarios que vendrán determinados por las dependencias existentes entre variables. El agente vendedor debe conocer estas dependencias para generar ofertas.

La evaluación de una oferta se realizará por una parte, en función a la proximidad que presente ésta con la oferta óptima del agente vendedor ($\mathcal{F}_{gen}(\mathcal{O}_x)$) y por otra parte, a su proximidad con las ofertas recibidas (H_o) del comprador de los otros posibles vendedores que negocien sobre el mismo producto o servicio ($\mathcal{F}_{ho}(\mathcal{O}_x)$). La función empleada para evaluar es:

$$\mathcal{F}(\mathcal{O}_x) = \mathcal{F}_{gen}(\mathcal{O}_x) + \mathcal{F}_{ho}(\mathcal{O}_x) \tag{5}$$

Tanto $\mathcal{F}_{gen}(\mathcal{O}_x)$ como $\mathcal{F}_{ho}(\mathcal{O}_x)$ se fundamentan en la aplicación de (3), adaptada para contemplar la prioridad (ω_i) asignada a cada una de las variables que entren en juego en las ofertas. El resultado se muestra en (6).

$$\mathcal{D}(\mathcal{O}_x, \mathcal{O}_y) = \frac{\sum_{i=1}^{n} d_N(\mathcal{O}_x, \mathcal{O}_y, v_i) \cdot \omega_i}{n} \tag{6}$$

Según lo anterior, $\mathcal{F}_{gen}(\mathcal{O}_x)$ y $\mathcal{F}_{ho}(\mathcal{O}_x)$ se calcularían mediante (7) y (8) respectivamente:

$$\mathcal{F}_{gen}(\mathcal{O}_x) = \mathcal{D}(\mathcal{O}_x, \mathcal{O}_{o\ ti\ \ a}) \tag{7}$$

$$\mathcal{F}_{ho}(\mathcal{O}_x) = \frac{\sum_{i=1}^{|H_o|} \mathcal{D}(\mathcal{O}_x, \mathcal{O}_i) \cdot \ _i}{|H_o|} \tag{8}$$

donde $_i$ prioriza la proximidad con las ofertas recibidas más recientemente. Este parámetro se calcula en función al instante de recepción asociado a cada oferta $(\mathcal{O}_i)$ perteneciente al conjunto de ofertas recibidas y al instante actual de negociación ($_{actual}$):

$$_i = \frac{(\mathcal{O}_i)}{_{actual}} \tag{9}$$

El algoritmo genético modifica la población de individuos en cada generación en base al valor asociado a cada individuo calculado mediante (5). El tamaño de la población ($\mathcal{N}$) se mantiene constante, eliminando aquellos individuos que presenten peor valoración.

En cada paso de la negociación, el agente vendedor aplicará el algoritmo esbozado anteriormente para la generación de una nueva oferta. Inicialmente la población consistirá en el conjunto de ofertas de entrada. Dicha población se irá evolucionando en m generaciones. La oferta seleccionada en cada paso será aquella que presente la mejor valoración según (5).

4 ESTRUCTURA DEL AGENTE VENDEDOR

El agente vendedor propuesto está orientado a la automatización de la fase de negociación en contextos competitivos B2C o B2B. Inicialmente el agente vendedor contará con información del producto o servicio sobre el que pretende negociar: los aspectos negociables, las dependencias entre ellos, la importancia de los mismos y las posibles ofertas de entrada, de abandono y óptimas de acuerdo para la negociación. Con toda esta información, el agente irá generando ofertas automáticamente, según lo descrito en secciones anteriores, hasta que finalice la negociación.

El hecho de que el agente vendedor tenga información de ofertas de otros vendedores en cada paso del proceso de negociación, se ha empleado para optimizar las ofertas generadas automáticamente. Todas estas ofertas conforme se van recibiendo van entrando a formar parte del *historico de ofertas* (H_o) del agente. Este histórico será inicializado al comienzo de un proceso de negociación. Un aspecto a tener en cuenta sobre el empleo de estas ofertas para generar una nueva oferta es por una parte, que las últimas ofertas recibidas deberán ser tenidas más en cuenta puesto que representarán los valores más actuales de la negociación y por otra parte, la propia evolución de las ofertas en el proceso de negociación (si las últimas distan de las primeras recibidas, si giran entorno a ciertos valores,...). Para recoger la actualidad de cada oferta del histórico se ha asociado a cada una de ellas un valor correspondiente al instante de la negociación en el que se ha recibido y para capturar el modo de evolución de la negociación se ha empleado un mecanismo de actualización de dicho valor asociado. Éste se fundamenta en el cálculo de la distancia media $(\overline{d}\ \ )$ entre todas las ofertas del histórico según (10) y actualizar al instante actual de la negociación el valor asociado a cada oferta $\mathcal{O}_x$, si el valor de la distancia entre esa oferta $\mathcal{O}_x$ y la última recibida $\mathcal{O}_n$ es inferior a la distancia media calculada. Esta actualización se efectuará cada vez que el agente reciba nuevas ofertas.

$$\overline{d}\ \ = \frac{\sum_{i=1}^{|H_o|} \sum_{j=i+1}^{|H_o|} \mathcal{D}(\mathcal{O}_i, \mathcal{O}_j)}{\sum_{k=1}^{|H_o|-1}} \tag{10}$$

El comportamiento que el agente vendedor presenta se describe a continuación:

1. Inicio de negociacion de un producto o servicio
   (a) Inicializar la poblacion de ofertas de punto de entrada (P_{gen}) para dicho producto o servicio a partir del conjunto base de ofertas de punto de entrada (C_{ob})
   (b) Inicializar a ´ el historico de ofertas recibidas (H_o)
2. Enviar una oferta de entrada
3. Mientras exista negociacion
   (a) Recibir ofertas e incrementar el tiempo (t_{actual})
   (b) Actualizar historico de ofertas (H_o)
      i. Analizar si la distancia de cada oferta es nula con cada una de las existentes. En caso aflrmativo no se incorporara dicha oferta.
      ii. Refrescar los $t(\mathcal{O}_i)$ de las ofertas segun (10)
   (c) Comprobar si una nueva variable ha entrado en juego en la negociacion. En caso aflrmativo, a cada una de las ofertas de la poblacion (P_{gen}) asignarle en dicha variable el valor que presente una oferta de (C_{ob}) seleccionada aleatoriamente
   (d) Generar nueva oferta
   (e) Si la nueva oferta generada presenta una distancia inferior a un cierto umbral (σ) del punto de abandono o no esta comprendida dentro del intervalo ZOPA requerir intervencion del vendedor

Un aspecto no tenido en cuenta en este trabajo es la necesidad de aplicar un protocolo de negociación que gobierne la interacción entre agentes. Es evidente la necesidad del establecimiento de un conjunto de reglas que definan los distintos estados de la negociación, los eventos que provocan las transiciones entre estados, las acciones permitidas en cada estado, etc. Este protocolo de negociación se investigará en futuros trabajos.

5 CONCLUSIONES

En el presente artículo, se ha propuesto una posible solución al problema de automatizar las intervenciones de un vendedor dentro un proceso de negociación automática, definiendo las características del contexto idóneo para su aplicación. Por una parte, en el diseño presentado, la capacidad de autonomía e independencia de la figura del vendedor ha sido representada mediante el concepto de agente inteligente [7]. Por otra parte, la estrategia de negociación se fundamenta principalmente en la aplicación de técnicas de exploración y optimización, como son los algoritmos genéticos [2] junto con técnicas difusas [6], con el objetivo de generar ofertas que mejoren o se acerquen a las propuestas por sus contricantes y a su vez le aporten el mayor beneficio.

En líneas futuras de trabajo se pretende profundizar sobre los siguientes aspectos: aplicación de ontologías dentro del marco de la negociación automática para la unificación de terminologías, modelización de estrategias de negociación y de protocolos de negociación que gobiernen la interacción entre vendedores y compradores en este contexto competitivo.

Agradecimientos

Este trabajo ha sido subvencionado por el Ministerio de Ciencia y Tecnología y la Junta de Comunidades de Castilla-La Mancha bajo los proyectos de investigación "DIMOCLUST" TIC2003-08807-C02-02 y "PREDACOM" PBC-03-004 respectivamente.

Referencias

[1] A.R. Lomuscio, M. Wooldrige and N.R. Jennings. A classification scheme for negotiation in electronic commerce. *Agent-Mediated Electronic Commerce: A European AgentLink Perspective*, F. Dignum and C. Sierra, Eds., Springer Verlag, pp.19-33, 2001.

[2] D. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Enero, 1989.

[3] J. Gómez-Pomar Rodríguez. Teoría y técnicas de Negociación. *Ed. Ariel S.A.*, 1991.

[4] J.J. Castro-Schez, J.L. Castro, and J.M. Zurita. Fuzzy repertory table, a method for acquiring knowledge about input variables to machine learning algorithms. *IEEE Transactions on Fuzzy Systems* 12(1):123-139, 2004

[5] J.J. Castro-Schez, N.R. Jennings, X. Luo and N.R. Shadbolt. Acquiring Domain Knowledge for Negotiating Agents:A case of Study. *International Journal of Human-Computer Studies*, 61(1):3-31, 2004

[6] J.J. Castro-Schez, R.Manjavacas, J.L. Castro, y J. Moreno. Design of a merchant broker agent based on fuzzy logic as mediator in e-commerce. *IPMU'04*, Perugia, Italia, Julio 2004.

[7] M. He, N.R. Jennings y H. Leung. On Agent-Mediated Electronic Commerce. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):1345-1363, 2003.

[8] M. Nissen. Knowledge-Based Reengineering: From Mysterious Art to Learnable Craft. *CITM*, Working Paper 96-WP-1012, 1996.

[9] M. Wooldridge y N.R. Jennings. Intelligent agents: theory and practice. *Knowledge Engineering Review*, 2(10):115-152, 1995.

[10] R. Guttman, A. Mouka and P. Maes. Agent-mediated electronic commerce: A survey. *Knowledge Engineering Review.* vol. 13, no. 2, pp.147-159, 1998.

[11] Sockol, Phyllips K. EDI. The Competitive Edge. *McGraw-Hill*, New York, 1989.

[12] S. Sierra, P. Faratin, and N.R. Jennings. A Service-Oriented Negotiation Model between Autonomous Agents. *Proceedings of the 8th European Workshop on Modeling Autonomous Agent in a Multi-Agent World.* Ronneby, Sweden, pp.17-35, 1997.

[13] X. Luo, N. R. Jennings, and N. Shadbolt. "Knowledge-based acquisition of tradeoff preferences for negotiating agents" *Proc. 5th Int. Conf. on Electronic Commerce*, Pittsburgh, USA, pp.138-144, 2003

HACIA UNA LÓGICA FUZZY EXPERIMENTAL

A. Sobrino
Dpto. de Lógica y Filosofía Moral. USC
lflglalex@usc.es

S. Fernández Lanza
Dpto. de Lógica y Filosofía Moral. USC
sflanza@usc.es

Resumen
Es sabido que la lógica fuzzy no es singular, sino plural. Existen diversas lógicas difusas y cada una de ellas se reclama como la más pertinente para modelar diferentes aspectos de la incertidumbre o vaguedad del lenguaje natural. En este trabajo se proponen distintos ámbitos en los que es posible contrastar diferentes propuestas de lógicas fuzzy, y se muestra un experimento que, con carácter simplemente ejemplificador, mostraría algunos aspectos de esta tarea.

Palabras clave: Experimentos, lógica fuzzy, contrastación de lógicas fuzzy

1 INTRODUCCIÓN

La lógica fuzzy es un instrumento de representación de proposiciones imprecisas y de gestión de las consecuencias aproximadas, frecuentes en el razonamiento de sentido común.

Dado que la vaguedad del lenguaje es poliédrica y plural, esta lógica ha tenido que desdoblarse en diferentes versiones y se ha visto obligada a desarrollar heterogéneas propuestas; cada una de ellas dice recoger tal o cual aspecto semántico o adaptarse mejor a tal o cual aplicación pretendida. Así, casi todos los manuales hablan de una lógica fuzzy en sentido restringido (*narrow*) o amplio (*extense*); de lógicas fuzzy basadas en distintas lógicas multivaloradas: de Kleene, de Bochvar, de Lukasiewicz,…; de familias de operadores fuzzy, como las t-normas o t-conormas, de lógicas fuzzy adecuadas a tareas como el control de procesos, etc. [4]

Si bien el número de soluciones fuzzy ha ido creciendo, no siempre se han encontrado buenas interpretaciones de lo que pretenden modelar. A veces parecen justificarse, fundamentalmente, en la solidez interna de sus pruebas, olvidándose de los ejemplos y de las aplicaciones pretendidas. En cualquier caso, se echa en falta un ámbito de contrastación compartido que permita evaluar entre posibles alternativas.

En este trabajo haremos referencia a algunos tipos de experimentos que permitan evaluar distintas lógicas fuzzy. Para ello, el artículo se estructura de la siguiente manera: en el apartado que sigue se reseñan los trazos de una metodología que invitaría a una concepción empírica de la matemática y, por ende, de la lógica. En el siguiente punto se analizan algunos rasgos de la tipología de los experimentos a los que se pueden someter diferentes propuestas de lógica difusa. A continuación se describe un posible experimento y algunos problemas derivados de su diseño sin más pretensión que la de ser una iniciación al tema. Finalmente, un apartado de conclusiones y de posibles vías de desarrollo servirá para concluir este artículo.

2. LA LÓGICA COMO CIENCIA CUASI-EMPÍRICA.

Hasta la década de los 80 se hacía una distinción clara entre ciencias formales y ciencias empíricas. La diferencia residía fundamentalmente en el método. De las ciencias formales se decía que eran deductivas; esto es, garantizaban la transmisión de la verdad de los axiomas al resto de proposiciones que se seguían lógicamente de ellos. En cambio, las ciencias de la naturaleza o ciencias físicas necesitaban de la observación y de la experimentación para verificar las verdades predichas por la teoría. La popularidad de las ciencias naturales y el carácter estable y atemporal del método matemático, hizo que las ciencias formales quedasen excluidas del debate epistemológico contemporáneo. Lakatos cambió esta tendencia, proponiendo una metodología empirista para la matemática [3].

Una aproximación *empirista* se caracteriza por distinguir como enunciados de la matemática a los axiomas y a otros cuya verdad no está siempre garantizada, sino supuesta. Si de esa suposición se siguiese alguna inconsistencia, habría que revisar la verdad en principio atribuida, teniendo entonces naturaleza conjetural. La matemática afrontaría problemas y ofrecería soluciones que alcanzarían más o menos corroboración según su grado de adecuación a los hechos; según su resistencia no sólo a falsadores lógicos, sino sobre todo heurísticos, que invitarían a reemplazar unas soluciones por otras. De este modo, la matemática pasaría a ser, al menos en parte, una ciencia experimental como la física.

3. TIPOS DE EXPERIMENTOS EN LÓGICA FUZZY.

El carácter conjetural de la matemática parece particularmente adecuado para el caso de la lógica fuzzy, porque, más que de una lógica, se trata de una familia de lógicas, cada una de ellas con una pretendida adecuación al análisis de la notable variedad de aspectos involucrados en la vaguedad del lenguaje. Se pueden diferenciar dos grandes ámbitos de contrastación para las conjeturas que ofrecen las diferentes lógicas fuzzy: las aplicaciones industriales y las oraciones y argumentos del lenguaje natural. Cada uno de ellos se describirá atendiendo a la siguiente estructura:

• ámbito conceptual al que va dirigido la lógica propuesta
• naturaleza de la contrastación
• tipología social de los agentes involucrados

3.1 CONTRASTACIÓN TECNOLÓGICA

Ámbito de proposición de la lógica. Se proponen lógicas para resolver problemas en ámbitos aplicados o manufacturados.

Contrastación. Las lógicas se prueban atendiendo a la adecuación o bondad de sus soluciones al ámbito pretendido.

Agentes de la contrastación. Especialistas en las aplicaciones y usuarios de los productos manufacturados.

3.2 CONTRASTACIÓN LINGÜÍSTICA

Se distingue una contrastación que tiene que ver con el significado aislado o puro de las oraciones que pretenden ser ejemplos de las diferentes propuestas formales (semántica), de una contrastación que refiere al significado de esas oraciones en su uso (pragmática).

3.2.1 SEMÁNTICA

• *Ámbito de proposición de la lógica.* La lógica fuzzy se propone con el ánimo de capturar el significado vago de las oraciones del lenguaje y de gestionar sus relaciones de consecuencia.
• *Contrastación.* Las lógicas o variedades de lógica se prueban atendiendo al grado en el que representan el significado que, una persona con cierto adiestramiento racional, diría que es el genuino de la oración.
• *Agentes de la contrastación.* Los sujetos que dirimen la adecuación de la lógica al significado de las oraciones suelen pertenecer al ámbito docente o profesional, con formación universitaria, con cierto entrenamiento racional y capaces de hacer una evaluación puramente racional –esto es, de ponerse, más o menos, en el contexto 'ideal' del que propone la lógica-, obviando posibles efectos perturbadores.

3.2.2 PRAGMÁTICA

• *Ámbito de proposición de la lógica.* Se propone, como en el caso anterior, una lógica para recoger el significado impreciso de las oraciones y sus relaciones de implicación.
• *Contrastación.* Las lógicas o variedades de lógica se prueban verificando su adecuación con el uso que una persona normal, sin especial adiestramiento racional, hace de la oración en cuestión.
• *Agentes de la contrastación.* Son sujetos racionales no entrenados en tareas de análisis formal y que en la evaluación no se abstraen del contexto que rodea al juicio lógico; al contrario, usan las intenciones subyacentes a la argumentación, las presuposiciones implícitas en el razonamiento, etc. [1]

Siguen, a continuación, algunos comentarios generales sobre estas modalidades de contrastación:

- Las contrastaciones tecnológicas tienen un gran poder decisor. Si una propuesta se diseña para resolver un problema concreto, la bondad o adecuación de la solución ofrecida será determinante a la hora de comparar alternativas. Dada la vocación aplicada de las lógicas fuzzy, 'los resultados mandan' y tienen gran autoridad. Pero poseen algunas limitaciones. Por el carácter casi siempre concreto de las aplicaciones, las propuestas formales suelen tener un fuerte sabor *ad hoc*, de propósito dedicado, y esto hace que las comparaciones tengan que ser relativas. Sería útil disponer de un catálogo de aplicaciones fuzzy a la tecnología y comprobar la variedad de las soluciones propuestas así como la 'pureza' de las lógicas empleadas, pero eso rebasaría el ámbito de este trabajo. A partir de ahora nos centraremos únicamente en los experimentos lingüísticos:
- Los experimentos semánticos a los que se han sometido las lógicas fuzzy han sido, hasta ahora, rudimentarios. Se basan, fundamentalmente, en la elección de oraciones distinguidas para mostrar las hipótesis que se desean defender. Pero no se ha propuesto una base de datos experimental suficientemente grande y 'neutral' a la que someter las diferentes ideas. Consensuar un lugar de contrastación puede ser beneficioso para evaluar distintas lógicas; para poder decir, con cierto criterio, que una es mejor o más adecuada que otra. Varias posibilidades se muestran a la hora de fijarse en ámbitos de los que pueden extraerse argumentaciones que incluyen lenguaje vago. Para la elaboración de este trabajo se han explorado algunos: a) en el ámbito técnico profesional, hemos observado las FAQ de Microsoft, que ofrecen una información muy estructurada, susceptible de convertir en oraciones condicionales, pero que directamente no lo son; b) en el terreno novelístico se han analizado los capítulos nº 2 de *Noticia de un secuestro*, de Gabriel García Márquez y *¿Quién mató a Palomino Molero*?, de Mario Vargas Llosa. Nos llamó la atención la absoluta escasez de estructuras condicionales 'si,...,entonces', así como la variedad de otros formatos condicionales; c) en el ámbito jurídico se evaluó una sentencia que resolvía una solicitud de nacionalidad y donde, para nuestra sorpresa no había oraciones de significado vago; todas ellas pretendían enunciar en términos categóricos; d) en un ámbito

perteneciente más al habla que a la escritura se analizó un discurso político. En ese texto nos encontramos con muchos pseudocondicionales, oraciones que parecían entrañar conclusiones, pero que no lo hacían; e) en la parcela académico-docente estudiamos en detalle el manual *Introducción a la música*, de Otto Károlyi. Es un texto sobre un conocimiento reglado que transmite sensaciones humanas; por tanto, imprecisas. De ahí que lo eligiéramos definitivamente para progresar en nuestro análisis.
- Los experimentos pragmáticos han sido desarrollados realizando entrevistas sobre la percepción que los sujetos tienen de la lógica implícita en las cuestiones que se le presentan [1]. La red permite hoy entrevistar a un gran número de usuarios y registrar sus datos. En este trabajo haremos uso de esta utilidad.

En el siguiente apartado se describe, a modo de ejemplo, una propuesta de experimento-tipo para comparar lógicas fuzzy, en los ámbitos lingüísticos reseñados:
1. Semántico: confeccionar una base de datos de oraciones y reglas de significado aproximado suficientemente amplia con el objeto de que sirva de base de contrastación para diferentes propuestas fuzzy -en este trabajo, nos limitaremos a las reglas-
2. Pragmático: Usando resultados del apartado anterior, elaborar una serie de formularios (HTML) que permitan comparar la racionalidad de las definiciones lógicas con el juicio de personas competentes en el uso condicional del lenguaje, aunque sin adiestramiento formal alguno.

4. APROXIMACIÓN A UN POSIBLE EXPERIMENTO

Se describe, a continuación, un prototipo de experimento. Se usan tecnologías de implementación de aplicaciones web, como Java y php, así como aquellas propias del procesamiento de expresiones regulares, como Perl. El esquema general de toda la propuesta es el siguiente:

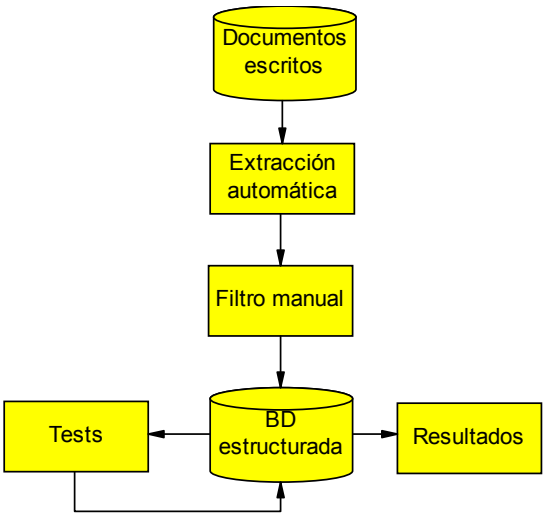


Fig.1: *Esquema general*

4.1 PRIMERA APROXIMACIÓN: HACIA UNA BASE DE ORACIONES (REGLAS) FUZZY

Como ya se indicó, para confeccionar la base de reglas se usó el libro *Introducción a la música*, de Otto Károlyi. El volumen total de texto escogido fue de 194 Kb y sobre él se procedió a la extracción automática de reglas [2], usando expresiones regulares con Perl. Se optó por un diseño poco rígido de estas expresiones regulares, con el fin de recuperar muchas expresiones, aun a riesgo de incurrir en fallos. Inicialmente, se realizó un primer preprocesado, utilizando técnicas automáticas y manuales, con el fin de evitar los casos críticos que surgen al considerar el punto (".") como final de una oración. En segundo lugar, el sistema procedió a dividir el texto en oraciones. En tercer lugar, se separaron aquellas oraciones que contienen expresiones como "si entonces", "por tanto", "dado que", etc., indicativas de contener estructuras condicionales.

Como se puede suponer, la flexibilidad del proceso y la polisemia y homonimia de algunas de las expresiones seleccionadas redundó en un alto número de falsos positivos que fueron eliminados posteriormente mediante procedimientos manuales. En la fase automática, partiendo de un total de 1488 oraciones, se recuperaron 189 susceptibles de tener estructura condicional. Tras la recuperación mecánica se procedió a un filtrado manual para eliminar aquellos casos erróneos. Para ello se emplearon formularios HTML que permitían incorporar en una base de datos (MySQL) los casos correctos. El esquema de este primer prototipo, desde el punto de vista de las tecnologías empleadas, es el siguiente:

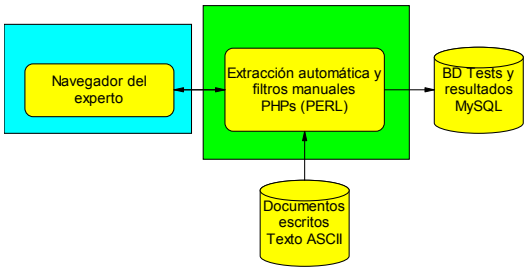


Fig. 2: *Tecnologías del prototipo 1*

El filtrado manual debe ser realizado por un experto que sepa distinguir las expresiones con estructura condicional de las que carecen de ella. La página web que permite al experto la selección es esta:

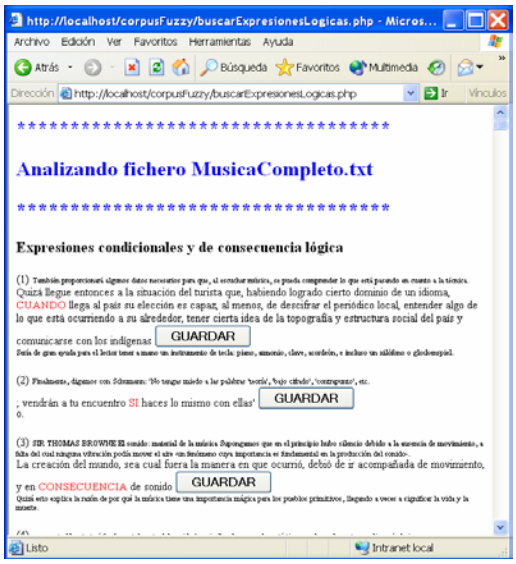


Fig3. : *Página web del Prototipo 1*

Dada la flexibilidad con la que fueron diseñadas las expresiones regulares, esta acción conllevó una reducción drástica: de las 189 recuperadas, sólo 88 contenían expresiones que eran claramente reglas, algunas precisas y otras borrosas. La selección realizada será usada en la siguiente fase, cuya arquitectura se describe a continuación.

4.2 SEGUNDA APROXIMACIÓN: ELEMENTOS PARA DISEÑAR UN TEST WEB

El objetivo concreto aquí es verificar si, dependiendo de la forma y del significado de las oraciones que conforman las reglas, los usuarios utilizan distintas funciones de implicación para justificar y valorar sus conclusiones. El test al que se somete a los sujetos experimentales responde, metodológicamente hablando, al patrón Modelo-Vista-Controlador (MVC), común en el diseño de aplicaciones web en Java. El esquema general es el siguiente:

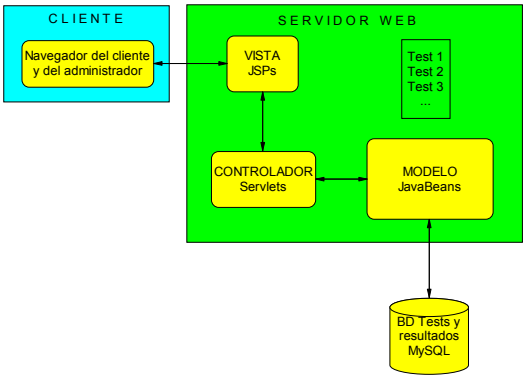


Fig. 4: *Patrón Modelo-Vista-Controlador del prototipo 2*

Para verificar qué tipo de regla borrosa usa un individuo cuando se le proporcionan datos sobre un razonamiento, se pueden realizar tres tipos de preguntas:

- Enunciar la regla. Dar una valoración del antecedente. Ofrecer una cualificación del consecuente. A tenor de la evaluación del antecedente y del consecuente, pedirle al sujeto que diga en qué grado se cumple la regla.
- Pedirle al individuo que cualifique la validez de la regla. Ofrecer una cualificación del antecedente. Pedirle que diga en qué grado se debería cumplir el consecuente.
- Ofrecer una evaluación de la regla y una cualificación del antecedente. Pedirle al sujeto del experimento que enuncie conclusiones compatibles con la evaluación ofrecida de la regla.

El análisis de las contestaciones supondrá tres acciones encadenadas:

• La primera acción consistiría en retomar el experimento anterior y separar, de las 88 reglas obtenidas, aquellas en las que todos los sujetos están de acuerdo en que son reglas borrosas. Esto evitará conflictos en los test posteriores cuando se inste a los mismos sujetos a proporcionar un grado de confianza en la regla o proponer una conclusión. En efecto, dados los resultados de un ensayo previo donde había una gran disparidad de opiniones sobre el carácter borroso o no de las reglas, nos pareció conveniente que sería de utilidad instruir a los entrevistados en las diferencias entre una regla borrosa y una regla precisa. Una vez asegurado que todos los individuos disponían de criterios para realizar tal distinción, se le mostró a cada sujeto el conjunto de oraciones seleccionadas, sobre las que deberían decidir entre una de estas alternativas:

1.- La oración no es una regla.
2.- La oración es una regla precisa.
3.- La oración es una regla borrosa.
4.- No lo sé.

significando esta última opción que el sujeto no distingue si es una regla o no, o que el sujeto sabe que es una regla pero no distingue si es precisa o borrosa. Dado que lo que se persigue es obtener el total de reglas borrosas, ambas opciones se pueden agrupar en un único punto.

• El segundo test tendría como objetivo evaluar cuál es el grado de confianza que los usuarios tienen en la regla. Este sería un dato necesario para una de las modalidades de pregunta que se propuso antes. Para ejecutar este experimento, el administrador del sistema debería revisar el total de argumentos borrosos separados en el proceso anterior y eliminar algunos ejemplos que ofrecen problemas por ser anafóricos. Por ejemplo, en la oración "Tal procedimiento implica generalmente una alteración cromática" la expresión "implica generalmente" pone de manifiesto que la regla es imprecisa; sin embargo, es difícil de evaluar su grado de confianza, puesto que no sabemos a que se refiere la expresión "tal procedimiento", teniendo que ser implementada alguna técnica de 'minería de texto' para hallar la respuesta. Por tanto, parece

conveniente usar para este segundo test sólo aquellas reglas cuya información esté autocontenida y sean comprensibles sin recurrir a otras partes del texto. Hecha esta selección, se le presentan las reglas que quedan a los sujetos del experimento y se le pide que indique el grado de confianza que les merece seleccionando alguna de estas opciones:

1.- La regla no falla nunca.
2.- La regla falla pocas veces.
3.- La regla falla en la misma medida que acierta.
4.- La regla acierta pocas veces.
5.- La regla falla siempre.

• El tercer test consiste en pedir al sujeto experimental que evalúe el grado de corrección de un argumento construido usando alguna de las reglas. A este respecto, caben dos posibilidades:
- usar todas las reglas y extraer conclusiones de los resultados
- seleccionar algunas reglas, sobre las que previamente se han hecho hipótesis sobre su adecuación a diferentes definiciones de implicación, y verificar si los sujetos aprecian lo mismo o no.

La primera posibilidad parecería la deseable, pero es engañosa. La diversidad de la vaguedad es tan esquiva y dependiente de contexto, que puede propiciar un caos de respuestas sin regularidad alguna, aunque quizás cada una de ellas esté avalada por ciertas razones. La selección previa de casos tampoco es fácil, al menos para algunas interpretaciones, pero dirige hacia algún horizonte al experimento [5]. Nosotros optamos por esta segunda vía.

La oportuna selección de oraciones sujeto de experimentación en el test 3 permitirá diagnosticar si, para casos diferentes, los usuarios eligen funciones de implicación distintas. En este trabajo proponemos tres reglas, que parecen ser representativas de otros tantos grupos de funciones de implicación; a saber,
- grupo Mamdani, que se caracteriza porque penaliza la falta de verdad en antecedente o consecuente. La definición de Mamdani, equivale a una conjunción: $v(p\rightarrow q)=\min(v(p), v(q))$ que es conmutativa. A este grupo pertenece también la implicación de Larsen: $v(p\rightarrow q)= v(p) \cdot v(q)$
- grupo Gödel. No se penaliza en ningún caso la falta de verdad de las premisas; en general, si la verdad del antecedente es menor o igual que la del consecuente, la implicación es verdadera en grado máximo. En cambio, si la verdad del consecuente es mayor que la del antecedente, la regla toma el valor del consecuente, sea este valor pequeño o grande: $v(p\rightarrow q)= v(p)\leq v(q))=1$; e.o.c., v(q). En este grupo se puede clasificar también la implicación de Gaines, $v(p\rightarrow q)= v(p)\leq v(q))=1$; e.o.c., v(q)/v(p).

En la definición de Gödel hay ausencia de conmutatividad, manifestada, en un caso extremo, en que si el antecedente fuese verdadero en grado 0,1 y el consecuente en grado 0,2 la regla vale 1, pero si se intercambiaran los valores, la regla valdría 0,1. Se diría que, aún siendo una definición para reglas borrosas, la implicación de Gödel no admite el Sorites: discrimina tan fino que una ligerísima diferencia entre los valores del antecedente y consecuente puede hacer, para valores bajos, que la regla varíe enormemente de valor de verdad.
- grupo Kleene-Dienes. Atiende a la definición del condicional o implicación material $v(p\rightarrow q)= \max(1-v(p), v(q))$. Es la definición que mejor refleja la similaridad que presentan muchas de las reglas seleccionadas para este trabajo que expresan proporciones vagas. A este grupo pertenece también la implicación de Kleene-Dienes-Lukasiewicz: $v(p\rightarrow q)=1-v(p)+x.y$.

La paráfrasis de cada la modalidad de implicación que da nombre a cada uno de los grupos sería así:
- Mamdani: la regla es válida o tiende a ser válida cuando antecedente y consecuente tienen valores altos o muy altos.
- Gödel: la regla es absolutamente válida cuando el valor del antecedente es menor o igual que el del consecuente; y tiende a ser válida cuando el valor del consecuente es alto.
- Kleene-Dienes: la regla tiende a ser válida cuando el valor del antecedente es bajo o el del consecuente es alto.

Atendiendo a esta caracterización, intentamos seleccionar de los ejemplos extraídos, alguno que nos pareciera que estaba en consonancia con los grupos de implicación arriba distinguidos. En una primera aproximación la tarea pareció sencilla, pero pronto pudimos comprobar que la vaguedad era escurridiza, contextual, y dependía no sólo de los conocimientos que cada cual tenía, sino también de un estado particular de nuestra percepción, estado que suele ser transformista porque cambia demasiado fácilmente la apreciación de en qué sentido o desde qué perspectiva una regla muestra tal o cual aspecto de la vaguedad. En este estado de incertidumbre, propio para juzgar temas de vaguedad, pero poco útil para clarificarlos, buscamos ejemplos y el resultado es el siguiente, que sometemos a debate y crítica, porque en estos casos nos parece que sólo hay candidaturas plausibles, nunca afirmaciones seguras.

En el texto hay varios ejemplos de implicaciones tipo Kleene-Dienes. Son reglas que guardan una proporción entre lo que se afirma en el antecedente y lo que debería seguirse como consecuente, por lo que parecería que una implicación en la que se diese este equilibrio debería arrojar un grado alto de verdad. Estos son algunos ejemplos:

◆ *Cuanto más intensa es la estimulación, más fuerte es el sonido producido.*

◆ *Cuanto más alta es la frecuencia del sonido, mayor será su altura de tono*

◆ *Cuando pulsamos la tecla de un piano producimos sonidos más fuertes o más suaves según la fuerza de*

nuestras pulsaciones.

En estos casos, si le diéramos al antecedente una evaluación de verdad en grado 0,3 y al consecuente en grado 0,2, parecería que la regla se cumpliría bastante, porque mantiene la proporción que enuncia, y así lo define Kleene-Dienes. Contrariamente, tanto una implicación tipo Mamdani como tipo Gödel le darían, en cambio, un valor muy bajo a la regla, lo que no parece plausible.

Como ejemplos de implicación tipo Mamdani hemos seleccionado estos dos, si bien con algunas restricciones:

- *Si el compás de una pieza está incompleto con respecto a su medida, el último debe proporcionar el valor del tiempo que le falta, completando así la simetría del conjunto.*
- *Evidentemente, el acorde de séptima dominante debe relajarse, o en términos de música, 'resolverse', tarde o temprano; por tanto, es necesario cambiar el acorde consonante a uno disonante.*

Elegimos estas dos frases por motivos distintos, aunque cada una de ellas tiene su talón de Aquiles. La primera la seleccionamos porque parece poder parafrasearse como una conjunción (el compás de una pieza está incompleto con respecto a su medida y el último compás debe proporcionar el valor del tiempo que le falta) y es conmutativa (...completando así la simetría...). En su contra tiene que es una oración en la que es difícil ver la vaguedad o, más bien, donde no tiene sentido quizás discriminar entre la vaguedad que supone completar un compás con una o dos notas –supuesto que quede completo con tres-. La segunda frase la elegimos porque, dada la aseveración de lo que se podría considerar como el antecedente, no parece tanto una condición como una aserción (Evidentemente...). En ese caso, es más fácil parafrasearla en términos de una conjunción. Por otra parte, la frase en este caso sí parece vaga. Lo que es más discutible es la conmutatividad antecedente-consecuente.

El ejemplo tipo Gödel nos parecía el más difícil de encontrar, justamente porque, tratándose de predicados vagos, es muy difícil imaginar una situación en la que no se dé el sorites. Y, efectivamente, así resultó. La estrategia que tuvimos que adoptar fue distinta a las anteriores. No se trató de conseguir un caso adecuado bajo cualquier aspecto, sino un ejemplo en el que el consecuente (c) estuviese contenido en el antecedente (a) y figurase como un elemento explícito de éste, de manera que (c) nunca pudiese ser mayor que (a). Así se evitaría tener que juzgar el caso raro en el que, para valores bajos, un (a) levemente más valorado que (c) conllevase un valor de verdad bajo; mientras que el caso contrario forzase su verdad absoluta. El siguiente ejemplo parece responder a esta intuición:

- *Cuando dos objetos poseen frecuencias iguales y uno de ellos está en vibración, el otro, sin ser tocado, vibra simpáticamente.*

En este caso, el valor de (a) tomará el mínimo de los valores de las dos oraciones de que consta y (c), dado que repite una de las condiciones de (a) –entendiendo la vibración simpática como una "especie" de identidad de vibraciones de los dos objetos–, nunca podrá ser mayor que el mínimo. En consecuencia, el valor de (a) será menor o igual que el de (c) y la implicación será absolutamente verdadera.

A continuación se muestra la página Web en la que se implementa este test para los ejemplos seleccionados.

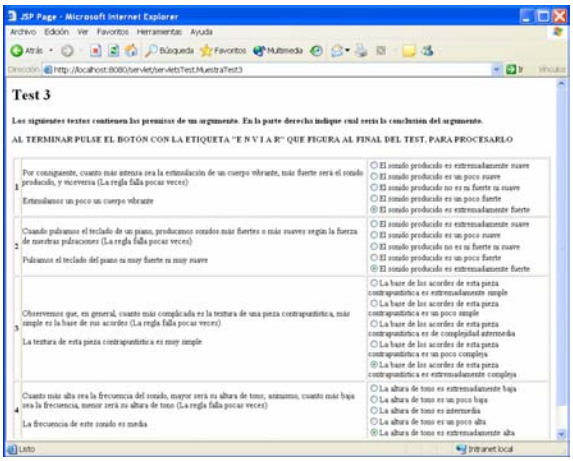


Fig5. : *Página web del Test 3*

5. CONCLUSIONES

En este trabajo se ha querido mostrar dos aspectos de una posible experimentación con lógicas difusas: 1) construir una base de datos de reglas aproximadas contra la que lanzar hipótesis sobre la adecuación de lógicas difusas, 2) mostrar a personas racionales ejemplos extraídos de 1) para que diagnostiquen si las distintas lógicas responden a aquello que dicen aprehender. Ambas tareas deberán ser completadas en el futuro, ampliando pasando estos tests a muestras representativas de personas y analizando los datos para extraer consecuencias útiles.

Agradecimientos
Proyecto PGIDT02PXIB30501PR de la Xunta de Galicia.

Referencias

[1] Ayuso, M. C. (comp.), *Razonamiento y racionalidad. ¿Sómos lógicos?*. Paidós, 1997.
[2] Kraft, D. et al. 'Rules and fuzzy rules in text: concept, extraction and usage', *Int. J. of Approximate Reasoning*, 34 (2003), 145-161.
[3] Lakatos, I., *Pruebas y refutaciones: la lógica del descubrimiento matemático*. Alianza, 1986 (reimp.).
[4] Zadeh, L.A., *Fuzzy Sets and Applications: Selected Papers*. John Wiley and Sons, 1987.
[5] Trillas, E., 'Una frase de Vila-Matas'. En estas Actas.

El INDICADOR *UNIVERSITAS SOCIALIS* : EL DIAGNÓSTICO DE LA RESPONSABILIDAD SOCIAL UNIVERSITARIA MEDIANTE UN SISTEMA EXPERTO BORROSO

Carlos Caño Alegre
http://sicodinet.unileon.es
ddecca@unileon.es

Enrique López González
http://sicodinet.unileon.es
ddeelg@unileon.es

Cristina Mendaña Cuervo
http://sicodinet.unileon.es
ddecmc@unileon.es

Resumen

La cuestión de la responsabilidad social de las universidades pasa por ser un tema trascendental en el desarrollo de cualquier país, ya que la educación de las generaciones futuras en unos valores, incide directamente el desarrollo sostenible del conocimiento, de la economía y de otros aspectos vitales de la vida en sociedad.

En este trabajo se desarrolla un indicador que se basa en un modelo de encuesta cuyos datos evalúa a diferentes niveles con el fin de obtener valoraciones lo más fieles posibles a la realidad aplicando un sistema experto borroso (SEB) implementado en la plataforma XFuzzy 3.0. Conviene resaltar que lo que se mide es el grado en que la propia responsabilidad social es percibida, con lo que se plasma la necesidad de evidenciar los esfuerzos realizados en este sentido y no solo de practicar políticas abstractas.

Palabras Clave: Tratamiento borroso de encuestas de muestreo, aplicaciones de sistemas expertos borrosos, XFuzzy 3.0, responsabilidad social universitaria, método de absorción-expansión de desviaciones, agregación reflexiva.

1 INTRODUCCIÓN

En los últimos años se están produciendo grandes cambios en las estructuras sociales, volcándonos hacia una economía basada en conocimientos, que pasa a ser el activo más importante. La forma de organizar el trabajo, el poder y la cultura se transforma rápidamente debido a la influencia de las modernas formas de intercambio de información.

Las universidades no son ajenas a este vaivén, evolucionando desde tradicionales actividades académicas localizadas en puntos fuertes de conocimiento hacia generación de conocimiento en campos interdisciplinarios con equipos de trabajo cuya sinergia aumenta con la posibilidad de interrelacionarse a través de medios rápidos entre ambientes diferentes.

Si embargo, la indudable mejora de los niveles de vida y desarrollo económico (en los países desarrollados y en vías de desarrollo, no así en la gran mayoría de países subdesarrollados), no lleva parejo el compromiso de las personas y su identificación con el resto del país. Se podría decir que la generación de conocimiento no ha ido acompañada de la evolución de los valores fundamentales, sino al contrario, en ocasiones, en detrimento de estos.

En este contexto, un interesante proyecto desarrollado en Chile denominado *Universidad: Construye País* facilita una aproximación al concepto de Responsabilidad Social Universitaria e intenta determinar un modelo de evaluación que permita conocer el grado de dicha responsabilidad percibido por PDI, PAS y estudiantes.

Con base al proyecto *Universidad: Construye País* este trabajo se orienta hacia el diseño de un indicador de Responsabilidad Social Universitaria aplicando técnicas del campo del soft-computing, con un diseño a varios niveles, como modelo para el desarrollo de indicadores sobre encuestas de muestreo mediante un sistema experto borroso.

El trabajo se estructura como sigue: En la sección 2 del presente trabajo se explica el método de absorción-expansión de desviaciones que hemos desarrollado. En la sección 3, se describe el procedimiento genérico para encuestas que se pretende aplicar a un caso concreto. En la sección 4, se detalla el modelo que se ha seguido para aplicar el procedimiento a un caso concreto de indicador de responsabilidad social universitaria.

2 MÉTODO DE ABSORCIÓN-EXPANSIÓN DE DESVIACIONES

2.1. OBJETIVO

Definidos una serie de conjuntos borrosos sobre una variable, y una batería de valoraciones crisp a aplicar sobre dicha variable, se intenta obtener un valor crisp representativo de dichas valoraciones. A su vez, se intenta, por un lado, que los valores que puedan ser considerados aberrantes no influyan demasiado en el valor final (absorción) y, por otro, permitir que se pueda realizar una clasificación más concisa del resultado amplificando su desviación hacia valores representativos cercanos (desviación).

2.2. DESARROLLO DEL MÉTODO

2.2.1. Representatividad de conjuntos

Se obtiene el porcentaje de representación de cada conjunto dentro de la variable, de acuerdo al peso que acumula su área tras aplicar toda la batería de valoraciones disponibles y su ponderación respectiva.

El hecho de incluir la ponderación de conjuntos p_i permite amplificar la importancia de ciertos conjuntos sin necesidad de modificar su función de pertenencia, puesto que la modificación de la función no solo modificaría el grado de relevancia sino por definición la pertenencia al conjunto de los valores del dominio. De este modo se plantea la independencia entre la forma del número borroso y la importancia que tendrá en una valoración reflexiva (se obtiene un valor a partir de una batería de valores, sobre una misma variable):

$$X_i = \frac{\sum_{j=1}^{n} p_i \cdot A_{ij}}{\sum_{j=1}^{n}\sum_{k=1}^{l} p_k \cdot A_{ij}} \cdot 100 \qquad (1)$$

En la fórmula (1) se tiene que: n es el número de valoraciones, l el número de etiquetas, X_i el porcentaje representativo del conjunto i-ésimo de la variable, p_i la ponderación del conjunto i-ésimo y Aij el área obtenida para el conjunto i-ésimo en la valoración j-ésima sobre la variable.

El cálculo del área se realiza por el procedimiento normal. Dada la función de pertenencia de un conjunto borroso $f(x)$ que toma valores en [0,1], primero se obtiene la función de pertenencia podada por el valor a (2) y luego se calcula el área de la parte inferior de la función mediante (3).

$$f'(x) = \begin{cases} f(x) \le f(a) \Rightarrow f'(x) = f(x) \\ f(x) > f(a) \Rightarrow f'(x) = f(a) \end{cases} \qquad (2)$$

$$A = \int f'(x) \qquad (3)$$

Llegados a este punto, se dispone de un porcentaje de representatividad asociado a cada conjunto borroso de la variable. Para una variable de 5 etiquetas lingüísticas, por ejemplo, se tendrían 5 porcentajes de relevancia. Si la mayoría de las valoraciones se concentraran en una determinada etiqueta, obviamente esta tendría una porcentaje de relevancia mucho mayor que las demás.

2.2.2. Cálculo de alturas

A partir de los porcentajes de representatividad se obtiene la altura a la que se debe cortar cada conjunto para que su área dentro del total se corresponda con la que debe tener según el porcentaje de representatividad calculado.

El proceso de obtención de las alturas tiene varios pasos:

1. Se fija una altura a 1 (para $t \rightarrow 0$ se fija la del conjunto de mayor representatividad) tomando ese conjunto como el de referencia.

2. Se calcula el área del conjunto de referencia.

3. Se obtiene el área que debe corresponder al conjunto i-ésimo mediante (4).

$$A_i = \frac{p_i}{p_{referencia}} \cdot A_{referencia} \qquad (4)$$

4. Se calcula la altura a la que se debe podar el conjunto i-ésimo para que su área sea la obtenida en relación al conjunto de referencia.

5. Si no se verifica que $0 \le h_i \le 1$ para todos los conjuntos, se regresa al paso 1 con $t \rightarrow t+1$.

Tras realizar el cálculo de alturas se obtiene un perfil de la variable en el que cada etiqueta ha sido podada por una altura determinada de modo que su área se corresponda con el porcentaje de relevancia que tiene.

2.2.3. Clarificación

Como método de clarificación tomamos el centro de masas, ya que es el más explícito en la concentración de valores. La función de pertenencia agregada sobre la que realizaremos el cálculo se define por (5).

$$F'(x) = Sup\left(f_i'(x)\right) \quad (5)$$

Entendiendo centro de masas como el punto *m* del dominio [a, b], para el que se verifica (6).

$$\int_a^m F'(x) = \frac{\int_a^b F'(x)}{2} \quad (6)$$

3 ESTRUCTURACIÓN DEL INDICADOR GENÉRICO PARA ENCUESTAS

3.1. DESCRIPCIÓN ESTRUCTURAL

El indicador tiene un desarrollo estructurado en capas, con el fin de que sea posible su aplicación directamente sobre una batería completa de encuestas.

Debido a que los puntos de interfaz entre las capas están reducidos, tenemos cierta independencia entre unas y otras, lo que nos puede permitir por ejemplo que no sea necesario tener encuestas respondidas por completo, sino que los encuestados pueden responder a aquellas cuestiones que conozcan y dejar sin respuesta aquellas que desconozcan, con lo que mejoramos la fiabilidad de los resultados.

3.2. CAPA 1: REPRESENTATIVIDAD MUESTRAL

En esta capa se evalúan los cuestionarios de forma transversal, esto es, se utilizan todas las respuestas obtenidas para una misma cuestión. El objetivo que se plantea es obtener un valor de respuesta para la cuestión, suficientemente representativo de todo el muestreo realizado.

El método de resolución utilizado aquí, será el definido como de absorción-expansión de desviaciones. Los criterios que marcarán los resultados obtenidos por la aplicación del método serán las funciones de pertenencia de cada conjunto y sus ponderaciones asociadas. El resultado final será un conjunto de entradas para el cuestionario que representarán al conjunto de la población de muestra.

3.3. CAPA 2: COHERENCIA MODULAR

En esta capa se realiza la evaluación modular del cuestionario a partir de valores representativos obtenidos. El objetivo es aplicar un cierto nivel de coherencia entre las cuestiones relativas a un mismo módulo o tópico, de forma que se obtenga una valoración para este lo más coherente posible a pesar de posibles disonancias en grupos minoritarios de cuestiones.

El método de resolución utilizado aquí, será el definido como de absorción-expansión de desviaciones. Los criterios que marcarán los resultados obtenidos por la aplicación del método serán las funciones de pertenencia de cada conjunto y sus ponderaciones asociadas. El resultado final será un valor resumen para cada módulo o tópico de la encuesta.

3.4. CAPA 3: MOTOR LÓGICO DE INFERENCIA

En esta capa se comienza la evaluación lógica de los valores obtenidos para cada módulo. El objetivo es obtener una valoración racional del concepto a evaluar, a partir de todos los conceptos abstractos que se relacionan con el.

El método de resolución utilizado aquí, será un sistema experto jerarquizado, cuyas bases de reglas, operadores y funciones de pertenencia de los conjuntos pertenecientes a cada variable determinarán cual ha de ser el valor resultante. El resultado final será el valor del indicador propuesto.

4 DISEÑO DE UN INDICADOR DE RESPONSABILIDAD SOCIAL UNIVERSITARIA BASADO EN UN SEB

4.1. CONCEPTUALIZACIÓN DEL INDICADOR

De acuerdo con el Equipo Coordinador del Proyecto Universidad Construye País, se entiende por Responsabilidad Social Universitaria la "capacidad que tiene la universidad, de difundir y poner en práctica un conjunto de principios y valores generales y específicos, por medio de cuatro procesos considerados claves en la universidad, como son la gestión, la docencia, la investigación y la extensión universitaria. Respondiendo socialmente así, ante la propia comunidad universitaria y el país donde está inserta."[7]. No obstante, cabe destacar que existen diferencias relevantes en los aspectos concretos en los que ha de manifestarse la responsabilidad entre los países desarrollados y los latinoamericanos. En los primeros, el compromiso se centra más en términos cívicos mientras que en el caso latinoamericano se atiende más al problema de las desigualdades sociales que caracterizan la realidad de la región.

En el presente nuestro caso de aplicación desarrollado a efectos ilustrativos del trabajo se ha partido del modelo de encuesta propuesto por *Universidad: Construye País* como base, y sobre el hemos realizado modificaciones y construido el proceso de evaluación del indicador según el modelo estructural genérico que proponemos para encuestas segmentado en capas.

4.2. REPRESENTATIVIDAD MUESTRAL

Al objeto de establecer la representación muestral se utiliza el método de absorción-expansión de desviaciones con la variable descrita en la tabla 1 y las ponderaciones de la tabla 2.

Tabla 1: Variable utilizada en representatividad

Etiqueta	Coordenadas			
Muy desacuerdo	-5	-5	17.5	25
Desacuerdo	15	30		45
A medias	35	50		65
De acuerdo	55	70		85
Muy de acuerdo	75	82.5	105	105

Tabla 2: Ponderaciones conjuntos borrosos

Etiqueta	$P_{ETIQUETA}$
Muy desacuerdo	1
Desacuerdo	1
A medias	1
De acuerdo	1
Muy de acuerdo	1

4.3. COHERENCIA MODULAR

Al objeto de establecer la coherencia modular se utiliza el método de absorción-expansión de desviaciones con la variable descrita en la tabla 3 y las ponderaciones de la tabla 4.

Tabla 3: Variable utilizada en coherencia

Etiqueta	Coordenadas			
Muy Malo	-5	-5	0	25
Malo	15	30		45
Normal	30	45	55	70
Bueno	55	70		85
Muy Bueno	75	100	105	105

Tabla 4: Ponderaciones conjuntos borrosos

Etiqueta	$P_{ETIQUETA}$
Muy Malo	0.6
Malo	0.8
Normal	0.5
Bueno	0.8
Muy Bueno	0.6

4.4. MOTOR LÓGICO DE INFERENCIA

El motor lógico de inferencia desarrollado se basa en un sistema experto basado en reglas y variables borrosas, construido con la herramienta XFuzzy3.0 [8], que nos otorga la posibilidad de generar código en java, c y c++, facilitándo la integración con un interfaz de usuario amigable.

La estructura jerárquica del sistema se indica en la figura 1. Los números se corresponden con los índices de las variables descritas en el apartado siguiente.

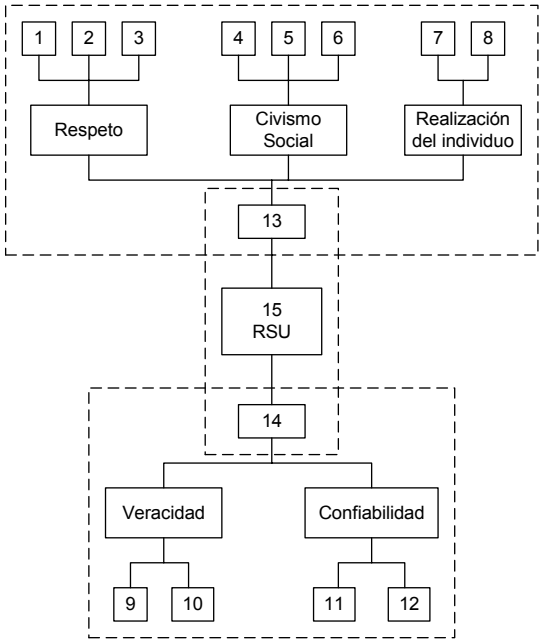


Figura 1: Mapa conceptual del sistema.

Para la construcción de este motor lógico de inferencia se definen las funciones de pertenencia de cada uno de los conjuntos pertenecientes a todas las variables. Asimismo, se definen también las reglas con las que trabajará el sistema y los operadores.

4.4.1. Variables

A continuación se detallan sólo las variables de entrada y salida de cada uno de los tres subsistemas utilizadas en el sistema, tal como se relacionan en la figura 1, ya que el resto son conceptos más abstractos que utilizamos para agrupar determinados tópicos y poder simplificar la implementación del indicador global que titula este trabajo.

1. **Dignidad de la persona.-** El ser humano nace libre e igual en derechos y deberes a sus semejantes. La persona debe ser capaz de construir relaciones solidarias apoyadas en una esfera de conocimientos. La universidad ha de favorecer el desarrollo de la autonomía, la razón y la conciencia.
2. **Igualdad entre sexos.-** Las tradicionales barreras sociales para la igualdad entre ambos sexos, son anacrónicas en una sociedad globalizada y en la que la fuerza del saber se encuentra por encima de la fuerza física.
3. **Aceptación y Aprecio de la Diversidad.-** Actitud de integración hacia el resto de individuos sin mediar discriminación por razones

de raza, sexo, edad, condición religiosa, social, política, etc.

4. **Ciudadanía, Democracia y Participación.**- El individuo ha de ser consciente de los derechos y deberes que adquiere por pertenecer a una comunidad social y/o política, desarrollando su espíritu reflexivo y una actitud participativa responsable. La universidad ha de gestionar sus órganos de gobierno y normativas internas definiendo claramente los derechos y deberes de sus miembros.
5. **Sociabilidad y Solidaridad.**- La persona a través de sus libertad tiene la capacidad de realizar actos en los que se cubren altruistamente las necesidades de los demás dentro de un sano ambiente de convivencia. La universidad debe favorecer el sentimiento de pertenencia universitaria y la crítica positiva, así como el reconocimiento de los aportes a la construcción de un favorable clima de convivencia.
6. **Desarrollo Sostenible y Medio Ambiente.**- Proceso que permite el desarrollo indefinido de las sociedades y culturas, y que los efectos de la actividad humana se mantengan dentro de sus límites. La universidad debe favorecer le desarrollo permanente de su capital humano y el perfeccionamientos de las relaciones internas.
7. Bien Común y Equidad Social.- Igualdad de oportunidades tanto para cubrir las necesidades básicas que plantea una forma digna de vida, como para favorecer el desarrollo y la mejora de la propia calidad de vida. La universidad debe mantener un desarrollo equilibrado de todas las instituciones que la componen.
8. **Libertad.**- La persona puede auto-realizarse en todas las dimensiones de su personalidad. La universidad debe respetar los derechos y libertades de todos los miembros de su comunidad.
9. Compromiso con la Verdad.- La verdad siempre esta en progreso y es el fruto de una relación de conocimientos. La honestidad para declarar los propios límites. La universidad debe primar la gratuidad del saber, para el uso o aplicación inmediata de ellos.
10. **Integridad.**- Cualidad de personas e instituciones que son coherentes con sus principios y valores, tanto en sus declaraciones como en sus acciones, siendo rectas, transparentes y honestas.
11. **Excelencia.**- Calidad superior de personas o instituciones que las distingue del resto.
12. **Interdependencia e Interdisciplinariedad.**- La relación entre la universidad y la sociedad es de absoluta interdependencia por los vínculos innegables que existen entre ambas. La universidad debe poner sus miras en los problemas de la sociedad que manifiestan la necesaria interdisciplinariedad.
13. **Valores generales de la sociedad.**- Principios y valores que no permitirán desarrollar los valores específicos de la universidad si no están presentes en la sociedad en la cual la universidad se desarrolla.
14. **Valores propios de la universidad.**- Principios y valores que tienen una vital importancia en la construcción de la universidad como comunidad de vida y de saber.
15. **Responsabilidad Social Universitaria**.- Capacidad que tiene la universidad, de difundir y poner en práctica un conjunto de principios y valores generales y específicos, por medio de cuatro procesos considerados claves en la universidad, como son la gestión, la docencia, la investigación y la extensión universitaria.

También se utiliza la posibilidad que ofrece el XFuzzy 3.0 de definir tipos y utilizarlos como propios de varias variables, ya que la valoración individual de los conceptos en sí es la misma. Para ello se empleará el tipo descrito en la tabla 5.

Tabla 5: Tipo genérico

Etiqueta	**Coordenadas**		
Muy Malo	0	0	25
Malo	17.5	30	42.5
Normal	35	50	65
Bueno	57.5	70	82.5
Muy Bueno	75	100	100

4.4.2. Bases de reglas

En este trabajo se ha estructurado el sistema en tres módulos de inferencia como se puede ver en la figura 1, con la finalidad de obtener salidas intermedias relevantes.

1. Sistema de valores sociales, que nos da el indicador de la apreciación que se tienen de valores generales de la sociedad, dentro de la universidad.
2. Sistema de valores universitarios, que nos da un indicador de la apreciación que se tiene de valores específicos de la universidad, dentro de la misma.
3. Sistema de responsabilidad social universitaria, propiamente dicho. Este valor es obtenido a partir del resultado de los otros dos módulos, ya que la evaluación de estos se considera un paso previo.

Aunque se obtienen los valores clarificados de los dos sistemas iniciales, para el cálculo del valor final no se utilizan estos valores, sino las expresiones difusas de las que derivan.

Las bases de reglas se construyen atendiendo a un proceso racional y primando implícitamente los conceptos que según la escena social española hemos considerando de mayor relevancia.

4.4.3. Operadores lógicos

Dado que los operadores definen en gran parte el resultado final del sistema, hemos tomado unos que representen específicamente las necesidades de obtener valoraciones de consenso entre todo el muestreo, tomando las opiniones mayoritarias pero manteniendo ciertos matices minoritarios.

Teniendo en cuenta las razones anteriores, utilizamos los operadores de la tabla 6.

Tabla 6: Operadores lógicos

Operación	Operador
And	$\min(h_a,h_b)$
Or	$\max(h_a,h_b)$
Defuzzificador	centro de masas

4.5. APLICACIÓN

A fecha de hoy, aún no se han realizado pruebas con el modelo completo, pero se esta desarrollando una encuesta sobre la herramienta PHPSurveyor que será "colgado" en la página web de nuestro grupo de investigación (http://sicodinet.unileon.es/universitassocialis) al objeto de que todos los interesados puedan realizar valoraciones sobre la responsabilidad social de las universidades y poder analizar los resultados.

Es de confiar que la página este funcionando en las fechas del congreso y ofrecer entonces algunos resultados reales del indicador.

5 CONCLUSIONES

El modelo planteado en este trabajo puede ser útil para su aplicación tanto en la propia creación de indicadores a partir de encuestas de muestreo, ofreciendo una alternativa más completa e interesante a tradicionales métodos estadísticos como en el diseño de indicadores de gestión.

En relación al caso ilustrativo analizado en este trabajo, el indicador Universitas Socialis, se considera viable como desarrollo futuro a corto plazo la construcción de una aplicación on-line que permita la realización de valoraciones, visualización de los resultados de las encuestas… Todo ello orientado a ofrecer información que sirva como un indicador del estado real y actual del nivel de responsabilidad social de las universidades españolas.

Agradecimientos

A los desarrolladores de la herramienta informática XFuzzy 3.0 (http://www.imse.cnm.es/Xfuzzy) por la maniobrabilidad que nos permite en la definición del sistema y la cantidad del tiempo ahorrado. Así mismo, a los revisores anónimos del Comité Científico de ESTYL'04 por sus consejos y correcciones.

Referencias

[1] H. Berenji. *Fuzzy Logic Controllers*. Incluido en "*An Introduction to Fuzzy Logic Applications in Intelligent Systems*", YAGER, R. y ZADEH, L. (eds.), Kluwer Academic, Boston, Pág. 69-96, 1993.

[2] E. Cárdenas, J. Castillo, O. Cordón, F. Herrera y A. Peregrín. *Influence of Fuzzy Implication Functions and Defuzzification Methods in Fuzzy Control*, Busefal, Vol. 57, Pág. 69-79, 1994.

[3] M. Emami, J. Türksen y A. Goldenberg. *Development of A Systematic Methodology of Fuzzy Logic Modelling*, IEEE Transactions on Fuzzy Systems, Vol. 6, Pág. 346-361, 1998.

[4] C. Lee. *Fuzzy Logic in Control Systems: Fuzzy Logic Controller* - Parts I and II, IEEE Trans. on Systems, Man and Cybernetics, Vol. 20, nº 2, Pág. 404-418 y 419-435, 1991.

[5] E. López González. *A Methodology for Building Fuzzy Expert Systems (FES) with Spreadsheet to Quality Function Deployment (QFD) of the Target Costing*. Incluido en Gil Aluja, J. (ed): "*Handbook of Management under Uncertainty*". Kluwer Academic Publishers. Dordrecht, Pág. 457-535, 2001.

[6] E. Mamdani. *Applications to fuzzy algorithms for simple dynamic plant*, Proceedings IEEE, Vol. 21, nº 12, Pág. 1585-1588, 1974.

[7] Equipo Coordinador del Proyecto Universidad Construye País. *Aproximación al concepto de responsabilidad social universitaria*, 2002 (http://www.construyepais.cl).

[8] F. J. Moreno-Velo, I. Baturone, S. Sánchez-Solano, A. Barriga. *XFUZZY 3.0: A Development Environment for Fuzzy Systems. Proc. International Conference in Fuzzy Logic and Technology*, Pág. 93-96, 2001.

[9] T. Takagi y M. Sugeno. *Fuzzy Identification of Systems and Its Applications to Modeling and Control*, IEEE Trans. on Systems, Man, and Cybernetics, Vol. 15, nº 1, Pág. 116-132, 1985.

[10] L. A. Zadeh. *Fuzzy Logic = Computing with words*, IEEE Trans. on Fuzzy Systems, Vol. 4, Pág. 103-11, 1996.

DISEÑO DE UN SISTEMA EXPERTO BORROSOS PARA EL CONTROL Y SEGUIMIENTO INTEGRADO DE DEPORTISTAS

Carlos Caño Alegre
http://sicodinet.unileon.es
ddecca@unileon.es

Cristina Mendaña Cuervo
http://sicodinet.unileon.es
ddecmc@unileon.es

Enrique López González
http://sicodinet.unileon.es
ddeelg@unileon.es

Resumen

La intensa carga de trabajo soportada por muchos deportistas es causa de numerosas lesiones. Esta idea ha motivado el siguiente desarrollo.

La posibilidad de realizar valoraciones sobre el estado de los jugadores en un momento concreto o visualizar el histórico de una serie de variables mejora indudablemente la accesibilidad a la información y nos permite mantener un control de posibles estados de riesgo por fatiga física o psicológica del futbolista.

Muchos de los tests o pruebas realizados a los deportistas presentan límites de aplicación amen de cierta inexactitud, por lo que el uso de métodos de manejo de incertidumbre como la lógica fuzzy pasa por ser una buena herramienta.

Palabras Clave: Seguimiento deportivo, fútbol, psicología, antropometría, funcional, valoración estado de riesgo, lógica difusa.

1 INTRODUCCIÓN

La evolución del deporte durante el siglo pasado, ha convertido actividades tradicionalmente lúdicas, realizadas por deportistas amateur, cuya finalidad era simplemente mantenerse en forma y pasar un rato agradable, en actividades cargadas de competitividad, que por ello exigen un elevado esfuerzo y dedicación, convirtiéndose en la forma de vida de la mayoría de los deportistas profesionales de la actualidad que persiguen alcanzar un cierto nivel en su respectivo deporte.

El aumento de la competitividad, implica a su vez un aumento considerable de la carga de trabajo que han de soportar estos deportistas, la cual está relacionada directamente con sus propias capacidades, limitaciones y las metas que deseen alcanzar.

Aunque en los tiempos que corren existen una gran variedad de deportes profesionales, un caso característico y que tradicionalmente está en el punto de mira informativo es el del fútbol. Los futbolistas de élite suelen soportar una gran carga de partidos a lo largo de una temporada, la cual en muchos casos resulta excesiva y deviene en lesiones que podrían evitarse tal vez con un calendario menos sobrecargado, la aplicación de rotaciones dentro de las plantillas y la mejor planificación de la carga de trabajo en los entrenamientos para permitir la recuperación. Algunas de estas medidas ya están siendo puestas en marcha, en unos casos con mayor éxito que en otros.

A pesar de todo esto, el deporte rey es un increíble negocio que mueve grandes cantidades de dinero y por ello los futbolistas son forzados a jugar en condiciones de riesgo, en muchas ocasiones desconocidas, con lo cual consideramos que tal vez una mejora de los mecanismos de control y seguimiento del estado de los deportistas sería satisfactoria tanto para ellos como para los clubes.

La finalidad que se persigue no es otra que aprovechar al máximo las capacidades que tiene un futbolista, pero sin llevarlo a situaciones de riesgo por sobrecarga de trabajo.

Aunque el caso más obvio que podemos plantearnos son las sobrecargas musculares y otros percances físicos, también existe un cansancio psicológico que se produce al tener que mantener un elevado grado de concentración durante largos periodos. Esto hace necesario expandir el control no solo a los niveles físicos, sino también a los psíquicos.

A partir de una serie de variables difusas encuadradas en tres aspectos bien diferenciados, se construirá un motor de inferencia mediante el software de desarrollo XFuzzy, para evaluar todas los niveles y obtener una valoración de cómo se encuentra el deportista.

En la sección 2 del trabajo, se explican los aspectos antropométricos a valorar dentro del estado del jugador referentes a la biometría.

En la sección 3, se detallan una serie de test que ayudan a valorar el estado funcional, esto es capacidad de aceleración, recuperación, etc…

En la sección 4, se describen variables que cuantifican el estado psicológico del jugador.

Finalmente, la sección 5 presenta ciertos aspectos sobre el desarrollo y la integración de la aplicación.

2 ASPECTOS ANTROPOMÉTRICOS DEL JUGADOR

La Cineantropometría engloba el análisis de diferentes aspectos del individuo, en especial los que se relacionan con su complexión física, como la forma, la proporcionalidad y la composición corporal. Sustituye, en términos de curso, a la Biometría que es impartida tradicionalmente en las Escuelas de Educación Física, y que trata sobre todo el área somática, utilizando la bioestadística para valorar los fenómenos biológicos envueltos.

Ya hace cuatro décadas Sheldon creó el término somatotipo y las técnicas fundamentales para su análisis. Sheldon expone la teoría de los tres componentes primarios del cuerpo humano, presentes en todos los individuos, en mayor o menor grado [12].

Varios aspectos del método de Sheldon fueron criticados o modificados y surgieron técnicas complementarias para perfeccionar la idea básica de expresar la forma humana a partir de tres componentes básicos. El concepto actual de somatotipo elaborado por Heath, expone que el somatotipo describe la configuración morfológica actual del individuo y que por tanto no se vincula estrictamente al potencial genético, siendo modificado por el crecimiento y por el entrenamiento.

2.1. CARACTERÍSTICAS ESTUDIADAS

Con el fin de efectuar el seguimiento y control del estado del deportista, tomamos una serie de características que el programa nos permitirá evaluar.

1. Distancia de Dispersión del Somatotipo.- La distancia de dispersión indica la distancia entre dos descriptores somatotípicos. En nuestro caso de estudio representará la distancia entre el somatotipo ideal para un futbolista de élite {2.1 – 5.3 – 2.1} y el somatotipo del individuo estudiado. Cuanto menor sea la distancia, mejor condición tendrá el jugador para practicar el fútbol.

2. Índice Córmico.- Relaciona la talla sentado con la estatura. Aquellos individuos clasificados según este índice como braquicórmicos son los más aptos.

3. Envergadura Relativa.- Relación entre la envergadura y la estatura. La similitud entre ambas medidas es muy propicia para la práctica del deporte.

4. Índice de Masa Corporal.- Índice establecido por la OMS[1] que relaciona el peso y la estatura. En la tabla 1 se ha realizado una descomposición borrosa de la variable, a partir de la clasificación rigurosa de la OMS, para flexibilizarla y permitir un manejo más natural.

Tabla 1: Variable IMC

Etiqueta	Coordenadas			
Bajo peso	0	0	19	21
Normal	19	22.5		26
Obesidad leve	24	27.5		31
Obesidad severa	29	35		41
Obesidad muy severa	39	50	70	70

2.2. COEFICIENTES AUXILIARES

Para completar la funcionalidad de la aplicación se usan otros campos en los que incluir coeficientes evaluables.

1. Valoración del antropometrista.- Permite incluir directamente la cualificada opinión del experto en antropometría, antes de realizar exámenes minuciosos. Nos sirve para incluir en la valoración, el carácter intuitivo del experto que no es replicable por métodos analíticos.

2. Otros.- Exámenes, mediciones o pruebas antropométricas que no tengan una presencia nominal en la aplicación.

Tabla 2: Tipo genérico para variables

Etiqueta	Coordenadas			
Muy Malo	0	0		25
Malo	20	27.5		40
Normal	35	50		65
Bueno	60	72.5		85
Muy Bueno	80	90	100	100

En la tabla 2 se muestran los descriptores del tipo utilizado tanto para los coeficientes auxiliares como para las variables de salida de cada base de reglas intermedia y la variable final.

[1] Organización Mundial de la Salud. www.who.int/es/

2.3. MOTOR DE INFERENCIA ANTROPOMÉTRICO

A partir de las características estudiadas y los coeficientes auxiliares se obtiene una valoración del perfil antropométrico del jugador.

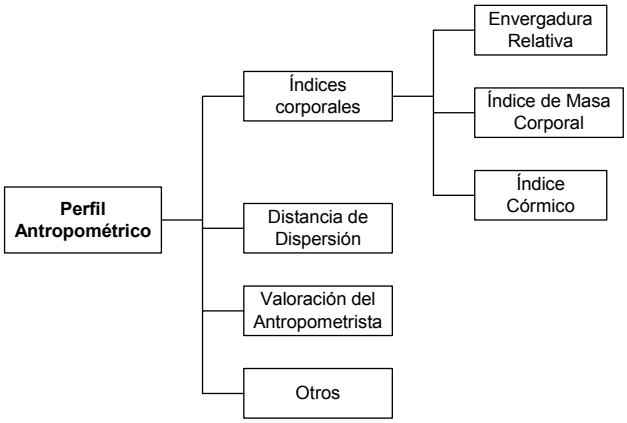


Figura 1: Mapa conceptual del perfil antropométrico.

En la figura 1 se muestra la estructura jerárquica del motor de inferencia utilizado para la evaluación antropométrica.

3 ASPECTOS FUNCIONALES DEL JUGADOR

Las cualidades fisiológicas fundamentales que debe poseer un jugador deben identificarse, en primer lugar, con las capacidades neuromusculares y, a continuación, con los sistemas bioenergéticos señalados para suministrar energía bioquímica, que se transforma en trabajo mecánico en los músculos.

Esta capacidades ayudan a desarrollar aceleraciones (realizar un lanzamiento), desaceleraciones (detenciones rápidas después de una fase lanzada), cambiar la dirección del movimiento, golpear de cabeza durante la ejecución de un salto, golpear el balón, etc.

Una de las características fundamentales del fútbol es la relacionada con la fuerza explosiva de los miembros inferiores (saltos, desplazamientos laterales, antero-posteriores, etc.). Las cualidades fisiológicas más interesantes son la capacidad de reclutamiento neuromuscular (frecuencia de los estímulos y sincronización de las unidades motoras incorporadas), el tipo de fibras musculares implicadas (fibras rápidas), la elasticidad muscular, que no debe identificarse con la flexibilidad, y las reservas energéticas de utilización inmediata y los enzimas relativos.

Para valorar la potencia explosiva de los miembros inferiores pueden utilizarse unos tests funcionales que, desgraciadamente, presentan límites de aplicación.

3.1. CARACTERÍSTICAS ESTUDIADAS

Las características funcionales del futbolista que evaluamos son el resultado de la aplicación de unos tests seleccionados.

1. Harvard Step Test.- También denominado índice de recuperación, indica la capacidad del individuo para restablecer las propias capacidades funcionales después del esfuerzo. En la tabla 3 se muestran los descriptores de cada una de las clasificaciones propuestas en el HST.

Tabla 3: Variable HST

Etiqueta	Coordenadas			
Pésimos	0	0		61
Mediocres	59	65		72
Buenos	69	75		82
Muy Buenos	79	85		92
Excelentes	89	95	100	100

2. Counter Movement Jump.- Sirve para obtener una valoración de la fuerza explosiva, necesaria en acciones como salto firme con contramovimiento (tiro de cabeza), capacidad de desaceleración, detención repentina con cambio de dirección.

3. Squat Jump.- Valora la fuerza explosiva que aporta capacidad de salto y capacidad de aceleración.

3.2. COEFICIENTES AUXILIARES

Para completar la funcionalidad de la aplicación se usan otros campos en los que incluir coeficientes evaluables referentes a las capacidades funcionales.

1. Valoración del preparador físico.- Permite incluir directamente la cualificada opinión del preparador físico, antes de realizar exámenes minuciosos. Nos sirve para incluir en la valoración, el carácter intuitivo del experto que no es replicable por métodos analíticos.

2. Otros.- Exámenes, mediciones o tests funcionales que no tengan una presencia nominal en la aplicación.

El tipo de variable utilizado es el que se muestra en la tabla 2.

3.3. MOTOR DE INFERENCIA FUNCIONAL

A partir de las características estudiadas y los coeficientes auxiliares se obtiene una valoración del perfil funcional del jugador.

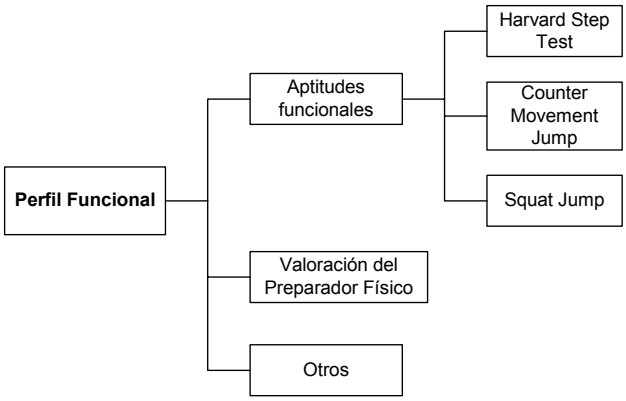


Figura 2: Mapa conceptual del perfil funcional.

En la figura 2 se muestra la estructura jerárquica del motor de inferencia utilizado para la evaluación funcional.

4 ASPECTOS PSICOLÓGICOS DEL JUGADOR

Los deportistas de élite han proporcionado numerosas evidencias anecdóticas de que las competiciones se ganan mediante la consecución de un "estado mental correcto" o se pierden como consecuencia del derrumbe ante las presiones de la competición. La relación entre el rendimiento y las puntuaciones en medidas psicológicas antes de la competición, especialmente el estado de ánimo y la ansiedad competitiva, han sido el foco de un esfuerzo investigador extensivo y continuado en la psicología del deporte [13].

4.1. CARACTERÍSTICAS ESTUDIADAS

Con el fin de efectuar el seguimiento y control del estado psicológico del deportista, tomamos una serie de características que el programa nos permitirá evaluar.

1. Inteligencia Emocional.- Indica los índices de autoconocimiento, autorregulación, motivación, empatía y habilidades sociales que posee el individuo.

2. Alteración Emocional.- Indica la rabia, tensión, depresión, vigor, fatiga y confusión. En la tabla 4 se propone una clasificación borrosa para la variable A.E.

Tabla 4: Variable Alteración Emocional

Etiqueta	Coordenadas		
Bajo	60	60	176
Normal	118	176	263
Alto	205	292	292

Esta variable de alteración es medida por el Test de POMS[2].

4.2. COEFICIENTES AUXILIARES

Para completar la funcionalidad de la aplicación se usan otros campos en los que incluir coeficientes evaluables referentes al estado psicológico.

1. Valoración del psicólogo.- Permite incluir directamente la cualificada opinión del psicólogo, antes de realizar exámenes minuciosos. Nos sirve para incluir en la valoración, el carácter intuitivo del experto que no es replicable por métodos analíticos.

2. Otros.- Exámenes o tests psicológicos que no tengan una presencia nominal en la aplicación.

El tipo de variable utilizado es el que se muestra en la tabla 2.

4.3. MOTOR DE INFERENCIA PSICOLÓGICO

A partir de las características estudiadas y los coeficientes auxiliares se obtiene una valoración del perfil psicológico del jugador.

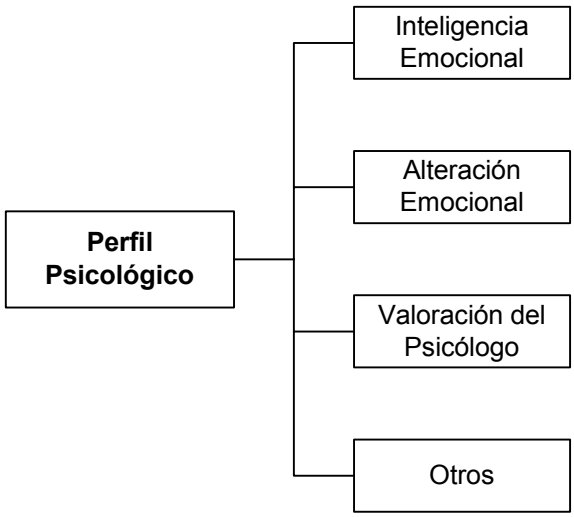


Figura 3: Mapa conceptual del perfil psicológico.

[2] Profile of Mood Status

En la figura 3 se muestra la estructura jerárquica del motor de inferencia utilizado para la evaluación psicológica.

5 ASPECTOS GENERALES DE LA APLICACIÓN

Los tres módulos se desarrollan y ejecutan en motores de inferencia independientes y sus resultados se agregan asignando un 25% al aspecto antropométrico, un 25% al aspecto psicológico y un 50% al aspecto funcional.

Las variables han sido diseñadas mediante la borrosificación de clasificaciones crisp, para permitir mayor flexibilidad a las valoraciones lingüísticas que la aplicación permite realizar.

Las bases de reglas se han construido correlacionando los valores de entrada-salida con relación conocida y los demás realizando interpretación de la naturaleza de las variables. Para dotar de mayor funcionalidad al sistema sería necesaria su revisión periódica por expertos.

La aplicación permite evaluar el perfil de un jugador en un momento concreto y obtener un gráfico del histórico de los valores para el repertorio de variables. Cabe destacar que no es necesario introducir todas las entradas para obtener las valoraciones, sino que es posible elegir que se quiere tener en cuenta para evaluar cada uno de los perfiles. A estos efectos se ha incluido una etiqueta singleton auxiliar en todos los tipos, que se utiliza cuando alguna variable no se desea tener en cuenta en la evaluación.

Los motores de inferencia son desarrollados en lenguaje XFL3, con ayuda del editor gráfico provisto por el XFuzzy 3.0.

Los datos introducidos en el sistema se almacenan en una base de datos relacional, para tener un acceso rápido y permitir consultas complejas directamente sobre el propio gestor de base de datos. Como SGBDR se han elegido PostgreSQL para linux y Microsoft Access para Windows.

El desarrollo se realiza en lenguaje java, por su completa transportabilidad y fácil integración con el código generado por el XFuzzy 3.0.

6 DISCUSIÓN Y CONCLUSIÓN

El sistema demuestra que es factible la utilización de sistemas expertos en este tipo de tareas de apoyo a la gestión deportiva, en aspectos en los que existe un conocimiento más o menos exhaustivo y que el añadir una etiqueta singleton, para señalar que una variable no se va a valorar, nos da una mayor flexibilidad que el obligar a introducir todos los valores, ya que en muchas ocasiones en circunstancias reales no se dispone de todos los datos y se han de tomar decisiones con los que hay disponibles.

Sobre el hecho de ayudar a la gestión deportiva con técnicas de soft-computing existe un proyecto denominado MilanLab impulsado por el A.C. Milan que se centra en el estudio del estado de los jugadores para poder optimizar su rendimiento, pero se orienta a la utilización de redes neuronales para el reconocimiento de circunstancias de riesgo.

Concluimos que la herramienta descrita en este trabajo puede facilitar el seguimiento del estado psicológico, físico y funcional de los futbolistas, en circunstancias en las que no se disponga de todos los datos, y con ello permitir detectar situaciones de riesgo que ayuden a aprovechar de una manera más responsable las capacidades deportivas.

6 LÍNEAS DE FUTURO

Con respecto a las líneas de desarrollo futuro a seguir se marcan varias tendencias:

1. Se debería completar y revisar la base de conocimiento con ayuda de expertos cualificados en un proceso de mejora iterativo.

2. Se pueden realizar versiones que permitan interactuar con el sistema a través de otros soportes distintos del ordenador, como por ejemplo el móvil. En este sentido cabe destacar que existe una versión del propio lenguaje java desarrollada para móviles y que muchos de los terminales ya soportan esta tecnología.

3. En situaciones en las que puedan disponerse de todos los datos podría considerarse para su estudio el entrenamiento de una red neuronal en contraste con el enfoque presentado.

Agradecimientos

A los desarrolladores de la herramienta informática XFuzzy 3.0 (http://www.imse.cnm.es/Xfuzzy) por la maniobrabilidad que nos permite en la definición del sistema y la cantidad del tiempo ahorrado.

A los revisores anónimos del Comité Científico de ESTYL'04 por sus consejos y correcciones.

Referencias

[1] H. Berenji. *Fuzzy Logic Controllers*. Incluido en "*An Introduction to Fuzzy Logic Applications in Intelligent Systems*", YAGER, R. y ZADEH, L. (eds.), Kluwer Academic, Boston, Pág. 69-96, 1993.

[2] C. Caño Alegre, E. López González. Sistema Inteligente de Gestión Deportiva. *Proc. Emergent Solutions for the Information and Knowledge Economy, Vol 2*. Pág. 355-368, 2003.

[3] E. Cárdenas, J. Castillo, O. Cordón, F. Herrera y A. Peregrín. *Influence of Fuzzy Implication Functions and Defuzzification Methods in Fuzzy Control*, Busefal, Vol. 57, Pág. 69-79, 1994.

[4] M. Emami, J. Türksen y A. Goldenberg. *Development of A Systematic Methodology of Fuzzy Logic Modelling*, IEEE Transactions on Fuzzy Systems, Vol. 6, Pág. 346-361, 1998.

[5] M. Gupta y J. Qi. *Theory of T-norms and Fuzzy Inference Methods, Fuzzy Sets and Systems*, Vol. 40, Pág. 431-450, 1991.

[6] M. Gupta y J. Qi. *Design of Fuzzy Logic Controllers Based on Generalized T-operators, Fuzzy Sets and Systems*, Vol. 40, Pág. 473-489, 1991.

[7] H. Hellendoorn y C. Thomas. *Defuzzification in Fuzzy Controllers, Journal of Intelligent and Fuzzy Systems*, Vol. 1, págs. 109-123, 1993.

[8] C. Lee. *Fuzzy Logic in Control Systems: Fuzzy Logic Controller* - Parts I and II, IEEE Trans. on Systems, Man and Cybernetics, Vol. 20, nº 2, Pág. 404-418 y 419-435, 1991.

[9] E. López González. *A Methodology for Building Fuzzy Expert Systems (FES) with Spreadsheet to Quality Function Deployment (QFD) of the Target Costing*. Incluido en Gil Aluja, J. (ed): "*Handbook of Management under Uncertainty*". Kluwer Academic Publishers. Dordrecht, Pág. 457-535, 2001.

[10] E. Mamdani. *Applications to fuzzy algorithms for simple dynamic plant*, Proceedings IEEE, Vol. 21, nº 12, Pág. 1585-1588, 1974.

[11] E. Mamdani y S. Assilian. *An experiment in linguistic synthesis with a fuzzy logic controller, International Journal of Man-Machine Studies*, Vol. 7, Pág. 1-13, 1975.

[12] A. Molina Ariño. *Iniciación a la medicina deportiva*, Pág. 29-34, 87-122, 1991.

[13] J.A. Mora Mérida, J. García Rodríguez, S. Toro Bueno y J.A. Zarco Resa. *Psicología aplicada a la actividad físico-deportiva*, Pág. 163-174, 2000.

[14] F. J. Moreno Velo, I. Baturone, S. Sánchez-Solano, A. Barriga. XFUZZY 3.0: A Development Environment for Fuzzy Systems. *Proc. International Conference in Fuzzy Logic and Technology*, Pág. 93-96, 2001.

[15] W. Pedrycz. *Fuzzy Modeling: Paradigms and Practice*, Kluwer Academic, Boston, 1996.

[16] M. Sugeno y T. Yasukawa. *A Fuzzy Logia-Based Approach to Linguistic Modeling* IEEE Trans. on Fuzzy Systems, Vol. 1, Pág. 7-31, 1993.

[17] T. Takagi y M. Sugeno. *Fuzzy Identification of Systems and Its Applications to Modeling and Control*, IEEE Trans. on Systems, Man, and Cybernetics, Vol. 15, nº 1, Pág. 116-132, 1985.

[18] L. A. Zadeh. Fuzzy Logic = Computing with words, IEEE Trans. on Fuzzy Systems, Vol. 4, Pág. 103-11, 1996.

[19] H. Zimmerman. *Fuzzy Sets Theory and Its Applications*, Kluwer Academic, 1996.

RELACIONES CONCEPTUALES PARA LA CONSTRUCCIÓN DE ONTOLOGÍAS CON GUMSE

Jesús Serrano-Guerrero, José A. Olivas y Javier de la Mata
Universidad de Castilla-La Mancha.
Paseo de la universidad 4, 13071-Ciudad Real, España.
Email: Jesus.Serrano3@alu.uclm.es
Joseangel.olivas@uclm.es
JavierL.Mata@alu.uclm.es

Resumen

El estudio aquí presentado trata sobre el descubrimiento de relaciones borrosas entre términos para la construcción automática de ontologías. Se proponen dos tipos de relaciones para dicha construcción: "relaciones físicas" y "relaciones semánticas", y un nuevo factor de influencia entre términos: "la distancia física". La distancia física entre dos términos determina el grado de relación entre ellos, así pues, dos términos que coexisten en la misma frase tienen más probabilidad de estar altamente relacionados que dos términos situados en distintos párrafos.

La construcción de estas ontologías tiene dos fases principales: "el procesamiento de textos" y "el cálculo de los coeficientes de relación entre términos".

Esta aproximación está siendo incorporada en el meta-buscador GUMSe[*] desarrollado en el marco del grupo de Investigación SMILe[†] como medio de ayuda para el refinamiento de la fase "*query-expansion*" del mecanismo de búsqueda en Internet.

1 INTRODUCCIÓN

En la actualidad Internet alberga una gran cantidad de información desordenada tanto física como conceptualmente, lo que dificulta la labor de los buscadores, provocando una masiva recuperación de información irrelevante para los usuarios. El principal problema reside en la búsqueda de información basada únicamente en aspectos léxicos, es decir, la aparición o no de los términos de una pregunta del usuario en el conjunto de documentos indexados por el motor de búsqueda utilizado. Los aspectos semánticos son difícilmente abordables, y para intentar cubrir este vacío surgen distintas técnicas entre las que destaca el uso las ontologías.

A día de hoy existen muchas ontologías referentes a distintos dominios que mejoran algunos aspectos de ciertas aplicaciones, pero todas ellas han sido creadas de forma manual siguiendo distintas metodologías como las de Gruninger [2] o Fernández [1]. Por el contrario la construcción automática de ontologías constituye un foco de investigación actual donde los logros obtenidos no han sido demasiado satisfactorios.

El sistema desarrollado por el agente constructor de ontologías de GUMSe constituye un cambio sobre la filosofía de construcción de ontologías basada únicamente en la co-ocurrencia de términos y la falta de una definición de "relación conceptual" consistente. Se propone una separación entre dos tipos de relaciones entre términos, las puramente semánticas y las físicas, las cuales influyen de distinta manera en el modelado de la ontología.

En las siguientes secciones se describen algunos de los principales aspectos más relevantes de la generación de ontologías con GUMSe. En la sección 2 se presenta la fase de tratamiento de textos para poder extraer los términos que modelarán la ontología buscada. La sección 3 trata el cálculo de los coeficientes físico y semántico necesarios para establecer las relaciones entre términos. El funcionamiento del sistema formal se muestra en la sección 4, la sección 5 habla sobre la plataforma GUMSe donde se está implementando este modelo, mientras que en la sección 6 se presentan las líneas de investigación futuras.

[*] GUMSe: Gum Search.

[†] SMILe: Soft Management of Internet e-Laboratory, ORETO-UCLM, Spain.

2 PROCESAMIENTO DE TEXTOS

Uno de los mayores problemas que presenta la construcción automática de ontologías es la extracción del conocimiento verdaderamente relevante de los textos que se procesan, especialmente si los textos candidatos para su construcción no han sido seleccionados previamente por personas expertas en el dominio sobre el que versa la ontología, sino que la información ha sido seleccionada directamente por otro proceso automático como son los buscadores.

La solución utilizada para la reconstrucción de textos, que facilita tanto el almacenamiento de información relevante como su posterior procesamiento consta de las siguientes fases:

A. Simplificación de textos

La mayoría de los textos utilizan un gran número de recursos lingüísticos como frases de relativo, formas pasivas, metáforas, etc., que aumentan el tamaño en líneas de texto pero que semánticamente no suelen aportar nada especialmente relevante. Sin embargo, todas las frases por muchos recursos lingüísticos que usen no pierden su estructura básica salvo las formas impersonales:

SUJETO INFINITIVO PREDICADO (1)

Esta idea es la base del funcionamiento del procesador de texto del *agente constructor de ontologías*.

Los verbos de las frases pueden aparecer en distintas formas verbales pero el aspecto semántico que aquí interesa es independiente de dicha forma verbal.

Jesús vivió en Puertollano (2)

La idea de tiempo no es relevante en esta frase, sólo nos interesa que Jesús independientemente de quien sea Jesús, está relacionado con Puertollano, independientemente de qué ciudad sea Puertollano.

Los conceptos son propios de la mente humana y la única forma de expresarlos es mediante palabras por lo que los conceptos que representan el predicado y el sujeto deben ser palabras, que tratadas adecuadamente mediante algoritmos de Stemming (ej: Porter [3]), una StopList, y WordNet [5] quedan reducidos a una única palabra, salvo excepciones que se comentarán a continuación. Así pues, la estructura de cada frase simplificada es:

TERMINO INFINITVO TERMINO *(3)*

Lo cual se inserta fácilmente en una base de hechos (por ejemplo en formato tipo PROLOG):

Vivir (Jesús, Puertollano) (4)

que permite extraer conocimientos posteriores y añadir más información al texto.

B. Adquisición de conocimiento implícito

No todo el conocimiento procesable es extraído únicamente de los textos, también se pueden usar *thesaurus*, en este caso, el utilizado es WordNet [5] mediante un agente especializado (*WordNet Agent*) de GUMSe.

1) Expansión del contexto

En el apartado anterior se hace alusión a que un concepto es más fácilmente procesable si es únicamente una palabra, pero no siempre es óptimo y en el caso en el que se propone a continuación más aún.

La mayoría de las frases en el lenguaje habitual abusan de términos implícitos y sobreentendidos por el contexto, lo cual es el principal problema para la creación automática de ontologías.

La solución aquí propuesta es la adquisición de dicho conocimiento a partir de la base de hechos anteriormente creada. Así pues, el texto simplificado en la fase anterior pasará a ser completado con el conocimiento implícito, por lo que ya no quedarán conceptos de una única palabra sino de más de una palabra. Este hecho empeora el almacenamiento y tiempo de procesamiento pero es sin duda la principal mejora que aporta este sistema ya que es esencial dicho nuevo conocimiento y su situación en el texto para el cálculo del coeficiente semántico que se verá a continuación.

Resumiendo, una frase de la forma:

“Sus obras eran escuchadas por la aristocracia con gran devoción”

en la fase de simplificación de textos quedaría de la forma:

Obras Conocer Aristocracia *(5)*

lo cual es muy simple pero no aporta un conocimiento real. Sin embargo, si le preguntamos a la base de hechos de quién o de qué se estaba hablando en el último

párrafo, y está nos responde que se hablaba de Beethoven, tendríamos:

Beethoven Obras Conocer Aristocracia (6)

lo cual es mucho más intuitivo y acerca mucho más la relación entre Beethoven y la aristocracia tanto semánticamente como en distancia física (ambas palabras están en el misma frase).

3 COEFICIENTES DE RELACIÓN

La principal mejora que propone el *agente de ontologías* es la separación de los tipos posibles de relaciones entre conceptos en dos:

- Relaciones puramente conceptuales y
- Relaciones puramente físicas

En la frase:

Las personas tienen piernas (7)

sin duda alguna, existe una relación muy fuerte entre los conceptos personas y piernas, ya que la relación es física además de ser un axioma. Por el contrario en la frase:

Las mujeres llevan joyas (8)

relaciona a las mujeres con las joyas, lo cual ni es un axioma ni es una verdad que se cumple en todos los casos y es muy dudoso que aporte un conocimiento de una gran relevancia para ningún contexto, por lo tanto este tipo de relación es calificada de puramente semántico.

Partiendo de la base de los dos tipos de relaciones anteriores y sus diferencias aparentes, se calculan los coeficientes relativos a cada uno de ellas de la siguiente manera.

A. Coeficiente semántico

Widyantoro [4] propone que las relaciones entre palabras están basadas en la cantidad de veces que co-ocurren las palabras que forman la relación en un documento, pero esta teoría presenta un problema que es la gran cantidad de documentos necesarios para poder estimar con cierta credibilidad la relación entre 2 conceptos, ya que sólo se procesan palabras sin contexto alguno.

El *Ontology Agent* de GUMSe trata de solventar esa gran demanda de documentos relevantes ya que mediante el descubrimiento de conocimiento implícito, se añaden al texto aquellas ideas y conceptos que verdaderamente predominan y que son los conceptos que determinan las relaciones más fuertes. Así en unos pocos documentos se tiene esa gran cantidad de palabras que demanda el cálculo propuesto por Widyantoro.

Pero además, tal y como se ha podido ver en el ejemplo 1 las relaciones semánticas entre conceptos pueden ser potenciadas por la distancia física que hay entre las palabras. En un documento no podrá haber la misma relación entre dos palabras que aparecen en la misma frase, como entre dos palabras que estén en párrafos distintos.

Así, dada una colección de páginas P = {a_1, a_2, a_3, …., a_n ,..} recuperadas por el agente *Page Retriever Broker* ante la consulta de un usuario, donde cada página a = {t_1, t_2, t_3, ….. t_n } está formada por un conjunto de términos t_js, el *Ontology Agent* calcula el coeficiente semántico como:

$$S = \sum_{a \in A} \left(\frac{\sum_{occur\,(t_i,t_j)} \frac{1}{\exp(\,frase\ *\ parrafo\,)}}{\max(\,ocurr\,(t_j), ocurr\,(t_i))} \right) \quad (9)$$

Frase y párrafo son medidas de distancia física, donde si los términos t_j, t_i están el párrafo actual su valor es 1 y si está en un párrafo posterior vale 2, y así sucesivamente. Por otro lado, si t_j, t_i están en la misma frase están a distancia 0, y si aparece t_i en una frase y t_j en la siguiente están a distancia 1 y así sucesivamente. Así la distancia física afecta de forma exponencial al coeficiente semántico, es decir, a mayor distancia menor valor.

La función *occur*(t_j, t_i) denota las veces que aparecen conjuntamente t_j, $_y$ t_i, mientras que *occur*(t_j) denota el número de veces que aparece el término t_j.

La división por *max(ocurr(tj,),ocurr(t_i))* es simplemente para normalizar, ya que el término que más veces se repite es el que probablemente más influya en el contexto ya que será sobre lo que versa el texto.

B. Coeficiente físico

La idea de tratar las frases como estructuras de la forma:

TERMINO INFINITIVO TERMINO (10)

permite identificar aquellos verbos que pueden expresar relaciones físicas, como agregaciones ("*have*", "*is part of*") o herencia ("*is a*", "*belong to*"), y como consecuencia aquellos conceptos que están relacionados físicamente.

El valor numérico de este coeficiente se calcula

$$F = \sum_{a \in P} \frac{ocurr\ (t_i, t_j)}{N\ (t_j)} \qquad (11)$$

donde $N(t_j)$ es el número de documentos evaluados que contiene a t_j, donde t_j es el sujeto de las estructuras:

TERMINO INFINITIVO TERMINO *(12)*

Este término $N(t_j)$ simplemente vale para normalizar el valor del coeficiente ya que t_j es el término que más puede influir en el contexto al ser el que tiene más posibilidades de definir sobre lo que se está hablando, al igual que en el caso del coeficiente semántico donde el término que más influye se calcula como

$$max(\ ocurr(t_j),\ ocurr\ (t_i)).$$

La elección del sujeto como término normalizador es evidente ya que si:

El conejo tiene patas *(13)*

El término más genérico es "*conejo"* y es sobre el que más posibilidades hay que el resto de páginas hablen.

Esta fórmula (11) es una simplificación de la anterior (9) ya que aquí la distancia física no existe porque los términos t_j, t_i se encuentran en la misma frase y en el mismo párrafo, así la influencia del la distancia viene medida por:

*Exp(frase * parrafo)* *(14)*

y estando en la misma frase y en el mismo párrafo:

*exp(1 * 0) = 1* *(15)*

por lo que la influencia por la distancia no hace falta ponerla.

Ambas fórmulas están normalizadas por el término más influyente en el contexto, y ambas representan básicamente el sumatorio de las co-ocurrencias de los términos tj, t_i, pero matizados unas veces por la distancia física y otras veces por el tipo de predicados en los que se encuentran dichos términos.

C. Solapamiento entre ambos coeficientes

Supongamos una frase en la que existiera aislada una relación de agregación y nada más relacionado en el documento:

Jesús Tener Bicicleta *(16)*

Donde *tj = Jesús* y t_i *= Bicicleta*.

Según el coeficiente físico

$$F = \sum_{a \in P} \frac{ocurr\ (t_i, t_j)}{N\ (t_j)} \qquad (17)$$

Ambas co-ocurren una vez en un documento entonces

F = 1.

Y según el coeficiente semántico

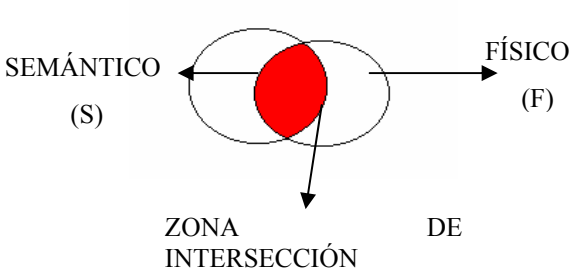
$$S = \sum_{a \in A} \left(\frac{\sum_{occur\ (ti,tj)} \frac{1}{\exp(\ frase\ *\ parrafo\ )}}{\max(\ ocurr\ (tj), ocurr\ (ti))} \right) \qquad (18)$$

Entonces *tj = Jesús* y t_i *= Bicicleta*, están en el mismo párrafo y en la misma frase, sólo aparecen una vez, luego *S = 1*.

Este ejemplo pone de manifiesto que ambas medidas están claramente relacionadas y no pueden trabajar aisladamente si se quiere tener todo el conocimiento de un conjunto de páginas. Pero además denota que la relación medida por el coeficiente físico absorbe a la medida por el coeficiente semántico ya que como es lógico, si un término está relacionado físicamente con otro implica que semánticamente también lo está.

Para procesar todo el conocimiento extraído debemos relacionar ambos coeficientes y la forma elegida por el *Ontology Agent* es mediante la unión borrosa (T-conorma de Luckasiewicz en este caso), ya que es la medida

idónea para eliminar la zona de solapamiento entre ambos coeficientes.



$$Relación = S \varphi F$$

Fig. 1

Donde φ representa la unión borrosa.

$$Relación = S + F - S*F, \qquad (19)$$

donde $S*F$ representa el grado de solapamiento que hay que eliminar de ambas relaciones.

4 PRUEBAS

En el ejemplo (16), donde las relaciones eran puramente físicas S valía 1 y F también valía 1, por lo que

$$Relación = S + F - S*F = 1+1 -1*1 = 1, \qquad (20)$$

es decir se sigue demostrando la fuerte relación entre ambos términos mediante los dos coeficientes.

Si por ejemplo no hay apenas relación física $(F = 0,05)$ y toda la carga sea puramente semántica $(S = 0.8)$, no habría apenas zona de intersección por lo que todo el valor sería el de la carga semántica:

$$Relación = S + F - S*F = 0.8+ 0.05 -0.8*0.05 = 0.81, \qquad (21)$$

que es el valor de carga semántica encontrado.

Si ambos coeficientes son similares $(S = F = 0,4)$, entonces deberían influirse positivamente hasta cierto grado como se demuestra:

$$Relación = S + F - S*F = 0.4+ 0.4 -0.4*0.4 = 0.64 \qquad (22)$$

5 GUMSE

GUMSe es un meta-buscador basado en agentes que, partiendo de los buscadores *google* y *altavista*, pretende abordar uno de los principales problemas de los buscadores actuales como es el factor "*recall*". La precisión de los buscadores actuales es relativamente buena, pero sin embargo hay muchas páginas que no son recuperadas por la falta de capacidades semánticas de los buscadores y que sin embargo son relevantes ante las preguntas del usuario.

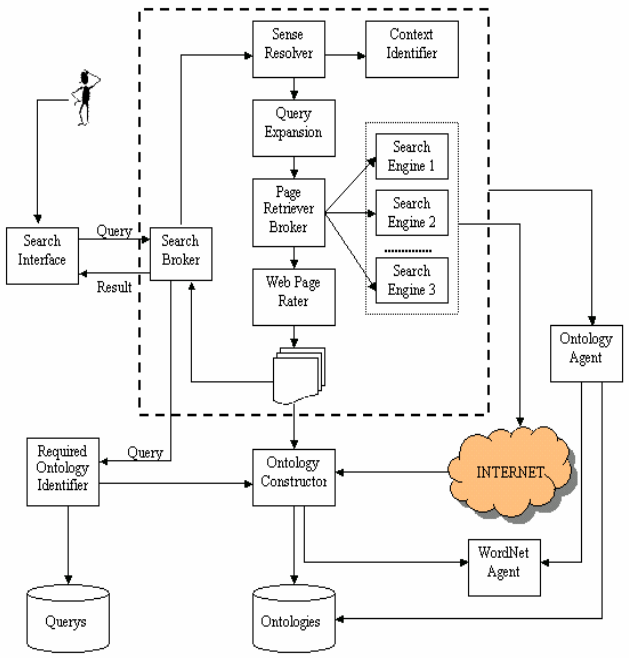


Fig. 2. Arquitectura del buscador GUMSe

GUMSE pretende dotar de estas capacidades semánticas mediante el modelo FIS-CRM, usado ya por el meta-buscador FISS [6], y el *Ontology Agent* aquí propuesto.

La finalidad del *Ontology Agent* es la de mantener un repositorio de ontologías generadas a partir de las páginas recuperadas de los distintos buscadores utilizados (*google* y *altavista*) en función de las consultas de los usuarios para poder descubrir relaciones entre los términos de la consulta y los documentos recuperados, y redefinir así la pregunta del usuario en la fase "*query expansion*", además de ser necesarias las ontologías para la aplicación del modelo FIS-CRM[6].

6 FUTUROS TRABAJOS

Las medidas actuales para el cálculo de los coeficientes tanto semántico como físico están basadas en los factores "frase" y "párrafo", pero también se proponen diversas

alternativas que deben ser estudiadas como la distancia a nivel de frases simples o frases compuestas.

El procesamiento de las frases puede ser guardado como grafos conceptuales tal y como propone Cao [7].

El cálculo de los coeficientes relacionales puede ser refinado mediante la identificación de otras relaciones entre términos como la hiponimia, antonimia, sinonimia [6, 8], etc., o el descubrimiento y tratamiento de axiomas.

La aplicación de distintos algoritmos de poda en función del valor de cada coeficiente trabajando independientemente o conjuntamente puede permitir la construcción de distintos tipos de ontologías.

La aplicación actualmente está desarrollada para lengua inglesa pero también debe ser extensible a otros idiomas distintos donde la forma de escribir puede ser distinta, como el caso del español, donde las frases son mucho más largas y la cantidad de recursos lingüísticos es mayor.

Así mismo son mejorables muchos de los algoritmos utilizados, y se pueden plantear nuevas formas de afrontar el problema.

Agradecimientos

Parcialmente subvencionado por los proyectos PBC-03-004 PREDACOM, JCCM, y TIC2003-08807-C02-02 DIMO-CLUST, MCYT.

Referencias

[1] M. Fernández López, A. Gómez-Pérez, J. Pazos Sierra, A. Pazos Sierra (1999): "Building a Chemical Ontology Using Methontology and the Ontology Design Environment", IEEE Intelligent Systems, v.14 n.1, 37-46.

[2] M. Gruninger,. M.S Fox, (1995), "Methodology for the Design and Evaluation of Ontologies", Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95, Montreal.

[3] M.F Porter (1980), An Algorithm for Suffix Stripping. *Program*, 14(3): 130-137.

[4] D. Widyantoro, J. Yen, (2001): "Incorporating fuzzy ontology of term relations in a search engine", Proceedings of the BISC Int. Workshop on Fuzzy Logic and the Internet, 155–160.

[5] WORDNET. A lexical database for the English language. http://www.cogsci.princeton.edu/~wn/

[6] J. A. Olivas; P. Garcés; F. P. Romero (2003): "An Application of the FIS-CRM Model to the FISS Metasearcher: Using Fuzzy Synonymy and Fuzzy Generality for Representing Concepts in Documents". Int. Jour. of Approx. Reasoning (Soft Computing in Recognition and Search) 34, 201-219.

[7] T. H. Cao (2001): "Fuzzy Conceptual Graphs for the Semantic Web", Proceedings of the BISC, International Workshop on Fuzzy Logic and the Internet, 74–79.

[8] S. Fernández (2001): Una contribución al procesamiento automático de la sinonimia usando Prolog, Tesis Doctoral, Universidad de Santiago de Compostela, España.

SISTEMAS DIFUSOS EVOLUTIVOS

Francisco Herrera
Dpto. de Ciencias de la Computación e Inteligencia Artificial
E.T.S. de Ingeniería Informática
Universidad de Granada
herrera@decsai.ugr.es

Resumen

Los sistemas difusos han demostrado su utilidad para resolver problemas en diferentes ámbitos de aplicación. El uso de los Algoritmos Genéticos en el diseño de sistemas difusos permite introducir la capacidad de aprendizaje y adaptación.

En esta presentación introduciremos los sistemas difusos evolutivos prestando atención a diferentes aspectos de los algoritmos genéticos que merecen ser destacados por su potencial aplicación en el diseño de sistemas difusos evolutivos. Posteriormente nos centraremos en los sistemas difusos evolutivos, en los modelos clásicos y en las nuevas propuestas que actualmente están siendo desarrolladas, junto con algunos problemas abiertos.

1 INTRODUCCIÓN

Los sistemas difusos se han aplicado con éxito a problemas descubrimiento de información, clasificación, modelado, control, etc., y en un considerable número de aplicaciones.

La Tabla 1 presenta el número de artículos disponibles en las revistas de Elsevier desde Enero de 2003 a Enero de 2004, realizando la búsqueda por las palabras clave "fuzzy rules/fuzzy systems", para las revistas clasificadas en un área científica concreta (energía, medicina, …).. Ésta nos muestra diferentes campos de aplicación donde podemos encontrar actualmente aplicaciones.

Un sistema difuso evolutivo es un sistema difuso al que se incorpora la capacidad de aprendizaje y adaptación mediante algoritmos genéticos (AGs) [8]. Los AGs [12,14,20] son algoritmos de búsqueda, basados en la evolución natural y la genética, que proporcionan un mecanismo de búsqueda y optimización robusto en sistemas complejos, y ofrecen una vía útil para problemas de optimización que requieran eficiencia y eficacia.

El análisis de la literatura especializada muestra que los sistemas difusos más estudiados y aplicados son los sistemas basados en reglas difusas (SBRDs) y asociados a éstos están los llamados sistemas basados en reglas difusas evolutivos (SBRDEs), que serán los modelos en los que centremos nuestra atención. La Figura 1 muestra un esquema gráfico de estos modelos.

Tabla 1. Artículos Disponibles desde Enero de 2003 a Enero de 2004 en Revistas de 10 áreas Temáticas de Elsevier

Agricultura y Ciencias Biológicas	39	Ciencias Ambientales	45
Ciencias de los Materiales	29	Inmunología / Microbiología	3
Ingeniería Química	26	Biotecnología	18
Química	16	Medicina	38
Energía	91	Física y Astronomía	44
Tierra y Ciencias Planetarias	32	Ingeniería (fuzzy) fuzzy rules: 93 fuzzy control: 199 fuzzy systems: 373	818

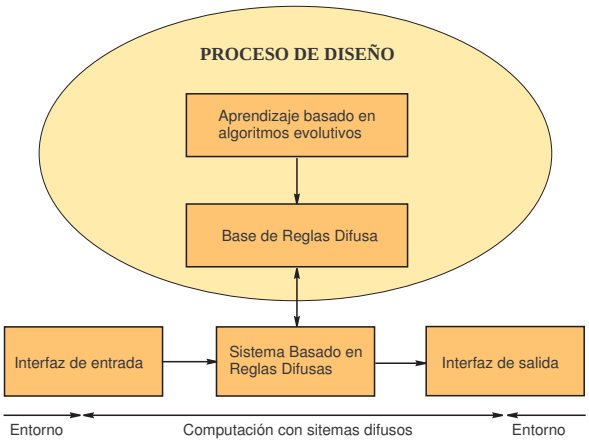


Figura 1. Sistemas Basados en Reglas Difusas Evolutivos

En esta presentación vamos a mostrar el estado del arte del desarrollo de los SBRDEs. En la Sección 2 introduciremos brevemente los AGs, prestando especial

atención a modelos específicos que merecen ser destacados por su potencial aplicación en el desarrollo de los SBRDEs. En la Sección 3 nos centraremos en los modelos clásicos de SBRDEs y en las propuestas que actualmente están siendo desarrolladas. También hablaremos de algunos problemas abiertos en el estudio de los SBRDEs. Unos breves comentarios finales se presentarán en la Sección 4.

2 ALGORITMOS GENÉTICOS

Los AGs [12,14,21] son procedimientos adaptativos para la búsqueda de soluciones en espacios complejos, inspirados en los procesos genéticos de los organismos naturales y en los principios de la evolución natural de poblaciones.

La idea básica de los mismos es mantener una población de cromosomas, los cuales representan soluciones candidatas a un problema concreto, que evolucionan con el tiempo a través de un proceso de competición y variación controlada. Cada cromosoma tiene una bondad o adaptación asociada, que describe la adecuación de la solución a la que representa. El proceso de competición, denominado mecanismo de selección, utiliza estas adaptaciones para determinar los cromosomas que se usan para crear otros nuevos. Los nuevos cromosomas se generan a través de los operadores genéticos denominados cruce y mutación.

Los AGs se han aplicado con éxito en problemas de búsqueda y optimización. Gran parte de este éxito se debe a su capacidad para explotar la información acumulada sobre un espacio de búsqueda, y así dirigir las siguientes búsquedas hacia los mejores subespacios. Esta es su principal ventaja, sobre todo en espacios grandes, complejos y parcialmente definidos donde las técnicas clásicas de búsqueda no son apropiadas. Los AGs ofrecen una aproximación válida a problemas que requieren técnicas de búsqueda eficaces y eficientes.

Bajo la formulación inicial de los AGs, las soluciones se han representado mediante cadenas binarias de longitud fija. Sin embargo, las buenas propiedades de los AGs no radican en el uso de cadenas de ceros y unos. Por esta razón, frecuentemente se han utilizado representaciones no binarias más adecuadas para cada uno de los problemas particulares de aplicación. Una de las representaciones alternativas más importantes es la representación real. Esta representación parece natural cuando se tratan problemas con variables continuas. El uso de nuevas representaciones y la construcción de nuevos operadores para manipularlas ha propiciado que la concepción actual de AG sea bastante diferente y más general que la idea original.

La estructura básica de un AG se muestra en la Figura 2.

Para la descripción básica de un AG hay que tener en cuenta los siguientes componentes:

- la representación de las soluciones
- el mecanismo de selección de soluciones para su preproducción
- los operadores genéticos de cruce y mutación
- los parámetros de los AGs (tamaño de la población, probabilidades de cruce y mutación, …)

Procedimiento Algoritmo Genético

Inicio

t=0

Inicializar P(t)

Evaluar P(t)

Mientras (No se cumpla la condición de parada) **Hacer**

Inicio

t=t+1

Seleccionar P(t) desde P(t-1)

Aplicar Cruce y Mutación sobre P(t)

Evaluar P(t)

Final

Final

Figura 2. Estructura de un AG Generacional

El esquema anterior es el conocido AG Generacional que reemplaza un conjunto de soluciones en cada iteración. Existe otro modelo, el AG estacionario (*steady state*) que en cada iteración solo reemplaza a uno o dos individuos. Este es un modelo que provoca mayor convergencia en el proceso de evolución cuando el elemento/s de la población sustituido/s es el peor de la misma. El algoritmo básico generando único descendiente es el siguiente:

- Seleccionar dos padres de la población.
- Crear un hijo utilizando cruce y mutación.
- Evaluar el hijo.
- Seleccionar un individuo en la población, que puede ser reemplazado por el hijo.
- Decidir si este individuo será reemplazado.

La descripción anterior de AG generacional y estacionario recoge las ideas básicas para la implementación de un AG. Pero desde su propuesta original, se han desarrollado una variedad creciente de estudios y modelos que permiten el uso de AGs en problemas donde se desea

utilizar un modelo eficiente, se requiere la obtención de múltiples soluciones a un mismo problema o la evaluación de múltiples objetivos, etc. Algunos de estos modelos están empezando a ser utilizados en el ámbito de los SBRDEs, y su uso puede mejorar el comportamiento de los mismos. Algunas de estas ideas y modelos se muestran a continuación.

- Modelos que introducen un buen equilibrio entre diversidad y convergencia en la búsqueda en espacios complejos. Existen múltiples estudios y propuestas sobre técnicas de diversidad que están permitiendo avanzar en el diseño de AGs con buen equilibrio entre diversidad/convergencia junto con buena eficiencia en diferentes problemas. Un modelo ya clásico es el AG CHC [11] que está siendo muy utilizado en diferentes ámbitos de aplicación, y que es un referente en este tipo de estudios. En el ámbito en el que nos encontramos, podemos destacar los AG difusos adaptativos que utilizan SBRDs para guiar el proceso de búsqueda adaptando diferentes componentes/parámetros de los AGs; en [16] se puede encontrar una breve revisión de éstos.
- Modelos que permiten obtener múltiples soluciones de un problema en una ejecución. Para ello se han desarrollado los llamados AG con nichos (*niching* GA), que permiten evolucionar diferentes soluciones a lo largo de una ejecución, cada una de ellas perteneciente a un nicho (subregión del espacio de soluciones). Estas soluciones pueden corresponden a diferentes soluciones de igual calidad o a diferentes óptimos locales del problema, como se muestra en la Figura 2. En [22] se puede encontrar un estudio comparativo reciente de diferentes modelos.

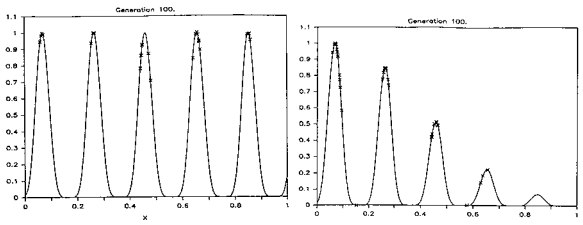


Figura 3. Ejemplos de problemas con múltiples soluciones de igual calidad y múltiples óptimos locales

- Modelos que permiten manejar múltiples objetivos, los algoritmos genéticos multiobjetivo [7,10]. El uso de estos algoritmos conduce a obtener un conjunto de soluciones no dominadas pertenecientes a la frontera del pareto (ver Figura 4). Su uso en el diseño de SBRDs permite introducir objetivos adicionales a la precisión tales como tamaño, medidas de interpretabilidad, etc.
- En términos de eficiencia y de manejo de problemas complejos podemos hablar de los AG paralelos/distribuidos que en los últimos años se han desarrollado ampliamente [1]. Estos algoritmos permiten reducir el tiempo de ejecución cuando se ejecutan sobre máquinas distribuidas, y permiten obtener mejoras de comportamiento por la separación espacial de soluciones que evita convergencia prematura en muchos problemas.

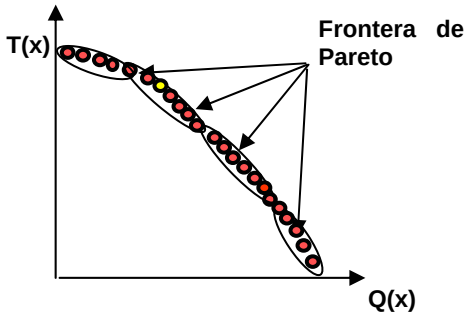


Figura 4. Frontera de pareto para dos funciones objetivo

- Abordar la solución de un problema como la resolución cooperativa de varios subproblemas es una metodología clásica en el diseño de algoritmos. En la actualidad se están desarrollando ampliamente los llamados AGs coevolutivos que permiten evolucionar en paralelo diferentes partes de un problema (soluciones parciales de subproblemas llamadas especies) con una evaluación global de las mismas. Estos algoritmos se están aplicando en problemas que pueden descomponerse conjuntamente con la solución, y pueden ser una buena alternativa para la evolución de diferentes componentes de los SBRDs. La Figura 5 muestra un modelo de evolución de especies (subproblemas).

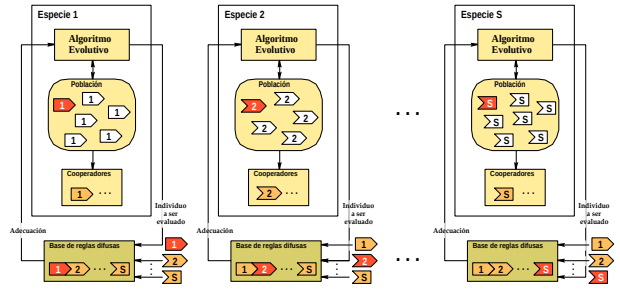


Figura 5. Algoritmo Genético Coevolutivo Cooperativo

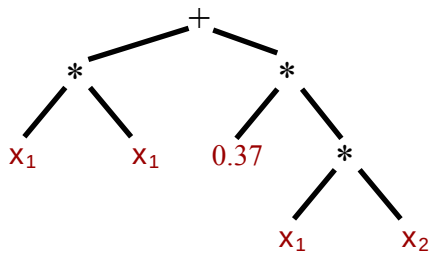
- Podemos definir la Programación Genética (PG) como un caso particular de AGs usados para inducir programas de ordenador de un modo automático, representados mediante un árbol sintáctico de expresiones o como secuencias de instrucciones de máquinas de estados finitos. La representación de programas en forma de árboles de expresión se basa en la existencia de una gramática libre de contexto que define las sentencias válidas del lenguaje.

Naturalmente, los operadores genéticos han de adaptarse a la representación empleada. Las Figuras 6 y 7 muestran un ejemplo con una gramática, un árbol sintáctico, el cromosoma y su expresión numérica asociada.

$$S \rightarrow x_1 \mid x_2 \mid N \mid + \; S \; S \mid * \; S \; S$$

$$N \rightarrow \Re$$

Figura 6. Gramática con dos variables y 2 operadores



Cromosoma: + * x_1 x_1 * 0.37 * x_1 x_2

Expresión: $x_1^2 + 0.37 * x_1 * x_2$

Figura 7. Árbol sintáctico, cromosoma y expresión

3 SISTEMAS DIFUSOS EVOLUTIVOS

Existe un gran número de trabajos dedicados a la generación automática de Bases de Conocimiento (BC) de un SBRDs con AGs. En [8] se puede encontrar un extenso estado del arte, y en [9] se describen aproximaciones recientes.

El aspecto central para el empleo de un AG para automatizar el diseño de una BC es que el proceso de diseño se pueda analizar como un problema de optimización o búsqueda.

Desde el punto de vista de optimización, el hecho de encontrar una BC apropiada para un problema es equivalente a parametrizar la BC, y así poder encontrar aquellos valores de los parámetros que sean óptimos con respecto al criterio de diseño. Los parámetros de la BC constituyen el espacio de optimización, el cual se transforma en una adecuada representación genética sobre la que opera el proceso de búsqueda.

El primer paso en el diseño de un SBRDE es decidir que parte de la BC será aprendida o ajustada. Básicamente tenemos dos componentes:

- la base de datos (BD) que contiene las funciones de escala de las variables y las funciones de pertenencia asociadas a las etiquetas lingüísticas, y
- una base de reglas (BR) que contiene la colección de reglas difusas.

Por otra parte, hemos de distinguir entre ajuste/adaptación y aprendizaje. El ajuste está asociado con la optimización de un SBRDs que ya existe, mientras que el aprendizaje consiste en un método de diseño automático para extraer reglas.

El proceso de ajuste asume la existencia de una BR predefinida y tiene como objetivo encontrar un conjunto óptimo de parámetros para las funciones de pertenencia y/o las funciones de escala, en definitiva los parámetros de la BD.

El proceso de aprendizaje actúa como un proceso de búsqueda sobre el espacio de posibles reglas (aprendiendo una BR) o sobre el espacio que representa la BC.

A continuación veremos los modelos clásicos de ajuste y aprendizaje. Posteriormente estudiaremos nuevas tendencias de SBRDEs y algunas cuestiones y problemas abiertos.

3.1. MODELOS CLÁSICOS

Los modelos de ajuste siguen un patrón común, mostrándose las diferencias en los diferentes parámetros codificados, en los tipos de función de pertenencia, o en el tipo de sistemas difusos (Mamdani, TSK, …), entre otros aspectos.

Dentro de los modelos de aprendizaje de SBRDs podemos distinguir tres modelos básicos, el aprendizaje de una BR (Figura 8), el aprendizaje de la BC donde conjuntamente se aprenden las funciones de pertenencia y la BR (Figura 9), y el aprendizaje de la BD mediante AG y el uso de un algoritmo base para el aprendizaje de la BR (actuando este último con procedimiento asociado a la función de evaluación del AG (Figura 10).

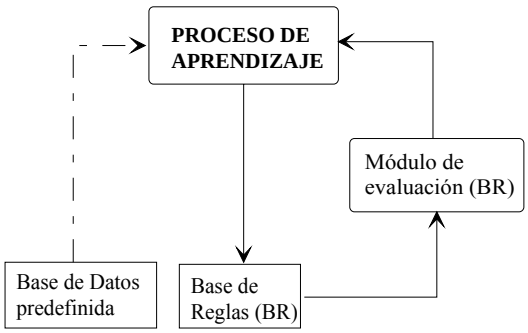


Figura 8. Esquema de aprendizaje evolutivo de la BR

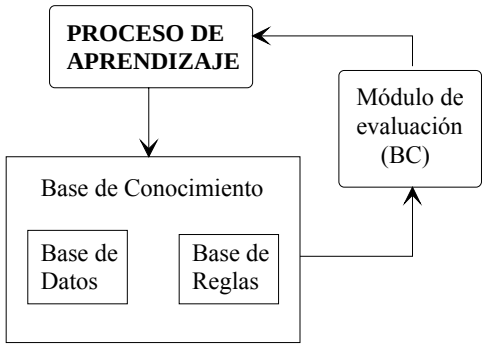


Figura 9. Esquema de aprendizaje evolutivo de la BC

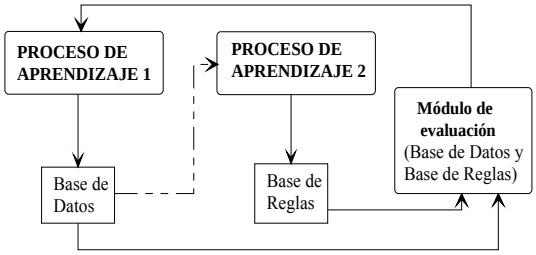


Figura 10. Esquema de aprendizaje evolutivo de la BD

En cuanto a cómo utilizar los AGs para el aprendizaje de la BR/BC, existen tres modelos generales que están fundamentados en el uso de los AGs para aprender SBRDs.

Modelo Pittsburgh. La BR/BC se codifica como un cromosoma. El AG aprende la mejor BR/BC asociada al problema.

Modelo Michigan. Utiliza una BR/BC con mecanismos de refuerzo sobre las reglas (recompensa y penalización de un peso), y sustitución de las reglas con peso bajo por nuevas reglas obtenidas utilizando AGs. Cada cromosoma codifica a una sola regla.

Modelo Iterativo. Es un modelo de aprendizaje evolutivo inductivo, a partir de ejemplos. Al igual que el modelo Michigan, los cromosomas representan reglas, pero al contrario que la anterior, se aprende sólo un individuo en cada ejecución, y se ejecuta el AG secuencialmente mientras no se verifique un criterio de cubrimiento de los ejemplos. No existe cooperación entre las reglas de la población, todos los cromosomas compiten para seleccionar el mejor que formará parte de la población. Posteriormente se aplican procesos de refinamiento para mejorar la cooperación entre las reglas mediante la selección de conjuntos de reglas cooperativas, y mediante el ajuste de los parámetros de las reglas.

El modelo Michigan fué utilizado por uno de los primeros autores que publicaron un modelo de SBRDEs (Valenzuela-Rendón [23]). Sin embargo, no ha sido adaptado con todo su potencial real al ámbito de los SBRDs (los modelos de SBRDEs Michigan existentes son muy simples). Existe modelos recientes de aprendizaje evolutivo Míchigan, tales como XCS [3,19], que son muy útiles para el modelado de sistemas dinámicos, robótica, minería de datos, etc., cuya adaptación a SBRDEs puede ser de gran utilidad en diferentes áreas como las anteriormente citadas.

3.2. NUEVAS TENDENCIAS

Los modelos clásicos que hemos comentado han sido ampliamente desarrollados. En la actualidad se están desarrollando trabajos en nuevas direcciones, explorando nuevos ámbitos de desarrollo de los SBRDs. Entre ellos vamos a destacar:

- Los modelos de selección evolutiva de reglas, encuadrados en el ámbito de la compactificación de SBRDs [18].
- El aprendizaje de la BR/BC utilizando programación genética. Se definen una gramática libre de contexto para la extracción de reglas y funciones y terminales dentro de la gramática que permiten la definición de reglas [2].
- La extracción evolutiva de características combinada con los SBRDs. Esto permite reducir el número de reglas en espacios de alta dimensionalidad, tanto por la reducción de variables dentro de una regla [15] como por la reducción global de variables [4].
- La extracción de SBRDs manteniendo un equilibrio entre interpretabilidad y precisión mediante el uso de AG multiobjetivo [17], donde diferentes objetivos representan los diferentes requerimientos del sistema que suelen ser contradictorios.
- Estudios recientes muestran como el uso de modelos de inferencia paramétricos permiten mejorar la precisión de los SBRDs [21]. Los AGs pueden ser utilizados para aprender los valores de los parámetros del sistema de inferencia o defuzzificación.
- Modelos de SBRDEs que utilizan diferentes estructuras de representación de reglas, incluyendo reglas con pesos, reglas con doble consecuente, reglas jerarquizadas, etc.

Hemos de indicar que estos nuevos modelos, junto con algunas de las diferentes propuestas asociadas a los modelos clásicos, utilizan la ventaja que proporcionan los AGs y por ende los SBRDEs frente a otras técnicas tales como redes neuronales, gradiente descendente, etc. Esta ventaja radica en la posibilidad de presentar en un cromosoma un espacio de soluciones que permite integrar variables de diferente naturaleza, binarias, discretas, reales, etc., con la definición de operadores específicos que permitan evolucionar estas estructuras. Esto permite

combinar mecanismos de selección de reglas con el ajuste de parámetros, aprender la forma de las funciones de pertenencia (gausianas, triangulares,…) representadas con variables discretas junto con la BR y funciones de pertenencia, etc. La Figura 11 muestra un ejemplo que incluye parámetros reales asociadas a cada variable para el ajuste (2 variables, 3 etiquetas), y genes binarios asociados a cada regla (3 reglas) para su selección.

P_1	M_1	G_1	P_2	M_2	G_2	R_1	R_2	R_3
0.2	0.4	0	-0.2	-0.3	-0.5	1	0	1

Figura 11. Esquema de codificación real y binaria

3.3. ALGUNAS CUESTIONES ABIERTAS

Podemos destacar algunas cuestiones abiertas referentes a los SBRDEs en particular y que son extensibles a los SBRDS en general.

El primer aspecto a destacar es el uso de SBRDEs/SBRDs en problemas con alta dimensionalidad, medida ésta tanto por la existencia de un gran número de patrones/ejemplos como por un gran número de variables, y con relaciones de entrada-salida fuertemente no lineales. El primer elemento de la dimensionalidad (gran número de patrones) puede provocar problemas de eficiencia de los algoritmo junto con el aumento del tamaño de la BR. El segundo elemento (gran número de variables) provoca directamente un aumento del número de reglas necesario para modelar el sistema. El aumento lineal de variables provoca un aumento exponencial en el número de reglas.

Este problema ha sido objeto de estudio reciente por diferentes autores utilizando AGs que introducen la selección de variables en reglas localmente o en el conjunto de variables globalmente. Los AGs están mostrando ser muy útiles tanto para la selección de características como para la selección instancias en el preprocesamiento de datos, y pueden ser una buena herramienta para la reducción de la dimensionalidad conjunta con la extracción de reglas fuzzy. Por otra parte, la definición de gramáticas libres de contexto adecuadas puede permitir el diseño de SBRDEs utilizando programación genética, extrayendo reglas que con pocas variables permitan modelar una parte importante del sistema cuando la dimensionalidad es alta.

Por la importancia de esta situación, abordar el uso de los SBRDs en problemas complejos de alta dimensionalidad, está siendo centro de atención de diferentes estudios, y merece ser estudiado diseñando modelos eficientes y eficaces para estos problemas.

Un segundo aspecto a destacar son los estudios recientes en el diseño de SBRDs que alcancen un buen equilibrio entre precisión e interpretabilidad, dos objetivos que suelen ser contradictorios en el modelado de sistemas [5,6].

Como ya se indicó anteriormente, los AG multiobjetivo se han utilizado recientemente para evolucionar SBRDs con ambos objetivos para obtener un conjunto de soluciones no dominadas.

Esta es igualmente una línea de trabajo que merece ser estudiada en profundidad por el creciente interés en la inclusión de varios objetivos, en particular objetivos asociados a la interpretabilidad del SBRD, una de las cualidades a resaltar de los SBRDs.

Por último, hay que hacer mención a un aspecto esencial para la evaluación de los SBRDs y los SBRDEs, y es la necesidad de procedimientos sistemáticos de evaluación de la calidad de éstos, utilizando procedimientos estadísticos de validación de resultados. La necesidad de problemas conocidos y estándar para la validación de resultados junto con bases de datos de resultados sobre ellos es igualmente una necesidad, lo cual permitiría a cualquier autor evaluar los resultados de sus propuestas frente a estudios previamente realizados.

3.4. KEEL 1.0 HERRAMIENTA SOFTWARE DE MINERÍA DE DATOS

Finalmente, me gustaría presentar brevemente el proyecto "KEEL: Entorno para la Extracción de Conocimiento basado en Algoritmos de Aprendizaje Genético y Evolutivo", TIC-2002-04036-C05. En el proyecto KEEL (Knowledge Extraction based on Evolucionar Learning) se está desarrollando la herramienta software KEEL 1.0 (en Java con código abierto y disponible para cualquier usuario). Ésta nos permite utilizar y construir diferentes modelos para Minería de Datos, y como característica importante hay que destacar que incluye una librería de algoritmos de aprendizaje evolutivo y de aprendizaje de SBRDs. En este proyecto, 3 de los 5 grupos de investigación están desarrollado software de aprendizaje de SBRDs (Granada, Jaén, Gijón).

También se está desarrollando una librería de problemas KEEL-dataset y una página WEB asociada que contiene las bases de datos de los problemas para aprendizaje no supervisado, clasificación y regresión, y que incluirá las particiones utilizadas junto con los resultados obtenidos por los diferentes algoritmos implementados en KEEL. La dirección WEB de acceso al proyecto es http://sci2s.ugr.es/KEEL y http://keel.ugr.es .

4 COMENTARIOS FINALES

Hay que resaltar la alta potencialidad de los SBRDEs debido a la alta capacidad de representación de diferentes componentes de los SBRDs. Existe un amplio desarrollo de estos modelos en los últimos años que muestra la madurez del tema, e igualmente existen múltiples

cuestiones y problemas abiertos que aventuran un interesante futuro en la evolución del mismo.

Me gustaría terminar con una frase de David Goldberg [13] sobre la integración diferentes metodologías/componentes en un Sistema Inteligente Híbrido, que no está en una simple combinación. Indica que el futuro de la computación inteligente "*lies in the careful integration of the best constituent technologies*". De acuerdo con Goldberg, el diseño de futuros SBRDEs debe integrar el potencial de abstracción y representación de conocimiento de los sistemas difusos junto con la potencia de los AGs para la búsqueda de buenas soluciones en espacios complejos, lo que nos lleva a un diseño sofisticado que va más allá de una mera puesta en conjunto de ambos.

Agradecimientos

La escritura de esta conferencia plenaria no habría sido posible sin el trabajo desarrollado en SBRDEs con diferentes coautores en estos 10 años dedicados a su estudio. Coautores a los que he tenido la satisfacción de dirigir la tesis doctoral en el desarrollo de SBRDEs (Oscar Cordón, María José del Jesus, Antonio Peregrin, Pedro Villar, Igor Zwir y Jorge Casillas), y otros coautores con los que ha sido un placer trabajar (Manuel Lozano, Rafael Alcalá, José Luís Verdegay, Luciano Sánchez, Luis Magdalena, Frank Hoffmann y Fernando Gomide).

Este trabajo ha sido escrito bajo la financiación de la CICYT con el proyecto TIC2002-04036-C05-01.

Referencias

[1] E. Alba, M. Tomassini. Parallelism and evolutionary algorithms. IEEE Transactions on Evolutionary Computation 6:5 (2002) 443-462.

[2] A. Bastian, Identifying fuzzy models utilizing genetic programming, Fuzzy Sets and Systems 113:3 (2000) 333-350

[3] E. Bernadó, J. M. Garrell. Accuracy-based learning classifier systems: models, analysis and applications to classification tasks. Evolutionary Computation 11:3 (2003) 209-238.

[4] J. Casillas, O. Cordón, M. J. del Jesus, F. Herrera. Genetic feature selection in a fuzzy rule-based classification system learning process for high dimensional problems. Information Science 136 (2001) 169-191.

[5] J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.). Accuracy Improvements in Linguistic Fuzzy Modeling. Springer-Verlag, 2003.

[6] J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.). Interpretability Issues in Fuzzy Modeling. Springer-Verlag, 2003.

[7] C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont. Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Pub., 2002.

[8] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena. GENETIC FUZZY SYSTEMS. Evolutionary tuning and learning of fuzzy knowledge bases. World Scientific, 2001.

[9] Cordón O., Gomide F., Herrera F., Hoffmann F., Magdalena L., Ten years of genetic fuzzy systems: Current framework and new trends. Fuzzy Sets and Systems 41:1 (2004) 5-31.

[10] K. Deb. Multi-Objective using Evolutionary Algorithms. John Wiley & Sons, 2001.

[11] J.J. Eshelman. The CHC adaptive search algorithm: How to have safe serach when engaging in nontraditional genetic recombination. Foundations of genetic Algorithms 1, G.J.E. Rawlin (Ed.), Morgan Kaufman, 1991, 265-283.

[12] D.E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, 1989.

[13] D. E. Goldberg. A meditation on the Computational Intelligence and its future. Technical Report IlliGAL Report #2000019, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 2000. Foreword of the Int. Symp. on Computational Intelligence, 2000.

[14] D.E. Goldberg. The Design of Competent Genetic Algorithms: Steps Toward a Computational Theory of Innovation. Kluwer Academic Pub., 2002.

[15] A. González, R. Pérez. Selection of relevant features in a fuzzy genetic learning algorithm. IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics 31:3 (2001) 417-425.

[16] F. Herrera, M. Lozano. Fuzzy adaptive genetic algorithms: design, taxonomy and future directions. Soft Computing 7:8 (2003) 545-562.

[17] H. Ishibuchi, T. Yamamoto. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining, Fuzzy Sets and Systems 141 (2004) 59-88.

[18] Y. Jin. Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement. IEEE Transactions on Fuzzy Systems 8:2 (2000) 212-221.

[19] P.L. Lanzi, An analysis of generalization in the XCS classifier system. Evolutionary Computation 7 (1999) 125-149.

[20] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, New York, 1992.

[21] C. Moraga, E. Trillas, S. Guadarrama. Multiple-valued logic and artificial intelligence fundamentals of cuzzy control revisited. Artificial Intelligence Review 20 (2003) 169-197.

[22] E. Perez, F. Herrera, C. Hernandez. Finding multiple solutions in job shop scheduling by niching genetic Algorithm. Journal of Intelligent Manufacturing 14:3-4 (2003) 223-239.

[23] M. Valenzuela-Rendón, The fuzzy classifier system: motivationand first results. H.P. Schewefel, R.Männner (Eds.), Proc. First Int. Conf. on Parallel Problem Solving from Nature (PPSN I), Springer, Berlín, 1991, 330-334.

FUZZY TECHNIQUES IN CONTROL AND DECISION MAKING FOR COMPLEX SYSTEMS

Da Ruan
Belgian Nuclear Research Centre (SCK•CEN)
Boeretang 200
2400 Mol, Belgium
druan@sckcen.be

Abstract

In this plenary lecture, an overview on fuzzy techniques in control and decision making for complex systems will be given mainly based on the 10 years progress of the FLINS forum (Fuzzy Logic and Intelligent Technology in Nuclear Science). After an outline of application areas of fuzzy techniques over the last four decades, the FLINS forum will be briefly introduced. Two concrete examples on nuclear reactor control operation and nuclear safeguards information management are demonstrated the potential use of fuzzy techniques in control and decision making for complex systems. Recommendations and future research directions on applications of fuzzy techniques in complex systems are suggested from the practical point of view.

Keywords: Fuzzy Techniques, Soft computing, FLINS, Control, Decision making, Nuclear engineering, Complex systems.

1 FUZZY TECHNIQUES: APPLICATION AREAS

Over the past four decades, fuzzy techniques have been successfully used in many application areas, from control to decision making [15]:

- Industrial control;
- Quality control;
- Elevator control and scheduling;
- Subway control;
- Traffic control;
- Loading crane control;
- Reactor control;
- Automobile transmissions;
- Automobile climate control;
- Automobile body painting control;
- Automobile engine control;
- Paper manufacturing;
- Steel manufacturing;
- Power distribution control;
- Software engineering;
- Expert systems;
- Operation research;
- Decision analysis;
- Financial engineering;
- Assessment of credit-worthiness;
- Fraud detection;
- Mine detection;
- Pattern classification;
- Oil exploration;
- Geology;
- Chemistry;
- Mathematics;
- Medicine;
- Biomedical instrumentation;
- Health-care products;
- Economics;
- Social sciences;
- Information retrieval;
- Internet.

One measure of the interest in fuzzy logic is the number of papers in the literature which contain the word 'fuzzy' in the title. Between 1970 and 2000 there were more than 11,000 such citations of articles in the Citation Index. Recently, Zadeh listed the following figures concerning fuzzy logic [15]:

- Number of fuzzy-logic-related patents applied for in Japan: 17,740;
- Number of fuzzy-logic-related patents issued in Japan: 4,801;
- Number of fuzzy-logic-related patents issued in the US: around 1,700.

Information about products which employ fuzzy-logic-related techniques from SIEMENS:

- Washing machines, 2 million units sold;

- Fuzzy guidance for navigation systems (Opel, Porsche);
- OCS: Occupant Classification System (to determine, if a place in a car is occupied by a person or something else; to control the airbag as well as the intensity of the airbag). Here fuzzy logic is used in the product as well as in the design process (optimization of parameters);
- Fuzzy automobile transmission (Porsche, Peugeot, Hyundai).

Some information about products which employ fuzzy-logic-related techniques from Japan:

- Omron: Fuzzy logic blood pressure meter, 7,4 million units sold, approximate retail value $740 million dollars;
- Sony: total product of Sony's fuzzy-logic-based camcorders 2.4 millions in 1995-1998;
- Fuzzy logic control is running at more than 10 places at 4 oil refineries of Idemitsu Kosan Co. Ltd including not only pure fuzzy logic control but also the combination of fuzzy logic and conventional control. They estimate that the effect of their fuzzy logic control is more than 200 million YEN per year and it saves more than 4,000 hours per year;
- Canon used (uses) fuzzy logic in their cameras, camcorders, copy machine, and stepper alignment equipment for semiconductor production and holds 31 established fuzzy logic patents in Japan and in the US;
- Minolta: one of the most famous fuzzy-logic-based single-lens reflex cameras was alpha-7xi whose name in the US market was Maxxum 7xi. It used six fuzzy logic systems in a camera and 30,000 were produced per month in 1991;
- Fuzzy logic plant controllers of Yamatake Corporation: Yamatake-Honeywell (Yamatake's former name) put FUZZICS, fuzzy software package for plant operation, on the market in 1992. It has been used for more than 10 years at the oil refineries, oil chemical, chemical, pulp, and other industries, where it is hard for conventional PID controllers to describe the plant process. They planned to sell 20-30 FUZZICS per year for a total of 200 million YEN. As this software runs on Yamatake's own control systems, the software package itself is not expensive compared to the hardware control systems.

From the above-mentioned facts, fuzzy techniques have been very successfully applied in the realms of consumer products, intelligent control (non-nuclear systems), and industrial systems.

Now how about fuzzy techniques in nuclear systems or in complex systems? This will be the key issue for this lecture.

2 ADVANTAGES AND DISADVANTAGES

According to a more popular article on fuzzy logic [2], some advantages of using fuzzy logic as a control system design methodology are:

- Fuzzy logic based control has been applied to complex plants where a complete mathematical model may not exist or is defined only poorly;
- Fuzzy logic uses if-then logic instead of differential equations;
- A fuzzy controller requires fewer, less expensive sensors than a conventional controller
- Software and hardware tools are readily available;
- Implementation requires less memory and less computing power than a conventional controller;
- Fuzzy systems have the ability to apply knowledge from experts using normal linguistics.

And some disadvantages are:

- The fundamental theory of fuzzy logic is not as thoroughly understood as more conventional methodologies;
- Retraining staff in a radical new technology when tried and tested procedures are already in place may be an expense that some managers may not wish to incur;
- In safety-critical applications it may be prudent to apply tried and trusted, albeit old-fashioned, solutions;
- As the number of inputs grows their mutual interactions can become very complex, and their combined impact on the output becomes very difficult to estimate;
- Realisation as an electronic circuit using conventional techniques can result in very complex circuits.

In one of his abstracts [15] on "A Critical View of the Foundations of Control and Decision Analysis," Zadeh points out as follows: *"Practitioners of classical control and decision analysis can point with pride to a multitude of brilliant successes, among them the conquest of space and optimal allocation of resources involving linear programs with tens of thousands of variables and constraints. But alongside the brilliant successes stand much less grandiose problems which so far have eluded solution…As a methodology, computing with words*

brings to light some basic flaws in the foundations of control and decision analysis."

3 THE FLINS FORUM AND ITS CONFERENCES

FLINS, an acronym originally for Fuzzy Logic and Intelligent Technologies in Nuclear Science [5], has now been extended to Applied Computational Intelligence [9] to advance the theory and applications of computational intelligence for complex systems in general and for nuclear/power research in particular. The main objectives/missions of FLINS are: (1) conducting research on applied computational intelligent systems for solving intricate problems pertaining to nuclear/power research and related complex systems; (2) bridging the gap between machine intelligence and complex systems via joint research with Belgian, European, and international research institutes and universities; and (3) encouraging interdisciplinary research and bringing multi-discipline researchers together via the international FLINS conferences on applied computational intelligence.

Under this framework, a series of international FLINS conferences have been successfully held in different cities in Belgium from 1994 to 2004.

Name	Date	place
FLINS'94	Sept. 14-16, 94	Mol
FLINS'96	Sept. 25-27, 96	Mol
FLINS'98	Sept. 14-16, 98	Antwerp
FLINS'00	Aug. 28-30, 00	Bruges
FLINS'02	Sept. 16-18, 02	Gent
FLINS'04	Sept. 01-03, 04	Blankenberge

4 NUCLEAR REACTORS OPERATION

The need for on-line reactor operator support systems has become evident after the Three-Mile-Island (TMI) accident in 1979. Since then, considerable attention has been paid by the engineering, scientific, economical and political communities and society at large to prevent this type of events by using State-of-the-Art artificial intelligence techniques [6]. Among the techniques available today, the use of fuzzy logic control (FLC) as a means of expressing linguistic expressions mathematically, has been recently applied to the nuclear reactor control.

Fuzzy logic was invented by Zadeh in 1965 [13]. Fuzzy logic control emerged in the 1970s following the work of Mamdani for the first fuzzy rule-based control system implemented on a laboratory-scale steam engine [3]. The paradigmatic solution developed by Mamdani is predicated on a single-layer rule-based architecture and acts as a direct replacement for conventional control algorithms, based on proportional (P) or proportional plus derivative or integral (PD) or (PI), formalisms, commonly used in industrial control applications under the broad category of (PID) control.

Perhaps the most impressive fact about the present success of fuzzy logic is the breadth of application of this paradigm, ranging from consumer products to industrial process control and automotive engineering. In spite of obvious differences in scope and/or manner of implementation, fuzzy logic plays a similarly central role in creating a suitable rule-based, linguistic, control strategy. Moreover, fuzzy logic bridges the gap between symbolic processing and numeric computation, thereby expanding the domain of application of control engineering to areas that have hitherto fallen outside this scope. And specifically, fuzzy logic forms the basis for implementation of control strategies in a wider sense to include decision making and supervisory control. Because FLC provides a pathway for transforming human abstractions into the numerical domain, it has the potential to assist nuclear reactor operators in the control room. With this transformation, linguistically expressed control principles can be coded into the fuzzy controller rule base. Having acquired the accumulated skill of many operators, FLC can assist an operator in controlling a complex system by using a set of linguistic expressions. One of the advantages of FLC is to derive a conceptual mode of the control operation, without the need to express the process as mathematical equations and to assist the human operator in interpreting and validating incoming plant variables and arriving at a proper control action.

4.1. THE MIT RESEARCH REACTOR IN THE US

One of the best-known research projects perhaps in this area is the successful application of FLC to the 5-MWt Massachusetts Institute of Technology (MIT) research reactor around the mid of 1980's [1]. A rule-based, digital, closed-loop controller that incorporates fuzzy logic has been designed and implemented for the control of power on the MIT research reactor. The advantage of rule-based systems is that they are generally more robust than their analytic counterparts in the above work. Therefore, the rule-based and analytic technologies should be used to complement each other, with rule-based systems being employed both as backups to analytic controllers and as a means of improving the man-machine interface by providing human operators with the rationale for automatic control action.

From the information obtained by the SCK•CEN-MIT liaison officer around the mid of 1990's, the fuzzy logic activities on the MIT research reactor was already completely stopped at the late 1980's. Some additional papers concerning the MIT-FLC project by Dr. Bernard

were obtained. The decision for ending this project was not due to any technical aspects, but with some human resources related issue. The experience for this FLC at the MIT has not been transferred to other research reactors yet. However, the reference of [1] has been frequently cited for other similar research work. The FLC project at the MIT was carried out in a relative short period but with a whole term's contribution from operators, engineers, and research scientists.

4.2. THE FUGEN HEAVY WATER REACTOR IN JAPAN

An FLC system was also developed for a feedwater control system of the 165 Mwe FUGEN Advanced Thermal Reactor (ATR) in Japan. A simulation study for this project was carried out from 1987 to 1988 (nearly at the same time as the MIT research reactor's FLC project) and the development of a prototype system that had an on-line support function from 1989 to 1991. The practical FLC system was introduced to the actual feedwater control system of FUGEN in July 1992. The operation of FLC system of FUGEN has successfully fulfilled its' function during the validation tests.

At a fuzzy logic conference some years ago in Germany, the author met Dr. Iijima and learned that the FUGEN reactor would reach its life end for the future operation. The FLC was a small part of the whole system's implementation, but has enabled operators at FUGEN to more effectively control the steam drum water level while compared to a conventional PI control system in this particular circumstance [4].

4.3. THE BR1 RESEARCH REACTOR IN BELGIUM

The application of FLC in the domain of nuclear industry presents a tremendous challenge. The main reason for this is the public awareness of the risks of nuclear reactors and the very strict safety regulations in force for nuclear power plants. The very same regulations prevent a researcher from quickly introducing novel control methods into this field. On the other hand, the application of FLC has, despite the ominous sound of the word ``fuzzy'' to nuclear engineers, a number of very desirable advantages over classical control, e.g., its robustness and the capability to include human experience into the controller. In an R&D project on FLC to the BR1 (The Belgian Reactor 1) for controlling the power level of a nuclear reactor, the study is intended to assess the applicability of FLC in this domain. The final goal is to develop an optimised and intrinsically safe controller. During the project, an FLC was proposed and first tested by comparing it with the classical controller of BR1. In the next step the BR1 at the SCK•CEN was used as a test bed to implement a PLC-based hardware controller. The BR1 reactor is internationally regarded as a nuclear calibration reference. It therefore provides an excellent environment for this type of experiments, because over the years considerable knowledge of the static and dynamic properties of the reactor has been accumulated.

The project (1995-1999) aimed at investigating the added value and technical limits of FLC for nuclear reactor operations. The progress made in these experiments including closed-loop experiments has been published [8]. By July 1998, a permission to carry on the FLC closed-loop test at BR1 was obtained. The very first closed-loop experiment was successfully carried out in September 1998. It is noted that that experiment and later other closed-loop experiments were only on the steady-state operation. The project as a feasibility study on FLC to nuclear reactors fulfils its function. Some additional remarks on the technical part of the project are as follows: These experiments demonstrate that FLC is able to approximate the classical controller of BR1 in the steady-state operation. Some problems however still exist. One of the main problems is the coarse resolution of the input signals. The hardware should be extended to create a finer resolution. At this moment, we think it is best to implement additional analogue hardware that calculates the signal DP (difference between the current power level and the designed power level) from the signals P (current power level) and Pd (designed power level) before the inputs are digitised by FLC.

Although the experiments showed good results in approximating the classical controller, they are still not yet complete. To really prove the correct behaviour of FLC in the steady-state operation, it is absolutely necessary to do closed-loop testing. It is therefore important to get the necessary licenses from the safety authority, which will allow us in the future to perform more closed-loop control experiments at BR1. The flexibility of FLC is demonstrated in the last fuzzy control closed-loop experiment. It is shown finally that FLC can easily be extended to include new inputs and rules, and thus make it applicable in new control situations. The final controller is expected to be extremely robust against variations of external parameters, such as variation of the reactor core temperature (important during the start-up and shut-down of the reactor), cooling rate, and ageing of the fuel elements. This, however, needs further R&D efforts.

4.4. THE TRIGA MARK III RESARCH REACTOR IN MEXICO

It is almost at the similar time period as the BR1 project at SCK•CEN in Belgium, Dr. Benitez at ININ in Mexico has been carrying out another research project on controlling neutron power of a TRIGA Mark III research reactor with fuzzy logic control with a slightly different approach.

Over the past years, the study for this project has mainly devoted to the comparative study of different fuzzy control algorithms for a nuclear reactor with some extended simulation results. The project has been partially supported by the National Council for Science and Technology of Mexico (CONACYT). ININ has recently upgraded its TRIGA Mark III Reactor's control room into a digital system, which makes it much easier for further fuzzy control and other intelligent control algorithms/tools to be implemented as a test bed. The new research plan for this project is foreseen under the support of CONACYT.

In 2003, ININ contacted SCK•CEN for a possible joint effort for nuclear applications of FLC and related intelligent control techniques. The global cooperation on this project is currently on the way, but technical aspects for such cooperation have been already started since 2003.

5 NUCLEAR SAFEGUARDS INFORMATION MANAGEMENT

Assurance of non-diversion of nuclear materials is the ultimate goal of safeguards. Many countries, concluding comprehensive safeguards agreements (INFCIRC/153 type) with the International Atomic Energy Agency (IAEA), are currently discussing the new additional protocol (INFCIRC/540). The protocol will make them to provide more information to the IAEA. Compared to the traditional regime, additional measures are taken into consideration in this strengthened safeguards regime. The new measures will essentially consist in having access to more information that can be very qualitative in nature.

The information collected by the IAEA comes from mainly three different sources: information provided by the State, information collected by the IAEA, and information obtained from open sources (like media, studies, provided by third parties, etc.). It is obvious that the amount of data is enormous and that not all information contributes to a better knowledge of the situation in a particular State. This information can be of very different nature: it can be relevant, or it can be uncertain, like fuzzy or vague (due to the imprecise boundary), incomplete (due to lack of information), abundant (due to the limited ability of human beings to perceive and process simultaneously large amounts of data), conflicting (due to different sources), fragmentary (the information usually related to a fragment of the problem, and different fragments can be covered by various information sources), not fully reliable (due to different sources for different purposes), and deficient. This information does not directly contribute to a better knowledge of the facts, nor does it facilitate decision-making.

The final conclusion on the non-proliferation commitment and on the absence of any undeclared activities of a State has to result from the balancing of all information in an integrated way. The IAEA has to process and evaluate a large amount of data, which is available from the different sources as described above.

Hence, there is an important need to establish a mathematical framework that provides a basis for synthesis across multidimensional information of varying quality, and provides an evaluation method that enables the IAEA to derive a final estimation on the possibility degree that “No nuclear material in a certain country is used for manufacture of nuclear weapons.”

Traditional study of such issues is conducted using probabilistic tools and techniques. However, it is not difficult to see that aspects related to imprecision or vagueness clearly have a non-probabilistic character since they are related to imprecision of meanings.

Usually, in a quantitative setting the information is expressed by means of numerical values. However, when we work in a qualitative setting, that is, with vague or imprecise knowledge, this cannot be estimated with an exact numerical value. Then, a more realistic approach may be to use linguistic assessments instead of numerical values, that is, to suppose that the variables that participate in the problem are assessed by means of linguistic terms.

Fuzzy logic as a modern computational tool provides a systematic way to interpret linguistic variables rather than numerical variables as well as a rule-based expert system in a natural decision-making procedure in nuclear safeguards applications. It can be viewed as complementary to traditional methods and can be a powerful tool to deal with imprecise information, especially linguistic information. Therefore, an important issue in the development of automated decision aids for safeguard is to handle fuzziness, since the evaluation involves human expertise and knowledge. This would enable the system to better emulate human evaluation processes.

A flexible and realistic linguistic assessment approach is developed to provide a mathematical tool for synthesis and evaluation analysis of nuclear safeguards indicator information. This symbolic approach, which acts by the direct computation on linguistic terms, is established based on fuzzy set theory. More specifically, a lattice-valued linguistic algebra model, which is based on a logical algebraic structure of the lattice implication algebra, is applied to represent imprecise information and to deal with both comparable and incomparable linguistic terms (i.e., non-ordered linguistic values). Within this framework, some weighted aggregation functions

introduced by Yager [11,12] are analyzed and extended to treat these kinds of lattice-value linguistic information. The application of these linguistic aggregation operators for managing nuclear safeguards indicator information is successfully demonstrated [7].

6 RECOMMENDATIONS

At the 2001 Halden workshop on Power Plant Surveillance and Diagnostics [6], one dark cloud hung over the conference and its participants. In a presentation the previous week at the Halden Project Summer School in Halden, Electricité de France (EDF) had initiated a moratorium on the installation of new plant operational support systems in their control rooms. The explanation for such drastic action was that the proliferation of such systems was introducing undue complexity in the instrumentation and processes systems of the plants and that the operations of such systems were diverting the operators from their other important duties. While this prohibition does not extend to engineering and maintenance systems, it is a “wake-up” call to those of us working in this field. The challenge to all the Workshop participants is to develop simpler, less intrusive and more useful and integrated operational support systems for the plant operators. They simply do not want multiple, independent plant operational support systems, each with its own special features and each requiring specialized training to operate. Equally important is for the plant operational support system developers to secure the “buy-in” and support of the plant operators of the systems that are installed, or they will not be used.

The application of fuzzy techniques and/or related soft computing techniques to nuclear systems/complex systems is to exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low solution cost [14]. At the same time, the nuclear power industry puts special demands on plant safety, surpassing all other industrials in its safety culture. The regulatory environments in which nuclear power plants operate reflect theses needs, and also the demands of the public for high levels of assurance about safety and regulatory compliance. This culture is not one which encourages innovation in control systems and philosophy, yet nowhere is there greater potential benefits from high reliability systems, automated fault recognition and rationally supported decision making. A demonstration of the use of intelligent control in an actual plant is a vital needed step in prototyping the next generation of nuclear power plants [10]. Issues related to validation of the complex program, reliability in a real application domain, understanding without "black-box," efforts in R&D, and license aspects are the hard part of fuzzy techniques/soft computing in nuclear/complex applications. The truly implementation of fuzzy techniques related technologies in a real plant remains a great challenge.

References

[1] J.A. Bernard, Use of a rule-based system for process control, *IEEE Control System. Mag*. 8 (5), 3-13, 1988.

[2] B. Cogan, Certainty and uncertainty in science, *Scientific Computing World*, Nov./Dec. 2001, pp. 28-30.

[3] E.H. Mamdani and S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, *International Journal of Machine Studies* 7 (1), 1-13, 1975.

[4] S. Terunuma, K. Kishiwada, H. Takahashi, T. Iijima, and H. Hayashi, Application of fuzzy algorithms for feedwater Fugen HWR, Proc. Int. Conf. *Man-Machine Interface in the Nuclear Industry*, Tokyo, Japan, Feb. 15-19, 1988, p. 463, IAEA, 1988.

[5] D. Ruan, P. D'hondt, P. Govaerts, and E.E. Kerre, Eds, *Fuzzy Logic and Intelligent Technologies in Nuclear Science*, the Proceedings of FLINS'94, World Scientific, Singapore, 1994

[6] D. Ruan and P. Fantoni (Eds.), *Power Plant Surveillance and Diagnostics*, Springer, 2002.

[7] D. Ruan, J. Liu, and R. Carchon, Linguistic assessment approach for managing nuclear safeguards indicator information, *Int. J. of Logistics Information Management* 16 (6), 401-419, 2003

[8] D. Ruan, Initial experiments on fuzzy control for nuclear reactor operations at the Belgian Reactor 1, *Nuclear Technology* 143, 223-240, 2003.

[9] D. Ruan, P. D'hondt, M. De Cock, M. Nachegael, E.E. Kerre, Eds., *Applied Computational Intelligence*, the Proceedings of the 6th Int. FLINS conference, Blankenberge, Belgium, Sept. 1-3, 2004.

[10] J.G. Williams and W.C. Jouse, Intelligent control in safety systems: criteria for acceptance in the nuclear power industry, *IEEE Trans. on Nuclear Science* 40 (6), 2040-2044, 1993.

[11] R.R. Yager, An approach to ordinal decision making, *Int. Journal of Approximate Reasoning* 12 , 237-261, 1993.

[12] R.R. Yager, Fusion of ordinal information using weighted median aggregation, *Int. Journal of Approximate Reasoning* 18, 35-52, 1998.

[13] L.A. Zadeh, Fuzzy sets, *Information and Control* 8 (3), 338-353, 1965.

[14] L.A. Zadeh, What is soft computing? *Soft Computing* 1, p1, 1997.

[15] L.A. Zadeh, email messages to BISC-Group, Dec. 17, 2003; Feb. 6, 2004; Feb. 13, 2004 etc.

LÓGICA BORROSA Y NARRATIVA: UN PÁRRAFO DE VILA-MATAS

Enric Trillas
Departamento de Inteligencia Artificial.
Universidad Politécnica de Madrid.
Campus de Montegancedo. Boadilla del Monte, 28660 Madrid
etrillas@fi.upm.es

Resumen

Con el estudio de un caso en términos elementales, se intenta sugerir que es posible llegar al establecimiento de una *ciencia experimental de la narrativa*. Que cabe empezar a pensar, con metodología proveniente de la lógica borrosa, en las relaciones entre conjeturar, narrar y computar.

1. INTRODUCCIÓN

En 1956 el programa *Logic Theorist* del luego premio Nobel de Economía Herbert Simon, y de Allen Newell, efectuó la prueba de un teorema del famoso libro *Principia Mathematica* de Bertrand Russell y Alfred N. Whitehead, que se considera más elegante que la del libro. Cabe decir que la narración de las razones del resultado fue hecha mejor por el programa informático (que corrió en el hoy prehistórico ordenador *Johnniac*, en Santa Mónica, California) que por Russell y Whitehead. Aunque la intención de Simon y Newell no residía en simular narraciones sino en la resolución de problemas concretos por ordenador, no hay duda de que para narrar también hay que plantear y resolver problemas; unos problemas que, siendo de naturaleza distinta a la de una *prueba* matemática, sin duda comportan conjeturas no siempre triviales ni claramente distintas a las de una *búsqueda* matemática. *Conjeturar, narrar y computar* son tres verbos que forman parte importante de lo que, continuamente, permite el progreso de la especie humana.

En *Ventanas de Manhattan*, Antonio Muñoz Molina afirma que "la literatura, el oficio, el gusto de leerla, también es en el fondo, una cosa algo rancia y bastante artesanal". En el caso de *Principia Mathematica*, si la ranciedad puede no ser aplicable, lo es el carácter artesanal al trabajo de Russel y Whitehead, y también al de Johnniac que, no obstante, sólo tardó unos minutos (¡en 1956!) en efectuar algo mejor la prueba citada. Además, el paso del tiempo también va añadiendo cierta ranciedad a lo que hacen los ordenadores. En principio, a lo escrito por los ordenadores también se les puede aplicar, por lo menos en parte, la frase de Muñoz Molina.

Vaya por delante, me interesa lo que escribe Enrique Vila-Matas y me gusta como lo hace. Entre los actuales es uno de mis escritores preferidos; no es, o no me lo parece, un escritor fácil, me hace pensar y, leyéndole, siempre aprendo algo. Me induce a reflexionar. En su libro, "Desde la ciudad nerviosa" (Alfaguara, Madrid, 2004), hay una frase condicional, aparentemente no contrasimétrica, que de tal manera me ha venido dando vueltas por la cabeza, que he acabado por tomar papel y lápiz para intentar entenderla. La frase condicional, que aparece en la página 208 del libro y dentro de la sección titulada "Un tapiz que se dispara en muchas direcciones", es la primera parte en bastardilla (añadida) del párrafo siguiente:

> "Siempre me había dicho que *si la vida no tiene sentido tampoco lo tiene leer*, pero de repente me pareció que la operación de leer y buscar artistas del No tenía mucho sentido. De pronto, sentí que la busca de *bartlebys* daba sentido a mi vida".

Mi primer escollo reside en la palabra sentido; el segundo, en qué quiere decir Vila-Matas con la frase condicional. El tercero, consecuencia de los dos anteriores, es la validez del razonamiento que exhibe el párrafo anterior. He aquí lo que he pensado.

Creo poder conjeturar que Vila-Matas usa la palabra sentido como imprecisa ó graduable; de lo contrario es difícil captar qué quiere decir con "tenía *mucho* sentido" (la bastardilla es mía). Quiero decir que a los enunciados del tipo "x tiene sentido" no los califica sólo con dos grados, como son *verdadero* o *falso*, sino que acepta grados intermedios como es, por lo menos, *muy verdadero*. *Sentido* aparece como el nombre de algo que exhibe magnitud; de ahí que pueda haber mucho. Por ello, he de aceptar también que el sentido de la vida deberá ser graduable; que el enunciado *la vida tiene sentido* vale en algún grado, al igual que *leer tiene sentido* y *si la vida no tiene sentido tampoco lo tiene leer.* En lo que sigue, todos los grados se suponen números entre cero y uno.

Ahora bien, para que a las frases 'x tiene sentido' se les pueda atribuir algún grado (en particular, verdadero o falso) es indispensable que, tanto x como *sentido*, puedan entenderse suficientemente; puedan describirse. En todo caso, por lo que se refiere a las palabras *vida* y *leer*, aunque no parezca que quepa colocarlas fácilmente en lugar de aquella x, supondré que ello puede hacerse en base a la autoridad en materia literaria de Vila-Matas.

Si *leer* es una acción que no me resultaría difícil describir y, por tanto, atribuir algún grado a *leer tiene sentido*, *vida* ni siquiera me parece una acción y, sea lo que sea, no está a mi alcance describir ese término en forma tal que el enunciado *la vida tiene sentido* sea inmediatamente graduable entre 0 y 1. Por ejemplo, el problema biológico de la muerte celular es uno que, según lo que sé, trae de cabeza a los investigadores. Tendré que limitarme, por tanto, a una descripción tan simple como que algo tiene vida (vive) si reacciona a estímulos del entorno; con lo que claro está, un termómetro vive. Pero conjeturo que, en el contexto de lo que escribe Vila-Matas, esa idea de tener vida, por excesiva que sea, no es muy descabellada. Por tanto, para avanzar, supondré que sé lo que es leer (incluso más allá del acto físico de hacerlo) y lo que es tener vida.

De la afirmación "leer y buscar artistas del No tenía mucho sentido" cabe seguir que el grado de *leer y buscar artistas del No tiene mucho sentido* es alto. Como ese grado es menor o igual que el de *leer tiene mucho sentido* y éste, a su vez, menor o igual que el de *leer tiene sentido*, sigue que el último grado es alto. Algo que en un contexto de lectores es razonable, como también lo es suponer que tal grado pueda ser directa e inmediatamente atribuible. En consonancia, tomaré grado(leer tiene sentido) como un parámetro numérico no inferior a grado(leer no tiene sentido). En cuanto al grado de *la vida tiene sentido*, lo consideraré como una variable.

Pero, antes de continuar, aún debo volver a la palabra *sentido*. Yendo a los diccionarios, entre la multitud de acepciones de ese término sólo encuentro, concordantes con lo que, a mi parecer, pueda querer decir Vila-Matas las que siguen: *Razón de ser, finalidad; significado; manera como hay que entender algo,* y *racionalidad*. De ellas las que parecen más claramente graduables son *finalidad* y *manera como hay que entender algo* y, como sea que la "búsqueda de *bartlebys* daba sentido a mi vida" puede entenderse como que esa búsqueda era una finalidad, creo que, por lo que respecta a *sentido* cabe identificarlo a *finalidad.* En cuanto a *manera como hay que entender algo*, es precisamente mi problema con la frase condicional; es decir, lo que Vila-Matas quiera decir con una frase condicional no necesariamente contrasimétrica del tipo "Si no y, entonces no x", es lo que no me permite dejar de pensar en la frase y el párrafo citados.

2. MODELOS DE LA FRASE CONDICIONAL

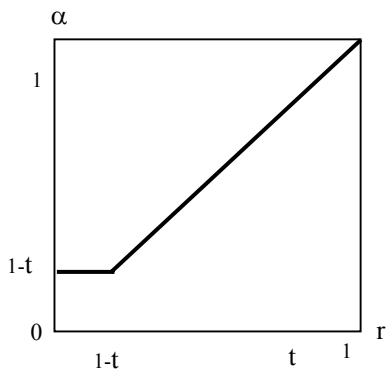
1. En la lógica formal clásica, las frases condicionales "Si x, entonces y" se entienden como equivalentes a "(no x) ó y", con lo que aceptando que "no (no x)" lo es a "x" (que no es siempre el caso) como se hace en aquella lógica, la frase "Si no y, entonces no x" sería equivalente a "no (no y) ó (no x)". Supuesto que "y ó (no x)" equivale a "(no x) ó y", resulta equivalente a "(no x) ó y"; en conclusión, equivalente a "Si x, entonces y".

Es decir, que Vila-Matas identificaría la frase "Si la vida no tiene sentido tampoco lo tiene leer", con su contrasimétrica "Si leer tiene sentido también lo tiene la vida". Con ello la frase valdría en un grado entre 0 y 1, dependiente de los grados en que pudiesen valer *leer tiene sentido* y *la vida tiene sentido* y si, por ejemplo, suponemos que la frase condicional es muy verdadera (0,9), entonces cabría convenir, como es usual, en que: 0,9 = max (grado (no (leer tiene sentido), grado (la vida tiene sentido)). De asignar (como lector apasionado) un grado alto a "leer tiene sentido", es decir, también aceptándola como muy verdadera (0,9), entonces de: grado (no (leer tiene sentido)) = 1 - grado (leer tiene sentido) = 1 - 0,9 = 0,1, resultaría, 0,9 = max (0,1, grado (la vida tiene sentido)), que llevaría a que fuese, grado (la vida tiene sentido) = 0,9; a que es muy verdadero que la vida tiene sentido. En el caso de que la frase condicional fuese poco verdadera (0,1), resultaría, con la misma hipótesis de lector apasionado, 0,1 = max (0,1, grado (la vida tiene sentido)), que llevaría a grado (la vida tiene sentido) $\leq$ 0,1; a que *la vida tiene sentido* valiese con un grado muy pequeño, a otorgarle poco valor.

En general, designando por t = grado (leer tiene sentido), por r = grado (la vida tiene sentido), y por α = grado (Si la vida no tiene sentido tampoco lo tiene leer)-Kleene-Dienes-, es: α = max (r, 1 - t). Por lo tanto:

- Si r $\leq$ 1 - t, es α = 1 - t
- Si 1 - t < r, es α = r.

Gráficamente,



que refleja la variación de α en función de r, para un valor t del parámetro.

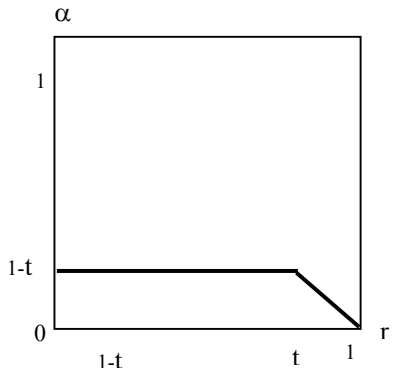
2. ¿Qué sucedería si Vila-Matas identificase, como es frecuente en el razonamiento ordinario y en el control borroso, las frases condicionales 'Si x, entonces y', con 'x e y'? En tal caso, "Si la vida no tiene sentido tampoco lo tiene leer" equivaldría a "la vida no tiene sentido y no tiene sentido leer", con lo que, supuesta la conjunción copulativa conmutativa, también valdría "Si leer no tiene sentido, tampoco lo tiene la vida". Es decir, que la frase de Vila-Matas sería, en realidad: "No tiene sentido leer" *si y sólo si* "No tiene sentido la vida". Para Vila-Matas las frases "No tiene sentido leer" y "No tiene sentido la vida" serían intercambiables. Lo que, en un lector apasionado, podría provenir de la exageración de identificar la finalidad de leer con la de vivir.

Si, en esas hipótesis, *convenimos* como es usual en que, grado (Si x, entonces y) = (min (grado (x), grado (y)) -Mamdani-, sería: grado (Si leer no tiene sentido tampoco lo tiene la vida) = min (grado (leer no tiene sentido), grado (la vida no tiene sentido)) = min (1 - gr (leer tiene sentido), 1 - grado (la vida tiene sentido)) = 1 - max (gr (leer), gr. (vida)), abreviando la escritura. Por lo tanto, con las mismas notaciones del párrafo anterior, es: $\alpha = 1-\max(t, r)$,

que lleva a:

- Si $t \leq r$, es $\alpha = 1 - r$
- Si $r < t$, es $\alpha = 1 - t$.

Gráficamente,



que, para el valor t, refleja la variación de α en función de r y muestra que es siempre $\alpha \leq 1-t$. Además, si es $r = 1$ es $\alpha = 0$.

3. Los modelos anteriores de 'Si x, entonces y' no son, desde luego, los únicos para identificar las frases condicionales con frases mezclando la conjunción, la disyunción y la negación. Un modelo más general, típico de algunas lógicas de la física cuántica, es: "Si x, entonces y" $\equiv$ (no x) ó (x e y), que lleva a:

"Si no x, entonces no y" equivale a (x ó (no x y no y)).

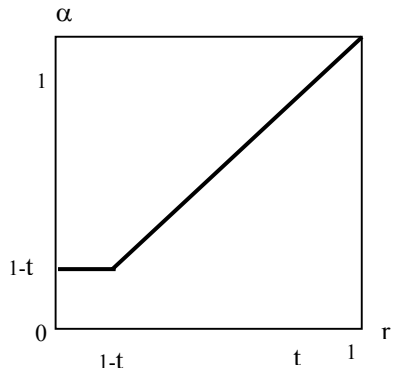
Es decir: "Si la vida no tiene sentido tampoco lo tiene leer", equivaldría en ese modelo a

" 'La vida tiene sentido' ó 'la vida no tiene sentido y leer no tiene sentido' ".

Entonces, conviniendo en que es α = max (gr (vida), min (gr (vida no), gr (leer no))) = max (r, min (1- r, 1- t)) = max (r , 1 - max (r, t)) -Early Zadeh-, resulta:

- Si $r \leq t$, es $\alpha = \max(r, 1 - t)$
- Si $t < r$, es α = max (r, 1 - r).

Gráficamente,



que muestra el mismo resultado del primer modelo.

4. Otro modelo, también usado en la lógica de la física cuántica, es:

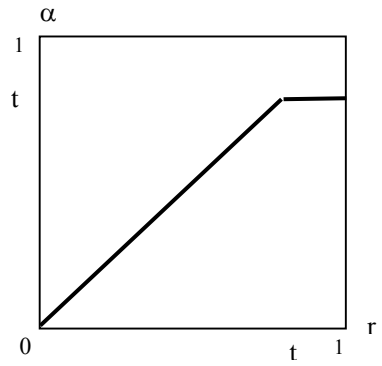
"Si x, entonces y" = "y ó (no x y no y)",

con el que la frase de Vila-Matas equivaldría a la frase incondicional "(no tiene sentido leer) ó (tiene sentido la vida y tiene sentido leer)".

En tal caso, sería $\alpha = \max(1 - t, \min(r, t))$ -Dishkant-, con lo que:

- Si $r \leq t$, es $\alpha = \max(1 - t, r)$
- Si $t < r$, es $\alpha = \max(1 - t, t) = t$.

Gráficamente,



que muestra que siempre es $\alpha \leq t$.

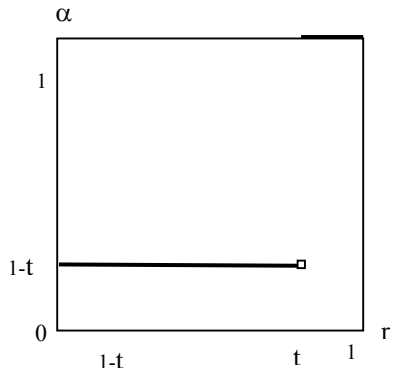
5. Hay otros modelos para calcular α que no provienen directamente de interpretar las frases condicionales "Si x, entonces y" por medio de enunciados incondicionales obtenidos usando y, o, no. El más simple de ellos es -Gödel-

$$\alpha = \begin{cases} 1, \text{ si grado (x)} \leq \text{grado (y)} \\ \text{grado (y), en otro caso,} \end{cases}$$

que, aplicado a "Si la vida no tiene sentido tampoco lo tiene leer", da:

$$\alpha = \begin{cases} 1, \text{ si } 1 - r \leq 1 - t, \text{ ó } t \leq r \\ 1 - t, \text{ si } r < t- \end{cases}$$

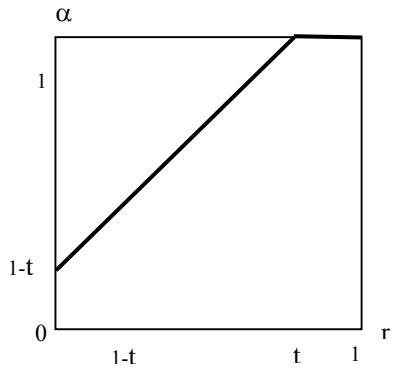
Gráficamente,



6. Otro modelo del tipo anterior viene dado por α = min (1, 1-grado (x) + grado (y)) -Lukasiewicz- que, aplicado a la frase de Vila-Matas, da: α = min (1, 1 - (1 - r) + (1 - t)) = min (1, 1 - t + r), con el que:

- Si $1 \leq 1 - t + r$ ($t \leq r$), es $\alpha = 1$
- Si $1 - t + r < 1$ ($r < t$), es $\alpha = 1 - t + r$

Gráficamente



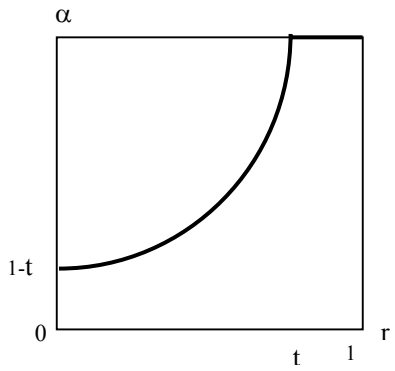
7. Finalmente, con el modelo de Menger-Goguen:

$$\alpha = \begin{cases} 1, \text{ si grado (x)} \leq \text{grado (y)} \\ \text{grado (y)/grado (x), en otro caso,} \end{cases}$$

se obtiene:

$$\alpha = \begin{cases} 1, \text{ si } t \leq r \\ 1-t/1-r, \text{ si } r < t, \end{cases}$$

con gráfica



3. DISCUSIÓN DE LOS MODELOS

De los modelos **1** a **7**, ¿Cuál o cuáles son los más adecuados para representar el grado α en que puede valer la frase de Vila-Matas?

En primer lugar, hay que suponer que si Vila-Matas no escribe "Si x, entonces y" sino "Si no y, entonces no x", es por cuanto distingue o, por lo menos, quiere distinguir en el texto citado, una frase de otra; por ello hay que rechazar los modelos **1** y **6**. Ya se vio que el **1** identifica las dos frases directamente. En cuanto al **6**, al ser grado (Si no y, entonces no x) = min (1, 1-grado (no y) + grado (no x)) = min (1, 1 - (1-grado (y)) + (1-grado (x))) = min (1, 1-grado (x) + grado (y)) = grado (Si x, entonces y), no establece diferencia alguna entre el grado de ambas frases; las representa por la misma función α.

En segundo lugar, supuesto que Vila-Matas no niegue la máxima validez ($\alpha = 1$) a su frase, también hay que rechazar los modelos **2** y **4**, en los que nunca es $\alpha = 1$. En cuanto al modelo **3**, lo rechazaré por llevar a la misma representación α que el **1**; además, aunque distingue entre las dos frases condicionales, se hace difícil aceptar la

extrema precisión de que sólo pueda ser $\alpha = 1$ cuando sea r = 1. Por lo que respecta al modelo **5**, parece excesivamente limitado que α sólo pueda tomar dos valores, los 1-t y t, con una discontinuidad en r = t; que, por ejemplo, fijado t = 0,9, fuese $\alpha = 0{,}1$ si r = 0,899, y $\alpha = 1$ si r = 0,9.

En tercer lugar, el modelo **7** no admite que la frase condicional pueda ser totalmente falsa ($\alpha = 0$) lo que, en los otros modelos solo se da en el **4** con r = 0, es decir cuando grado(la vida no tiene sentido) = 1. Si eso puede ser una objeción, parece poco importante: ¿Por qué Vila-Matas iba a decirse a sí mismo algo que, eventualmente, pudiese ser totalmente falso? Por consiguiente y final pero provisionalmente, me inclino a conjeturar que Vila-Matas sigue el modelo **7,** de Menger-Goguen, cuyas no contrasimetría y no linealidad parecen más acordes con la complejidad de sus escritos. Con ello, además, Vila-Matas no reduciría su frase condicional a una incondicional resultante de mezclar *la vida no tiene sentido* y *leer no tiene sentido* por medio de las conectivas y, o, no, sino que la consideraría una frase irreducible y distinta de su contrasimétrica.

4. UN MODELO DEL RAZONAMIENTO DE VILA-MATAS

Detrás del párrafo de Vila-Matas citado en la introduccion**,** se esconde un razonamiento que muestra que después de creer que "si la vida no tiene sentido tampoco lo tiene leer", la búsqueda de escritores del No (una lectura, en realidad) dio sentido a su vida; que, en fin, le llevó a concluir la frase contrasimétrica "si leer tiene sentido también lo tiene la vida". Ello evidencia la distinción ya citada de las dos frases condicionales y permite modelar aquel razonamiento con la frase triplemente condicional: ' Si "si la vida no tiene sentido tampoco lo tiene leer", entonces "si leer tiene sentido también lo tiene la vida" '. ¿Hasta dónde vale ese modelo del razonamiento?

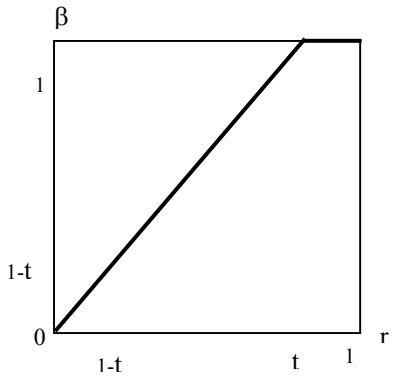
De aceptar el modelo **7** para las tres frases condicionales, con β = grado(si leer tiene sentido también lo tiene la vida) y con γ el grado de la anterior frase triplemente condicional, es:

$$\gamma = \begin{cases} 1, \text{ si } \alpha \le \beta \\ \beta/\alpha \text{ , si } \alpha > \beta. \end{cases}$$

Como sea que es:

$$\beta = \begin{cases} 1, \text{ si } t \le r \\ r/t, \text{ si } r < t, \end{cases}$$

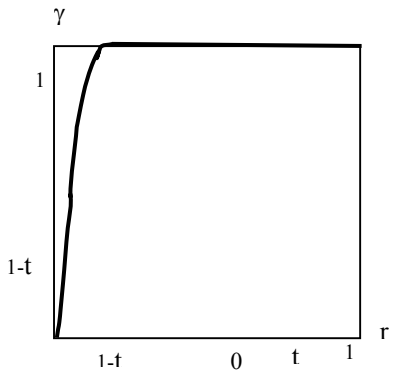
se obtiene la gráfica



Basta comparar esta gráfica con la de α en el modelo **7**, para ver que $\alpha \le \beta$ equivale a $1 - t \le r$. Por lo tanto se obtiene:

$$\gamma = \begin{cases} 1, \text{ si } 1 - t \le r \\ r(1 - r)/t(1 - t), \text{ si } 1 - t > r, \end{cases}$$

cuya gráfica



muestra que el razonamiento de Vila-Matas sería totalmente válido para la mayoría de valores de r; que solo sería falso si r = 0 y que, para valores pequeños de r (entre 0 y 1 -t) su grado de validez crecería rápidamente hacia 1. Por ejemplo, con t = 0.9 (1 - t = 0.1), si r = 0.05 es $\gamma = 0.53$, y si r = 0.09 es $\gamma = 0.91$.

5. CONCLUSIÓN

¿Qué concluir, en general? En primer lugar, que la literatura es una fuente considerable de reflexión acerca de las relaciones entre los verbos narrar, conjeturar y computar. Las narraciones literarias se basan en conjeturas del autor sobre las que el lector añade, si lee reflexivamente, las suyas. Y conjeturar es una actividad racional de primera magnitud.

En segundo lugar, que los modelos lógico-borrosos de las frases condicionales, al autorizar el estudio de la variación de sus grados de validez, permiten analizar su significado en función de lo que pueda ser básico para su comprensión. Y la comprensión de los enunciados hipotéticos es, tanto en la ciencia como en la literatura, esencial para la de la narración, sea de lo real, sea de lo imaginario.

En tercer y último lugar, que cuanto se ha hecho ha sido sobre lo dicho en la introducción; el análisis de la frase condicional de Vila-Matas es, naturalmente, posterior a su libro. La literatura escrita por personas es un universo donde buscar fenómenos observables para, a través de su estudio *more geometrico*, llegar a una experimentación literaria mediante ordenadores; al establecimiento, cuando sea posible, de una ciencia primero experimental y luego computacional, de la narrativa. En realidad, las diversas funciones α obtenidas son representaciones gráficas de la frase de Vila-Matas; de forma análoga a como Parsifal dice a Gurnemanz, "Zum Raum wird hier die Zeit" (aquí el tiempo se hace espacio), aquí la narración se hace matemáticas, se re-presenta como un modelo matemático.

Si Leopardi aconsejaba sabiamente "quitarle al lenguaje su peso" con la literatura, para el análisis matemático de la literatura parece conveniente añadirle, a lo escrito, tanto el peso de los modelos lógicos (sintaxis), como el de los grados de validez de los significados (semántica). Si "un libro no existe hasta que alguien lo haya leído" (Josefina Aldecoa, *En la distancia*), ese libro no existirá como material de observación en el sentido anterior si no se analizan sintáctica y semánticamente algunos de sus contenidos. Y ese análisis no tiene sólo utilidad como un paso hacia una ciencia experimental de la narrativa, para que los ordenadores lleguen a escribir narraciones como las personas, sino también para que quienes escriben libros puedan reflexionar sistemáticamente sobre lo que vayan escribiendo.

Finalmente, si Vila-Matas leyese este artículo, ¿Qué modelo diría que siguió? ¿Sigue siempre el mismo modelo de frases condicionales? ¿Cuándo sigue uno u otro?. Imagino que su respuesta, quizás tras hacerse una pregunta como la que J. M. Coetzee formula en su "Elisabeth Costello",

> "¿Qué hombre o mujer joven con sangre en las venas quiere pasarse la vida hurgando en los archivos o llevando a cabo *explications du texte* sin fin?,

podría estar próxima a la de Jorge Luis Borges a Herbert Simon:

> "I am not one who plays with symbols. But if there is some transcendental explanation of one of my stories, it is not for me to discover it, that is the task of the critics and the readers".