SISTEMA NEURO-FUZZY PARA LA DETECCIÓN DE DESVIACIONES EN LA MEDIA Y EN LA VARIABILIDAD DE UN PROCESO DE FABRICACIÓN.

Javier Puente García
David de la Fuente García
Raúl Pino Diez

E.T.S. Ingenieros Industriales de Gijón.
Dpto. Administración de Empresas y Contabilidad. Universidad de Oviedo, España

RESUMEN

En este estudio se presenta una alternativa a las técnicas tradicionales para controlar la calidad de los procesos de fabricación, como los gráficos X-R. Aplicando la teoría de Redes Neuronales Artificiales, conjuntamente con la teoría de Conjuntos Borrosos (Fuzzy), hemos construido un sistema que cumple dos objetivos fundamentales en cuanto al control de calidad de un proceso: por un lado detecta inmediatamente las desviaciones inadmisibles de la media y la variabilidad del proceso y por otro lado, evita las "falsas alarmas" que puede dar el gráfico "X-R" cuando se está fabricando con un nivel de calidad adecuado y las mediciones indican aparentemente lo contrario.

Palabras clave: Control de Calidad, Redes Neuronales, Lógica Fuzzy.

1. INTRODUCCIÓN.

El control estadístico de procesos (SPC) trata de mostrar y supervisar el ajuste de los procesos de fabricación a las normas y tolerancias garantizadas desde el diseño de producto y proceso. De esta manera se trata de evitar que las medidas a controlar sobrepasen los límites de control fijados en torno a la media de las medias o al rango medio (según hablemos del gráfico X o el R) de las muestras que se obtengan del proceso, y en el caso de que se sobrepasen, efectuar las correcciones oportunas en los procesos.

En SPC se consideran variaciones admisibles aquellas que sólo pueden achacarse al azar o al comportamiento normal del proceso productivo: máquinas, proceso y personal. Este comportamiento regular recoge los efectos aleatorios sistemáticos, es decir la parte predecible de los datos observados de acuerdo con el modelo dinámico y de referencia. Representa el estándar interno de calidad y en el diseño del proceso éste deberá ajustarse a las expectativas exteriores de calidad. Cualquier desviación del estándar interno debería considerarse como "controlable, desviación del proceso o desviación común" en la terminología de Deming.

Para la detección de estas variaciones se han utilizado tradicionalmente los gráficos de control "X-R". La principal regla para la detección de una situación de "fuera de control", es que un punto representativo de las mediciones de la característica de calidad a controlar esté fuera de los límites de control situados a $+3\sigma$ y -3σ respecto a la característica media. En otro caso, el proceso está bajo control.

Investigaciones como las de Pugh (1991), Guo y Dooley (1992) o Chang y Aw (1994, 1995) demuestran con éxito la posible aplicación de las redes neuronales a la identificación y clasificación de desviaciones en la media de un proceso.

En nuestro trabajo, proponemos un sistema neuro-fuzzy para detectar desviaciones en la media y en la variabilidad de un proceso de fabricación. A partir de la medición de la característica de calidad de ocho muestras, se alimentan dos redes neuronales previamente entrenadas para proporcionar como salida la media y la desviación típica del proceso. Posteriormente se emplean dos clasificadores que utilizan conceptos de lógica difusa, para identificar el estado del proceso de fabricación en función de la salida de la red neuronal.

2. DISEÑO DE REDES NEURONALES PARA DETECCIÓN DE DESVIACIONES EN LA MEDIA Y VARIABILIDAD DE UN PROCESO.

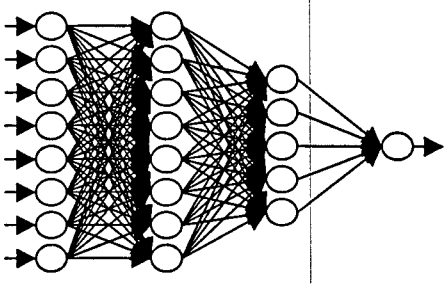
Las redes neuronales (o redes de neuronas artificiales), son modelos matemáticos simplificados de las redes de neuronas que constituyen el cerebro humano. Estos

modelos están compuestos por un conjunto de "neuronas artificiales" o conjunto de unidades que procesan e intercambian información. Las neuronas de una red están estructuradas en distintas capas, de forma que cada neurona de una capa están conectada con todas las de la capa siguiente a través de unos enlaces caracterizados por un peso que da idea de la importancia de la conexión.

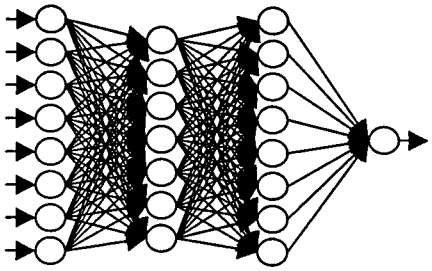
Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo son capaces de aprender de la experiencia, de generalizar a partir de casos anteriormente aprendidos, de abstraer características esenciales a partir de entradas que presentan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que esta tecnología se esté aplicando en múltiples áreas. Estas ventajas incluyen: Aprendizaje adaptativo, Autoorganización, Tolerancia a fallos y operación en tiempo real entre otras.

Aprovechando estas características, se han construido numerosas aplicaciones de redes neuronales que se han demostrado muy útiles en los campos de reconocimiento de patrones y generalización. En este trabajo, utilizamos redes neuronales aplicadas al control de calidad de un proceso, por lo que se entra dentro de las dos categorías: por un lado se reconoce un patrón aprendido a partir de información incompleta o defectuosa, y por otro lado se obtendrán unos resultados utilizando la capacidad de generalización a partir de los patrones previamente aprendidos.

La red neuronal propuesta es, en ambos casos, un PERCEPTRÓN MULTICAPA, constituido por cuatro capas de neuronas, con la particularidad de que las neuronas de la capa de entrada están conectadas a todos los nodos de las otras tres capas (en la figura 1 no se han dibujado todos los enlaces para evitar confusiones). Este diseño tiene la ventaja de que el tiempo de entrenamiento disminuye considerablemente sin que sea demasiado alto en número de neuronas en las capas ocultas.



RED 8-8-5-1 para las MEDIAS



RED 8-7-8-1 para las DESVIACIONES

Figura 1

La capa de entrada está compuesta por ocho neuronas, cada una de ellas para una observación o medida cuya media y/o variabilidad se quiere controlar. Se tienen dos capas ocultas; para el control de las medias, con 8 y 5 neuronas y para el control de las desviaciones con 7 y 8 neuronas respectivamente. El número de capas ocultas y el de nodos en cada capa se ha elegido por ser los que daban un menor tiempo de entrenamiento y mejores resultados. En la capa de salida se tiene una única neurona que será la que saque al exterior el resultado del procesado de la red. La salida de cualquier nodo de las capas ocultas o de salida, se calcula mediante la función no lineal siguiente:

$$V_j = \frac{1}{1 + e^{-\sum_{i=1}^{N_j} W_{ij} \cdot V_i}} \qquad (1)$$

donde:
V_i = salida de una neurona de una capa anterior,
V_j = salida de la neurona,
W_{ij} = peso de la conexión entre una neurona i de una capa anterior, y la neurona j, y
N_j = número de neuronas de capas anteriores a las que está conectada la neurona cuya salida se desea calcular.

El algoritmo de entrenamiento utilizado es el "backpropagation", que se basa en la modificación de los pesos de las conexiones entre las neuronas en función del error cometido en el cálculo de la salida de cada uno de los ejemplos de entrenamiento frente a la salida deseada en cada caso. Después de cada pasada de los ejemplos de entrenamiento, se hace una corrección de los pesos de forma que el error va disminuyendo.

El conjunto de ejemplos de entrenamiento está constituido por 1800 casos tanto para la media como para la desviación, ordenados aleatoriamente, y cada uno de ellos formado por 8 entradas y 1 valor de salida que será la media o la desviación típica de la Distribución Normal con la que se generaron los 8 valores de entrada, como se indica en las tablas adjuntas 1 y 2:

Grupo	N° de ejemplos	Distribución	Salida deseada
A	200	N(-4, 1)	0.1
B	200	N(-3, 1)	0.2
C	200	N(-2, 1)	0.3
D	200	N(-1, 1)	0.4
E	200	N(0,.1)	0.5
F	200	N(1, 1)	0.6
G	200	N(2, 1)	0.7
H	200	N(3, 1)	0.8
I	200	N(4, 1)	0.9

Tabla 1: Casos de entrenamiento para la red de "Medias"

Grupo	N° de ejemplos	Distribución	Salida deseada
A	360	N(0,0.5)	0.1
B	360	N(0,1.0)	0.3
C	360	N(0,1.5)	0.5
D	360	N(0,2.0)	0.7
E	360	N(0,4.0)	0.9

Tabla 2: Casos de entrenamiento para la red de "Desviaciones"

Antes de presentar los casos a la red para el entrenamiento, se deben normalizar entre 0.1 y 0.9 tanto las entradas como la salida para evitar la saturación de la sigmoide que se utiliza como función de transferencia.

Como procedimiento para el entrenamiento de la red se han dividido los casos de ejemplo anteriores en dos grupos: el 80% de ellos se utilizará para el entrenamiento, mientras el 20% restante se usará para la validación.

Para evitar el "overfitting" o sobreentrenamiento se ha utilizado la metodología "early-stopping" que se basa en realizar una pasada por los ejemplos del conjunto de validación después de cierto número de pasadas de los casos del conjunto de entrenamiento, de forma que se pueda hacer un seguimiento tanto del error de entrenamiento como del de validación; cuando se aprecie un aumento apreciable del error de validación se detendrá el proceso de entrenamiento de la red neuronal, puesto que se puede asegurar que si continuara, se produciría el mencionado "overfitting" con lo que la capacidad de generalización y predicción de la red neuronal se vería claramente perjudicada. En la Figura 2 puede observarse genéricamente el instante de parada del entrenamiento.

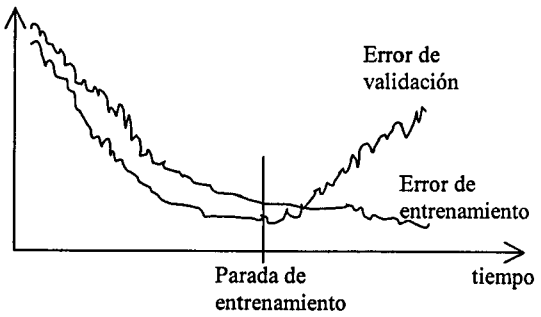


Figura 2

Para la construcción y entrenamiento de la red neuronal se utilizó el programa comercial "SNNS v 4.0", que permite el diseño de redes neuronales de muy diferentes tipos y configuraciones. Después de pasar por la red todos los ejemplos de entrenamiento en orden aleatorio 150.000 veces para la media, el error cuadrático medio en la salida bajó hasta 0,0016. Para la red de desviaciones se pasaron los ejemplos de entrenamiento en 200.000 ciclos, bajando el error hasta 0.0088. Con estos valores las redes se consideraron entrenadas y listas para poder clasificar nuevos casos.

3. CLASIFICADORES FUZZY DEL ESTADO DEL PROCESO.

Según la lógica tradicional, cuando un elemento determinado pertenece a un subconjunto de un referencial, debe hacerlo en un grado absoluto. De la misma manera, si el elemento no pertenece al subconjunto, no lo hará en modo alguno.

La lógica borrosa permite la introducción de matices en cuanto a la pertenencia de un elemento a un subconjunto difuso, de forma que un elemento pueda pertenecer a aquel sólo en cierto grado. Así, un subconjunto difuso "F" de un determinado referencial "U" estará caracterizado por una función de pertenencia μ_F que toma valores en el intervalo [0,1]:

$$\mu_F: U \rightarrow [0,1] \quad (2)$$

Como se observa en la figura 3, a medida que los valores de x crecen por encima de x_i, el grado de pertenencia al subconjunto F borroso disminuye gradualmente; o lo que es lo mismo, a medida que disminuye el nivel de seguridad (α) los intervalos de confianza para los valores de "x" aumentan (se hacen más imprecisos).

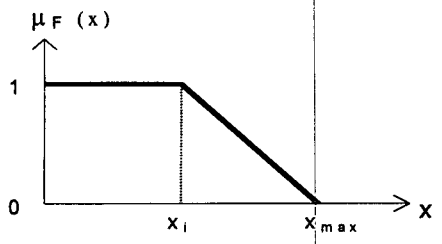


Figura 3

Nosotros proponemos los siguientes esquemas de subconjuntos fuzzy, para clasificar las salidas de la redes neuronales (ver figura 4 y figura 5):

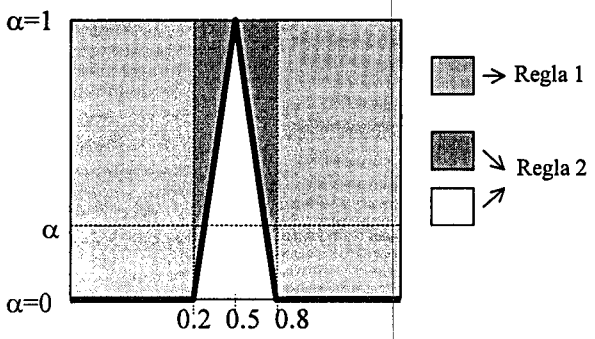


Figura 4: Clasificador fuzzy para las medias

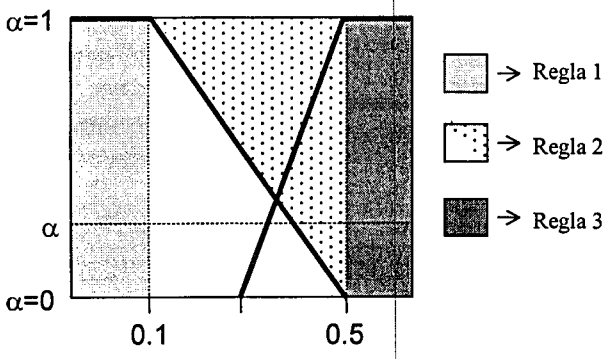


Figura 5: Clasificador fuzzy para las desviaciones

Como puede apreciarse en la figura 4, el valor de salida 0.5 se correspondería con una N(0,1), por lo tanto los valores que se han introducido a la red en su entrenamiento deben haber sido tipificados. El valor 0.2 correspondería a un valor de una N(-3, 1) y el 0.8 a una N(3, 1). En este clasificador, según la regla 1 para un nivel de seguridad α, valores de salida de la red por debajo de 0.2 o superiores a 0.8 representarán muestras fuera de control según la media. Según la regla 2, para valores de salida de la red entre 0.2 y 0.8, aquellos que caigan dentro del triángulo serán considerados bajo control mientras que los valores exteriores al triángulo entre esos límites, harán que tengamos que evaluar una nueva muestra antes de tomar la decisión. Si el valor obtenido para la siguiente muestra cae dentro del triángulo decidiremos que la situación está bajo control, y si vuelve a caer fuera del triángulo decidiremos que la situación estará definitivamente fuera de control.

El clasificador fuzzy para las desviaciones proporcionadas por la red de desviaciones se interpreta de manera análoga al clasificador de las medias. Así, para salidas de la red de desviaciones por debajo de 0.1, correspondiente a desviaciones de una N(0, 0.5), consideramos que la desviación estará bajo control para cualquier nivel de seguridad α (Regla 1). Para valores superiores a 0.5, correspondiente a desviaciones de una N(0, 1.5) consideraremos que la desviación es inadmisible (Regla 3). Por último para valores entre 0.1 y 0.5 (Regla 2), cuando estemos en la zona no punteada, aceptaremos la desviación de la muestra, y en caso contrario tendremos que esperar a tomar la decisión según el valor de la siguiente muestra. Si el valor obtenido para la siguiente muestra vuelve a caer en la zona punteada, o en la zona de la Regla 3, consideraremos una situación de fuera de control para la desviación; en otro caso, consideraremos el sistema bajo control.

El proceso de análisis conjunto será de tipo secuencial. Se comienza observando el resultado obtenido para la media y a continuación se pasa a observar la desviación de cada muestra. Para determinar que el proceso está bajo control debe verificarse que tanto la media como la desviación están bajo control.

4. ANÁLISIS DE LOS RESULTADOS OBTENIDOS.

Para comparar el sistema clasificatorio neuro-fuzzy propuesto con el gráfico X-R-Shewart de medias y rangos, utilizaremos el concepto de "longitud media hasta fallo" (LHF) definiéndolo como número medio de muestras seguidas que es necesario analizar hasta encontrar una "inaceptable", esto es, con media o desviación tal que ponga al sistema fuera de control. Así, la meta de un sistema de control de calidad de procesos es doble, por un lado debe detectar cambios repentinos en la media o variabilidad tan pronto como sea posible (LHF pequeño), y por otro mantener el proceso bajo control cuando no existan problemas (LHF grande). Esto se corresponde con minimizar los errores tipo I y tipo II, siempre considerados en los sistemas estadísticos de control de calidad.

4.1 Errores tipo I y tipo II

Los errores tipo I caracterizan las falsas alarmas del sistema de control; indicarían que el proceso no se encuentra bajo control cuando en realidad sí lo está. Por

tanto al comparar los gráficos X-R con nuestro sistema de control neuro-fuzzy, aquel que propicie mayor valor de LHF para un mismo conjunto de muestras (consideradas aceptables a priori), será el más adecuado para el control al minimizar este error tipo I. Los errores tipo II, por el contrario, identificarían situaciones en las que no se detectan a tiempo cambios en el sistema y que lo llevarían a ponerse fuera de control. Por tanto en la comparación entre ambos métodos, aquel que propicie un menor valor de LHF para un mismo conjunto de muestras (consideradas inaceptables a priori), será el más adecuado para el control al minimizar el error de tipo II.

Por tanto, evaluaremos la capacidad de control de nuestro proceso según gráficos X-R y según el modelo neuro-fuzzy eligiendo como mejor opción para el control aquel que nos proporcione una mejor solución de compromiso en la reducción simultánea de ambos tipos de errores.

Profundizando en lo anterior, para medidas de la característica de calidad a controlar que hayan sido generadas según una distribución aceptable tanto para la media como para la desviación, nos interesará que el valor LHF sea lo más alto posible (lo ideal y estadísticamente imposible sería que nunca se encontrase fallo), ya que al ser valores generados como "aceptables" a priori, el sistema no debería detectar puntos fuera de control. Por el contrario, cuando consideramos medidas generadas según distribuciones no aceptables tanto para la media como para la desviación, el sistema de control adoptado debe detectar cuanto antes estas situaciones que ponen al sistema fuera de control, siendo el óptimo valor de LHF la unidad.

Una vez calculados los valores de LHF para distintos niveles de α según el esquema propuesto, resulta interesante la comparación entre éstos y los obtenidos para el gráfico X-R:

α	0.3	0.5	0.7	0.8	X-3sigma
N(-4, 1)	1,00	1,00	1,00	1,00	1,00
N(-3, 1)	1,03	1,02	1,01	1,00	1,00
N(-2, 1)	1,20	1,17	1,17	1,15	1,08
N(-1, 1)	8,93	6,58	3,64	3,48	4,5
N(-0.5, 1)	122,30	68,41	31,25	24,14	34,72
N(0, 1)	2535,82	1815,83	614,45	385,24	370,46
N(0.5, 1)	116,24	59,54	29.14	21,55	34,72
N(1, 1)	8,62	6,05	3,48	3,24	4,5
N(2, 1)	1,15	1,15	1,12	1,12	1,08
N(3, 1)	1,00	1,00	1,00	1,00	1,00
N(4, 1)	1,00	1,00	1,00	1,00	1,00

Tabla 3. Valores de LHF para distintas distribuciones y valores de α. "Medias"

α	0.3	0.5	0.7	0.8	R-3sigma
N(0, 0.5)	13000,0	3250,3	590,9	200,0	142,8
N(0, 1.0)	6,217	3,475	1,794	1,486	2,323
N(0, 1.5)	1,597	1,324	1,108	1,079	1,083
N(0, 2.0)	1,154	1,089	1,032	1,021	1,018
N(0, 4.0)	1,011	1,002	1,000	1,000	1,000

Tabla 4. Valores de LHF para distintas distribuciones y valores de α. "Desviaciones"

A la vista de los resultados de la tabla 3 puede observarse que para valores de medidas generadas según distribuciones normales centradas (de media 0), el esquema propuesto mejora notablemente al gráfico X, ya que incluso para altos valores de seguridad α, el número medio de muestras seguidas que se necesita obtener para detectar que el proceso está fuera de control es relativamente mayor que con el gráfico X ($385_{\alpha=0.8}$ frente a 370). Cuando las medidas generadas no son tan centradas (medias entre –2 y +2), los resultados que se obtienen mejoran también al gráfico X para valores de α relativamente altos (por ejemplo para α=0.8, con valores de la N(-0.5, 1) el clasificador fuzzy da valores de LHF menores que los proporcionados por el gráfico X: 24 frente a 34, lo que significa que con el clasificador fuzzy detectamos más rápidamente el descentramiento de nuestra medida objetivo). Por último para valores altos de descentramiento el comportamiento es prácticamente igual con el gráfico X que con el clasificador fuzzy.

Por otra parte, a la vista de los resultados de la tabla 4 puede observarse que para valores de medidas generadas según una N(0, 0.5) el esquema propuesto mejora notablemente al gráfico R de desviaciones, ya que incluso para altos valores de seguridad (α), el número medio de muestras seguidas que se necesita obtener para detectar que el proceso está fuera de control es bastante mayor que con el gráfico R ($200_{\alpha=0.8}$, frente a 142). Cuando los valores analizados corresponden a muestras con pequeñas dispersiones, pero mayores que 0.5 (caso de la N(0, 1.0), el clasificador fuzzy propuesto también mejora notablemente el comportamiento respecto al gráfico R. En la tabla puede observarse como para altos niveles de seguridad (α=0.7, α=0.8) los valores LHF mejoran sustancialmente al obtenido según al gráfico R (1.8 y 1.5 frente a 2.32), lo que indica que se detecta más rápidamente el fallo con el clasificador fuzzy que con el gráfico R. Para valores de muestras con dispersiones mayores que 1, el comportamiento del clasificador fuzzy para valores altos de α, responde aproximadamente igual que el gráfico R.

5. CONCLUSIONES.

Se ha propuesto un sistema neuro-fuzzy para la detección de desviaciones en la media y variabilidad de un proceso de fabricación. Dos redes neuronales de tipo "perceptrón multicapa" se ha entrenado off-line para transformar las medidas de la característica de calidad a controlar en valores de la media y la desviación típica del proceso. Después se proponen sendos clasificadores fuzzy para catalogar la salida de la red neuronal e informar sobre el estado del proceso. Los resultados preliminares mejoran a los obtenidos mediante la utilización de gráficos X-R tradicionales en términos de LHF (longitud media hasta el fallo), sobretodo cuando la desviación de la media o de la variabilidad es pequeña. Es especialmente destacable el hecho de que el valor de LHF es mucho mayor para el caso de que el proceso se encuentre bajo control (media o desviación típica alrededor de la óptima), ya que no se producen las falsas alarmas características de los gráficos X-R.

Como ampliación de este trabajo, estamos realizando un programa que permita el procesado de todos los datos de las distribuciones de manera sencilla, con objeto de que éstos puedan ser capturados directamente según se originan las medidas en los procesos reales de control de calidad. Asimismo, investigaremos el efecto de diferentes distribuciones (Gamma...) sobre las redes neuronales entrenadas con ejemplos obtenidos de la distribución Normal.

BIBLIOGRAFÍA.

CHANG, S.I. and AW, C.A. (1993) "A neural network control chart for detecting positive process mean shifts". Proceedings of the 7th Oklahoma Symposium on Artificial Intelligence, Stillwater, Oklahoma, Nov 18-19, pp 35-44.

CHANG, S.I. and AW, C.A. (1994) "A process mean control chart using neural networks". Proceedings of the 3rd Industrial Engineering Research Conference, May 18-19, Atlanta, Georgia, pp 337-342.

DE LA FUENTE, D.; HERNÁNDEZ, C. y DEL OLMO, R. (1989) "Control Adaptativo de Calidad en Procesos Industriales". QÜESTIIÓ, Vol 13, n°1,2,3 pp 105-125.

GUO, Y. and DOOLEY, K.J. (1992) "Identification of change structure in statistical process control" International Journal of Production Research, 30 (7), 1655-1669.

HILERA, J.R. y MARTÍNEZ, V.J. (1995) "REDES NEURONALES ARTIFICIALES: Fundamentos, modelos y aplicaciones". RA-MA Editorial.

KAUFMANN, A. y GIL ALUJA, J. (1986) "Introducción a la teoría de subconjuntos borrosos a la gestión de las empresas". Ed. Milladoiro. Santiago de Compostela.

KAUFMANN, A. y GIL ALUJA, J. (1987) "Técnicas operativas de gestión para el tratamiento de la incertidumbre". Hispano-Europea. Barcelona

KAUFMANN, A. y GIL ALUJA, J. (1990) "Las matemáticas del azar y la incertidumbre". Ceura. Madrid.

PUGH, G.A. (1991) "A comparison of neural networks to SPC Charts". Computers and Industrial Engineering, 21, 253-255.

YAGER, R.R. and FILEV, D.P. (1994) "Essentials of fuzzy modeling and control". (New York: Wiley).

ZADEH, L.A. (1969) "Toward a theory of fuzzy systems". E.R.L. Report 69, Berkeley University. California

Una Arquitectura para Controladores Neuro-Difusos CMOS Mixtos de Alta Complejidad

Rafael Navas-González
Dto. de Electrónica, UMA
Complejo Tecnológico,
Campus de Teatinos
29071-Málaga
email: rnavas@ctima.uma.es

Fernando Vidal Verdú
Dto. de Electrónica, UMA
Complejo Tecnológico,
Campus de Teatinos
29071-Málaga
email: vidal@ctima.uma.es

Angel Rodríguez Vázquez
Dto. de Diseño de Circuitos
Analógicos y de Señal Mixta
C N M-Universidad de Sevilla
Edificio CICA, C/Tarfia s/n,
41012-Sevilla
email: angel@cnm.us.es

Resumen

En aplicaciones para las que los requerimientos de precisión son de tipo medio o bajo, los circuitos analógicos resultan ser más eficiencientes que sus competidores digitales desde el punto de vista de la razon area/consumo de potencia [1]. Sin embargo, el límite en la precisión de los circuitos analógicos impone un límite en la complejidad abordable con los mismos. De ahí que los controladores difusos analógicos se hayan orientado habitualmente a sistemas con requerimientos de precisión media-baja, muy alta velocidad, bajo consumo y baja-media complejidad. En este artículo se presenta una estrategia de diseño de controladores difusos que emplea técnicas de señal mixta para conservar la mayoría de las ventajas de una implementacion analógica, a la vez que premite incrementar de manera notable la complejidad del sistema. En concreto, la estrategia propuesta consiste en aprovechar el carácter local de las reglas en el algoritmo de inferencia difusa. Así, se implementa con técnicas analógicas el algoritmo de inferencia para un número reducido de reglas, aquéllas que en cada momento determinan realmente las salida del sistema en cada punto concreto del universo de discurso. Este bloque, que llamaremos núcleo analógico, es programado dinámicamente para realizar el cómputo asociado a cualquier conjunto de reglas sobre cualquier punto del universo de discurso completo. Los datos para programar este núcleo analógico se almacenan en una memoria, que constituye la base de conocimiento del sistema difuso, a modo de conjunto de la reglas virtuales. Para mostrar la viabilidad de la propuesta se muestran simulaciones de HSPICE de un sistema ejemplo.

Palabras claves: Circuitos Digitales, Analógicos y Mixtos , Controladores Neuro-Difusos

1 INTRODUCCIÓN

El auge que han experimentado las aplicaciones basadas en la Lógica Difusa ha servido de impulso para la proliferacion de circuitos electrónicos de aplicación especifica (ASIC), los cuales aceleran el computo de los algoritmos difusos en aplicaciones de control donde se requiere velocidad [2][3]. Tales circuitos pueden ser digitales [3][4][11] o analógicos [5][7][8]. En los primeros, las técnicas de "pipeline" han permitido realizar el computo de un gran numero de reglas difusas por segundo, salvando así las restriciones de ancho de banda que impone el tiempo de propagación de las celdas en las tecnologías estándar empleadas. Sin embargo, por lo que respecta al retardo entrada-salida, que es el parámetro más importante a tener en cuenta en el control en tiempo real, éste es mayor en las realizaciones digitales que en las analógicas.

Los controladores difusos analógicos han demostrado ser ventajosos en aplicaciones de baja complejidad (p.e., aplicaciones con pequeño número de entradas y reglas) que requieren alta velocidad de operación y bajo consumo de potencia. En estos casos su baja resolución, no resulta un problema inportante, y en muchas de ellas se puede trabajar con resoluciones por debajo de los cuatro bits [8]. Además, los controladores analógicos ofrecen un interfaz natural a sensores y actuadores, ya que no es necesario la presencia de conversores A/D, D/A en el camino de las señales de control y realimentación.

Muchos de los problemas de control en los que un ASIC puede ser empleado tienen un nivel de complejidad no superior a 100 reglas y 4 entradas [3]. Entre los controladores difusos analógicos más complejos que aparecen en la literatura estan los reportados en [5] con 13-reglas y 3 entradas, y en [6] con 16-reglas y 2 entradas, de mucha menor complejidad que la anteriormente citada. Uno de los principales obstáculos para incrementar la complejidad de los controladores difusos analógicos es la existencia de nudos de computación global [9][10], (como los que aparecen al realizar el proceso de "deffuzzificación" empleando el método del centro de gravedad). En estos nudos se suman también los errores (sistemáticos o aleatorios) causados por los circuitos que realizan los antecedentes de regla, de modo que el error global final se hace mayor, lo que supone, entre otras cosas una degradación en su precisión. Esto es especialmente cierto en aplicaciones de control en las cuales es bastante usual emplear una partición en rejilla del universo de discurso [10], lo cual da lugar a un incremento exponencial de error global debido principalmente al crecimiento exponencial del número de reglas con el número de entradas. Así, para sistemas de complejidad media las implementaciones analogicas comienzan a tener problemas.

Este artículo presenta una nueva arquitectura para un chip controlador difuso de señal mixta en el que se

preservan las ventajas de una implementación analógica (p.e. paralelismo computacional, optima eficiencia en cuanto a la razon area/velocidad/consumo), a la vez que se independiza la acumulación del error del número de reglas, lo que permite un incremento en la complejidad. Esta arquitectura explota el caracter local de las reglas difusas; de modo que, para cada región en el universo de discurso, la salida puede ser obtenida a partir del conjunto de reglas que contribuyen con un valor no nulo en la región considerada. Llamaremos a estas reglas, reglas activas [11]. El controlador presentado tiene un núcleo analógico que realiza computo de la salida en base a las reglas activas sobre una región dada. Este núcleo es programado dinámicamente para situarse en cualquier intervalo de interpolación, utilizando técnicas de multiplexado digital.

2 ARQUITECTURA Y DESCRIPCIÓN FUNCIONAL

Considérese a modo de ilustración la partición en rejilla bidimensional de la Fig.1(a). Si sobre ella realizamos una nueva partición en base a aquellas regiones del universo de discurso para las cuales el conjunto de reglas activas es diferente (llamaremos intervalo de interpolación a cada una de esas regiones), obtendremos una partición como la de la Fig.1(b). Así por ejemplo, a cualquier pareja de entradas (x_1,x_2) en el intervalo punteado $C_{ij} = [(\varepsilon_{1i},\varepsilon_{1i+1}), (\varepsilon_{2j},\varepsilon_{2j+1})]$, le corresponde una salida que estará determinada por las reglas que aparecen en el recuadro sombreado que encierra a dicho intervalo, mientras que el resto de las reglas no tienen influencia en la salida. Por tanto, para el computo de la salida en esa región del universo de discurso, solamente son necesarias las funciones de pertenencia que corresponden a las reglas activas, así como sus valores "singleton" asociados. Es más, como se ilustra en la Fig.1(b), solamente los trozos

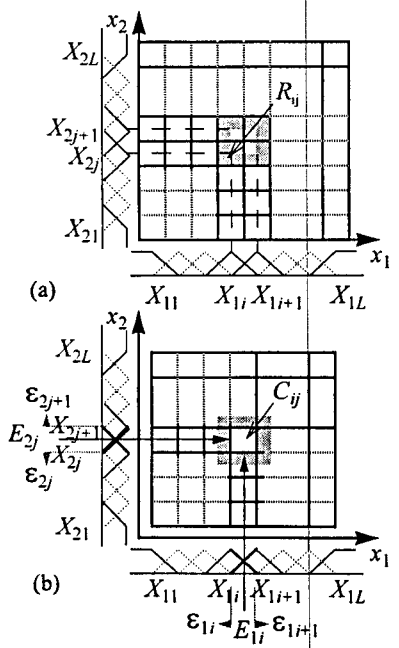


Fig. 1: Illustración del conjunto de reglas activas: (a) Partición en Rejilla; (b) Intervalos de Interpolación

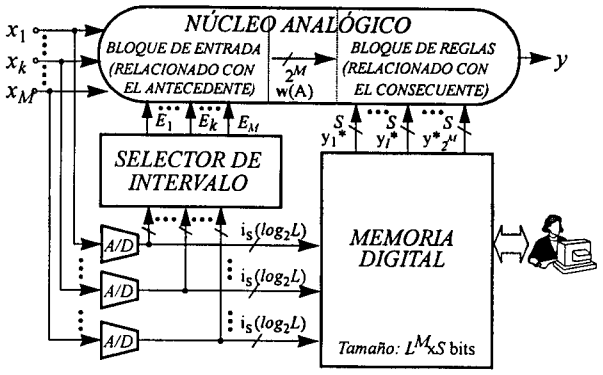


Fig. 2: Arquitectura global de un Controlados con Núcleo analógico Multiplexado.

de las funciones de pertenencia asociadas a las etiquetas X_{1i}, X_{1i+1}, X_{2j} y X_{2j+1}, y dibujados con trazo grueso, así como los valores "singleton" asociados a los consecuentes de las cuatro reglas activas $\mathbf{y}^*_{ij}$, $\mathbf{y}^*_{i(j+1)}$, $\mathbf{y}^*_{(i+1)j}$ y $\mathbf{y}^*_{(i+1)(j+1)}$, son necesarios para generar la salida en el intervalo C_{ij}.

La Fig.2 muestra una arquitectura global capaz de realizar el procedimiento descrito arriba para un controlador con M entradas y L etiquetas por entrada (esto es,un total de L^M reglas), en ella aparecen los siguientes bloques de alto nivel:

- *Núcleo Analógico*: este es el bloque que realiza el procedimiento de inferencia difusa. Posee un conjunto de entradas de programación que son proporcionadas por los bloques Selector de Intervalo y *Memoria Digital*. Estas entradas configuran al Núcleo Analógico para trabajar con el conjunto de reglas activas, lo que supone especificar las funciones de pertenencia asociadas a los antecedentes de regla, asi como los valores "singleton" asociados a los consecuentes de regla.
- *Convertidores A/D*: este bloque permite identificar el intervalo de interpolación C_{ij} al que pertenece una entrada dada y así determinar el conjunto de reglas activas. Está formado por M convertidores analogico/digital, uno por entrada, con una resolución de $i_s(log_2L)$ bits (esto es, el menor entero superior a log_2L). En conjunto, proporcionan una palabra de $M \times i_s(log_2L)$ bits, que codifica a cada uno de los intervalos de interpolación. Esta palabra digital es usada para seleccionar las funciones de pertenencia de los antecedentes de regla por el bloque *Selector de Intervalo*, y para acceder al bloque de *Memoria Digital* para obtener los valores "singleton" de los consecuentes de regla.
- *Selector de Intervalo*: Este bloque selecciona, a partir de la palabra digital proporcionada por el bloque de Convertidores, los valores de tensión $E_1,\ldots E_k,\ldots E_M$ que configuran el *Bloque de Entrada* del *Núcleo Analógico* para operar con las funciones de pertenencia asociadas a las reglas activas en cada intervalo.
- *Memoria Digital*: este bloque selecciona, a partir de la palabra digital proporcionada por el bloque de *Convertidores A/D*, los valores "singleton" $y_1^*,\ldots,y_l^*,\ldots y_{2^M}^*$ que configuran el *Bloque de Reglas*

del *Núcleo Analógico para operar con los consecuentes de las reglas activas*. La salida de este bolque es una plabra digital de tantos bit como son necesarios para codificar el conjunto de valores "singleton" requeridos. El bloque de memoria proporciona tambien un interfaz externo para accesos de lectura y escritura.

Es importante observar que la arquitectura mostrada en la Fig.2 resulta más ventajosa a medida que se incrementa el número de entradas y etiquetas. Por otra parte, el número de salidas del controlador podría ampliarse sin demasiado esfuerzo debido principalmente a que muchos de los bloques descritos podrían ser compartidos por la circuitería dedicada a generar cada salida. Más concretamente, cada salida adicional necesitaría un nuevo bloque de *Memoria Digital* y 2^M *Bloques de Regla.*

3 BLOQUES FUNCIONALES

La sección anterior describe brevemente la funcionalidad de los bloques de la Fig.2. En esta sección proponemos una realización CMOS para tales bloques.

3.1 NÚCLEO ANALÓGICO

La arquitectura del *Núcleo Analógico* se muestra en la in Fig.3. Esta es la misma que la del controlador difuso descrito en [12], aunque en este caso se necesitan 2^M reglas, y solamente dos funciones de pertenencia por entrada. Los bloques de diseño son tambien bastante similares a los descritos en [12], donde el lector puede encontrar más detalle sobre ellos.

Sin embargo, es importante destacar algunas de las diferencias, derivadas principalmente de la estrategia específica de multiplexado, aquí presentada. En primer lugar, como se dijo en la sección 2., solamente son necesarios los trozos de las funciones de pertenencia marcados en trazo grueso en la Fig.1(b) para evaluar las salida en cada intervalo. Por tanto, el circuito generador de funciones de pertenencia debe ser capaz de generar dos trozos con pendientes de signo opuesto. Esto puede realizarse de una manera simple por medio de un par diferencial como se muestra en la figura Fig.4(a). Es más, el mismo par diferencial proporciona dos curvas complementarias una de la otra, hecho que puede ser explotado para realizar directamente la operación de complementación, si, como es el caso, realizamos el operador mínimo por medio de un circuito de máximo más

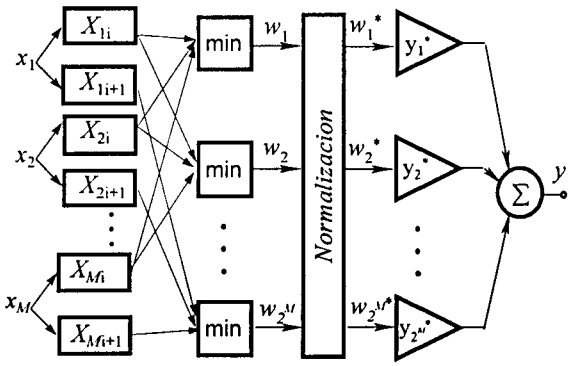


Fig. 3: Arquitectura del Núcleo Analógico

complementario, siguiendo las leyes de De Morgan [7]. La Fig.4(b) muestra uno de los *Bloques de Entrada*, que es uno de los bloques de diseño de alto nivel que realiza la mayoría de los circuitos asociados a cada una de las entradas en la primera y segunda capas de la Fig.3. La realización propuesta tiene como entradas valores de tensión, mientras que el computo posterior se realiza en modo corriente. Las corrientes de salida del par diferencial se replican para generar las 2^M salidas que se requieren para realizar las 2^M reglas que determinan la salida dentro de un intervalo de interpolación específico. Tales corrientes se convierten de nuevo en tensiones en la celda de entrada del circuito de mínimo, sombreado en la Fig.4(b). Las 2^M salidas de este bloque se conectan a las correspondientes proporcionadas por el resto de los bloques de entrada. Como resultado se tienen 2^M antecedentes de regla. La salida de los antecedentes de regla se conectan a los 2^M *Bloques de Regla*, como se muestra en la Fig.4(c). Este es otro de los bloques de diseño de alto nivel, y realiza los circuitos correspondientes a las capas 3 y 4 de la Fig.3 así como la etapa de salida del circuito de mínimo. Esta etapa, que pertenece al bloque de mínimo (Fig.3), es la que realmente actúa como interfaz entre los dos bloque de alto nivel mencionados *Bloque de Entrada* y *Bloque de Reglas*. A modo de ilustración, la Fig.4(d) muestra un ejemplo de interfaz para construir la regla R_{ij} de la Fig.1. En el ejemplo de la Fig.1, se necesitan 2 *Bloques de entrada*, los cuales proporcionan cuatro salidas de tensión cada uno (nótese que las salidas de cada par diferencial son replicadas para compartir el circuito y ahorrar area y consumo de potencia). Para construir la regla R_{ij}, las salidas $V_{G1(i+1)}$ y $V_{G2(j+1)}$, correspondientes a las funciones de pertenencia $X_{1(i+1)}$ y $X_{2(j+1)}$, se conectan a la celda de salida del circuito de mínimo del *Bloque de Regla* asociado a R_{ij}. La Fig.4(d) es de hecho un circuito de mínimo donde se ha ahorrado realizar el complemento a la entrada porque el par diferencial ya proporciona la señal complementada. Observese que $X_{1(i+1)}$ and $X_{2(j+1)}$ son respectivamente los complementos de X_{1i} and X_{2j} en el intervalo de interpolación C_{ij} de la Fig.1.

El bloque de alto nivel de la Fig.4(c) es muy similar al bloque de reglas descrito en [12]. Sin embargo, el circuito de ponderación de "singleton" es ligeramente diferente. Debido a que esta celda es reconfigurada dinámicamente por una palabra digital $s_{l0}...s_{lS-1}$, los cambios en esta palabra generan transitorios en la corriente de salida. Tales transitorios suponen grandes excursiones de la corriente de salida. La principal causa de ésto, es la demanda de corriente por las ramas que no contribuyen a la corriente de salida, y cuyos transistores entran en la región óhmica, donde son forzados a aportar corriente a la salida. La solución propuesta trata de mantener todos los transistores en saturación proporcionando un camino de corriente alternativo a través de conmutadores de corriente controlados por señales de control complementarias $s_{l0}...s_{lS-1}$. Un mayor consumo de potencia es, de nuevo, el precio a pagar para tener un mejor comportamiento dinámico. Finalmente, la salida global del sistema se obtiene sumando las salidas de los *Bloques de Regla*,

$$y = \sum_{l=1...2^L} y_l \qquad (1)$$

lo que se realiza uniendo directamente los nodos de salida de dichos bloques, ya que la salida del *Bloque de Regla* es una corriente.

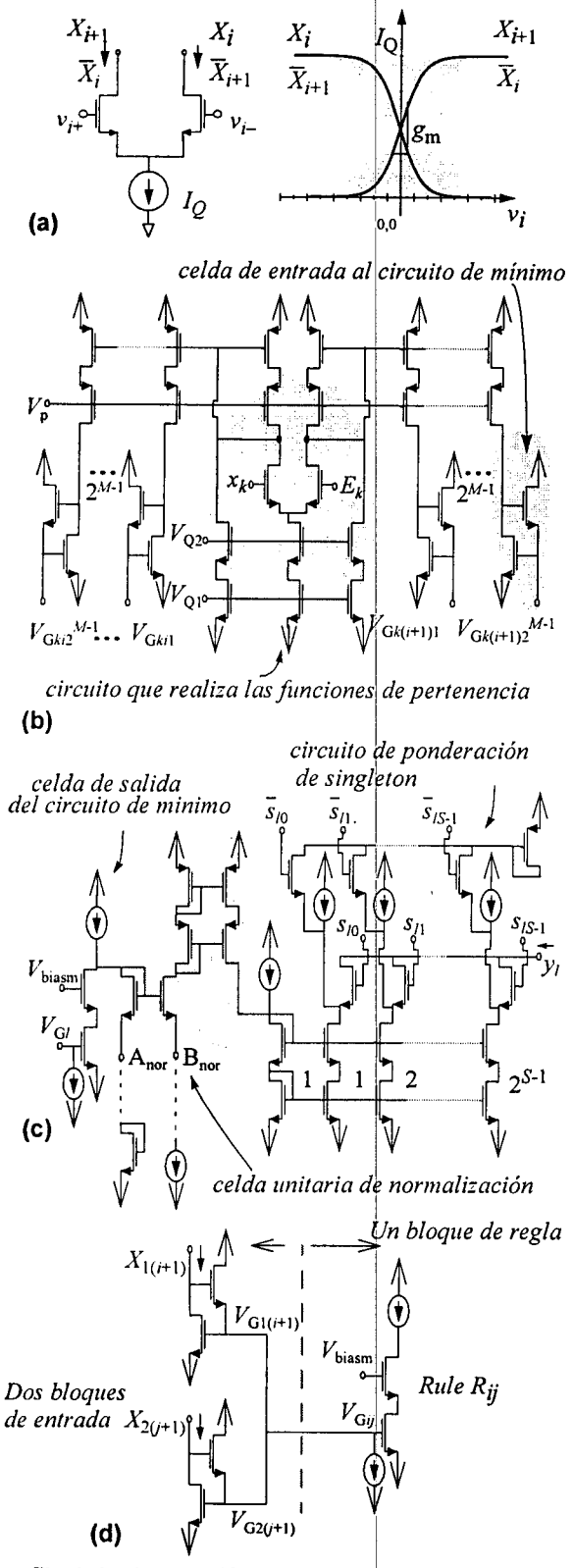


Fig. 4: Implementación del Núcleo Analógico; (a) Generación de la función de pertenencia, (b) Bloque de entrada (c) Bloque de reglas y (d) Interfaz entre ambos.

3.2 CONVERTIDORES A/D

En la literatura se pueden encontrar diferentes implementaciones de convertidores A/D que pueden ser usadas para realizar este bloque. Sin embargo, debido por un lado a la baja resolución requerida (se requieren $i_s(log_2 L)$ bits y L raras veces es mayor que siete u ocho), y por otro a la necesidad de velocidad, (el retardo en la identificación de intervalo de interpolación influye directamente en el retardo global) se ha optado por emplear convertidores del tipo "flash", como el que se muestra en la Fig.5. Aunque el bloque básico es un convertidor "flash" común, merece la pena comentar ciertos detalles sobre la realización global del *Bloque de Convertidores*. En primer lugar, si realizamos una partición en rejilla idéntica para cada entrada en su universo de discurso, debido a la alta impedancia de entrada de los comparadores, puede compartirse una única serie de resistores lineales para generar todos los valores de referencia que definen los límites de los diferentes intervalos de interpolación (ε_{ki} en la Fig.5), con el consiguiente ahorro en área y consumo. Además, dicha serie de resistencias puede ser diseñada para proporcionar también los valores de tensión E_{ki} que constituyen los diferentes valores de programación de los antecedentes de regla de los *Bloques de Entrada* en el *Núcleo Analógico*, de manera que en conjunto podemos decir que constituyen una especie de memoria analógica de solo lectura. En segundo lugar, para proporcionar cierto grado de inmunidad al ruido en las entradas del sistema, que pudiese provocar inestabilidad a la salida, sobre todo para valores de entrada próximos a los límites entre intervalos de interpolación, se ha dotado de histéresis a los comparadores empleados en los conversores. Por último, se ha utilizado un codificador Gray para convertir la escala en termómetro, formada por las salidas de los comparadores, en un codigo Gray. Esta codificación de las salidas permite minimizar el número de líneas que conmutan simultaneamente, lo que reduce los riesgos debidos a carreras y minimiza el ruido inyectado por estas líneas en el resto del sistema. Recuérdese que estas líneas se utilizan para seleccionar los valores de programación del *Núcleo Analógico*.

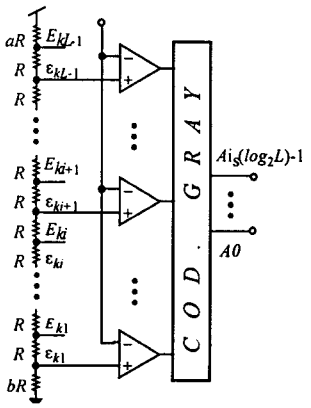


Fig. 5: Convertidor A/D de tipo "Flash"

3.3 SELECTOR DE INTERVALOS

Este bloque está formado básicamente por un bus analógico, $E_1 \ldots E_M$ en la Fig.2, cuyos valores se seleccionan digitalmente de entre los valores de referencia generados en la serie de resistores lineales descrita en el *Bloque de*

Convertidores A/D. La Fig.6 muestra una de las M celdas (una por entrada) que constituyen este bloque. Un decodificador Gray proporciona las señales de control que actúan sobre las puertas de transmisión, a partir de las salidas del *Bloque de Convertidores A/D* que identifican al intervalo de interpolación correspondiente a las entradas actuales.

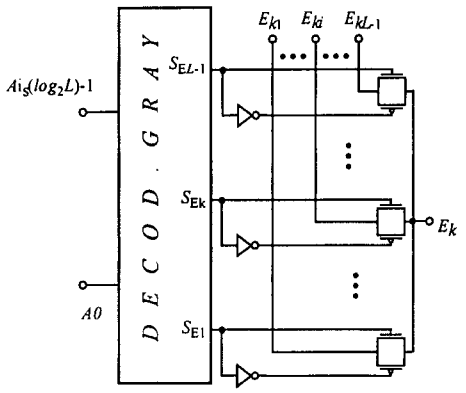


Fig. 6: Celda k del Selector de Intervalos

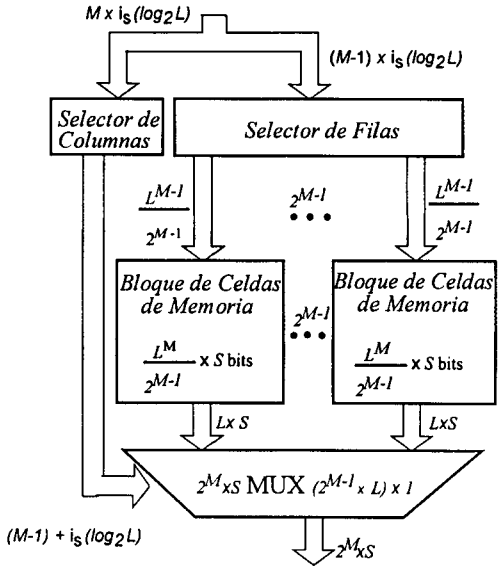


Fig. 8: Arquitectura del bloque Memoria Digital

3.4 MEMORIA DIGITAL

La palabra digital de $M \times i_s(\log_2 L)$ bits proporcionada por el *Bloque de Convertidores A/D* y asociada a cada intervalo en el espacio de entrada, es usada para direccionar una memoria digital. Esta memoria proporciona los valores "singleton" que son necesarios para generar la salida en cada intervalo. Por tanto, el *Bloque de Memoria Digital* proporciona una palabra de 2^M x S bits, donde S es el número de bits por valor "singleton y 2^M es el número de valores "singleton" que son necesarios para programar el *Bloque de Reglas* del *Núcleo Analógico*. Es importante observar que la posición que ocupan los valores "singleton" en esa palabra viene impuesta por los requerimientos de programación del *Núcleo Analógico*. Para explicar esto, veamos el ejemplo de la Fig.7. En ella se muestra, para dos intervalos de interpolación adyacentes cuál es la palabra que debe proporcionar el bloque de *Memoria*. Como puede observarse, ambos intervalos comparten dos valores

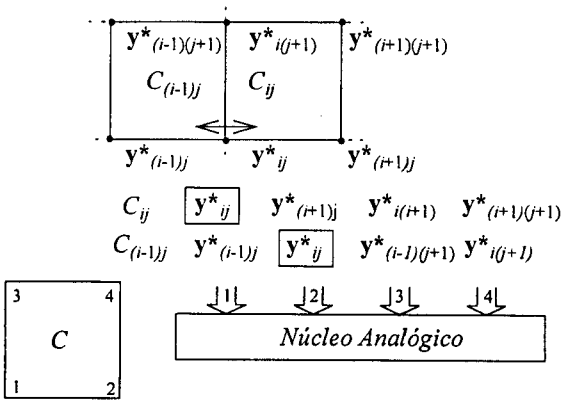


Fig. 7: Interfaz Núcleo Analógico - Memoria

"singleton" (y^*_{ij} and $y^*_{i(j+1)}$). Estos dos valores "singleton" deben aparecer en diferente orden en cada una de esas palabras. Una posible estrategia para realizar el bloque de *Memoria* consiste en almacenar una palabra para cada intervalo con los valores "singleton" en su posición correcta. Sin embargo, esto implica replicar cada valor "singleton" tantas veces como intervalos de los que forma parte haya, esto es 2^M. Una segunda estrategia consiste en organizar apropiadamente los valores "singleton" en distintos bloques de memoria y multiplexar el bus de salida para evitar redundancia, que es la estrategia propuesta para el controlador de este trabajo. Este multiplexado de datos no se realiza en el tiempo, todos los valores "singleton" aparecen en la posición correcta en un sólo paso. La Fig.8 muestra la arquitectura propuesta para el *Bloque Memoria Digital* que sigue la segunda estrategia descrita anteriormente. Los valores "singleton" son distribuidos en 2^{M-1} bloques de celdas de memoria de $L^{M-1}/2^{M-1}$ filas y L columnas cada uno. Cada bloque de celdas de memoria almacena parejas de valores "singleton" que necesitan ser direccionados simultáneamente. El selector de filas selecciona 2^{M-1} filas simultáneamente, una por cada bloque de celdas de memoria, para obtener así un conjunto de $L \times 2^{M-1}$ valores "singleton". El selector de columnas selecciona los valores "singleton" correctos a partir de este conjunto, y controla el multiplexor para colocarlos en la posición correcta para formar la palabra de 2^M x S bits de salida, todo en un sólo paso. Para ilustrar la distribución de datos en el *Bloque de Memoria* y seguir su funcionamiento podemos analizar la situación para un caso bidimensional (dos entradas). La Fig.9(a) muestra cuatro intervalos genéricos adyacentes $C_{ij}, C_{(i-1)j}, C_{(i-1)(j-1)}, C_{i(j-1)}$ con sus correspondientes valores "singleton" asociados. Esos datos se han distribuido en dos bloques de celdas de memoria como muestra la parte superior de la Fig.9(b). El bloque de Celdas de Memoria 1 contiene las filas con el índice j impar, mientras que el bloque de Celdas de Memoria 2 contiene las filas con índice j par. Ambos bloques tienen L columnas.

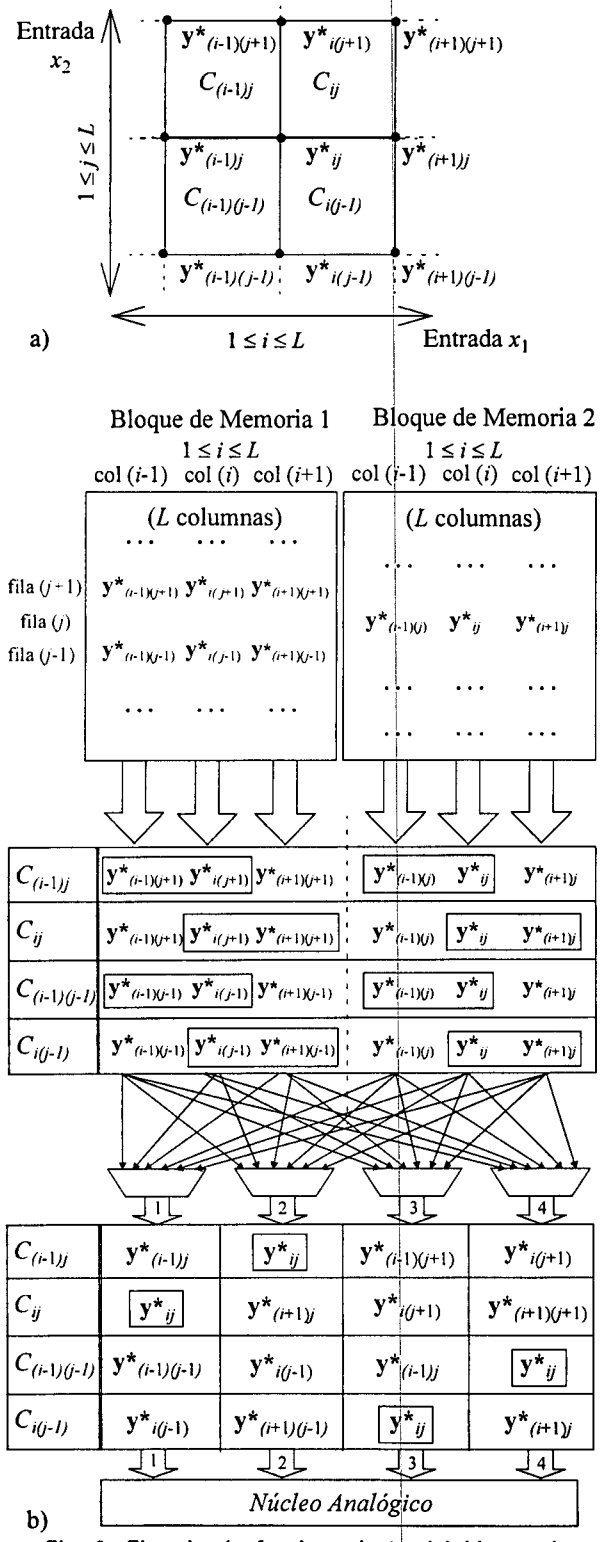


Fig. 9: Ejemplo de funcionamiento del bloque de memoria:a) Organización de datos, b) Datos proporcionados para accesos a intervalos adyacentes.

Debajo, en la parte central de la Fig.9(b) se muestra el resultado de la selección para cada uno de los cuatro intervalos de la Fig.9(a). Todos los valores "singleton" necesarios para programar el *Núcleo Analógico* (marcados con un recuadro en la figura) están en la palabra resultante de la selección por filas. Por último el selector de columnas, selecciona los valores correctos en cada caso y los coloca adecuadamente en el bus de salida, como puede verse en la parte inferior de la Fig.9(b). Obsérvese que cada valor "singleton" aparece en su posición correcta (ver Fig.7). Para accesos externos, todos los bloques de memoria son configurados de manera que el bloque completo se comporta como una memoria RAM convencional.

4 RESULTADOS Y CONCLUSIONES

Este artículo presenta una estrategia para realizar controladores de media-alta complejidad manteniendo las ventajas del diseño analógico. Se muestran algunos resultados de simulación con HSPICE de un ejemplo de controlador para demostrar la viabilidad de la propuesta. La arquitectura de controlador de la Fig.2 se ha particularizado para L=8, M=2 and S=4. La Fig.10(a) muestra la salida del controlador (superficie de control) cuando los valores "singleton" correspondientes a los consecuentes de regla toman alternativamente sus valores máximo y mínimo; de manera que así se aprecian claramente todos los intervalos de interpolación. Para ilustrar el comportamiento dinámico, la Fig.10(b) muestra el resultado HSPICE de un análisis transitorio en el que la entrada x_1 describe una trayectoria sobre la superficie de control de la Fig.10(a), como muestra la gráfica central de la Fig.10(b), en la que transita por diferentes intervalos de interpolación. Las curvas que aparecen en la grafica superior de la Fig.10(b) se obtienen de repetir dicha trayectoria para diferentes valores estáticos de la entrada x_2 :2.80V, 2.90V, 2.95V and 3V. El grafico inferior de la Fig.10(b) pone de manifiesto los cambios en los valores "singleton" como consecuencia del cambio de intervalo de interpolación y por tanto muestra la reprogramación dinámica del nucleo analógico. El controlador se ha diseñado en tecnología CMOS 0.7µm.

Es claro que para esta estrategia, tanto el area como el consumo de potencia crece con la complejidad del sistema. A fin de proporcionar una figura de merito para medir sus ventajas frente a una implementación completamente analógica permítasenos el siguiente razonamiento: Consideremos que el area y el consumo de potencia de los bloques *Selector de Intervalo* y *Convertidores A/D* es despreciable cuando se compara con el *Núcleo Analógico* y el *Bloque de Memoria Digital*. (recuérdese que solo se necesita una serie de resistores lineales para todos los convertidores que haya en el sistema). Además, supongamos un área de interconexión similar en ambos casos (las cual es realmente inferior en la estrategia propuesta). Por último, ya que el tamaño de memoria digital es el mismo en ambas implementaciones, las suposiciones anteriores nos permiten estimar el ahorro en área y potencia como la razón entre el número total de reglas (L^M) que deberían ser implementadas sin multiplexación, y el número de reglas activas (2^M), que es el implementado fisicamente en la estrategia propuesta. Tal razón es,

$$\alpha = (L/2)^M \qquad (2)$$

que depende fuertemente del número de entradas y el número de etiquetas por entrada, ésto es, de la complejidad del sistema. De aquí se tiene que mientras mayor sea el valor de α en (2), más adecuada es la estrategia de multiplexación. Finalmente, recordar que solamente las reglas implementadas, las reglas activas, contribuyen al error en la salida, lo cual permite la expansión del sistema para un límite de error dado, con respecto a una realización completamente analógica.

5 Referencias

[1] K.A. Nishimura, "*Optimum Partitioning of Analog Mixed-Signal Circuits for Signal Processing*". Memorandum No. ECB/ERL M93/67, University of California at Berkeley. 1993

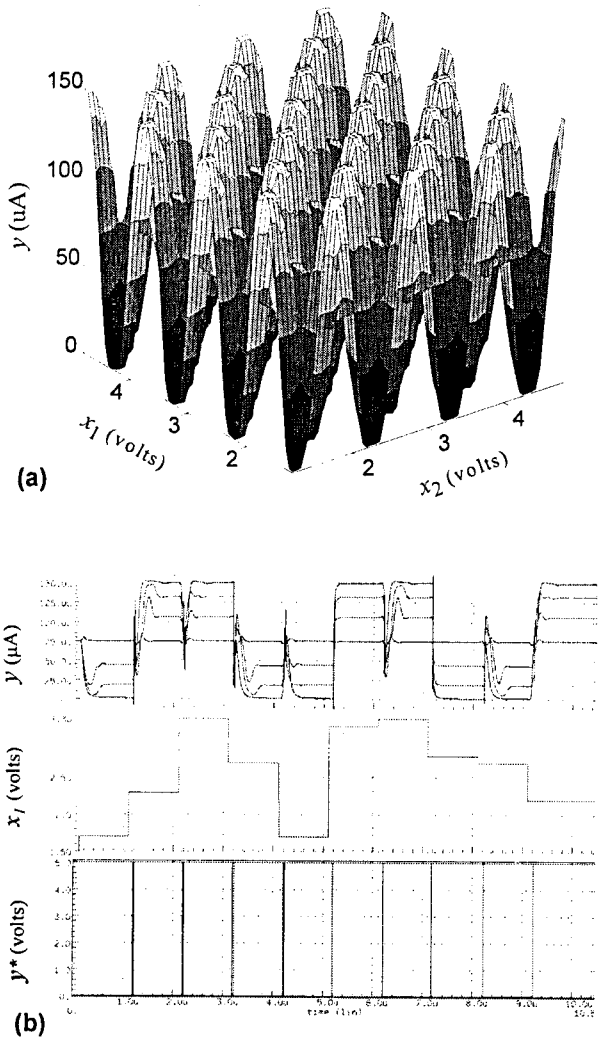


Fig. 10: Resultados de un Controlador ejemplo; (a) Ejemplo de superficie de control bidimensional, (b) Transitorio entre diferentes intervalos.

[2] P.P. Bonissone et al., "Industrial Applications of Fuzzy Logic at General Electric", *Proceedings of the IEEE,* Vol. 83, pp. 450-465, March 1995

[3] A. Costa, A. de Gloria, P. Faraboschi, A. Pagni and G. Rizzotto, "Hardware Solutions for Fuzzy Control". *Proceedings of the IEEE,* Vol. 83, pp. 422-434, March 1995.

[4] H. Eichfeld, M. Klimke, M. Menke, J. Nolles and T. Künemund, "A General-Purpose Fuzzy Inference Processor", *Proc. of the 4th Int. Conf. on Microelectronics for Neural Networks and Fuzzy System* , Sept. 1994

[5] S.Guo, L. Peters, and Surmann, "Design and Application of an Analog Fuzzy Logic Controller". *IEEE Trans. on Fuzzy Systems,* Vol. 4, pp. 429-438, November 1996

[6] F. Vidal-Verdú, R. Navas and A. Rodríguez-Vázquez, "A 16Rules@2.5 Mflips Mixed Signal Programmable Fuzzy Controller CMOS-1μm Chip" *Proc. of the ESSCIR'96,* pp. 156-159, Sep.1996.

[7] F. Vidal-Verdú and A. Rodríguez-Vázquez, "CMOS Design of Analog Neuro-Fuzzy Controllers using Building Blocks" *IEEE Micro*, August 1995.

[8] T. Yamakawa, "A Fuzzy Inference Engine in Nonlinear Analog Mode and Its Application to a Fuzzy Logic Control". *IEEE Trans. on Neural Networks,* Vol. 4, pp. 496-522, May 1993.

[9] J.S.R. Jang and C.T. Sun, "Neuro-Fuzzy Modeling and Control". *Proceedings of the IEEE,* Vol. 83, pp. 378-406, March 1995.

[10] M. Brown and C. Harris, *NeuroFuzzy Adaptive Modeling and Control*, prentice Hall International, 1994.

[11] M. Masetti, E. Gandolfi, A. Gabrieli and F. Boschetti, "4 Input VLSI Fuzzy Chip Design able to process an Input Data Set every 320ns", *Proceedings of the JCIS'95,* Wrightsville Beach, North Carolina, USA, October 1995

[12] F. Vidal-Verdú, R. Navas and A. Rodríguez-Vázquez, "A Modular CMOS Analog Fuzzy Controller", *Proceedings of the FUZZ-IEEE'97*, Barcelona, Spain, July 1997.

Red Neuro-Fuzzy para Clasificación de Patrones

Francisco Javier López Aligué, Miguel Macías Macías, Isabel Acevedo Sotoca, Horacio González Velasco y Carlos Javier García Orellana
Departamento de Electrónica e Ingeniería Electromecánica. Universidad de Extremadura.
Facultad de Ciencias, Badajoz, Spain

Resumen

En este trabajo se presenta un nuevo modelo de neurona fuzzy especialmente indicado para el reconocimiento y clasificación de caracteres. Se trata, básicamente, de un modelo de medida de diferencias entre los prototipos que definen las clases y las entradas externas.
El modelo conduce a una estructura abierta de redes neuronales de gran tamaño, dotadas de unos algoritmos de aprendizaje sumamente sencillos y rápidos. Se muestran, a modo de evaluación, los resultados obtenidos en una base de datos.

Palabras Clave: Reconocimiento de patrones, neurona fuzzy, aprendizaje supervisado.

1 INTRODUCCIÓN

Desde un punto de vista práctico, un sistema fuzzy de clasificación de patrones es un ente desligado del soporte material en el que se ejecuta. Por regla general, se implementa sobre un ordenador mediante un programa que realiza las funciones del algoritmo de clasificación que es, a fin de cuentas, el responsable de la característica fuzzy del proceso. Dado que la construcción de estos algoritmos por métodos informáticos convencionales impone, si se quiere que funcionen adecuadamente, unas limitaciones excesivamente severas, se buscan otros métodos para construirlos; entre éstos, las Redes Neuronales ofrecen una posibilidad muy atractiva dado que la estructura resultante es totalmente iterativa y el aprendizaje de las clases de pertenencia no exige una programación específica.

Sin embargo, la utilización indiscriminada de redes neuronales en los mecanismos de clasificación no suele traer consigo unos beneficios significativos, por que ciertos parámetros, tales como el tiempo necesario para que se lleve a efecto el aprendizaje, provocan un exceso de tiempo o de coste que los hace inaceptables, salvo para casos muy limitados, y en esos casos, los algoritmos tradicionales presentan una competencia muy fuerte.
En este trabajo presentamos un modelo de función neuronal, que no dudamos en considerar neuro-fuzzy, especialmente indicada para llevar a cabo procesos de clasificación fuzzy de patrones con gran velocidad de aprendizaje y programación.

2 LA FUNCIÓN NEURONAL

La idea de partida consiste en asignar a cada neurona la misión de calcular el "parecido" existente entre las entradas externas $\vec{z}_i = (z_{1i}, z_{2i}, \ldots, z_{ni})$ que recibe y la configuración de los pesos de conexionado asociados a dichas entradas (las sinapsis). Identificamos ese parecido mediante una función que denominaremos $a_i(\vec{z}_i, \vec{w}_i)$ o función de "activación" (por razones que más adelante veremos) y que viene definida por:

$$a_i(\vec{z}_i, \vec{w}_i) = 1 - \frac{1}{n}\sum_{k=1}^{n}|z_{ki} - w_{ki}| = \frac{1}{n}\sum_{k=1}^{n}\left[1 - |z_{ki} - w_{ki}|\right] \quad (1)$$

donde "n" es el número de entradas y pesos, es decir, la dimensión de los vectores $\vec{z}_i$ y $\vec{w}_i$. Por regla general, tanto las entradas como los pesos sinápticos están acotados entre 0 y 1. Para aquellas, estos límites identifican la máxima y mínima actividad, la iluminación máxima y la total obscuridad para el caso de patrones ópticos, mientras que, para los pesos, son el resultado del proceso de aprendizaje que, como veremos más adelante, obedece a parecidos criterios. Pues bien, como puede verse, el valor de la función de activación también está comprendido entre 0 y 1. El valor 0 corresponde al caso en que las entradas y los pesos se encuentren en extremos opuestos, es decir cuando uno vale 0 y el otro 1 para todos los

componentes, y 1 cuando ambos elementos, entradas y pesos, sean exactamente iguales, también para todas las componentes.
En las demás situaciones, el valor estará comprendido entre ambos extremos. No es difícil demostrar, a partir de estos hechos, el carácter fuzzy de esta función. En realidad, se trata de una función que no efectúa ninguna clasificación, sino que se limita a medir el parecido entre dos vectores: uno que ha adquirido por aprendizaje (los pesos) y otro que recibe del exterior en "tiempo real".
A primera vista, esta definición puede parecer muy lejana de la más convencional de la sumatoria de los productos de entradas por pesos, a la que suele escribirse de la forma:

$$a_i(\vec{z}_i,\vec{w}_i)=\sum_{k=1}^{n} z_{ki}\, w_{ki} \qquad (2)$$

pero, en realidad, no se trata de otra cosa que de un producto escalar de ambos vectores que, a fin de cuentas, es una medida de su coincidencia. Es decir, se trata de destacar, y facilitar, el cálculo de la semejanza existente entre ambos vectores lo que, a fin de cuentas, es el punto de partida para el cálculo de la mayoría de las funciones de pertenencia.
Es importante notar que, en esta función, no existen las conexiones inhibidoras (de peso sináptico asociado negativo) apriorísticas. Es decir, no hay entradas que, por definición, aporten cantidades negativas al valor de la actividad. Este concepto, tan biológicamente justificable, aparecerá de forma natural y espontánea al llegar a la función de salida de la neurona. Aquí queremos dejar claro que nuestra neurona calcula simplemente el parecido entre vectores y, como cualquier función fuzzy, está comprendido entre 0 (disimilares – no pertenece) y 1 (iguales – pertenece totalmente).
Sin embargo, antes de pasar al cálculo de la salida de la neurona, derivada a partir de la función de actividad, es necesario hacer algunas consideraciones de orden práctico. Aunque todavía no hemos hablado del mecanismo de aprendizaje, podemos anticipar que se trata de un método derivado del LVQ de Kohonen, de forma que los pesos corresponden a los valores de ciertas fracciones de los prototipos. Así, si durante el entrenamiento un determinado pixel de la imagen del prototipo no está iluminado, lo que para patrones de tipo de caracteres gráficos correspondería a aquellos puntos no cubiertos por el carácter impreso (la letra o el número a aprender en cuestión), el peso asociado a ese punto de la imagen será nulo.
Cuando la red esté en funcionamiento, recibirá entradas externas correspondientes a patrones que deberá reconocer. En varios casos, es posible que el valor de la entrada (componente z_{ki}) sea también nulo. En esta situación, tendremos que evitar que se produzca ninguna contribución a la actividad de la neurona, puesto que se trata de un punto donde no hubo señal y donde ahora tampoco la hay. Aunque insistiremos sobre este aspecto, pensemos que de no evitarlo, la ausencia de entrada (la oscuridad total) sería, probablemente, la entrada con más similitudes detectadas.
Para ello, consideremos tan sólo el número de pesos activados (es decir, distintos de cero) durante el aprendizaje, y sea éste r_i. En consecuencia, ahora deberemos tomar como "positivas" aquellas entradas que coincidan con las del prototipo, resultando "negativas" las que no lo hagan. Así, la nueva expresión de la actividad pasa a ser:

$$a_i(\vec{z}_i,\vec{w}_i)=\frac{1}{r_i}\sum_{k=1}^{n} I_{ki} \quad \text{con} \qquad I_{ki}=\begin{cases} 1-|z_{ki}-w_{ki}| & si \quad z_{ki}>\theta \\ 0 & si \quad z_{ki}\le\theta \end{cases} \qquad (3)$$

donde hemos tenido en cuenta que el número máximo de entradas positivas será r_i y, además, hemos evitado la contribución del ruido de fondo de la obscuridad mediante la imposición del umbral θ, o umbral de actividad, por debajo del cual se considera nulo el valor de cualquier entrada.
Existe una última situación, de orden eminentemente práctica que debe considerarse. Se trata de aquellas neuronas que durante el aprendizaje no recibieron ninguna entrada activa y que, por lo tanto, tienen sus pesos todos nulos. Eso significa que $r_i=0$, es decir, que la ecuación (3) nos da un valor de $a_i=\infty$ siempre. Para evitarlo, es conveniente introducir esta salvedad y definir la actividad de la forma:

$$a_i(\vec{z}_i,\vec{w}_i)=\begin{cases} \dfrac{1}{r_i}\displaystyle\sum_{k=1}^{n} I_{ki} & si \quad r_i>0 \\ 0 & si \quad r_i\le 0 \end{cases} \qquad (4)$$

Por último, existe una posibilidad, como veremos más adelante, de que la suma de las actividades parciales sea un número negativo. Para evitar esta nueva contingencia, la salida de la neurona se deriva, mediante una función del tipo escalón (sigmoide, umbral,...) que permita, además, establecer ciertas limitaciones tales como la "inercia" a generar una salida de valor considerable cuando la entrada es aun pequeña, o que a partir de cierto valor de la actividad, la salida presente una clara tendencia a saturarse, manteniendo el rango acotado entre los valores extremos 0 y 1 para cualquier caso. De esta forma, la función de salida quedará expresada como:

$$y_i = U\left[a_i(\vec{z}_i,\vec{w}_i)\right] \qquad (5)$$

3 EL ALGORITMO DE APRENDIZAJE

Como hemos dicho al principio, uno de los aspectos más notables de nuestra Red es la facilidad y rapidez con que se lleva a cabo el aprendizaje. El punto de partida consiste en emplear el concepto ampliamente aceptado de que los pesos sinápticos de las conexiones neuronales constituyen la memoria de la red. Por otra parte, el objetivo final consiste en utilizar la red para que calcule cuanto se parece una entrada determinada a los distintos prototipos con los que se definen las clases, para poder efectuar la asociación final.

Así, el aprendizaje, de manera similar al algoritmo LVQ, consiste en almacenar los prototipos a lo largo y ancho de los pesos sinápticos. Para ello, tomaremos los valores de los pixels de cada uno de los prototipos con los que deseemos definir las distintas clases, y los iremos distribuyendo por los pesos de las neuronas de la red.

Es, como en el caso del LVQ, un aprendizaje supervisado en el que el conjunto de las clases, definidas por sus respectivos prototipos, se conoce de antemano y en el que cada neurona de la capa de salida está asociada con alguna de ellas, a las que denominaremos conjunto de entrenamiento.

El aprendizaje se lleva a cabo en dos partes: Primero, se asocian una serie de neuronas a cada prototipo. Ello supone conocer de antemano cuantos prototipos hay, es decir conocer el conjunto de las clases, y después establecer un reparto de las neuronas entre las clases. Este reparto puede ser cualquiera, desde aleatorio hasta totalmente rígido. Como resultado, si la Red tiene N neuronas y debe dedicarse a la clasificación de las entradas entre M clases, a cada clase se le asocian P neuronas, de forma que $P=N/M$, pero no obligatoriamente. Lo importante es conocer cuales son y que ninguna clase se quede sin asociarse a un subconjunto de neuronas de las N totales posibles.

La segunda etapa del aprendizaje consiste en igualar el vector peso $\vec{w}_i$ de la i-ésima neurona a su vector entrada $\vec{z}_i$ cuando esa entrada es el prototipo, para todas aquellas neuronas asociadas a la clase, manteniendo sus pesos inalterables el resto de neuronas.

Como resultado, cada neurona tiene almacenados, en sus pesos sinápticos, los valores de algunos de los pixels del prototipo al que se la asoció. Dado que a cada prototipo se le asocia un elevado número de neuronas, sus pixels estarán repartidos sobre todas ellas y, dado que las entradas de las neuronas se distribuyen geográficamente sobre parte del plano de entrada, cada pixel del prototipo está almacenado en varias entradas de varias neuronas, utilizando un esquema redundante muy similar al que la propia naturaleza emplea en sus sistemas biológicos.

Sin embargo, esta identificación de los pesos con las entradas no corresponde a una igualación total. En efecto, hay una situación, en la que establecemos un convenio diferente: nos referimos al caso en el que el valor de la entrada no supera el umbral de actividad θ. En este caso, vamos a dar al peso un valor -1, orientado a provocar que cuando estemos en fase de reconocimiento, la fase normal de funcionamiento después del aprendizaje, una entrada nula suponga un incremento nulo de la actividad neuronal y una entrada activa (un pixel iluminado donde el prototipo tenía un pixel apagado) sea un factor negativo para la actividad de la neurona.

Así, la descripción completa del proceso de aprendizaje, vendrá definida de la siguiente manera:

$$w_{ki} = \begin{cases} z_{ki} & si \quad z_{ki} > \theta \\ -1 & si \quad z_{ki} \leq \theta \end{cases} \qquad (6)$$

En definitiva, todo el aprendizaje es de tipo supervisado, con clases definidas a priori y se lleva a cabo en un solo paso. Como consecuencia, el proceso es de máxima rapidez y puede ser repetido a menudo, adaptando así la red a las nuevas clases que vayan estableciéndose a lo largo del tiempo, sin que ello suponga una merma importante de su funcionamiento.

4 ESTUDIO DE LA ACTIVIDAD DE LA NEURONA

Podemos comprobar que el carácter excitador o inhibidor de las conexiones de las entradas a la neurona viene impuesto por el proceso de aprendizaje. Para comprobarlo, estudiaremos la contribución a la actividad según las posibles combinaciones entrada-peso sináptico:

- *Caso 1* . $z_{ki} > \theta$ y $w_{ki}>0$:
  Como z_{ki} y w_{ki} son en este caso positivos y comprendidos en el intervalo $[\theta,1]$, tendremos que $I_{ki}=1-|z_{ki}-w_{ki}|$ es positivo y tomará valores en el intervalo $[\theta,1]$. A este tipo de entradas a la neurona donde, tanto la entrada como el peso asociado a ella superan el umbral de actividad les llamaremos excitadoras ya que contribuyen con una cantidad positiva, perteneciente al intervalo $[\theta,1]$, al valor de la actividad interna de la neurona.

- *Caso 2.* $z_{ki} \leq \theta$ y $w_{ki}>0$:
  En esta situación, como la entrada a la neurona no supera el umbral de actividad, la ecuación (4) indica que $I_{ki}=0$, lo que parece indicar que dicha entrada no afecta al valor de la actividad interna. Sin embargo, la realidad es que el hecho de que

w_{ki} sea positivo supone que, en el momento del aprendizaje, esta entrada obligó a aumentar en una unidad el parámetro r_i, lo que, en definitiva, significa un aumento del denominador de (4) y, en consecuencia, una disminución de la actividad. A este tipo de conexiones les llamaremos débilmente inhibidoras.

- *Caso 3.* $z_{ki} > \theta$ y $w_{ki} < 0$:
  Dada la forma del algoritmo de aprendizaje, la única posibilidad de que el peso sea negativo es que $w_{ki}=-1$, es decir, entrada nula durante la fase de aprendizaje, con lo cual quedaría:
  $$I_{ki} = 1 - |z_{ki} - (-1)| = -z_{ki}$$
  por lo que la contribución de esta entrada a la actividad de la neurona se considera inhibidora ya que el valor de I_{ki} estará comprendido en el intervalo $[-1,-\theta]$.

- *Caso 4.* $z_{ki} \leq \theta$ y $w_{ki} < 0$:
  En este caso, independientemente del valor del peso, la contribución a la actividad es cero. Además, el valor del peso indica que dicha entrada en el proceso de aprendizaje no contribuyó al valor de la constante de normalización r_i, con lo que el resultado final es que no excita ni inhibe, es decir, no altera el valor de la actividad interna de la neurona. Por esto, a estas conexiones les denominamos neutras.
  En la tabla 1 puede observarse un resumen del tipo de entrada y de su contribución a la actividad interna de la neurona en función de los valores numéricos de dicha entrada y del peso asociado a ella.

Tabla 1

Entrada	Peso	Tipo	Contribución
$z_{ki} > \theta$	$w_{ki} > 0$	excitadora	$+k,\ k \in [\theta,1]$
$z_{ki} \leq \theta$	$w_{ki} > 0$	débilmente inhibidora	0
$z_{ki} > \theta$	$w_{ki} < 0$	inhibidora	$-z_{ki}$
$z_{ki} \leq \theta$	$w_{ki} < 0$	neutra	0

5 LA RED

Una vez que disponemos de las funciones neuronales y de los algoritmos de aprendizaje, llega el momento de construir la red completa, para lo que necesitamos definir la estructura del interconexionado entre ellas y entre la red total y el mundo exterior, del que procederán las entradas primarias y al que se ofrecerán las salidas finales. Como en todas estas estructuras, nuestra red es del tipo multicapa, es decir, está formada por grupos de neuronas denominados capas encadenadas unas detrás de otras e identificadas por el funcionamiento síncrono de sus componentes. Cada una de estas capas estará compuesta por un determinado número de neuronas y pueden presentar conexionado hacia delante, interacción lateral y realimentación.

La conexión entre dos capas viene definida por las entradas de cada neurona de la capa destino que recibe de la capa origen. Esto suele hacerse mediante una distribución de conexiones tomando las salidas de las neuronas de la capa origen localizadas en un cierto entorno y el tamaño de dicho entorno. Utilizando una descripción geográfica, podemos imaginar las capas como planos yuxtapuestos, de forma que cada neurona viene localizada por sus coordenadas *(i,j)* en su plano. Así, las entradas a esa neurona proceden de las neuronas de la capa origen, que no tiene porque ser la capa anterior obligatoriamente, situadas alrededor de la misma vertical y a una distancia R del punto *(i,j)* de esa capa origen, y que constituyen el entorno antes citado.

Por otra parte, nuestras redes neuronales son de funcionamiento *off-line*, por lo que hay que distinguir entre la fase de aprendizaje y la fase de reconocimiento. En la fase de aprendizaje se presentan de forma secuencial el conjunto de prototipos a la red seleccionando, en la forma que más nos convenga, a aquellas neuronas que decidimos asociar a cada prototipo. En la fase de reconocimiento, ante la presencia de un patrón de entrada nuestra red produce una salida, formada por las salidas de un determinado grupo de neuronas, usualmente pertenecientes a la capa final o de salida, que contiene información sobre la posible pertenencia a todas y cada una de las clases definidas en la fase de aprendizaje, puesto que todas las neuronas de esa capa de salida, en mayor o menor medida, responden.

Nos encontramos, por tanto, ante una situación de clasificación difusa, donde la información de pertenencia viene dada por el número de neuronas de salida asociadas a cada clase que responden y a la intensidad de salida con que lo hacen. Para interpretar adecuadamente esta información, podemos definir una función de pertenencia de la siguiente forma: llamando n_j al número de neuronas de la capa de salida asociadas a la clase C_j, N_j al conjunto de los índices de dichas neuronas e y_k a la salida de la k-ésima neurona, se define el grado de pertenencia $\xi(x_i, C_j)$ (en tantos por ciento) de un patrón de entrada x_i a la clase C_j, mediante la ecuación:

$$\xi(x_i, C_j) = 100 \frac{\sum_{k \in N_j} y_k}{n_j} \qquad (7)$$

Para seleccionar, a partir de esta función, la clase a la que asignamos la entrada, y por motivos de sencillez, en este trabajo consideraremos que un patrón x_i pertenecerá a aquella clase C_j que maximice la cantidad $\xi(x_i, C_j)$.

Además para mejorar los resultados obtenidos en la fase de reconocimiento anteponemos a nuestra red neuronal una etapa de preprocesamiento que normaliza el tamaño, la posición y la orientación de los patrones de entrada a la red neuronal.

6 APLICACIÓN PRÁCTICA

Debido a lo expuesto en el apartado dedicado al conexionado de la red, el número de parámetros que definen nuestras redes neuronales es grande por lo que el número de redes neuronales que podemos implementar será prácticamente infinito, esto hace que a la hora de abordar un caso práctico como puede ser el reconocimiento de caracteres manuscritos o cualquier otro sea necesario, en primer lugar, definir cuales son las arquitecturas neuronales más idóneas para la implementación de nuestro modelo. Para llevar a cabo este estudio, hemos definido una figura de mérito η que hemos denominado "grado de reconocimiento de la red" y que se calcula en función de un determinado número de caracteres que van a formar parte de la fase de reconocimiento (figura 1).

Definimos el Grado de Reconocimiento η como el producto de dos factores identificativos del proceso: el Indice de Separabilidad i_s y el Grado de Acierto g_a.

$$\eta = i_S \times g_a \qquad (8)$$

Definimos el Grado de Acierto de la red g_a, como el cociente entre el número de patrones reconocidos n_r y el número de patrones utilizados en la etapa de reconocimiento n. Este cociente se eleva al cuadrado para penalizar más el hecho de que la red no clasifique de forma correcta determinados patrones.

$$g_a = \left(\frac{n_r}{n}\right)^2 \qquad (9)$$

Por otro lado, no sólo pretenderemos que una determinada red neuronal reconozca todos los patrones de entrada, sino también que los reconozca con suficiente claridad, es decir, que el grado de pertenencia $\xi(x_l, C_j)$ de un determinado patrón de entrada a su clase sea lo mayor posible en relación con los grados de pertenencia a las demás clases establecidas en el proceso de aprendizaje. Para medir esta "claridad" en el reconocimiento, definimos el índice de separabilidad i_s

$$i_s = \frac{\sum_{l=1}^{n} \xi(x_l, C_l)}{\sum_{j=1}^{N} \sum_{l=1}^{n} \xi(x_l, C_j)} \qquad (10)$$

donde N es el número de clases establecidas en el aprendizaje.

El caso más favorable corresponde a que la red neuronal reconozca todos los patrones de entrada ($g_a=1$) y que el grado de pertenencia de los patrones a las clases a las que no pertenecen sean todos cero ($i_s=1$) por lo que el grado de reconocimiento de la red valdría la unidad.

Una vez definida la figura de mérito para medir la calidad del reconocimiento, vamos a tratar de mejorarla. En primer lugar parece lógico pensar que a mayor tamaño de la red (número de conexiones entre las neuronas que la integran) mejor reconocimiento. Si lo anterior es cierto, nos encontramos ante el dilema de decidir cual será la forma más rentable de llevar a cabo este aumento: incrementando el número de entradas a cada neurona, aumentando el número de capas ó introduciendo en la red la interacción lateral y la realimentación.

Como ejemplo práctico, para el cálculo del grado de reconocimiento se han establecido 25 clases, correspondientes a las 25 letras del abecedario, y se han tomado los patrones de la figura 1 en la fase de reconocimiento. El estudio de la respuesta de la red se ha hecho individualizando su análisis para cada uno de los factores citados, tal como se describe a continuación.

6.1 GRADO DE RECONOCIMIENTO DE LA RED FRENTE AL NÚMERO DE ENTRADAS DE LA NEURONA.

Para acometer este estudio hemos implementado redes neuronales monocapa (ya que todavía no sabemos si el aumento del número de capas será aconsejable o no) con y sin interacción lateral y en cada caso con 50, 100, 200 y 400 entradas a cada neurona. En la tabla 2 se pueden observar los valores del grado de reconocimiento de la red (en función de los patrones de la figura 1) y el tanto por ciento de patrones reconocidos en función del número de entradas a cada neurona para una red de una capa con y sin interacción lateral.

Tabla 2

	Con int. lateral		Sin int. lateral	
N. de entradas	η	Reconocidos (%)	η	Reconocidos (%)
400	0.572	100	0.223	100
200	0.505	100	0.208	100
100	0.424	100	0.184	100
50	0.257	91.6	0.121	91.6

En la figura 2, se puede observar la representación gráfica del grado de reconocimiento de la red en función del número de entradas por neurona para los casos con interacción lateral y sin ella. Podemos ver como, en ambos casos, el grado de reconocimiento de la red aumenta al aumentar el número de entradas a cada neurona y cómo este aumento es brusco al

principio para luego tender a la estabilidad. También se pone de manifiesto que a igualdad en el número de conexiones, la red monocapa con interacción lateral presenta mejores resultados que la red sin interacción lateral, lo que por otra parte era esperable y lógico. Por otro lado, este aumento del número de entradas a cada neurona conlleva un irremediable aumento en la duración de la fase de aprendizaje, por lo que buscaremos un punto óptimo en el que se aumente en la medida de lo posible al grado de reconocimiento de la red minimizando la duración de la fase de aprendizaje.
Si observamos la figura 2, podemos ver que el punto óptimo para redes monocapa sin interacción lateral está entorno a las 100 entradas por neurona, ya que un aumento del número de entradas a partir de este punto, no tiene prácticamente influencia sobre el grado de reconocimiento de la red, mientras que la tiene en la duración de la fase de aprendizaje. Para redes con interacción lateral también habría un punto óptimo, que estaría más a la derecha, en torno a las 200 entradas por neurona.

6.2 GRADO DE RECONOCIMIENTO DE LA RED FRENTE AL TAMAÑO DEL CAMPO RECEPTIVO DE LA NEURONA.

Para acometer este estudio hemos implementado una red neuronal de 1 capa sin interacción lateral, con 200 entradas a cada neurona y se han hecho simulaciones para tamaños del campo receptivo de radios 7 (radio mínimo para albergar 200 conexiones), 15, 25 y 50.
En la tabla 3, se puede observar el grado de reconocimiento de la red (en función de los patrones de la figura 1) y el tanto por ciento de patrones reconocidos para cada una de las anteriores simulaciones y en la figura 3 hemos representado dicho grado de reconocimiento en función del radio del tamaño del campo receptivo de la neurona. En dicha gráfica se pone de manifiesto el aumento del grado de reconocimiento de la red frente al aumento del radio del campo receptivo de la neurona.

Tabla 3

Radio del campo receptivo	η	Reconocidos (%)
50	0.208	100
25	0.123	91.6
15	0.091	83.3
7	0.031	58.3

A diferencia de lo que ocurría en el caso anterior, el aumento del tamaño del campo receptivo de la neurona no tiene influencia en la duración de la fase de aprendizaje, por lo que siempre utilizaremos el máximo posible.

Como conclusión a los dos últimos apartados podemos decir que nuestro modelo neuronal en redes monocapa se comporta tanto mejor cuanta más información tenga cada neurona del patrón de entrada y lo realmente interesante es que esta información sea una información global de todo el patrón de entrada.

6.3 GRADO DE RECONOCIMIENTO DE LA RED FRENTE AL NÚMERO DE CAPAS.

Para este apartado, se han implementado redes neuronales de 1, 2 y 4 capas y, en cada caso, se ha distinguido entre conexionado hacia delante exclusivamente, con interacción lateral y cuando, además, presenta realimentación. En la tabla 4 se representa el grado de reconocimiento de la red (en función de los patrones de la figura 1) y el tanto por ciento de patrones reconocidos para cada una de las situaciones y en la figura 4 se representa la variación del grado de reconocimiento de la red frente al número de capas para redes únicamente de conexionado hacia delante, con interacción lateral y con interacción lateral y realimentación.

Tabla 4

	Feed-forward		F-f + int. lateral		F-f + int. Lateral + feed-back	
N. capas	η	Reconocidos (%)	η	Reconocidos (%)	η	Reconocidos (%)
1	0.184	100	0.424	100	-	-
2	0.269	91.6	0.320	100	0.308	100
4	0.403	91.6	0.222	83.3	0.191	83.3

Ahora, podemos observar que para redes multicapa de conexionado hacia delante, el grado de reconocimiento de la red aumenta al aumentar el número de capas de la misma, cosa que no ocurre en los otros dos casos, lo que nos obliga a que se limite la interacción lateral únicamente a redes monocapas.

6.4 ELECCIÓN DE LA ARQUITECTURA IDÓNEA.

En los anteriores apartados, vimos que el grado de reconocimiento de nuestra red aumenta en las siguientes arquitecturas:

- Redes monocapa sin interacción lateral aumentando el número de entradas a cada neurona.
- Redes monocapa con interacción lateral aumentando el número de entradas a cada neurona.
- Redes multicapa de conexionado hacia adelante aumentando el número de capas.

Para escoger la configuración óptima, en la figura 5 hemos representado el grado de reconocimiento para

las arquitecturas anteriores frente al número de conexiones totales entre las neuronas. En dicha gráfica, puede observarse que, a igual número de conexiones, los mejores resultados se obtienen para una red monocapa con interacción lateral. En segundo lugar, está la red multicapa de conexionado únicamente hacia delante y, por último, se encuentra la red monocapa sin interacción lateral. Esta última arquitectura, a pesar de presentar los peores resultados, es aconsejable en aquellas situaciones en las que se requiera un tiempo de ejecución mínimo.
En definitiva, si lo que importa es conseguir buenos resultados en el aprendizaje sin importar demasiado el tiempo de ejecución es aconsejable utilizar una red monocapa con interacción lateral y con un número entre 200 y 400 entradas a cada neurona (punto óptimo obtenido en el apartado 6.1), y si lo que importa es realizar la clasificación en un tiempo mínimo conviene utilizar una red monocapa sin interacción lateral con 100 entradas por neurona (punto óptimo obtenido en el apartado 6.1)

7 RESULTADOS OBTENIDOS

Para probar el modelo y las concusiones anteriores, lo hemos aplicado al reconocimiento de caracteres manuscritos. Debido a la escasa resolución espacial del NIST, hemos generado nuestra propia base de datos con 400 caracteres numéricos manuscritos correspondientes a 10 escritores diferentes. Estos caracteres están digitalizados con una resolución espacial 100x100 y a dos niveles de grises, y se han precedido de un preprocesador que eliminaba las distorsiones geométricas. Para el reconocimiento, utilizamos una red monocapa sin interacción lateral de 10.000 neuronas con 100 entradas cada una. Dicha red alcanzó un 93% de aciertos con un tiempo de reconocimiento por debajo de 1 segundo y un tiempo de aprendizaje de 1 segundo en un sistema multiprocesador con 8 CPUs Motorola de la serie 68030.

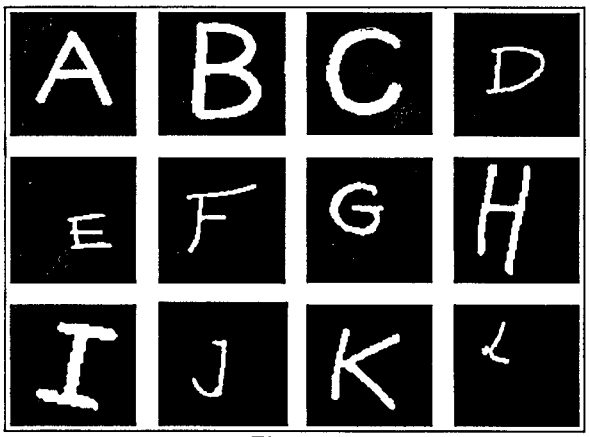


Figura 1

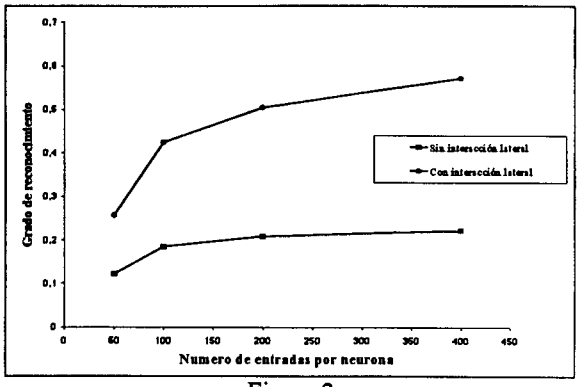


Figura 2

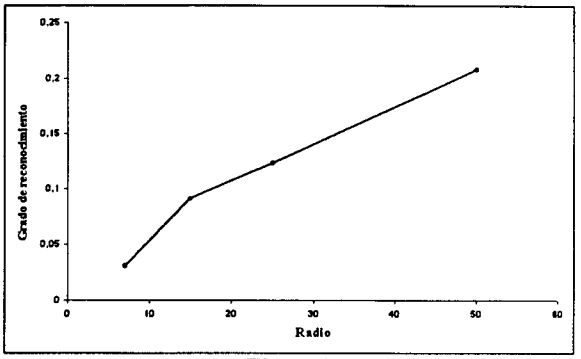


Figura 3

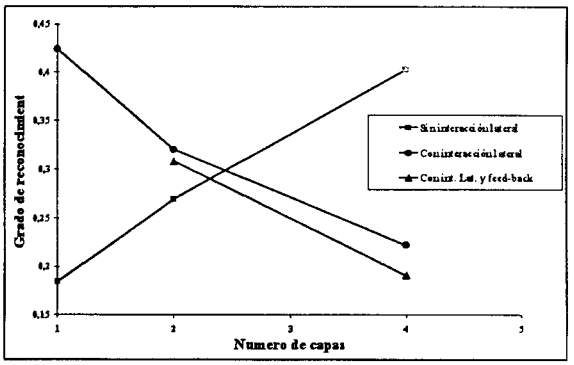


Figura 4

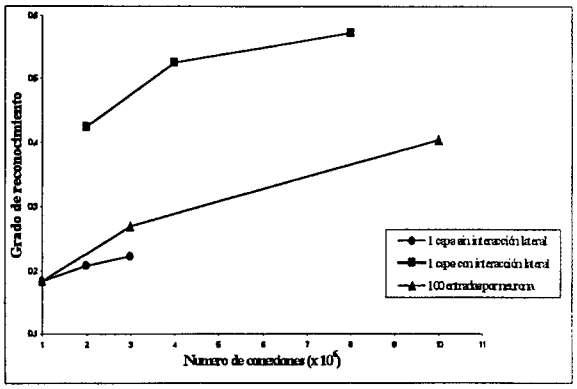


Figura 5

8 CONCLUSIÓN

El aspecto más destacable de todo lo que hemos expuesto radica, a nuestro entender, en dos puntos clave del reconocimiento de patrones: En primer lugar, la gran rapidez de ejecución de los procesos de entrenamiento y clasificación y, en segundo lugar, la idoneidad mostrada por cada una de las topologías en cada caso práctico.

Otro aspecto, no descrito en el trabajo pero igualmente consecuencia de los modelos planteados, es una especial adaptación a la construcción de grandes redes neuronales y una gran resistencia a la degradación de un buen número de sus componentes. Con este modelo neuro-fuzzy, el algoritmo LVQ originario puede alcanzar elevadas cuotas de acierto con unos mínimos requerimientos de tiempo, lo que lo hace idóneo para la implementación de redes dinámicas.

Agradecimientos

Los autores desean hacer constar su agradecimiento a la C.I.C.Y.T. y a la Junta de Extremadura por su ayuda a través de la financiación de los Proyectos TIC 97-0268 y PRI 96-06D007, respectivamente, gracias a los cuales a podido ser llevado a cabo este trabajo.

Referencias

[1] Barto, A.G."Reinforcement learning and adaptive critic methods". Handbook of Intelligent Control. pp 469-491. Van Nostrand-Reinhold. 1992.

[2] Bishop, C.M. Neural Networks for Pattern Recognition. Oxford University Press, 1995.

[3] Jang, J., Sun, T. y Mizutani, E."Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence". Prentice Hall, 1997.

[4] Kohonen, T. "Learning Vector Quantization". Abstr. of the First An.l INNS Meeting. 1988.

[5] López Aligué, F.J. "A Neural model for Pattern Recognition". Artificial Intelligence in Engineering. Computational Mechanics Pub. Southampton, pp 395-402, 1995.

Agregación de Intervalos Borrosos mediante el Principio de Extensión*

Adolfo R. de Soto
Area de Lenguajes y Sistemas Informáticos
Universidad de León
ddears@unileon.es

Enric Trillas
Dpto. de Inteligencia Artificial
Universidad Politécnica de Madrid
etrillas@fi.upm.es

Resumen

El principio de extensión puede utilizarse para obtener funciones de agregación sobre la clase de intervalos borrosos y respecto al orden reticular definido por las operaciones MAX, MIN obtenidas, a su vez, mediante la extensión de los operadores max y min . En el trabajo se utiliza además la representación de tipo LR para intervalos borrosos mediante funciones de forma, para calcular la extensión de algunas medidas de agregación.

Palabras Clave: Agregación, Intervalos Borrosos, Variables Lingüísticas

1 Introducción

Algunas etiquetas lingüísticas pueden considerarse como valores medios de otras, por ejemplo, *regular* puede ser la etiqueta "media" de *mal* y *bien; mediano* (la palabra misma lo dice) respecto a *alto* y a *bajo*, o respecto a *corto* y *largo*. Las personas somos capaces de realizar esta "agregación" de etiquetas lingüísticas en nuestro quehacer diario; por ejemplo, un profesor puede valorar un examen con las etiquetas básicas *mal*, *regular*, *bien* y añadiendo luego algunas otras mediante modificaciones de estas como *regular alto*, *bien bajo*, etc, para cubrir mejor la variedad de resultados. La pregunta es cómo modelizar la agregación de 5 o 6 valoraciones lingüísticas de este tipo.

En este trabajo se pretende presentar un método de agregación de etiquetas lingüísticas mediante el principio de extensión [7], con el que se obtiene realmente una función de agregación entre etiquetas lingüísticas. Para modelizar las etiquetas se utilizan intervalos borrosos, un tipo de conjuntos borrosos definidos sobre la recta real que incluyen a todos los conjuntos borrosos utilizados habitualmente para representar etiquetas lingüísticas.

Trabajo financiado parcialmente por la CICYT bajo el proyecto: TIC96-1393-C06.

2 Intervalos borrosos y el principio de extensión

Un intervalo borroso μ es un conjunto borroso convexo sobre la recta real, es decir que verifica, para todo $x, y, z \in I\!R$ con $x \leq z \leq y$, la desigualdad

$$\mu(z) \geq \min(\mu(x), \mu(y)),$$

y normal e.d. tal que existe $x \in I\!R$ con $\mu(x) = 1$. Un intervalo borroso es convexo si y sólo si todos sus α-cortes

$$[\mu]_\alpha = \{x \in I\!R \ : \ \mu(x) \geq \alpha\}$$

son intervalos convexos de la recta real. Un intervalo borroso semicontinuo superiormente [3] verifica que todos sus α-cortes son intervalos cerrados. Denotaremos por $\mathfrak{F}(I\!R)$ al conjunto de todos los intervalos borroso semicontinuos superiormente. De acuerdo a esta definición, todos los α-cortes de un intervalo borroso son de la forma $[a, b], [a, +\infty), (-\infty, a]$ y $(-\infty, +\infty)$ con $a, b \in I\!R$. La terminología para números borrosos no esta muy unificada, algunos autores (ver por ejemplo [6]) utilizan el concepto de número borroso para referirse a lo que aquí denominamos intervalo borroso y para otros (p. e. [5], [4]) un número borroso es un elemento de $\mathfrak{F}(I\!R)$ con soporte acotado y un único valor modal, es decir, un único punto donde vale 1.

Utilizaremos el concepto de intervalo borroso para capturar los tres tipos de subconjuntos borrosos μ que se utilizan como etiquetas lingüísticas habitualmente: los conjuntos borrosos de tipo Z, que son aquellos para los que existe un $x_0 \in I\!R$ tal que $\mu(x) = 1$ para todo $x \leq x_0$ y son no crecientes en el intervalo $[x_0, +\infty)$; los de tipo S, para los que existe un x_0 tal que $\mu(x) = 1$ para todo $x_0 \leq x$ y son no decrecientes en el intervalo $(-\infty, x_0]$; y los de tipo campana, para los que existen $x_0, x_1 \in I\!R$ de tal forma que son no decrecientes

en $(-\infty, x_0]$, no crecientes en $[x_1, +\infty)$ y valen 1 en $[x_0, x_1]$.

Dada una función $f : I\!R^n \to I\!R$, el principio de extensión [7] permite construir una nueva función $F : \mathfrak{F}(I\!R)^n \to \mathfrak{F}(I\!R)$ mediante la expresión

$$F(\mu_1, \ldots, \mu_n)(y) = \sup_{f(x_1,\ldots,x_n)=y} \min(\mu_1(x_1), \ldots, \mu_n(x_n)),$$

y $F(\mu_1, \ldots, \mu_n)(y) = 0$ cuando y no tiene antiimagen por f.

A continuación damos algunos resultados ya conocidos que necesitaremos posteriormente. Designaremos por $\leq_\Pi$ al orden producto, o puntual de $I\!R^n$; es decir al definido como $(x_1, \ldots, x_n) \leq_\Pi (y_1, \ldots, y_n)$ si y sólo si $x_1 \leq y_1, \ldots, x_n \leq y_n$.

Teorema 1 *[2]Sean μ y η dos intervalos borroso tales que para todo $\alpha \in (0, 1]$, $[\mu]_\alpha \neq I\!R$ y $[\eta]_\alpha \neq I\!R$. Sea f una función, continua y monótona no decreciente respecto al orden producto $\leq_\Pi$, es decir:*

$$\forall (x, y) \leq_\Pi (x', y'),\ f(x, y) \leq f(x', y'),$$

entonces

$$[F(\mu, \eta)]_\alpha = f([\mu]_\alpha, [\eta]_\alpha),$$

donde $f([\mu]_\alpha, [\eta]_\alpha) = \{z \in I\!R : \exists x \in [\mu]_\alpha, \exists y \in [\eta]_\alpha,\ f(x, y) = z\}$.

El teorema anterior prueba que $\mathfrak{F}(I\!R)$ es cerrado bajo la extensión de cualquier función f contínua y monótona no decreciente respecto al orden producto. El teorema también es válido cuando f tiene un número finito de argumentos.

Mediante el principio de extensión se pueden "extender" las operaciones aritméticas habituales dando lugar a una aritmética borrosa [4] muy relacionada con la aritmética de intervalos. También pueden extenderse las operaciones min y max mediante:

$$\begin{aligned} & MIN(\mu_1, \ldots, \mu_n)(y) \\ = & \sup_{\min(x_1,\ldots,x_n)=y} \min(\mu_1(x_1), \ldots, \mu_n(x_n)) \\ & MAX(\mu_1, \ldots, \mu_n)(y) \\ = & \sup_{\max(x_1,\ldots,x_n)=y} \min(\mu_1(x_1), \ldots, \mu_n(x_n)). \end{aligned}$$

Teorema 2 *$\mathfrak{F}(I\!R)$ es un retículo distributivo con las operaciones MIN y MAX definidas previamente.*

La demostración de este teorema puede verse en muchas referencias para números borrosos, como p.e. en [4, 5], pero cualquiera de las demostraciones del teorema que aparecen en las referencias son válidas también para intervalos borrosos no acotados por un extremo, como, por ejemplo, los de tipo Z o S.

El retículo $\mathfrak{F}(I\!R)$ tiene el orden reticular $\preceq$ definido de la manera habitual. Si $\mu, \eta \in \mathfrak{F}(I\!R)$:

$$\mu \preceq \eta \Leftrightarrow MIN(\mu, \eta) = \mu \Leftrightarrow MAX(\mu, \eta) = \eta.$$

Este orden queda caracterizado por el siguiente teorema [6]:

Teorema 3 *Sean $\mu, \eta \in \mathfrak{F}(I\!R)$, es $\mu \preceq \eta$, si y sólo si*

$$\begin{aligned} \inf[\mu]_\alpha &\leq \inf[\eta]_\alpha,\ y \\ \sup[\mu]_\alpha &\leq \sup[\eta]_\alpha \text{ para todo } \alpha \in I\!R. \end{aligned}$$

El orden $\preceq$ ordena linealmente las etiquetas lingüísticas *muy bajo* $\preceq$ *bajo* $\preceq$ *mediano* $\preceq$ *alto* $\preceq$ *muy alto* si se definien de la forma habitual a como se hace en control borroso, pero, en general, es sólo un orden parcial.

3 Función de agregación inducida para intervalos borrosos

Dado un retículo $(R, \leq)$, una función $f : R^n \to R$ es una función de agregación n-dimensional si

1. f es idempotente, es decir, $f(x, \ldots, x) = x$ para todo $x \in I\!R$.
2. f es monótona respecto del orden producto $\leq_\Pi$ de R^n.

Mediante el principio de extensión se puede "extender" funciones de agregación a intervalos borroso obteniéndose funciones de agregación n-dimensionales sobre $\mathfrak{F}(I\!R)^n$.

Teorema 4 *Sea $f : I\!R^n \to I\!R$ una función continua, monótona no decreciente respecto del orden producto e idempotente; la función $F : \mathfrak{F}(I\!R)^n \to \mathfrak{F}(I\!R)$, inducida mediante el principio de extensión a partir de f, es una función de agregación sobre el retículo $\mathfrak{F}(I\!R)$ de los intervalos borrosos.*

Demostración. Veamos en primer lugar la idempotencia de F :

$$\begin{aligned} F(\mu, \ldots, \mu)(y) &= \sup_{f(x_1,\ldots,x_n)=y} \min(\mu(x_1), \ldots, \mu(x_n)) \\ &\geq \min(\mu(y), \ldots, \mu(y)) \\ &= \mu(y), \end{aligned}$$

por ser f idempotente. Por otra parte si $x_1, \ldots, x_n \in I\!R$, sean $m = \min(x_1, \ldots, x_n)$ y $M = \max(x_1, \ldots, x_n)$, y pongamos $y = f(x_1, \ldots, x_n)$. Por ser f una función de agregación es $m \leq y \leq M$ y, por ser μ convexa, $\mu(y) \geq \min(\mu(m), \mu(M))$. Ahora

bien, como se verifica que $m \leq x_i \leq M$ para todo i, tendremos que $\mu(x_i) \geq \min(\mu(m), \mu(M))$ y así $\min(\mu(x_1), \ldots, \mu(x_n)) \geq \min(\mu(m), \mu(M))$. Como $\min(\mu(x_1), \ldots, \mu(x_n)) \leq \min(\mu(m), \mu(M))$ resultará que $\min(\mu(x_1), \ldots, \mu(x_n)) = \min(\mu(m), \mu(M))$. Por tanto,

$$\begin{aligned} F(\mu, \ldots, \mu)(y) &= \sup_{f(x_1,\ldots,x_n)=y} \min(\mu(x_1), \ldots, \mu(x_n)) \\ &= \sup_{f(x_1,\ldots,x_n)=y} \min(\mu(m), \mu(M)) \\ &\leq \mu(y). \end{aligned}$$

Veamos ahora la monotonía respecto del orden $\preceq_\Pi$. Sean $\mu, \eta \in \mathfrak{F}(\mathbb{R})$ dos intervalos borroso con $\mu \preceq \eta$, para probar que F es monótona no decreciente es suficiente mostrar que

$$F(\mu, \mu_2, \ldots, \mu_n) \preceq F(\eta, \mu_2, \ldots, \mu_n)$$

es válido para cualesquiera $\mu_2, \ldots, \mu_n \in \mathfrak{F}(\mathbb{R})^n$, puesto que para las otras componentes la prueba es análoga. Debido al teorema 3, será suficiente probar que

$$\begin{aligned} \sup[F(\mu, \mu_2, \ldots, \mu_n)]_\alpha &\leq \sup[F(\eta, \mu_2, \ldots, \mu_n)]_\alpha, \text{ y} \\ \inf[F(\mu, \mu_2, \ldots, \mu_n)]_\alpha &\leq \inf[F(\eta, \mu_2, \ldots, \mu_n)]_\alpha, \end{aligned}$$

para todo $\alpha \in [0, 1]$. Pero por el teorema 1 se verifica

$$[F(\mu, \mu_2, \ldots, \mu_n)]_\alpha = F([\mu]_\alpha, [\mu_2]_\alpha, \ldots, [\mu_n]_\alpha).$$

Sea $y = f(x, x_2, \ldots, x_n)$ un punto con $x \in [\mu]_\alpha$ y $x_i \in [\mu_i]_\alpha$ para $i = 2, \ldots, n$, como $\sup[\mu]_\alpha \leq \sup[\eta]_\alpha$, existirá $x' \in [\eta]_\alpha$ con $x \leq x'$, por lo que $y \leq y' = f(x', x_2, \ldots, x_n)$ puesto que f es una función monótona no decreciente respecto del producto. Asi que para cada $y \in [F(\mu, \mu_2, \ldots, \mu_n)]_\alpha$, existe un $y' \in [F(\eta, \mu_2, \ldots, \mu_n)]_\alpha$ tal que $y \leq y'$ y consecuentemente

$$\sup[F(\mu, \mu_2, \ldots, \mu_n)]_\alpha = \sup[F(\eta, \mu_2, \ldots, \mu_n)]_\alpha.$$

La parte con el ínfimo es análoga. ■

4 Intervalos borroso de tipo LR

La siguiente representación de intervalos borrosos [2] incluye todos los subconjuntos borrosos utilizados habitualmente como etiquetas lingüísticas de una variable lingüística.

Sea $L : [0, \infty) \to [0, 1]$ una función continua tal que:

1. $L(0) = 1$;
2. L es estrictamente decreciente salvo en $L^{-1}(0)$
3. $L(1) = 0$ o ($L(u) > 0$ y $\lim_{u\to\infty} L(u) = 0$);
4. $\forall u < 1$, $L(u) > 0$.

L recibe el nombre de función de forma. Algunos ejemplos de funciones de forma son $\max(1 - x^p, 0)$, $\max(1 - x, 0)^p$, e^{-x}, e^{-x^2}, etc.

Consideraremos la clase de intervalos borrosos $\mathfrak{F}_{LR}(\mathbb{R})$ semicontinuos superiormente dados por la representación siguiente: $\mu \in \mathfrak{F}_{LR}(\mathbb{R})$ si existen dos funciones de forma L, R y cuatro parámetros $(\underline{\mu}, \overline{\mu}, \alpha_\mu, \beta_\mu)$ con $\underline{\mu} \leq \overline{\mu}$ y $\alpha_\mu, \beta_\mu > 0$, tal que

$$\mu(x) = \begin{cases} L\left(\frac{\underline{\mu} - x}{\alpha_\mu}\right) & \text{si } x \leq \underline{\mu} \\ 1 & \text{si } x \in [\underline{\mu}, \overline{\mu}] \\ R\left(\frac{x - \overline{\mu}}{\beta_\mu}\right) & \text{si } x \geq \overline{\mu}. \end{cases}$$

En ese caso escribiremos $\mu = (\underline{\mu}, \overline{\mu}, \alpha_\mu, \beta_\mu)_{LR}$. Esta representación cubre todos los intervalos borrosos cuyas funciones de pertenencia son tales que

$$\lim_{x\to-\infty} \mu(x) \in \{0, 1\}, \quad \lim_{x\to+\infty} \mu(x) \in \{0, 1\},$$

asi como todos los intervalos borrosos con soporte acotado y, por tanto, cubre a los conjuntos borrosos de tipo S, los de tipo Z y los que son en forma de campana. Como ejemplo, si μ es un trapecio borroso, entonces $L = R = \max(0, 1 - x)$ y $(\underline{\mu} - \alpha_\mu, \overline{\mu} - \beta_\mu)$ es su soporte. Adoptaremos como convención:

- Si μ es un intervalo borroso de tipo S, entonces $\mu = (\underline{\mu}, +\infty, \alpha_\mu, +\infty)_{LR}$, para cualquier R.
- Si μ es un intervalo borroso de tipo Z, entonces $\mu = (-\infty, \overline{\mu}, -\infty, \beta_\mu)_{LR}$, para cualquier L.

Es posible calcular el intervalo borroso resultante de aplicar, por ejemplo, la media aritmética $ma(x, y) = \frac{x+y}{2}$ extendida a dos intervalos borrosos $\mu = (\underline{\mu}, \overline{\mu}, \alpha_\mu, \beta_\mu)_{LR}$ y $\eta = (\underline{\eta}, \overline{\eta}, \alpha_\eta, \beta_\eta)_{LR}$. Para ello utilizaremos el siguiente resultado [2, 5]. Por μ^+ y η^+ se denota a las restricciones de μ y η a $(-\infty, \underline{\mu}]$ y a $(-\infty, \underline{\eta}]$ respectivamente, y por μ^- y η^- sus restricciones a $[\overline{\mu}, +\infty)$ y a $[\overline{\eta}, +\infty)$.

Teorema 5 *Sean $\mu, \eta \in \mathfrak{F}(\mathbb{R})$ sin partes constantes fuera de 0 y de 1 y tales que sus α-cortes $[\mu]_\alpha = [\underline{\mu}_\alpha, \overline{\mu}_\alpha]$, $[\eta]_\alpha = [\underline{\eta}_\alpha, \overline{\eta}_\alpha]$ son conjuntos compactos para todo $\alpha \in (0, 1]$. Dada una función $f : \mathbb{R}^2 \to \mathbb{R}$ continua y estrictamente creciente respecto al orden producto (si $x > x'$ e $y > y'$, entonces $f(x, y) > f(x', y')$), el intervalo borroso $F(\mu, \eta)$ obtenido mediante el principio de extensión puede calcularse resolviendo las dos ecuaciones siguientes*

$$(F(\mu, \eta)^+)^{-1} = f((\mu^+)^{-1}, (\eta^+)^{-1}) \quad (1)$$
$$(F(\mu, \eta)^-)^{-1} = f((\mu^-)^{-1}, (\eta^-)^{-1}). \quad (2)$$

Por tanto, para aquellos elementos de $\mathfrak{F}_{LR}(\mathbb{R})$ cuyos α-cortes, para $\alpha \in (0,1]$, sean intervalos acotados, $F(\mu,\eta)$ puede calcularse separadamente su parte creciente y su parte decreciente, puesto que las funciones de forma L y R son estrictamente decrecientes. Para la media aritmética y para la parte creciente de μ y η, para todo $y \in (0,1]$ existen dos únicos números reales x_μ e x_η tales que

$$L\left(\frac{\underline{\mu} - x_\mu}{\alpha_\mu}\right) = y = L\left(\frac{\underline{\eta} - x_\eta}{\alpha_\eta}\right) \tag{3}$$

lo que es equivalente a

$$\begin{aligned} x_\mu &= \underline{\mu} - \alpha_\mu L^{-1}(y), \\ x_\eta &= \underline{\eta} - \alpha_\eta L^{-1}(y). \end{aligned}$$

ecuaciones que son las equivalentes a (1) y (2). Por tanto para $ma(x,y) = \frac{x+y}{2}$ se tendrá que

$$z = \frac{x_\mu + x_\eta}{2} = \frac{\underline{\mu} + \underline{\eta}}{2} - \frac{\alpha_\mu + \alpha_\eta}{2} L^{-1}(y)$$

de lo que se deduce que

$$L\left(\frac{z - \frac{\underline{\mu}+\underline{\eta}}{2}}{\frac{\alpha_\mu+\alpha_\eta}{2}}\right) = y.$$

Haciendo lo mismo para la parte decreciente se obtiene que

$$MA(\mu,\eta) = \left(\frac{\underline{\mu}+\underline{\eta}}{2}, \frac{\overline{\mu}+\overline{\eta}}{2}, \frac{\alpha_\mu+\alpha_\eta}{2}, \frac{\beta_\mu+\beta_\eta}{2}\right)_{LR}. \tag{4}$$

El último resultado no sólo muestra la forma de calcular la media aritmética extendida sobre intervalos borroso LR, sino que además demuestra que la media aritmética extendida conserva la forma de los conjuntos borrosos a los que se aplica. Así la media aritmética extendida de trapecios con soporte acotado, será un trapecio agregado de estos dos. Además, si $\alpha_\mu = \alpha_\eta = \alpha$ y $\beta_\mu = \beta_\eta = \beta$, entonces la media aritmética extendida de dos intervalos borroso de tipo LR tendrá $\alpha_{MA(\mu,\eta)} = \alpha$ y $\beta_{MA(\mu,\eta)} = \beta$.

Cuando μ o η son del tipo S o del tipo Z, su soporte no es acotado y es evidente que la existencia del elemento x_μ o del elemento x_η de la ecuación (3) no tiene sentido. No obstante la solución dada por (4) sigue siendo válida para cuando alguno de los parámetros de μ o η es $\pm\infty$. Por ejemplo si $\mu = (-\infty, \overline{\mu}, -\infty, \beta_\mu)_{LR}$ y $\eta = (\underline{\eta}, \overline{\eta}, \alpha_\eta, \beta_\eta)_{LR}$ entonces es evidente que $MA(\mu,\eta)(z) = \sup_{z=ma(x,y)} \min(\mu(x), \eta(y)) = 1$ para todo $y \in (-\infty, \frac{\overline{\mu}+\overline{\eta}}{2}]$ puesto que si $y \leq \frac{\overline{\mu}+\overline{\eta}}{2}$, entonces, tomando $x = 2y - \overline{\eta}$, tendremos que es $y = \frac{x+\overline{\eta}}{2}$ con $x \leq \overline{\mu}$ por lo que $\min(\mu(x), \eta(\overline{\eta})) = 1$. Los demás casos son análogos.

Algunas consecuencias de esto son:

1. La agregación con la media aritmética extendida de un intervalo borroso de tipo S o de tipo Z con uno en forma de campana, es un intervalo borroso de tipo S y tipo Z respectivamente. Por ejemplo si $\mu = (\underline{\mu}, +\infty, \alpha_\mu, +\infty)_{LR}$ y $\eta = (\underline{\eta}, \overline{\eta}, \alpha_\eta, \beta_\eta)_{LR}$, entonces $MA(\mu,\eta) = (\frac{\underline{\mu}+\underline{\eta}}{2}, +\infty, \frac{\alpha_\mu+\alpha_\eta}{2}, +\infty)$. En el caso en el que los parámetros α y β sean constantes, este resultado admite una interpretación en términos de modificadores lingüísticos puesto que $MA(\mu,\eta)$ es una traslación de μ [1]. El resultado de agregar, por ejemplo, *alto* (un predicado representado habitualmente por un conjunto borroso de tipo Z) *y mediano* (un predicado en forma de campana) sería *bastante alto,* si se entiende por esta expresión una debilitación de *alto*, es decir, algo así como *casi alto.*

2. La agregación mediante la media aritmética extendida de un intervalo borroso del tipo Z con uno de tipo S es $\mathbb{R}$. Ello puede considerarse realmente un caso patológico, teniendo en cuenta que siempre se tiene la desigualdad $\mu_Z \preceq \mathbb{R} \preceq \mu_S$, cuando μ_Z es un conjunto borroso de tipo Z y μ_S es un conjunto borroso de tipo S. Parece adecuado pensar que la agregación de una etiqueta de tipo Z con una agregación de una etiqueta de tipo S debería ser la etiqueta en forma de campana intermedia de las dos, al menos para el caso en el que estas dos etiquetas no tuviesen ningún punto de intersección.

Referencias

[1] **De Soto, A.R.; Trillas, E.; Herrero, M.J.** (1997) "Antónimos y Modificadores Lingüísticos". *Actas del VII Congreso Español sobre Tecnologías y Lógica Fuzzy*, pp. 7–14, Tarragona.

[2] **Dubois, D.; Prade, H.** (1993) "Fuzzy Numbers: An overview", *Readings in Fuzzy Sets for Intelligent Systems,* Vol. I, pp. 3–39, Morgan Kauffman, San Mateo, CA.

[3] *Encyclopedic Dictionary of Mathematics*, MIT Press, 1980.

[4] **Klir, G. J.; Yuan, B.** (1995) *Fuzzy Sets and Fuzzy Logic: Theory and Applications.* Prentice-Hall, Inc.; New York.

[5] **Novák, V.** (1989) *Fuzzy Sets and Their Applications*, Adam Hilger, Bristol.

[6] **Ramík, J.; Rímánek, J.** (1985) "Inequality Relation between Fuzzy Numbers and its use in Fuzzy Optimization", *borroso Sets and Systems* 16, pp. 123-138.

[7] **Zadeh, L. A.** (1965) "Fuzzy Sets", *Information and Control*, Vol. 8, pp. 338–353.

NOTA SOBRE EL CONCEPTO DE IMPLICACIÓN BORROSA*

Enric Trillas
Dep. Inteligencia Artificial
Univ. Politécnica de Madrid
email:etrillas@fi.upm.es

Cristina del Campo
Dep. Estadística e I.O. II
Univ. Complutense de Madrid
e-mail: campocc@ccee.ucm.es

Susana Cubillo
Dep. Matemática Aplicada
Univ. Politécnica de Madrid
e-mail: scubillo@fi.upm.es

Resumen

Tradicionalmente, en la lógica borrosa tanto se ha considerado que la implicación debe generalizar a la clásica implicación material, como no se ha ofrecido una clara distinción (como se hace en la lógica clásica) entre los conceptos de implicación y condicional. En este artículo se plantean algunos criterios para justificar determinadas funciones de "implicación" borrosas que no generalizan a la implicación material y se avanza algo en la distinción entre los caracteres de implicación y de condicional de dos de ellas.

Palabras Clave: función de implicación, T-condicional, t-norma.

1 Introducción

Uno de los problemas de la Lógica Borrosa es el tratamiento de la inferencia con información imprecisa y, en particular, el de la obtención de modelos para los enunciados condicionales, es decir, los del tipo "Si x es A entonces y es B", donde x e y son elementos del universo y A y B son predicados vagos sobre el mismo. Con este fin, se han definido operadores a partir de conceptos análogos de las lógicas bivaluada y multivaluada.

A la hora de buscar generalizaciones para la Lógica Borrosa, las posibilidades son múltiples, pero debe mantenerse siempre que, al aplicar el nuevo concepto al caso límite de dos valores, el resultado coincida con el dado por el concepto original. Así, por ejemplo, a la hora de modelizar la conjunción se buscará una operación $T : [0,1] \times [0,1] \longrightarrow [0,1]$ que, al menos, cumpla las condiciones de frontera $T(0,0) = T(1,0) = T(0,1) = 0$ y $T(1,1) = 1$ de acuerdo con la tabla de verdad de la operación "y" en la lógica clásica.

Este trabajo está parcialmente financiado por la CICYT dentro del proyecto PIC96-1393-C06

De un modo similar, para representar los enunciados condicionales, los operadores de implicación deben buscarse entre aquellos que verifiquen las condiciones de frontera de las implicaciones bivaluadas. Así , a partir del condicional material, $p \rightarrow q = p' + q$ cuya tabla de verdad viene dada por

$p'+q$	0	1
0	1	1
1	0	1

se obtiene la usual definición de implicación [1], una vez añadidas otras condiciones de monotonía y la propiedad del intercambio típicas del cálculo proposicional clásico.

En este artículo se propone un estudio de diferentes conceptos de implicación borrosa atendiendo a las distintas posibilidades clásicas. Se analizan, además, algunos ejemplos de los que también se estudia su comportamiento respecto a la regla de inferencia del Modus Ponens.

2 Funciones de implicación

En un Álgebra de Boole $(B, +, \cdot, ')$ se dice que una operación $\rightarrow$: $B \times B \longrightarrow B$ es una implicación si, para todo x, y en B, es $x \cdot (x \rightarrow y) \leq y$, desigualdad que es equivalente a $x \rightarrow y \leq x' + y$. Por tanto, la implicación material no es la única implicación clásica, aunque sí la más grande. En [6] se demuestra que, además de la material, sólo existen seis funciones "booleanas" que cumplen dicha condición. A saber: $x \cdot y$, $x' \cdot y$, $x' \cdot y'$, y, x' y $x \cdot y + x' \cdot y'$ cuyas tablas de verdad son, respectivamente:

$x'+y$	0	1
0	1	1
1	0	1

$x \cdot y$	0	1
0	0	0
1	0	1

$x' \cdot y$	0	1
0	0	1
1	0	0

$x' \cdot y'$	0	1
0	1	0
1	0	0

y	0	1
0	0	1
1	0	1

x'	0	1
0	1	1
1	0	0

$x \cdot y + x' \cdot y'$	0	1
0	1	0
1	0	1

Estas implicaciones clásicas pueden ser tomadas como base para distintos modelos de la implicación en lógica borrosa. Así, desde un principio [9] atendiendo a las propiedades de frontera, monotonía e intercambio de la implicación material, se consideró la:

Definición. *Una función* $I : [0,1] \times [0,1] \longrightarrow [0,1]$ *es una función de implicación si verifica para todo* x, x', y, y', z *en* $[0,1]$ *los siguientes axiomas:*

1. $I\left(x', y\right) \leq I(x,y)$ *si* $x \leq x'$, *es decir* I *es decreciente en la primera variable.*

2. $I(x,y) \leq I\left(x, y'\right)$ *si* $y \leq y'$, *es decir* I *es creciente en la segunda variable.*

3. $I(0,y) = 1$.

4. $I(1,y) = y$.

5. $I(x, I(y,z)) = I(y, I(x,z))$.

De ellas se dirá que son funciones de implicación $\mathrm{FI}_{x'+y}$.

Ejemplos de operadores que son funciones de implicación $\mathrm{FI}_{x'+y}$, son las R-implicaciones y las S-implicaciones, que se definen, respectivamente, como $I^T(x,y) = \sup\{z \in [0,1] : T(x,z) \leq y\}$ donde T es una t-norma continua, e $I(x,y) = S(N(x), y)$ con S una t-conorma continua y N una función de negación fuerte [5]. Caracterizaciones de estas funciones de implicación aparecen en [1].

Sin embargo se usan otras funciones "de implicación" que no son de estos tipos; por ejemplo la implicación exponencial [4] $E(x,y) = y^x$, que no es una R-implicación, ya que si existiese una t-norma T tal que $E(x,y) = I^T(x,y)$, sería $x^x = I^T(x,x) = 1$ para todo x. Por otro lado, si E fuese una S-implicación, tomando $y = 0$, para todo $x > 0$ se tendría $y^x = 0 = S(N(x), 0) = N(x) = 0$ para todo $x > 0$, por lo que N no podría ser una negación fuerte.

No obstante, aunque esta definición de función de implicación $\mathrm{FI}_{x'+y}$ ha sido comúnmente aceptada, en Control no es frecuente utilizar la implicación material como modelo para representar enunciados condicionales, sino que parece más adecuado elegir la operación $x \cdot y$ como modelo de implicación ya que sólo es verdadera si lo son ambos x e y. Al igual que en el caso anterior, atendiendo a las condiciones de frontera, a la monotonía de ambas variables y a la propiedad del intercambio, se obtiene un nuevo modelo.

Definición. *Una función* $I : [0,1] \times [0,1] \longrightarrow [0,1]$ *es una función de implicación* $FI_{x \cdot y}$ *si verifica para todo* x, x', y, y', z *en* $[0,1]$ *los siguientes axiomas:*

1. $I(x,y) \leq I(x',y)$ *si* $x \leq x'$, *es decir* I *es creciente en la primera variable.*

2. $I(x,y) \leq I\left(x, y'\right)$ *si* $y \leq y'$, *es decir* I *es creciente en la segunda variable.*

3. $I(0,y) = 0$.

4. $I(1,y) = y$.

5. $I(x, I(y,z)) = I(y, I(x,z))$.

Un ejemplo muy sencillo de función de implicación $\mathrm{FI}_{x \cdot y}$, es la implicación de Mamdani, definida por $M(x,y) = Min(x,y)$. Esta función, que tiene gran aplicación en el control borroso, no es, sin embargo, una función de implicación $\mathrm{FI}_{x'+y}$.

Resumiendo, el concepto de implicación borroso no está unívocamente determinado por la definición dada en [1] sino que dependerá de la implicación clásica que se generalice. Como ya se ha dicho, en lógica bivaluada una función $\rightarrow$ es implicación si $x \rightarrow y \leq x' + y$, por lo que en el caso borroso una función de implicación I deberá verificar la ley $I(x,y) \leq I_{x'+y}(x,y)$ y, en particular, será $I(1,0) \leq I_{x'+y}(1,0) = 0$. Además, es razonable pensar que si, dada una regla ("*Si* x *es* A, *entonces* y *es* B"), se tiene el antecedente ("x *es* A"), entonces necesariamente se tendrá el consecuente ("y *es* B"). Por lo tanto el valor de verdad de I si $x = 1$, $y = 1$, debe de ser 1. Por ello, condiciones frontera necesarias para que I sea función de implicación son $I(1,0) = 0$ y $I(1,1) = 1$.

Algunos autores han propuesto otras implicaciones que sólo generalizan parcialmente a alguna de las implicaciones booleanas mencionadas. Así, por ejemplo, (ver [2] y [3]) es el caso de la llamada implicación de fuerza (*Force Implication*), $F(x,y) = x \cdot (1 - |x - y|)$, que verifica las condiciones de frontera $F(0,y) = 0$ y $F(1,y) = y$, por lo que verifica la tabla de verdad de la implicación booleana $x \cdot y$. Sin embargo, dicho operador no verifica ni los axiomas de monotonía ni la

propiedad del intercambio de la implicación $\mathrm{FI}_{x\cdot y}$.

3 Funciones de implicación y T-condicionales

El objetivo en esta sección es relacionar los conceptos de función de implicación y T-condicional, particularizando para algunos ejemplos aparecidos en la sección anterior.

En el cálculo proposicional clásico los enunciados condicionales del tipo "*Si p entonces q*" vienen representados por $p \longrightarrow q = p' + q$, que es una operación en el álgebra de los enunciados. Cuando esta implicación es afirmada, es decir, cuando $p' + q = 1$, se obtiene una relación entre los elementos del álgebra. Naturalmente, no toda relación $\Rightarrow$ entre enunciados tal que $p \Rightarrow q$ represente al enunciado verdadero "*Si p, entonces q*", vendrá dado por la forma "$p \Rightarrow q$ si y sólo si $p \longrightarrow q = 1$" para alguna implicación $\longrightarrow$. En general, una tal relación $\Rightarrow$ es un condicional respecto a un conjunto V de enunciados tomados como "verdaderos", si se verifica la Meta-Regla del Modus Ponens: "$p \Rightarrow q$ y $p \in V$: $q \in V$", la cual puede comprimirse en la inecuación: $Min(\varphi_V(p), \varphi_{\Rightarrow}(p,q)) \leq \varphi_V(q)$ para cualesquiera enunciados p y q. Todo ello lleva de forma natural a la siguiente definición.

Definición. *Dado un universo de discurso E, un subconjunto borroso $\mu : E \longrightarrow [0,1]$ y una t-norma T, una relación borrosa $R : E \times E \longrightarrow [0,1]$ es un $\mu - T$-condicional si para todo x, y de E se verifica que $T(\mu(x), R(x,y)) \leq \mu(y)$. En particular cuando $E = [0,1]$ y μ es la identidad, se dirá que R es una función generadora de T-condicional o, abreviadamente, un T-condicional.*

Trivialmente si R es un T-condicional y $T_1 \leq T$ entonces R es también un T-condicional. Además, R es un T_1-condicional si y sólo si $R \leq I^T$.[8]

Un primer teorema ayudará a obtener funciones de implicación que no son condicionales para ninguna t-norma de la familia del producto, y por lo tanto tampoco para el mínimo. Se recuerda que una t-norma T_1 pertenece a la familia de t-normas de T ($T_1 \in \mathcal{F}(T)$) si existe una aplicación biyectiva $\varphi : [0,1] \longrightarrow [0,1]$ con $\varphi(0) = 0$ y $\varphi(1) = 1$ tal que $T_1 = \varphi^{-1} o T o (\varphi \times \varphi)$.

Teorema 1. *Sea $I : [0,1]\times[0,1] \longrightarrow [0,1]$ un operador para el que existe un $a \neq 0$ tal que $I(a,0) \neq 0$. I no es T-condicional para ninguna t-norma $T \in \mathcal{F}(Prod)$.*

Demostración: Si $T \in \mathcal{F}(Prod)$ se tiene que $T(x, I(x,y)) = \left[\varphi^{-1} o\, Prod\, o\, (\varphi \times \varphi)\right](x, I(x,y)) = \varphi^{-1}(\varphi(x) \cdot \varphi(I(x,y)))$. Si I fuese T-condicional sería $T(x, I(x,y)) \leq y$ para todo x, $y \in [0,1]$. En particular para $y = 0$, para todo $x \in [0,1]$ es $\varphi^{-1}(\varphi(x) \cdot \varphi(I(x,0))) = 0$, es decir $\varphi(x) \cdot \varphi(I(x,0)) = 0$. Por tanto o bien $\varphi(x) = 0$ o bien $\varphi(I(x,0)) = 0$. Para $x = a$, es $\varphi(x) = \varphi(a) \neq 0$, por tanto es $\varphi(I(a,0)) = 0$, lo que implica $I(a,0) = 0$ que está en contradicción con la hipótesis. ■

Por lo que respecta a las funciones de implicación $\mathrm{FI}_{x'+y}$, ya en [7] y [8] se hizo un estudio de la T-condicionalidad de las S-implicaciones y de las R-implicaciones, para cualquier t-norma T. Ahora bien, aplicando el teorema anterior se puede llegar más rápidamente al mismo resultado que en los citados artículos; es decir, a que se trata de funciones de implicación que no son T-condicionales para ninguna t-norma de la familia del producto ni para el mínimo. Pero evidentemente el estudio queda abierto para aquellas I que no son ni S-implicaciones ni R-implicaciones. Respecto a la implicación exponencial, el teorema no puede ser aplicado ya que para todo $x > 0$ es $E(x,0) = 0^x = 0$. Sin embargo se tienen los siguientes resultados:

Teorema 2. *Si $T = \varphi^{-1} o\, Prod\, o\, (\varphi \times \varphi)$, donde $\varphi(x) = x^\alpha$, $\alpha \in \mathbb{R}$, entonces E no es T-condicional. En particular E no es $Prod$-condicional.*

Demostración: Si E fuese T-condicional se tendría que verificar la desigualdad $T(x, E(x,y)) \leq y$ para todo x, $y \in [0,1]$. Como contraejemplo basta tomar $x = \dfrac{1}{2}$ e $y = \dfrac{1}{9}$, y se tiene que

$$T(x, E(x,y)) = \left[\varphi^{-1} o\, Prod\, o\, (\varphi \times \varphi)\right](x, E(x,y)) = \sqrt[\alpha]{x^\alpha \cdot E(x,y)^\alpha} = \frac{1}{2} \cdot \frac{1}{9}^{\frac{1}{2}} = \frac{1}{6} > \frac{1}{9}$$

Es decir, la implicación exponencial no es un $Prod$-condicional y por tanto tampoco un Min-condicional. ■

Queda sin embargo abierto el problema para otras expresiones de φ.

Denotando por W la t-norma de Łukasiewicz se tiene:

Teorema 3. *La implicación exponencial es W-condicional.*

Demostración: Se tiene que probar que para todo x, y en $[0,1]$ se verifica la desigualdad $W(x, E(x,y)) \leq y$; es decir que es $Max(0, x + y^x - 1) \leq y$. Considérese

para cualquier y en $[0,1]$ la función $f_y(x) = x + y^x - 1$. La finalidad de lo que sigue es comprobar que el máximo en $[0,1]$ de cada función f_y está en el punto $x = 1$.

a) Si $y = 0$, $f_0(x) = x - 1$ es una función estrictamente creciente cuyo máximo en $[0,1]$ es $x = 1$.

b) Si $y \neq 0$, la función f_y es contínua y derivable y su derivada es $f_y'(x) = 1 + y^x \cdot \ln y$ y si se iguala a cero se obtiene el valor $x_0 = \dfrac{\ln\left(\ln\dfrac{1}{y}\right)}{\ln\dfrac{1}{y}}$, que la anula.

La segunda derivada es $f_y''(x) = y^x \cdot \ln^2 y$, que es siempre positiva; es decir, el punto x_0 obtenido es un mínimo y, por lo tanto, el máximo se alcanza en uno de los extremos del intervalo. Ahora bien es $f_y(0) = 0$ y $f_y(1) = y$.

Entonces para todo y en $[0,1]$, es $f_y(x) \leq f_y(1)$; es decir, que es $x + y^x - 1 = f_y(x) \leq f_y(1) = y$. Por tanto es $Max(0, x + y^x - 1) \leq y$ y la implicación exponencial es un W-conditional. ■

Queda abierto el problema para otras t-normas de la familia de Łukasiewicz.

Los conceptos de función de implicación y de T-condicional son generalizaciones de dos conceptos distintos en la lógica clásica y sucede lo mismo con dichos conceptos borrosos. Por ejemplo la implicación de fuerza, de la que ya se vio en la sección anterior que no es función de implicación, es un W-condicional. La siguiente pregunta sería si cualquier función de implicación es T-condicional y la respuesta, que es negativa, viene dada por el siguiente ejemplo.

El operador I definido por

$$I(x,y) = \left\{ \begin{array}{ll} y, & \text{si } x = 1 \\ 1, & \text{en otro caso} \end{array} \right\}$$

es una función de implicación $\mathrm{FI}_{x'+y}$, pero no es Z-condicional, donde Z es la t-norma mínima definida por $Z(x,y) = Min(x,y)$ si $Max(x,y) = 1$; $Z(x,y) = 0$ en otro caso. En efecto, tomando $0 < x < 1$ e $y = 0$, resulta $Z(x, I(x,0)) = Z(x,1) = x > 0$. Al no ser este operador I Z-condicional, no es T-condicional para ninguna t-norma T.

Respecto a las funciones de implicación $\mathrm{FI}_{x \cdot y}$, por ser crecientes en la primera variable, junto con las condiciones frontera, se tiene $I(x,y) \leq I(1,y) = y$, y por tanto para cualquier t-norma T se da la desigualdad $T(x, I(x,y)) \leq I(x,y) \leq y$. Así se obtiene

Teorema 4. *Toda función de implicación $FI_{x \cdot y}$ es T-condicional para cualquier t-norma T.*

Por último, aunque ya se ha visto que la implicación de fuerza no es una función de implicación $\mathrm{FI}_{x'+y}$ en el sentido de este trabajo, conviene hacer un estudio sobre su condicionalidad. En primer lugar se puede aplicar el terorema 1, ya que tomando $0 < x < 1$ se tiene que $F(x,0) = x(1-x) > 0$, de donde se deduce inmediatamente, debido a tal teorema que F no es T-condicional para ninguna $T \in \mathcal{F}(Prod)$. Tampoco lo es, como consecuencia, para la t-norma del mínimo. Sin embargo

Teorema 5. *La implicación de fuerza es W-conditional.*

Demostración: Hay que probar que vale $W(x, F(x,y)) \leq y$ para todo x, $y \in [0,1]$. Para $x \leq y$ la desigualdad es trivial, por lo que basta considerar $y < x$. Entonces

$$W(x, F(x,y)) = Max(0, x + x(1 - x + y) - 1) = Max\left(0, \left(2x - x^2 - 1\right) + xy\right) \leq xy \leq y$$

Es decir, F es W-condicional. ■

Para las demas t-normas de la familia $\mathcal{F}(W)$ el estudio no resulta tan evidente y permanece abierto.

4 Conclusiones

En este artículo se han establecido unos criterios para una nueva definición de función de implicación, partiendo de implicaciones clásicas distintas de la implicación material. Además, se ha estudiado la condicionalidad de operadores definidos en la literatura y que no habían sido considerados en trabajos anteriores.

Referencias

[1] C. Alsina, J.L. Castro and E. Trillas, *On the Characterization of S and R Implications*, VI I.F.S.A. World Congress, São Paulo (Brazil), 1995, Vol.1, 317-319.

[2] Ch. Dujet and N. Vincent, *Force Implication: A new Approach to Human Reasoning*, Fuzzy Sets and Systems, 69 (1995), 53-63.

[3] Ch. Dujet and N. Vincent, *A Suggested Conditional Modus Ponnens*, International Journal of Uncertainty, Fuzzyness and Knowledge-Based Systems, Vol. 5, No. 1 (1997), 93-106.

[4] R.Yager, *An Approach to Inference in Approximate Reasoning*, International Journal of Man-Machine Studies, 13 (1980), 323-338.

[5] E. Trillas, *Sobre funciones de negación en la teoría de conjuntos borrosos*, Stochastica, Vol. III, nº 1 (1979), pp.47-60.

[6] E. Trillas y S. Cubillo, *Modus Ponens on Boolean Algebras Revisited*, Mathware & Soft Computing, 3 (1996), pp. 105-112.

[7] E. Trillas, S. Cubillo y C. del Campo, *A Few Remarks on Some T-Conditional Functions*, Sixth IEEE International Conference on Fuzzy Systems, Barcelona, 1997, Vol. 1, pp. 153-156.

[8] E. Trillas, A.R. de Soto y S. Cubillo, *On Implication and T-Conditional Functions*, en el libro "*Discovering the World with Fuzzy Logic*", V. Novak e I. Perfilieva (eds.), pendiente de publicación.

[9] E. Trillas y L. Valverde, *On Mode and Implication in Approximate Reasoning*, en el libro "*Approximate Reasoning in Expert Systems*", M.M. Gupta, A. Kandel, W. Bandler y J.B. Kiszka (eds.), Elsevier Science Publishers B.V., North-Holland, 1985.

Particiones, Indistinguibilidades y Variables Lingüísticas

Adolfo R. de Soto
Area de Lenguajes y Sistemas Informáticos
Facultad de Económicas y Empresariales
Universidad de León
ddears@unileon.es and

Jordi Recasens*
Secció Matemàtiques i Informàtica
E.T.S.A.V.
Universidad Politécnica de Cataluña
jordi.recasens@ea.upc.es

Resumen

En este trabajo se muestra un método para obtener una jerarquía de particiones del universo $[0,1]$ de forma que cada una de ellas es compatible con la indistinguibilidad de Lukasiewicz. Las clases de cada partición de un determinado nivel presentan una relación de antonimia entre si. Además la partición de un nivel puede verse como el refinamiento de un nivel anterior mediante una clase de modificadores lingüísticos, por lo que estas particiones parecen adecuadas para modelizar etiquetas lingüísticas de una variable lingüística.

Palabras Clave: Modelización Lingüística, Granularización Fuzzy, Indistingüibilidad, Particiones Fuzzy.

1 Introducción

Es muy frecuente que entre los predicados graduados pertenecientes a una misma variable lingüística [8] exista una relación de antonimia. De hecho los valores lingüísticos permitidos para muchas variables lingüísticas pueden generarse a partir de un par de predicados antónimos y una serie de modificadores lingüísticos. Por ejemplo, para la variable lingüística *temperatura,* los valores *frío, caliente, helado, muy caliente, bastante frío,* etc. pueden ser generados a partir del predicado *frío,* de su antónimo *caliente* y de los modificadores lingüísticos como *muy, bastante* etc. El uso de las conectivas lógicas conjunción, disjunción o negación permiten combinar estas etiquetas para obtener otras como, por ejemplo, *templado = ni frío ni caliente.* En [1] se mostró, mediante una representación adecuada de los modificadores lingüísticos y del antónimo en la teoría de conjuntos borrosos, una relación existente entre estos dos fenómenos, relación que se correspondía adecuadamente con el uso de estos términos lingüísticos en el lenguaje natural. En el mencionado trabajo se percibe que la antonimia es un fenómeno propio de la variable lingüística más que de ciertos predicados concretos. Siguiendo con el ejemplo de la variable *temperatura*, la antonimia existente entre *frío-caliente* parece la misma que la existente entre *muy frío-muy caliente* o entre *helado-abrasando,* si nos referimos por ejemplo al estado de un café.

La importancia de las relaciones de antonimia entre predicados se manifiesta en la cantidad de conocimiento que el hombre adquiere gracias a la existencia de predicados antónimos; basta pensar en el aprendizaje de los niños. Quizá esto no sea más que otra manifestación más de la tendencia del hombre a dualizar, presente también en el fenómeno de la negación. Además cuando se transmite información mediante alguno de estos contrarios, se supone casi siempre la presencia implicita del otro y de la posición intermedia. Por ejemplo, al decir que *el cafe está frío* se contrapone a *el café está caliente* y las dos situaciones son extremos de *el cafe está bien.*

Zadeh al definir una variable lingüística utilizó gramáticas generativas de tal forma que, a partir de ciertos operadores, todos los términos lingüísticos de la variable están dados; son los que se pueden construir con dichos operadores. De la misma forma, si se tiene una semántica asociada a los términos lingüísticos básicos mediante conjuntos borrosos, se tiene una semántica asociada a todos los términos lingüísticos válidos, sin más que asociar a dichos operadores operaciones adecuadas entre conjuntos borrosos. Sin embargo, y a pesar de que en el lenguaje natural tengamos a nuestra disposición todos los términos lingüísticos válidos para una variable lingüística, al menos potencialmente, consideramos que la asignación de un significado a esos términos mediante conjuntos borrosos se realiza de otra forma, con base en los términos ímplicitos existentes cuando se formula una

El autor ha sido financiado parcialmente por el proyecto número PB94-1208 de la DGICYT.

predicación. Así al usar *frío* se contrapone a *caliente* y puedo considerar que la variable lingüística la forman únicamente los términos *frío-caliente* o quizás *frío-templado-caliente*. Si en cambio utilizo el término *muy caliente* estoy suponiendo implicitamente la existencia de, al menos, los términos *muy frío-frío-caliente-muy caliente*; si utilizo *helado* supongo *helado-muy frío-frío-caliente-muy caliente-abrasando*, etc. De esta forma los términos lingüísticos de una variable lingüística constituyen una jerarquía de varios niveles, cada uno de ellos presentando diferente granularidad. Esto permite establecer distintos niveles de precisión marcados por el uso de varias etiquetas que se supone dividen el universo base de la variable lingüística.

En este trabajo se mostrará una forma de crear una jerarquía de clases de conjuntos borrosos que sirva para modelizar los términos lingüísticos de una variable lingüística. La clase de conjuntos borrosos de cada nivel se obtendrá como una partición borrosa del universo de discurso compatible con una indistinguibilidad; a su vez cada nivel se obtiene como un refinamiento de la indistinguibilidad natural asociada a la t-norma de Lukasiewicz, lo que permite que para todos los niveles exista una relación constante de antonimia.

El trabajo comienza con un recordatorio sobre indistinguibilidades y el estudio de un conjunto de propiedades que parece razonable imponer a una partición borrosa para que muestre cierta compatibilidad con una indistinguibilidad. Una indistinguibilidad puede refinarse operándola consigo misma a través de la t-norma asociada. Esta operación hace aumentar el número de clases de la partición compatible. En concreto, se ve que la indistinguibilidad natural E_W para la t-norm W de Lukasiewicz, genera una partición de $[0,1]$ en dos clases antónimas entre si, y que pueden interpretarse, por ejemplo, como (*falso, verdadero*). E_W^2 refina esta partición en 3 clases, las cuales admiten la interpretación (*falso, ni falso ni verdadero, verdadero*), y donde, de nuevo, está presente la misma relación de antonimia.

El estudio de las indistinguibilidades E_T^n es especialmente interesante; en primer lugar, sus clases son los elementos de la partición de nivel n, por lo que adquieren un significado; en segundo lugar, permiten un estudio eficaz de las particiones que se usan en control fuzzy. En efecto, dada una partición en un nivel de precisión determinado, se puede generar una aplicación f creciente, continua y lineal a trozos, entre el intervalo I, universo de discurso, y $[0,1]$ de modo que si la partición tiene n clases, se ajusta a las n clases generadas en $[0,1]$ por E_T^{n-1}. Así se corresponden a la partición generada en I por la T-indistinguibilidad $E(x,y) = E_T(f(x), f(y))$.

2 Las Indistinguibilidades E_T^n

En primer lugar recordemos que una T-indistinguibilidad $E : X \times X \to [0,1]$ es una relación borrosa reflexiva, simétrica y T-transitiva, siendo T una t-norma. Dada una t-norma continua T, la T-indistinguibilidad natural E_T sobre $[0,1]$ se define como

$$E_T(x,y) = T(\widehat{T}(x,y), \widehat{T}(y,x)),$$

donde $\widehat{T}$ es la operación definida por residuación

$$\widehat{T}(x,y) = \sup\{z \in [0,1] \ : \ T(x,z) \leq y\}.$$

Las T-indistinguibilidades han sido estudiadas ampliamente en el marco de la teoría fuzzy, ver por ejemplo [5, 3]. Para las t-normas continuas mínimo (min), producto ($\Pi(x,y) = xy$) y Lukasiewicz ($W(x,y) = \max(0, x+y-1)$), las T-indistinguibilidades naturales asociadas sobre $[0,1]$ son:

- $E_{\min}(x,y) = \begin{cases} 1 & \text{si } x = y \\ \min(x,y) & \text{e.o.c.} \end{cases}$
- $E_\Pi(x,y) = \min(\frac{x}{y}, \frac{y}{x})$
- $E_W(x,y) = 1 - |x-y|$

El siguiente resultado es inmediato.

Lema 1 *Si E_T es una T-indistinguibilidad sobre un universo X, entonces $E_T^n(x,y) = T(E(x,y), \overset{n}{\ldots}, E(x,y))$ con $n \geq 1$, también es una T-indistinguibilidad en X.*

Ejemplo 2 *Sobre* $[0,1]$

- *Para la t-norma mínimo se tiene $E_{\min}^n = E_{\min}$ para toda n.*
- *Para la t-norma producto $E_\Pi^n(x,y) = \min(\frac{x^n}{y^n}, \frac{y^n}{x^n})$.*
- *Para la t-norma de Lukasiewicz $E_W^n(x,y) = 1 - n|x-y|$, donde $x - y = \max(0, x-y)$.*

En general, las distintas potencias de E refinan la indistinguibilidad puesto que se tiene $E_T^n \leq E_T^{n-1}$ y por tanto permiten una granuralidad más fina.

3 Particiones compatibles con una T-indistinguibilidad

Consideraremos la siguiente definición de partición borrosa.

Definición 3 *Una familia finitamente enumerada $P = \{\mu_i\}_{i=1,\dots,n}$ de conjuntos borrosos sobre un universo X constituye una T-partición borrosa si cumple las dos condiciones siguientes:*

1. $T(\mu_i(x), \mu_j(x)) = 0$ *para* $i \neq j : 1, \dots, n$ *y cualquier* $x \in X$.
2. $S(\mu_1(x), \dots, \mu_n(x)) = 1$ *para todo* $x \in X$ *y* S *la t-conorma* N*-dual de* T *respecto a alguna negación fuerte* N.

Denominaremos clase i de la partición P al conjunto borroso μ_i.

La noción de compatibilidad de una partición con una T-indistinguibilidad viene dada por la siguiente definición.

Definición 4 *Una T-partición $P = \{\mu_i\}_{i=1,\dots,n}$ se dice compatible con una T-indistinguibilidad E_T si se verifica, para cualquier $i = 1, \dots, n$ y cualesquiera $x, y \in X$, que*

1. $T(\mu_i(x), \mu_i(y)) \leq E(x, y)$, *y*
2. *existe un* $a_i \in X$ *con* $E(x, a_i) \leq \mu_i$.

La primera propiedad implica que todo par de elementos x e y que pertenezcan a la misma clase μ_i son elementos equivalentes, mientras que la segunda asegura la existencia de un elemento prototípico para cada clase de la partición.

Los siguientes resultados son directos.

Lema 5 *Para cada clase μ_i existe un $a_i \in X$ con $\mu_i(a_i) = 1$.*

Demostración. Basta tomar $x = a_i$ en 4.2.

Lema 6 *Sea $a_i \in X$ tal que $\mu_i(a_i) = 1$, entonces $\mu_i(x) = E(x, a_i)$ para todo $x \in X$.*

Demostración. Por 4.1 $\mu_i(x) = T(\mu_i(a_i), \mu_i(x)) \leq E(a_i, x)$ y por 4.2 $\mu_i(x) \geq E(a_i, x)$.

Lema 7 *Si E es separadora, es decir, $E(x, y) = 1$ implica que $x = y$, entonces para toda $i = 1, \dots, n$ existe un único $a_i \in X$ tal que $\mu_i(a_i) = 1$.*

Demostración. Por 4.1, si $\mu_i(x) = \mu_i(y) = 1$, entonces $T(\mu_i(x), \mu_i(y)) \leq E(x, y)$ y por tanto $x = y$.

Lema 8 *Las clases de una partición P compatible con una T-indistinguibilidad E_T son T-estados lógicos [6], o generadores[1] [2], de la T-indistinguibilidad T.*

[1]Otras denominaciones para el mismo concepto son "observables" o "extensionales".

Demostración. Por el lema 6 las clases de la partición son las columnas de la T-indistinguibilidad E_T, y las columnas son T-estados lógicos, o generadores, de E.

4 Una familia de Particiones compatibles con E_W^n

Si se considera la t-norma de Lukasiewicz W y el universo $[0, 1]$, para cada elemento de la familia de W-indistinguibilidades E_W^n existe una única partición compatible.

Teorema 9 *La única partición compatible con E_W en $[0, 1]$ es la dada por los subconjuntos $t(x) = x$ y $f(x) = 1 - x$.*

Demostración. Como es fácil de comprobar, para cualquier par de números x, y de $[0, 1]$ se cumple que si $W(x, y) = \max(0, x + y - 1) = 0$ y $W^*(x, y) = \min(1, x + y) = 1$, entonces $x + y = 1$, donde W^* es la t-conorma dual de la t-norma de Lukasiewicz. Por tanto, una W-partición $P = \{\mu_i\}_{i=1,\dots,n}$ verificará que

$$\sum_{j=1}^{n} \mu_i(x) = 1,$$

para todo $x \in [0, 1]$. Tomando el prototipo a_i de una clase de la partición, se tiene que

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} \mu_i(x) = 0,$$

es decir, $\mu_j(a_i) = 0$ para todo $j = 1, \dots, n$ y $j \neq i$. Todas las clases de P son columnas de E_W, pero las únicas columnas que se anulan en algún punto son $E_W(0, x)$ y $E_W(1, x)$. Así la única partición compatible es la formada por las clases del 0 y del 1.

Más general,

Teorema 10 *La única partición compatible con E_W^n está formada por las $n + 1$ columnas $\{E_W^n(a_k, x)\}_{k=0,\dots,n}$, siendo $a_k = \frac{k}{n}$.*

En la figura 1 pueden verse las clases de la particiones compatibles para E_W, E_W^2, E_W^3, E_W^4 y E_W^8.

5 Las particiones de E_W^n y el antónimo

Existe una fuerte relación entre la negación y el antónimo. El antónimo puede considerarse una negación interna dependiente del universo de discurso mientras que la negación habitual es una negación externa independiente del universo de discurso. La

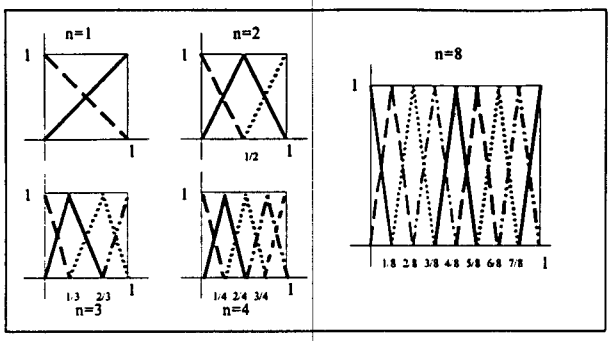


Figura 1: Algunos ejemplos de particiones compatibles con E_W^n para diferentes n

distinción entre negación interna y externa viene por la forma de operar del antónimo y de la negación habitual. Sobre un universo X dos conjuntos borrosos μ, η se dicen antónimos si existe una aplicación $ant : X \to X$ involutiva y decreciente tal que $\mu(x) = \eta(ant(x))$. Es evidente que ant tiene las mismas características que una negación, basta con tomar $X = [0,1]$. La diferencia reside en que, dada una negación $n : [0,1] \to [0,1]$, el conjunto borroso negado de μ es $no(\mu)(x) = n(\mu(x))$; al aplicarse n a los valores del conjunto borroso y no a los elementos del universo, la negación no depende del universo del discurso, en cambio el antónimo sí.

Dada una t-norma T, mediante la operación definida por residuación $\widehat{T}$ se obtiene una negación a través de la expresión $n_T(x) = \widehat{T}(x,0)$. Esta negación es en muchos casos débil, pero para la t-norma de Lukasiewicz resulta en cambio la negación habitual $n_T(x) = 1 - x$, puesto que $\widehat{W}(x,y) = \min(1, 1-x+y)$. De esta forma puede considerarse que E_W tiene asociada la negación habitual ya que también $E_W(x,0) = n(x) = 1 - x$. Sobre $[0,1]$, un antónimo viene dado por una función de negación, resultando que sobre todas las particiones compatibles con las indistinguibilidades E_W^n existe una relación de antonimia definida mediante esta negación.

Teorema 11 *Si μ_i es una clase de una partición compatible con E_W^n, entonces su antónimo respecto de la aplicación $ant(x) = 1 - x$ es también una clase de la partición.*

Demostración. Para todo $i \in \{1, \ldots, n\}$ se tendrá que

$$\begin{aligned} ant(\mu_i)(x) &= \mu_i(ant(x)) = E_W^n(ant(x), a_i) \\ &= 1 - n|(1-x) - a_i| \\ &= 1 - n|1 - a_i - x| \\ &= 1 - n\left|1 - \frac{i}{n} - x\right| \\ &= 1 - n\left|\frac{n-i}{n} - x\right| \\ &= \mu_{n-i}(x). \end{aligned}$$

Es interesante observar que el antónimo de una clase de una partición será la misma clase si y sólo si $i = n-i$, es decir, cuando $i = \frac{n}{2}$, y esto puede ocurrir únicamente cuando n sea par, con lo que $\frac{n}{2}$ será la etiqueta central.

La relación de antonimia entre las clases de una partición compatible para un nivel n se mantiene para cualquier nivel, este fenómeno se corresponde con la aparición de etiquetas lingüísticas a pares en una variable lingüística, como, por ejemplo, la ya mencionada *helado-abrasando*, o para *altura*, partiendo de la pareja básica *bajo-alto* se puede añadir, además de todas las parejas generadas por modificadores lingüísticos, *enano-gigante*.

La aplicación $n(x) = 1 - x$ operando como una negación, es decir externamente, no da el mismo resultado que operando internamente como es fácil de comprobar. Esta regla presenta una excepción con $E_W(x,y)$, donde sólo hay dos clases una la negación, y también el antónimo, de la otra.

La negación de una clase de una partición no es otra clase de la partición pero se puede escribir como combinación del resto de clases, puesto que se verifica la igualdad

$$1 - \mu_i = W^*(\mu_1, \ldots, \mu_{i-1}, \mu_{i+1}, \ldots, \mu_n).$$

Considerando W y W^* como las conectivas lógicas conjunción y disjunción respectivamente, son evidentes ciertas relaciones interesantes como, por ejemplo, que la unión de clases consecutivas de una partición es un trapecio que las incluye, o que la su intersección es siempre cero.

6 Paso de E_W^n a E_W^{2n}

Los modificadores lingüísticos fueron definidos por Zadeh en sus primeros trabajos como operadores sobre conjuntos borrosos, así el modificador *muy* para un predicado borroso P con función de pertenencia μ_P da lugar a la función de pertenencia $\mu_{muyP}(x) = \mu_P^2(x)$. Fundamentalmente con esta operación se consigue una intensificación del grado de μ_P, de tal forma que $\mu_{muyP} \leq \mu_P$ puntualmente. Pero esta intensificación no es quizá una propiedad obligatoria para un modificador lingüístico, aunque se pueda dar para algunos. Otras definiciones de modificadores lingüísticos son posibles. La definición de Zadeh utiliza el operador $m(x) = x^2$ operando de forma externa, se pueden definir operadores que modelizan adecuadamente los modificadores lingüísticos operando de forma interna (ver [1] para más detalles).

A partir de este momento denominaremos por $t_{r:n}$ a la clase r de la partición compatible con E_W^n. Tomemos la familia de operadores $m_k(x) = kx$ con $k \geq 1$, que operaran sobre los conjuntos borrosos de forma interna, es decir, $m_k(t_{r:n})(x) = t_{r:n}(kx)$. Los conjuntos $t_{0:n}$ pueden escribirse como $t_{0:n}(x) = \max(0, 1 - nx)$ [2]. Es evidente entonces que

$$m_k(t_{0:n})(x) = t_{0:n}(kx) = 1 - nkx = t_{0:nk}(x),$$

por lo que cualquier $t_{0:n}$ puede verse como un conjunto borroso modificado, a través de $m_{n/r}$, de $t_{0:r}$ para $r \leq n$. Además, en este caso se cumple que $t_{0:n} \leq t_{0:r}$ puntualmente cuando $r \leq n$.

Los conjuntos borrosos $t_{n:n}$ son los antónimos de $t_{0:n}$ por lo que

$$\begin{aligned} t_{n:n}(x) &= ant(t_{0:n})(x) = t_{0:n}(ant(x)) \\ &= t_{0:n}(1 - x) = 1 - n(1 - x). \end{aligned}$$

De nuevo se verifica que $t_{n:n} \leq t_{r:r}$ para $r \leq n$ y $t_{n:n}$ puede verse como un conjunto borroso intensificado de $t_{r:r}$.

La partición asociada a E_W^2 puede obtenerse del siguiente modo a partir de E_W:

$$\begin{aligned} t_{0:2} &= m_2(t_{0:1}) \\ t_{2:2} &= ant(t_{0:2}) \\ t_{1:2} &= \min(1 - t_{0:2}, 1 - t_{2:2}) = W(1 - t_{0:2}, 1 - t_{2:2}). \end{aligned}$$

$t_{1:2}$ es la etiqueta central de E_w^2 y correspondería al predicado *ni P ni antP* cuando P es un predicado polar [1].

Calculando $m_2(t_{1:2})$ obtenemos

$$\begin{aligned} m_2(t_{1:2})(x) &= t_{1:2}(2x) = \min(1 - t_{0:2}(2x), 1 - t_{2:2}(2x)) \\ &= \min(4x, 2(1 - 2x)) \\ &= t_{1:4}(x), \end{aligned}$$

como se puede comprobar calculando la expresión funcional de $t_{1:4}$. En este caso no se verifica que $t_{1:4} \leq t_{1:2}$ pero en cierto sentido si es una intensificación de un predicado graduado, por ejemplo si $t_{1:2}$ representara *templado*, $t_{1:4}$ podría ser *más frío que templado*.

Las clases de la partición E_W^4 son:

$$\begin{aligned} t_{0:4} &= m_2(t_{0:2}) \\ t_{1:4} &= m_2(t_{1:2}) \\ t_{3:4} &= ant(t_{1:4}) \\ t_{4:4} &= ant(t_{0:4}) \\ t_{2:4} &= \min(1 - (t_{1:4} + t_{0:4}), 1 - (t_{3:4} + t_{4:4})), \end{aligned}$$

y se obtienen como modificadas de las clases de la partición E_W^2. Generalizando el proceso para cualquier $n = 2k$ con $k \geq 1$, se tiene entonces que

$$\begin{aligned} t_{i:n} &= m_2(t_{i:n/2}) \\ t_{n-i:n} &= ant(t_{i:n}) \end{aligned}$$

para todo $i = 1, \ldots, 2k - 1$, mientras que la etiqueta central sería

$$t_{n/2:n} = \min(1 - \sum_{j=0}^{2k-1} t_{j:n}, 1 - \sum_{j=0}^{2k-1} t_{n-j:n}). \quad (1)$$

De esta forma la partición asociada a E_W^{2n} puede considerarse como un refinamiento de la partición asociada a E_W^n a través del modificador m_2. La expresión dada en (1) es equivalente a $t_{n/2:n} = W(1 - W^*(t_{1:n}, \ldots, t_{2k-1:n}), 1 - W^*(t_{n-1:n}, \ldots, t_{n-2k-1:n}))$ por lo que las clases de la partición compatible con E_W^{2n} se generan a partir de las clases asociadas a E_W^n, el modificador lingüístico m_2 y la terna de De Morgan $(W, W^*, 1 - j)$. La etiqueta intermedia de nuevo es *ni las etiquetas de la izquierda ni las etiquetas de la derecha*. El uso de W como conjunción y de W^* como disjunción es coherente con la definición de T-indistinguibilidad, donde la t-norma T se utiliza para "agregar" los valores $E(x, y)$ y $E(y, z)$ en la generalización de la propiedad transitiva, y con la definición de partición donde T y S se usan como intersección y unión de clases respectivamente.

7 Interpretación de las particiones de control fuzzy

En control fuzzy, se particiona un intervalo $I = [a, b]$ en una colección finita de números triangulares y trapezoidales de forma que en cada punto de I sólo intervienen dos clases (sólo pertenece al soporte de dos clases) y de manera que cuando uno llega a valer 0 el siguiente toma el valor 1 (máximo solapamiento).

Teorema 12 *Sea $P = \{\mu_i\}_{i=1,\ldots,n}$ una partición como las anteriores del intervalo $I = [a, b]$. Existe una única función f no decreciente, continua, lineal a trozos $f : I \to [0, 1]$ tal que si $\{t_{i:n}\}$ es la partición de $[0, 1]$ asociada a E_W^{n-1}, entonces $\mu_i(x) = t_{i:n}(f(x))$ para todo $x \in I$.*

Ejemplo 13 *En la figura 2 se da una partición con cuatro clases y la función f que permite trasvasar la información a $[0, 1]$ dando la partición asociada E_W^3, que es el caso para $n = 3$ de la figura 1*

En el intervalo I se puede definir la correspondiente relación de indistinguibilidad E.

[2] Para simplificar la notación se supondrá que las expresiones que utilizaremos para $t_{r:n}$ son realmente de la forma $\max(0, \min(1, t_{r:n}))$.

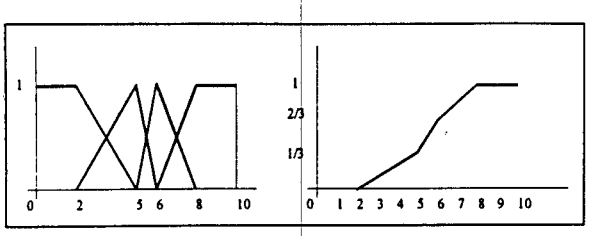


Figura 2: Una partición en $[0,10]$ *y la función lineal a trozos que pasa la partición asociada a* E_W^3 *en* $[0,1]$.

Definición 14 *Con las notaciones anteriores,*

$$E(x,y) = E_W^{n-1}(f(x), f(y)).$$

E es una relación de indistinguibilidad homómetra a E_W^{n-1} [7]. Además E permite generar otros conjuntos borrosos, sus columnas, de forma directa traspasando los de $[0,1]$ con E_W^{n-1}. $T(E,E)$ refina la partición en caso necesario.

E parece la relación de indistinguibilidad más natural que se puede generar con la partición de partida y no coincide con la que definen Klawonn y Kruse [4], que es la mayor posible que las contiene como estados lógicos.

8 Conclusiones

En el trabajo se ha presentado una forma de refinar una T-indistinguibilidad operándola consigo misma de tal forma que, para la t-norma de Lukasiewicz, se obtiene una familia jerárquica de particiones borrosas del intervalo $[0,1]$. Esta familia posee varias propiedades que hacen especialmente sencilla su interpretación como etiquetas lingüísticas de una variable lingüística. Así existe una relación de antonimia para toda la familia y la partición asociada E_W^{2n} puede verse como un refinamiento de la partición asociada E_W^n mediante cierta familia de modificadores lingüísticos.

Por otra parte, a través de una función lineal a trozos estas particiones pueden trasvasarse a cualquier intervalo compacto I de la recta real, dando lugar a las particiones habitualmente usadas en control fuzzy. Esta aplicación permite definir una indistinguibilidad en el intervalo I, lo que posibilita una interpretación dentro de la lógica borrosa del control fuzzy.

Referencias

[1] DE SOTO, A. R.; TRILLAS, E.; HERRERO, M. J. (1997) "Antónimos y Modificadores Lingüísticos". Publicado en las *Actas del VII Congreso -Español sobre Tecnologías y Lógica Fuzzy*, pp. 7–14, Tarragona.

[2] JACAS, J.; RECASENS, J. (1994) "Fixed Points and Generators of Fuzzy Relations". *J. Math. Anal. Appl.* 186, pp. 21–29.

[3] JACAS, J.; RECASENS, J. (1995) "Fuzzy T-transitive Relations: Eigenvectores ang Generators". *Fuzzy Sets and Systems*, 72, pp. 147–154.

[4] KLAWONN, F.; KRUSE, R. (1993) "Equality Relations as a basis for Fuzzy Control". *Fuzzy Sets and Systems*, 54, pp. 147–156.

[5] TRILLAS, E.; VALVERDE, L. (1984) "An Inquiry on Indistinguishability Operators", *Aspects of Vagueness* Editado por H.J. Skala, S. Termini and E. Trillas, pp. 231 256, Reidel, Dordrecht.

[6] TRILLAS, E (1993) "On Logic and Fuzzy Logic". *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 1/2 107–137.

[7] TRILLAS, E.;ALSINA, C. (1978) "Introducción a los espacios métricos generalizados". *Fund. J. March. Serie Universitaria.* Madrid.

[8] ZADEH, L. A. (1976) "The Concept of a Linguistic Variable and its Application to Approximate Reasoning, I,II and III " en *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh* editado por Yager, R.R. et al., pp. 219–366, John Wiley and Sons, New York.

On the learning of weights in some aggregation operators

Vicenç Torra
Departament d'Enginyeria Informàtica (ETSE), Universitat Rovira i Virgili
Carretera de Salou s/n, E-43006 Tarragona
vtorra@etse.urv.es

Abstract

In this paper we consider the learning of weights in the weighted mean and in the OWA operator. In relation to the latter one, we compare the results obtained with our method and another one in the literature. We consider also some guidelines in relation to the WOWA operator.
Keywords: Aggregation operators, WOWA operators

1. Introduction

Aggregation operators are used nowadays in several applications in the Artificial Intelligence framework to combine information from different information sources (opinions of experts, data from sensors, results from computational systems). We find their implementation in several systems. For example: vision [10], robotics [3], knowledge based systems [14]. Some examples of well known aggregation methods are the weighted mean (characterized in [1, 2]), the OWA operator (defined in [17, 18] and characterized in [7]), the WOWA operator ([15]) and the Choquet integral (see [9] for a review of fuzzy integrals).

Although these operators have been used in several applications, one of the problems any user has to deal with to use one of them is to fix the parameters that are associated with each one of the methods. For example, to use a weighted mean or an OWA operator the user has to settle the weights of each information source. The determination of these weights is usually done in an heuristic way (after trial and error) or asking an expert to supply them.

In this work we present a solution to this problem based on a machine learning approach: we assume that there is a set of examples and we are interested in learning the model. In our case, an example consists on a set of values corresponding to the values supplied by the information source (the input values) and the resulting value after the aggregation (the output value). Learning the model consists on determining the weights of the aggregation function: for example, in the case of the weighted mean, the weights of each information source.

We consider in this paper, how to learn to determine the weights in the weighted mean and the weights of the OWA operator. Besides of that, we also consider the learning of weights in the WOWA operator.

This work begins in section 2 with some preliminaries where we review the aggregation operators that are considered then in the next section to discover the model. In section 3, we consider the learning of the weights in the weighted mean. In section 4, we consider the learning of the weights in the OWA and the WOWA operators. We finish in section 5 with some conclusions.

2. Preliminaries

We define below the aggregation functions that we consider in the next sections in relation to learning the weights. The weighted mean (with the quasi-arithmetic mean that is a general form of it) was studied and characterized in [1, 2]. The OWA operator is defined in [17, 18]. Both functions, the weighted mean and the OWA operator, are a lineal combination of the values according to a set of weights. The difference is the meaning of these weights in each function. On one hand, the weighted mean allows the system to compute an aggregate value from the ones corresponding to several sources taking into account the reliability of each information source. Alternatively, the OWA operator permits of weighting the values in relation to their ordering. See [15] for detailed comparison of both aggregation functions.

Definition. A vector $\mathbf{v}= [v_1\ v_2\ ....\ v_n]$ is a *weighting vector* of dimension n if and only if

$$v_i \in [0,1] \qquad \Sigma_i\ v_i = 1$$

Definition. Let **p** be a weighting vector of dimension n, then a mapping WM: $\mathbb{R}^n \rightarrow \mathbb{R}$ is a *weighted mean* of dimension n if $WM_p(a_1,...,a_n) = \Sigma_i\ p_i\ a_i$.

Definition [17, 18]. Let **w** be a weighting vector of dimension **n**, then a mapping OWA_w: $\mathbb{R}^n \rightarrow \mathbb{R}$ is an *Ordered Weighted Averaging (OWA) operator* of dimension n if

$$OWA(a_1,...,a_n) = \Sigma_i\ w_i\ a_{\sigma(i)}$$

where $\{\sigma(1),...,\sigma(n)\}$ is a permutation of $\{1,..,n\}$ such that $a_{\sigma(i-1)} \geq a_{\sigma(i)}$ for all i=2, ..., n. (i.e., $a_{\sigma(i)}$ is the i-th largest element in the collection $a_1,..., a_n$).

The WOWA operator, defined in [15], is a combination function that allows to consider both the relevance of the information sources (the weights in the weighted mean)

and the relevance of the values (the weights in the OWA operator).

Definition. Let **p** and **w** be two weighting vectors of dimension n, then a mapping WOWA:$\mathbb{R}^n$ -> $\mathbb{R}$ is a *Weighted Ordered Weighted Averaging (WOWA) operator* of dimension n if

$$\text{WOWA}(a_1,...,a_n) = \Sigma_i\, \omega_i\, a_{\sigma(i)}$$

where $\{\sigma(1),...,\sigma(n)\}$ is a permutation of $\{1,..,n\}$ such that $a_{\sigma(i-1)} \geq a_{\sigma(i)}$ for all i=2, ..., n. (i.e., $a_{\sigma(i)}$ is the i-th largest element in the collection $a_1,..., a_n$), and the weight ω_i is defined as

$$\omega_i = w^*(\Sigma_{j\leq i}\, p_{\sigma(j)}) - w^*(\Sigma_{j<i}\, p_{\sigma(j)})$$

with w^* a monotone increasing function that interpolates the points $(i/n, \Sigma_{j\leq i}\, w_j)$ together with the point (0,0). w^* is required to be a straight line when the points can be interpolated in this way. From now on, ω represent the set of weights $\{\omega_i\}$, i.e., $\omega=[\omega_1,..., \omega_n]$.

An interpolation method to built the function w^* is described in [16].

3. Learning weights in the Weighted Mean

In this section, we consider learning weigts for the weighted mean. We assume that the dimension of the weighted mean is settled to N. This is, we consider examples where the number of information sources (the number of elements to combine) is N. We assume there that the number of examples is M. Therefore, we will have a set of examples in the form of the ones in table 1.

a^1_1	a^1_2	a^1_3	...	a^1_N	|	m^1
a^2_1	a^2_2	a^2_3	...	a^2_N	|	m^2
...						
a^M_1	a^M_2	a^M_3	...	a^M_N	|	m^M

Table 1. Data examples

3.1. Learning the weights for the weighted mean

Considering a set of examples, the ideal weights are the ones that approximate with a minimum error the solutions of the examples. Using the usual distance to measure the difference between the ideal value (m) and the calculated one ($\Sigma\, a_i p_i$) for each example j, we define an expression that we want to minimize.

$$D(\mathbf{p}) = \sum_{j=1}^{M} \left(\sum_{i=1}^{N} a_i^j\, p_i - m^j \right)^2 \qquad (1)$$

In fact, using the Euclidean norm $\|\mathbf{x}\| = \sqrt{(\mathbf{x}'\,\mathbf{x})}$, we can express D(p) alternatively as $D(p)= \| H\, p - d \|^2$. Here H stands for the set of measurements obtained from the information sources, this is $H=\{a^j_i\}$, and d stands for the "ideal" solutions that should give the system. This is, $d=(m^1, ..., m^M)$. Note also that we use A' to denote the transpose of A. To obtain a solution we have to minimize this distance. This problem is the method of linear least squares [13].

However, expressing the distance in this way and minimizing it is not enough to get a correct weighting vector. We have to minimize this expression, but in order that the solution p is a weighting vector, we need that the weights p_i to be positive and add to one. Therefore, we have a minimization problem with constraints. This constrained problem can be formalized in the following way:

$$\text{Min} \quad D(\mathbf{p}) \qquad (2)$$

such that

$$\sum_{i=1}^{N} p_i = 1 \qquad (3)$$

$$p_i \geq 0 \qquad (4)$$

Using the expression $D(p) = \| H\, p - d\|^2 = (H\, p - d)(H\, p - d)$ and the fact that this expression gets its minimum when we have the minimum of $(1/2)\, p'\, H'\, H\, p - H'\, d\, p$, we can reformulate this problem in the following way (where $Q = H'\, H$ and $c = - H'\, d$).

$$\text{Min} \quad (0.5)\, p'\, Q\, p + c'\, p \qquad (2)$$

such that

$$\sum_{i=1}^{N} p_i = 1 \qquad (3)$$

$$p_i \geq 0 \qquad (4)$$

The problem formulated in this way, is a typical optimization problem. In fact, it is a linear constrain quadratic program. It can be solved with several algorithms (see, for example, [11, 8]). As an example, it can be solved with active set methods.

According to [11], we can say that "the idea underlying active set methods is to partition inequality constraints into two groups: those that are to be treated as active and those that are to be treated as inactive. The constraints treated as inactive are essentially ignored." In fact, active constraints are required to be equal to the limit. In our case, these constraints would be a selection of the weights that are required to be zero.

The idea of active set methods is to define at each step of an algorithm a set of constraints that are to be treated as active. The algorithm then proceeds to move on the surface defined by the working set of constraints to an improved point. In this movement, maybe some new constraints are added to the working set. This is in our case to fix some new weights to be zero. Also, in each iteration, some active constraints are removed from the working set.

The main step in each iteration is to find a local optimum for the constraints in the working set. This is, to find the solution of the following optimization problem (See [11] for details):

minimize $(1/2)\ d'_k\ Q\ d_k + g'_k\ d_k$
subject to the fact that all constraints in the working set are zero

Here, $g_k = c + Q\ p_k$, p_k is the solution found in the previous iteration, and the working set consists on some weights settled to zero and the constraint that all weights add to zero. The solution d_k corresponds to the step to perform from the last solution p_k. As, we do not want to violate the constraints in the working set when the step d_k is added to p_k, we require that d_k does not modify the constraints. That is why it is required that $\sum d_i = 0$ because then $\mathbf{1}(\mathbf{p}+\mathbf{d}) = 1$).

This is, the previous problem is stated as follows (when W_k corresponds to the working set at the iteration k):

minimize $(1/2)\ d'_k\ Q\ d_k + g'_k\ d_k$
subject to $a'_i\ d_k = 0$ for all $i \in W_k$.

This can be solved with the following system of linear equations [11]:

$$\begin{aligned} Qx + A'\lambda &= g_k \\ Ax &= 0 \end{aligned} \qquad (5)$$

where λ are the Lagrange multipliers, and A is the matrix formed with all the coefficients of the active constraints.

The existence of a single or more solutions of this quadratic problem depends on the rank of Q and those of the constraints (the matrix A). In the next section we analyse in some detail aspects related on how to obtain the solution of a given problem in the framework of learning the weights, and the singularity of Q.

3.2. Aspects related to the optimization problem.

When optimization methods are used to learn the weights from examples we are assuming that the problem to minimize has either a single solution or all the solutions are equally adequate for the problem. Moreover, several optimization methods, as the ones based on active sets, assume that the matrix Q that define the problem is nonsingular (a square matrix whose columns are linearly independent). Both cases are not always true in relation to learning weights for an arithmetic mean.

1. All solutions (weighting vectors) are not equally adequate.

   This is so because usually we have that information sources (experts or sensors) are redundant. In this case, several different solutions with the same D(p) can be obtained. In this case, it is better to distribute weights among all the sources than to have the weights accumulated in a single information source. If $p_1=(0.5, 0.5)$ and $p_2=(0,1)$ and $D(p_1)=D(p_2)$ then the p_1 is a better solution because the final aggregated value would be less influenced by the error of a single source. In other words, it is always preferable that the weights are as much distributed as possible.

2. Quadratic problems can have singular matrices.

   We will study this case with more detail below. However, in general, if we have redundant information sources, we have that the output of some of them can be deduced from the values given by the other sources. In this case, the matrix Q is singular.

We analize the latter case below.

1.If the columns of H are linearly independent, then the matrix $Q = H'H$ is nonsingular [13]. In this case, the least squares problem has a single solution. When, we combine this minimization problem with the restrictions, it seems that there is still a single best solution (this has not been proved yet although graphically it seems so). In any case, the linear system used in active set methods to solve the minimization problem has a single solution and is solved by a non-singular matrix. [See, Ref. 11, p.424].

2. When the matrix H has columns that are linealy dependent, then Q is singular. We establish this result in the following proposition.

Proposition. If there is a column k in H that is a linear combination of the other ones in H, then the matrix $Q = H'H$ has the column k and the row k (Q is simmetric) that is a linear combination of the other ones, and the vector $c=H'd$ has the column k that is also a linear combination of the other ones.

In this case, the minimization problem subject to constraints can have several solutions with the same minimization value. The simplest case, is when the column c_k is a linear combination of other columns in such a way that

$$c_k = \sum_{i \neq k} \alpha_i\ c_i$$

with $\sum_i \alpha_i = 1$ for $\alpha_i \geq 0$.

In this case, although there are several solutions, the following proposition holds.

Proposition. If there is a column c_k in H that is a linear combination of the other columns c_i in such a way that:

$$c_k = \sum_{i \neq k} \alpha_i c_i$$

with $\sum_i \alpha_i = 1$ for $\alpha_i \geq 0$, then if the vector p is a solution of the minimization of D(p) with p such that (3) and (4) hold and $p_k \neq 0$, then there is, at least, another vector p' such that D(p')=D(p) such that (3) and (4) also hold.

Proof. Let $p'_i = p_i + \alpha_i p_k$ and $p'_k = 0$. As $\sum \alpha_i = 1$, then it holds, $\sum_i p'_i = 1$ and $p_i \geq 0$. Moreover, as c_k is a linear combination in H, then the column k in Q and d is also a linear combination. This help to prove that D(p)=D(p').

This proposition implies that we can eliminate the column corresponding to k in the matrix H, and solve the minimization problem without it. The minimal D(p) will be the same.

In general, however, it is not true that the linear combination of c_k is obtained with $\sum \alpha_i = 1$. If this condition does not hold, then it is possible that there is a single solution of the minimization problem, and the algorithm can find this solution because in (5) we need Q to be definite positive but only on the subspace M={x: Ax=0}. Let us consider the following example:

Example. Let us learn the weights from a single observation H=(1 3) and d=(1). In this case, the matrix Q and the vector c are defined as follows:

$$Q = \begin{pmatrix} 1 & 3 \\ 3 & 9 \end{pmatrix} \quad c = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

In this case, we have that $c_1 = c_2/3$ and thus the previous proposition does not apply. In this example there is a single vector p that minimizes D(p). The solution is p=(1 0).

In fact, this problem can be solved by active set methods because, although the matrix Q is singular, when the constraint corresponding to the weights is added, it is possible to find a solution. If we consider the system of equations in (5) for this problem, with p_k=(1/2, 1/2), $d_k = (x_1, x_2)$ and considering in the working set only the equality constraint that $x_1 + x_2 = 0$, we will get g_k=(3 9) and the following system of equations:

$$\begin{aligned} x_1 + 3x_2 + \lambda &= 1 \\ 3x_1 + 9x_2 + \lambda &= 3 \\ x_1 + x_2 &= 0 \end{aligned}$$

This system has a single solution: x_1=1.5, x_2=-1.5, λ=0. d=(1.5, -1.5) corresponds to the direction to move from p=(1/2,1/2). Due to the constraints, we arrive to p=(1,0) that is the single solution of the problem. ❑

Example: We consider below an example of the application of the method described above to determine the weights of the weighted mean. The example consists on a data matrix H (with the values of the information sources) with 10 examples and 5 weights, and its corresponding solution vector d. We show the results of three cases. In the first one, we have a known weighting vector (p=(0.1, 0.2, 0.3, 0.4, 0.0). In the second and three case, we have introduced error in the data. We have introduced the error keeping the matrix constant and modifying the solution vectors. This has been done, instead of what would be the more natural alternative: matrices with white noise and the same solution vector, to make the calculus simpler and to show the results easily.

0.3	0.4	0.5	0.1	0.2	0.3	0.32	0.2
0.2	0.1	0.4	0.1	0.5	0.2	0.25	0.1
0.2	0.5	0.8	0.0	0.1	0.36	0.37	0.2
1.0	0.5	0.3	0.6	0.7	0.53	0.6	0.4
0.2	0.1	0.1	0.1	0.1	0.11	0.12	0.03
0.7	0.7	0.7	0.7	0.7	0.7	0.76	0.5
0.4	0.8	0.2	0.8	0.6	0.58	0.64	0.3
0.3	0.2	0.1	0.4	0.3	0.26	0.29	0.1
0.6	0.8	0.7	0.2	0.5	0.51	0.56	0.3
0.1	0.5	0.2	0.6	0.4	0.41	0.45	0.3

Table 2. Data matrix H and three solution vector

Below are given the results corresponding to these three solution vectors. As expected, the weights in the first example are the ones used to generate the solution vector. In the second and third column, as larger errors were considered, the resulting weighting vector is less similar to the first one. Also, in relation to computational issues, the first two results were obtained in a single step, while in the third case three steps were needed. In all three cases, the initial weighting vector was p=(.2, .2, .2, .2, .2).

p = (0.100, 0.200, 0.300, 0.400, 0.000)
p = (0.080, 0.350, 0.153, 0.218, 0.198)
p = (0.058, 0.000, 0.420, 0.522, 0.000)

In the last case, to find a feasible solution two constraints have to be added in the working set. These relate to the weights p_2 and p_5. This is due to the fact that the minimum is obtained with negative weights.

3.3. On linearly dependent columns in matrix H.

In the previous section we have considered the fact that in real problems, the existence of linearly dependent columns is something possible due to the existence of redundant information sources. It has been said that in this case, it is better not to accumulate the weights into a single source but to distribute this weight among the sources (maximum dispersion on the values).

A measure that have already been used in several works [17, 12, 5] to express the dispersion of weights is the entropy. This is defined for a weigting vector p in the following way:

$$E(p) = - \Sigma_i p_i \ln p_i$$

It is convenient, however, to extend the domain so that zero values (zero weights) are allowed: an expression which appears formally as 0 log 0 is defined to be zero [4].

The entropy is defined so that maximal entropy (with the restriction that $\Sigma\ p_i = 1$) corresponds to maximal dispersion and is achieved when all the weights are defined as $p_i = 1/N$. Thus, with a maximum entropy, the influence of a particular information source is minimized. Thus, if we have that there exist several solutions of the quadratic problem stated above with the same minimal distance Δ, then we want to select the one with maximal entropy.

Due to the fact that *a priori* we do not know which is the minimal distance Δ, we need to solve the general problem in two steps. In the first step, we solve the minimization problem stated above, determine if there is a single solution or there exist several ones and settle the minimal distance Δ. In the second step, we find the weighting vector p that maximizes the entropy and has a minimal distance Δ. The second step is a non-linear optimization problem with non-linear constraints.

1. Solve the minimization problem:

Min $D(p) = (0.5)\ p'\ Q\ p + c'\ p$ (2)

such that

$$\sum_{i=1}^{N} p_i = 1 \qquad (3)$$

$$p_i \geq 0 \qquad (4)$$

2. If there is a single solution p that minimizes D(p), then return p

3. If there are more than a single solution, then

Let $\Delta = D(p)$ for one of the solutions of the previous problem.

Return the weighting vector p such that

Min $- E(p) = \Sigma_i p_i \ln p_i$

such that

$$(0.5)\ p'\ Q\ p + c'\ p = \Delta$$

$$\sum_{i=1}^{N} p_i = 1$$

$$p_i \geq 0$$

Althogh it is possible that some columns in H are linearly dependent of some others, in practical situations, this should not be frequent. This is so because the values supplied by the sensors are subject to errors.

4. Learning weights in the OWA and the WOWA operators

The learning of weights in the OWA operator is analogous to the case of the weighted mean. The OWA operator is a weighted mean once the elements to weight are ordered. Therefore, the method described above can be used to learn the weights in this case.

In [6] it is applied the gradient descent technique to find the weights of the OWA operator. It is not necessary to use such an approximation technique because with the general procedure explained above more exact results can be obtained. To show the applicability of our approach to the OWA operator we use the same example used in [6]. We will show that our method gives better results.

Example [6]: We assume the collection of samples in Table 3. Each sample consists of 4 information sources and the corresponding aggregated value:

0.4	0.1	0.3	0.8	0.24
0.1	0.7	0.4	0.1	0.16
1.0	0.0	0.3	0.5	0.15
0.2	0.2	0.1	0.4	0.17
0.6	0.3	0.2	0.1	0.18

Table 3. Example in [6]

In order to apply the previous learning algorithm, as the weights in the OWA have to be considered in relation to the position of the elements, we need to order the elements. After ordering them we get:

0.8	0.4	0.3	0.1	0.24
0.7	0.4	0.1	0.1	0.16
1.0	0.5	0.3	0.0	0.15
0.4	0.2	0.2	0.1	0.17
0.6	0.3	0.2	0.1	0.18

Table 4. After having ordered the values in each row

Now, we can learn the weights with the previous result. The resulting weighting vector is the following one:

$$p = (0.6677,\ 0.2293,\ 0.0,\ 0.1031)$$

This solution have been found in two steps if we begin with an initial weighting vector p equal to (.25, .25, .25, .25).

For the same example, and with the same initial weighting vector p=(.25, .25, .25, .25) in [6] it is given the following solution after 150 iterations:

$$p = (0.08,\ 0.11,\ 0.14,\ 0.67)$$

It is easy to see that our weighting vector is more precise in reproducing the aggregated value when we calculate the distance D(p) with the OWA operator. This is, the distance defined as:

$$D(p) = \sum_{j=1}^{5} \left(\sum_{i=1}^{4} a^j_{\sigma(i)}\, p_i - m^j \right)^2$$

with $\sigma(i)$ a permutation as in section 2. If we calculate the distance with the two weights we obtain the results given next that show that our method gives a better performance.

[6]:	0.002156
Our method:	0.001264

4.1. Learning the weights in the WOWA operator.

The use of least squares method can be applied also to the WOWA operator. In this case, however, the distance has to be defined in terms of the operator with the two sets of weights. Thus, the general form of the method is as follows:

$$\text{Min} \qquad D(\mathbf{p}) = \sum_{j=1}^{M} \left(\sum_{i=1}^{N} \text{WOWA}_{w,p}(a^j) - m^j \right)^2$$

such that

$$\sum_{i=1}^{N} p_i = 1$$

$$\sum_{i=1}^{N} w_i = 1$$

$$p_i \geq 0,\ w_i \geq 0$$

The problem formulated in this way can be solved with optimization algorithms, and, as in the case of the weighted mean, there exists software that can solve this problem. They usually use the gradient vector and the Hessian matrix. Both elements can be built from the definition of the WOWA operator and the interpolation method defined to build the quantifier from the weights w_i.

5. Conclusions

In this paper we have studied some aspects related with the learning of weights of the weighted mean, the OWA operator and the WOWA operator. The paper is focused to the case of the weighted mean although similar considerations can be applied to the OWA operator. In the case of the latter we have compared the results obtained with our method with the ones in given in [6].

In the case of the WOWA operator only some general guidelines have been considered and further work is planed on the practical determination of the weights.

6. Acknowledgements

The author acknowledge partial support of the CYCIT project SMASH (TIC96-1138-C04-04). The authors are also indebted to Jordi Castro for helpful comments and suggestions.

7. References

[1] Aczél, J., On weighted synthesis of judgements, *Aequationes Math.*, 27 288-307 (1984).

[2] Aczél, J., Alsina, C., On synthesis of Judgements, *Socio-Econom Plann. Sci.*, 1986.

[3] Amat, J., Esteva, F., López de Mántaras, R., Autonomous Navigation Troup for Cooperative Modelling of Unknown Environments, in *Proceedings of the Int. Conference on Advanced Robotics* (ICAR'95), Sant Feliu de Guixols, Catalunya, Spain, pp. 383- 389, 1995.

[4] Ash, R., Information theory, Interscience publishers, (1965).

[5] Carbonell, M., Mas, M., Mayor, G., On a class of Monotonic Extended OWA Operators, Proceedings of the Sixth IEEE International Conference on Fuzzy Systems (IEEE-FUZZ'97), Barcelona, Catalunya, Spain, (1997), 1695-1699.

[6] Filev. D. P., Yager, R. R., On the issue of obtaining OWA operator weights, *Fuzzy Sets and Systems*, 94 (1998) 157-169.

[7] Fodor, J., Marichal, J.-L., Roubens, M., Characterization of the Ordered Weighted Averaging Operators, *IEEE Trans. on Fuzzy Systems*, 3:2, 236-240 (1995).

[8] Gill, P.E., Murray, W., Wright, M. H., *Practical Optimization*, Academic Press, (1981).

[9] Grabisch, M., Nguyen, H.T., Walker, E.A., *Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference*, Kluwer Academic Publishers, Dordreht, The Netherlands, (1995).

[10] López, B., Álvarez, S., Millán, P., Puig, D., Riaño, D., Torra, V., Multistage vision system for road lane markings and obstacle detection, Proceedings of *Euriscon '94*, Malaga, pp. 489-497, (1994).

[11] Luenberger, D. G., *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Menlo Park, California, (1973).

[12] O'Hagan, M., Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic, *Proceedings of the 22nd Annual IEEE Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 1988, 681-689.

[13] Stoer, J., Bulirsch, R., Introduction to numerical analysis, Springer-Verlag, (1980).

[14] Torra, V., Cortés, U., Towards an automatic consensus generator tool: EGAC, *IEEE Transactions on Systems, Man and Cybernetics*, 25:5 888-894 (1995).

[15] Torra, V., The Weighted OWA operator, *International Journal of Intelligent Systems*, 12 (1997) 153-166.

[16] Torra, V., The WOWA operator and the interpolation function W*: Chen and Otto's interpolation method revisited, *Fuzzy Sets and Systems* (in press).
[17] Yager, R.R., Families of OWA operators, *Fuzzy Sets and Systems*, 59 125-148 (1993).
[18] Yager, R.R., On ordered weighted averaging aggregation operators in multi-criteria decision making, *IEEE Trans. on SMC*, 18 183-190 (1988).

UN ORDEN TOTAL ENTRE NÚMEROS GRADUALES USADO EN PROBLEMAS DE DECISIÓN

José A. Herencia
Dep. Matemáticas
Facultad de Ciencias, Univ. de Córdoba
E-mail: malhegoj@uco.es

M. Teresa Lamata
Dep. Ciencias de la Computación e I. A.
E.T.S.I. Informática, Univ. de Granada
E-mail: mtl@decsai.ugr.es

Resumen

Al resolver un problema de decisión clásico con recompensas difusas se requiere un método de ordenación total de números difusos, con objeto de seleccionar la alternativa correspondiente a la mejor recompensa difusa. Aquí se considera una definición de tal orden total basada en el grado de optimismo/pesimismo y la preferencia entre riesgo/seguridad del decisor. También se usan los números graduales (dada su analogía con los números difusos), que permiten extender directamente los cálculos efectuados sobre intervalos reales.

Palabras clave: Números difusos, números graduales, relación de orden, problemas de decisión, recompensas difusas.

1 INTRODUCCIÓN

En muchas aplicaciones prácticas, el conocimiento de las cantidades o medidas manejadas no es un número real exacto sino que se modeliza mejor por el concepto de número difuso. El principio de extensión de Zadeh ofrece, a partir de las operaciones con números reales, una aritmética con números difusos respecto a cuyo uso existe un gran consenso. Sin embargo, el orden usual de la recta real $\mathbb{R}$ no se extiende de forma natural a los números difusos; lo cual origina gran variedad de métodos de ordenación de números difusos [1, 2, 3, 5, 8, 10, 11, 17] ... Parte de estos métodos mantiene la vaguedad inherente al concepto de número difuso ofreciendo un orden difuso entre los números a ordenar. En cambio hay otros métodos que ofrecen un orden "crisp" (o nítido) entre los números difusos.

Algo análogo ocurre con la formalización teórica de los problemas de decisión. Los distintos aspectos del problema (alternativas, estados de la Naturaleza, ambiente en que se presentan estos estados, recompensas, objetivos, ...) suelen estar más o menos afectados de cierto grado de ambigüedad, vaguedad o incertidumbre. La conocida formulación de Bellman y Zadeh [4] expresa tanto las condiciones del problema como su solución en términos difusos. En cambio, en la extensa literatura sobre Teoría de la Decisión en ambiente difuso, se reflejan situaciones prácticas donde algunos aspectos son difusos y otros no.

En este artículo consideramos las herramientas necesarias para resolver un problema de decisión clásico (con un número finito de alternativas y un número finito de estados de la Naturaleza), pero donde las recompensas vienen dadas por números graduales de Zadeh. Estos números se definen por analogía a las familias de α-cortes de números difusos de Zadeh. Así, cada número gradual de Zadeh es una familia decreciente de intervalos reales acotados [13, 14]. La ventaja que presentan respecto a los números difusos es que dicha familia no está sujeta a las condiciones del Teorema de Representación de Negoita y Ralescu [16], permitiendo extender sin dificultad los distintos conceptos y operaciones usuales definidos sobre intervalos. Como tales extensiones son análogas a las comúnmente usadas para α-cortes de números difusos, hay un gran paralelismo entre las propiedades de ambos tipos de números [6]. El uso de números graduales en los problemas de decisión ya fue sugerido hace dos años [15]; aquí continuamos esa línea profundizando en los métodos de comparación de tales números necesarios para la solución del problema. Los resultados considerados son aplicables (por los motivos antedichos) a problemas de decisión con recompensas difusas.

Los métodos usuales de resolución del problema de decisión planteado en sus distintos ambientes (criterios maximin, maximax y de Hurwicz en ambiente de incertidumbre, criterio de la esperanza matemática en ambiente de riesgo, etc.) requieren efectuar operaciones y ordenaciones entre los números graduales fijados como recompensas. Tal como se ha indicado antes, hay una pauta clara respecto a la aritmética con números graduales, mientras que no ocurre lo mismo con su ordenación. Como el orden que establezcamos se traducirá en un orden entre alternativas, debe tratarse además

de un orden total con objeto de seleccionar la alternativa considerada como "la mejor". Por tanto, el aspecto fundamental a estudiar es el establecimiento de un orden nítido y total entre números graduales. Con tal finalidad se usan dos parámetros subjetivos, dependientes del decisor: el "grado de optimismo o pesimismo" y la "preferencia entre riesgo y seguridad". En el uso del primer parámetro seguimos las ideas ya estudiadas por González y de Campos para definir un orden parcial entre números difusos [7, 11] [12]. El segundo parámetro se introduce en este trabajo y proporciona un método para obtener el orden total requerido (a partir del orden parcial determinado por el grado de optimismo).

En la sección 2 hacemos una breve revisión sobre los números graduales de Zadeh, considerando sólo las operaciones necesarias para aplicar los criterios antes mencionados al resolver un problema de decisión. En la sección 3 consideramos ciertos órdenes (parciales y totales) entre intervalos reales para, posteriormente (sección 4) extender estos órdenes a los números graduales de Zadeh, llegando por último a un orden total entre ellos con el que completamos las herramientas necesarias para cubrir nuestro objetivo.

2 ASPECTOS BÁSICOS SOBRE LOS NÚMEROS GRADUALES

Los números graduales de Zadeh se definen como familias decrecientes de intervalos reales acotados, indexadas por el intervalo $[0,1]$. Sus operaciones se definen como una extensión natural de las operaciones con intervalos, estableciéndose así fácilmente resultados análogos a los generalmente usados con las familias de α-cortes de números difusos. Un estudio detallado de tales definiciones y resultados aparece en [13] y [14]. Aquí nos limitamos a enunciar la definición de número gradual de Zadeh (usando intervalos compactos) y de las operaciones entre los mismos que se usan para resolver un problema de decisión. En la sección 4 consideraremos el orden entre tales números, que es el aspecto central de este artículo.

Se llama **número gradual de Zadeh** a toda aplicación $\psi : [0,1] \to \{[A,B] : A \leq B \in \mathbb{R}\}$, que asigna a cada $\alpha \in [0,1]$ el intervalo $[a(\alpha), b(\alpha)]$ verificando que $\forall \alpha, \beta \in [0,1], [\alpha < \beta \Rightarrow a(\alpha) \leq a(\beta) \leq b(\beta) \leq b(\alpha)]$. Por tanto dichos números, que en lo sucesivo denominaremos simplemente **números graduales** (ya que son los únicos que usamos aquí), están determinados por una pareja de funciones $a, b : [0,1] \to \mathbb{R}$, que verifican las tres condiciones siguientes: la función $a(\alpha)$ es creciente, la función $b(\alpha)$ es decreciente y $a(1) \leq b(1)$.

Consideremos ahora las operaciones que aquí nos interesan en el conjunto de números graduales (que en lo sucesivo notaremos por $\mathcal{G}_Z(\mathbb{R})$). Para números graduales arbitrarios $\psi(\alpha) = [a(\alpha), b(\alpha)]$, $\psi_1(\alpha) = [a_1(\alpha), b_1(\alpha)]$, $\psi_2(\alpha) = [a_2(\alpha), b_2(\alpha)]$ y cualquier número real P, se define:

$$(P\psi)(\alpha) := \begin{cases} [Pa(\alpha),\ Pb(\alpha)], & \text{si } P \geq 0 \\ [Pb(\alpha),\ Pa(\alpha)], & \text{si } P < 0 \end{cases} \quad (1)$$

$$(\psi_1 + \psi_2)(\alpha) := [a_1(\alpha) + a_2(\alpha), b_1(\alpha) + b_2(\alpha)] \quad (2)$$

NOTA: Los órdenes totales entre números graduales que obtendremos en la última sección son los que pueden usarse para ordenar cualquier conjunto finito de tales números, proporcionando así el correspondiente mínimo y máximo. No obstante, también puede usarse el siguiente concepto de mínimo y máximo, obtenido como la extensión de las correspondientes operaciones con números reales:

$$\min\{\psi_1, \psi_2\}(\alpha) := [\min\{a_1(\alpha), a_2(\alpha)\}, \min\{b_1(\alpha), b_2(\alpha)\}] \quad (3)$$

$$\max\{\psi_1, \psi_2\}(\alpha) := [\max\{a_1(\alpha), a_2(\alpha)\}, \max\{b_1(\alpha), b_2(\alpha)\}] \quad (4)$$

El conjunto $\mathcal{G}_Z(\mathbb{R})$ es demasiado amplio en el sentido de que las funciones $a(\alpha)$ y $b(\alpha)$ que definen cada número gradual están sujetas a muy pocas restricciones. De esta forma, podemos considerar funciones que raramente aparecerían en números graduales usados como recompensas de un problema de decisión. Para tales aplicaciones bastará considerar funciones bastante más específicas. A modo de ejemplo, definamos dos tipos concretos de números graduales:

DEFINICIÓN 2.1 *Denominamos:*

- **número gradual triangular** *a cada número gradual determinado por dos funciones lineales $a, b : [0,1] \to \mathbb{R}$, cuyas gráficas forman un triángulo. Más precisamente, cualesquiera tres números reales $A \leq M \leq B$, determinan el número gradual triangular $\tau_{(A,M,B)}$ definido por las funciones:*

  $$a(\alpha) = A + \alpha(M - A), \qquad b(\alpha) = B - \alpha(B - M).$$

- **número 3-gradual** *a cada número gradual determinado por dos funciones escalonadas $a, b : [0,1] \to \mathbb{R}$, que son constantes sobre los intervalos $[0, 1/3]$, $(1/3, 2/3]$ y $(2/3, 1]$. Así, para cualesquiera números reales $A \leq A' \leq A'' \leq$*

$B'' \leq B' \leq B$, *tenemos el número 3-gradual* $\xi_{(A,A',A'',B'',B',B)}$ *definido por las funciones:*

$$\begin{aligned} a(\alpha) &= A, \quad b(\alpha) = B, \quad si \ \ 0 \leq \alpha \leq 1/3, \\ a(\alpha) &= A', \quad b(\alpha) = B', \quad si \ \ 1/3 < \alpha \leq 2/3, \\ a(\alpha) &= A'', \quad b(\alpha) = B'', \quad si \ \ 2/3 < \alpha \leq 1. \end{aligned}$$

Es inmediato efectuar las operaciones (1) y (2) con estos tipos de números graduales, obteniendo:

PROPOSICIÓN 2.2 *1. Para los números graduales triangulares se verifica:*

$$\tau_{(A,M,B)} + \tau_{(A',M',B')} = \tau_{(A+A',M+M',B+B')},$$

$$P\tau_{(A,M,B)} = \begin{cases} \tau_{(PA,PM,PB)}, & si \ P \geq 0 \\ \tau_{(PB,PM,PA)}, & si \ P \leq 0 \end{cases}$$

2. Para los números 3-graduales arbitrarios $\xi = \xi_{(A,A',A'',B'',B',B)}$, $\xi_1 = \xi_{(A_1,A_1',A_1'',B_1'',B_1',B_1)}$ *y* $\xi_2 = \xi_{(A_2,A_2',A_2'',B_2'',B_2',B_2)}$, *se verifica:*

$$\xi_1 + \xi_2 = \xi_{(A_1+A_2,A_1'+A_2',A_1''+A_2'',B_1''+B_2'',B_1'+B_2',B_1+B_2)}$$

$$P\xi = \begin{cases} \xi_{(PA,PA',PA'',PB'',PB',PB)}, & si \ P \geq 0 \\ \xi_{(PB,PB',PB'',PA'',PA',PA)}, & si \ P \leq 0 \end{cases}$$

NOTAS:

1. Ni el mínimo (3) ni el máximo (4) de dos números graduales triangulares tiene por qué ser otro número gradual triangular (ejemplo: $\tau_{(0,3,4)}$ y $\tau_{(1,2,5)}$).

2. Tanto el mínimo (3) como el máximo (4) de dos números 3-graduales es otro número 3-gradual (determinado por los mínimos o máximos, respectivamente, de los 6 parámetros correspondientes), pero que no tiene por qué coincidir con ninguno de los dos números de partida. Por ejemplo, dados los números $\xi_1 = \xi_{(0,3,6,7,10,12)}$ y $\xi_2 = \xi_{(4,8,9,9,10,10)}$, su mínimo y su máximo son, respectivamente, $\xi_{(0,3,6,7,10,10)}$ y $\xi_{(4,8,9,9,10,12)}$.

3. Basándonos en los tipos usuales de números difusos que aparecen en las más variadas aplicaciones, resulta inmediato generalizar los dos tipos de números graduales antes definidos. Así podemos considerar:

- Los *números L-R-graduales* (análogos a los números L-R-difusos introducidos por Dubois y Prade [9]), tomando ciertas funciones L y R como base para definir las funciones $a(\alpha)$ y $b(\alpha)$. Las cuales quedan entonces determinadas por los 4 parámetros $a(0)$, $a(1)$, $b(1)$ y $b(0)$.
- Los *números n-graduales*, que pueden definirse fijando n valores del intervalo $[0,1]$: $0 < \alpha_1 < \alpha_2 < \ldots < \alpha_n = 1$ y tomando funciones escalonadas a y b que sean constantes sobre los n subintervalos correspondientes. Este procedimiento es análogo al uso del retículo $L = (\alpha_1, \alpha_2, \ldots, \alpha_{n-1}, 1)$, en lugar del intervalo unidad, cuando se considera el concepto de conjunto L-difuso definido por Goguen.

3 ORDENACIÓN DE INTERVALOS

En esta sección consideramos distintos órdenes en el conjunto de intervalos reales compactos $\mathcal{C} = \{[A,B] : A \leq B \in \mathbb{R}\}$, que nos servirán de base para obtener el orden total que nos interesa establecer entre números graduales. Teniendo en cuenta el problema que motiva nuestro estudio, podemos pensar en los intervalos dados como estimaciones imprecisas de las recompensas que pueden obtenerse en ciertas circunstancias.

Dados cualesquiera dos intervalos $I_1 = [A_1, B_1]$ e $I_2 = [A_2, B_2]$, el orden entre ambos está claro cuando $A_1 \leq A_2$ y $B_1 \leq B_2$: en ese caso es obviamente preferible I_2 a I_1, lo que puede notarse por $I_1 \leq I_2$ (por simetría, tendríamos igualmente dado el caso $I_2 \leq I_1$). La situación no resulta tan obvia cuando uno de los intervalos está estrictamente incluido en el otro; es decir cuando $A_2 < A_1 \leq B_1 < B_2$ (o el caso simétrico que se trataría de forma idéntica). Dada esta situación, podemos considerar el grado de optimismo del decisor, quien mostrará sus preferencias atendiendo más hacia un extremo u otro del intervalo. Así, si se trata de un individuo totalmente optimista, esperará obtener la máxima recompensa posible, atendiendo exclusivamente a los valores B_1 y B_2, por lo que preferirá I_2 (estableciendo que $I_1 \leq I_2$). En cambio, un decisor totalmente pesimista preferiría I_1 (considerando que $I_2 \leq I_1$) al atender exclusivamente a los valores A_1 y A_2. Tanto estas dos situaciones extremas como todos los casos intermedios entre ambas pueden recogerse considerando un grado de optimismo $\lambda \in [0,1]$, según el cual la recompensa representada por el intervalo $[A,B]$ será estimada por el número $\lambda B + (1-\lambda)A$.

Para englobar todos los casos considerados en el párrafo anterior bastaría con fijar el grado de optimismo $\lambda \in [0,1]$ (asociado a cada decisor) y establecer la siguiente definición de orden $\leq_\lambda$ en $\mathcal{C}$ (que indica la preferencia de tal decisor):

$$[A_1,B_1] \leq_\lambda [A_2,B_2] \ :\Leftrightarrow \ [A_1 \leq A_2 \text{ y } B_1 \leq B_2] \text{ ó}$$

$$\lambda B_1 + (1-\lambda)A_1 < \lambda B_2 + (1-\lambda)A_2. \qquad (5)$$

Con objeto de simplificar la expresión anterior, interesa distinguir dos situaciones:

(1^a) Sea $\lambda \in \{0,1\}$. Entonces, fijado el valor $\lambda = 0$, observemos que el caso $A_1 < A_2$ (resp. $A_2 < A_1$) implica que $[A_1,B_1] \leq_0 [A_2,B_2]$ (resp. $[A_2,B_2] \leq_0 [A_1,B_1]$), mientras que en el caso $A_1 = A_2$ se establece la comparación entre ambos intervalos únicamente a través de los valores de B_1 y B_2. Para $\lambda = 1$ tenemos una situación análoga. En definitiva, se trata de los órdenes lexicográficos sobre la pareja de números formada por los extremos del intervalo, comenzando por ordenar el extremo inferior si $\lambda = 0$ ó el extremo superior si $\lambda = 1$. Por tanto, se trata de órdenes totales en ambos casos.

(2^a) Sea $\lambda \in (0,1)$. Si suponemos que $A_1 \leq A_2$ y $B_1 \leq B_2$ siendo estricta alguna de las dos desigualdades (es decir, tratándose de intervalos distintos $I_1 \neq I_2$) entonces, cualquiera sea el valor de $\lambda \in (0,1)$, se cumple que $\lambda B_1 + (1-\lambda)A_1 < \lambda B_2 + (1-\lambda)A_2$. Por tanto, la definición (5) equivale a que sea $I_1 = I_2$ o que se cumpla la última desigualdad estricta. Considerando esta otra expresión equivalente es inmediato comprobar las tres propiedades reflexiva, antisimétrica y transitiva que nos aseguran que $\leq_\lambda$ es una relación de orden. Pero, para $\lambda \in (0,1)$, este orden no es total ya que son no comparables cualesquiera dos intervalos distintos $I_1 \neq I_2$ tales que $\lambda B_1 + (1-\lambda)A_1 = \lambda B_2 + (1-\lambda)A_2$.

En conclusión, queda justificada la siguiente definición que recopila los argumentos previos:

DEFINICIÓN 3.1 *En el conjunto $\mathcal{C}$ de intervalos reales compactos se definen las siguientes relaciones de orden (expresadas para elementos arbitrarios $I_1 = [A_1, B_1]$ e $I_2 = [A_2, B_2]$):*

- *Para cada $\lambda \in (0,1)$, tenemos el orden parcial $\leq_\lambda$ dado por:*

  $I_1 \leq_\lambda I_2 \quad :\Leftrightarrow \quad I_1 = I_2$ *ó* $\lambda B_1 + (1-\lambda)A_1 < \lambda B_2 + (1-\lambda)A_2$

- *Para $\lambda \in \{0,1\}$, tenemos los dos órdenes totales (lexicográficos) $\leq_\lambda$ definidos como sigue:*

  $I_1 \leq_0 I_2 \;\Leftrightarrow\; A_1 < A_2$ *ó* $[A_1 = A_2$ *y* $B_1 \leq B_2]$.

  $I_1 \leq_1 I_2 \;\Leftrightarrow\; B_1 < B_2$ *ó* $[B_1 = B_2$ *y* $A_1 \leq A_2]$.

Los órdenes $\leq_\lambda$ son totales cuando $\lambda \in \{0,1\}$ porque, en caso de darse la igualdad $\lambda B_1 + (1-\lambda)A_1 = \lambda B_2 + (1-\lambda)A_2$, que corresponde a la igualdad de un extremo del intervalo, se atiende al valor del otro extremo para establecer el orden. Lo que equivale a atender a la amplitud del intervalo (prefiriéndose el intervalo más amplio cuando $\lambda = 0$ y el más estrecho cuando $\lambda = 1$). Este hecho sugiere considerar un método análogo para extender los órdenes $\leq_\lambda$ cuando $\lambda \in (0,1)$, convirtiéndolos también en órdenes totales, que son los que nos interesan. El método consiste en atender a otra característica subjetiva asociada con el decisor: su preferencia por el riesgo o por la seguridad. A saber, ante intervalos distintos pero indiferentes (por cumplir la igualdad $\lambda B_1+(1-\lambda)A_1 = \lambda B_2+(1-\lambda)A_2$), un decisor arriesgado preferirá el intervalo de mayor amplitud donde tiene posibilidades de mayor recompensa; mientras que un decisor cauto preferirá el intervalo de menor amplitud, donde tiene mayor seguridad de obtener sus posibles valores (y menos riesgo de posibles pérdidas). En definitiva, además del "*grado de optimismo*" $\lambda \in [0,1]$, consideramos la característica "*preferencia entre riesgo y seguridad*" que notamos por $x \in \{r, s\}$. De esta forma, junto a los órdenes totales $\leq_0$ y $\leq_1$, podemos definir otros órdenes totales $\leq_{\lambda r}$ y $\leq_{\lambda s}$ (asociados a cada $\lambda \in (0,1)$ y a cada decisor "*arriesgado*" y "*seguro*", respectivamente), de la siguiente manera:

DEFINICIÓN 3.2 $\forall \lambda \in (0,1)$, $\forall x \in \{r, s\}$, *se define en $\mathcal{C}$ el orden total $\leq_{\lambda x}$ por las siguientes expresiones (donde se usa la Definición 3.1):*

$I_1 \leq_{\lambda r} I_2 \;:\Leftrightarrow\; I_1 \leq_\lambda I_2$ *ó* $[\lambda B_1 + (1-\lambda)A_1 = \lambda B_2 + (1-\lambda)A_2$ *y* $B_2 - A_2 > B_1 - A_1]$.

$I_1 \leq_{\lambda s} I_2 \;:\Leftrightarrow\; I_1 \leq_\lambda I_2$ *ó* $[\lambda B_1 + (1-\lambda)A_1 = \lambda B_2 + (1-\lambda)A_2$ *y* $B_2 - A_2 < B_1 - A_1]$.

Obsérvese que la denominación de "*riesgo*" dada a la preferencia por intervalos de mayor amplitud no tiene nada que ver con el "*ambiente de riesgo*", que es la denominación atribuida al caso en que los estados de la Naturaleza en un problema de decisión están determinados por una distribución de probabilidad.

Por otro lado, téngase en cuenta que si $\lambda = 0$ entonces, una vez fijado el extremo inferior $A = \lambda B + (1-\lambda)A$ del intervalo, la preferencia por intervalos de mayor amplitud no entraña ningún riesgo ya que dicha amplitud sólo puede ampliar el valor del extremo superior B. Por tal motivo, siempre que λ sea un valor próximo a cero, es lógico elegir el parámetro $x = r$. Razonando de igual forma se deduce que para cualquier λ próximo a uno, es lógico elegir el parámetro $x = s$. En cambio, para valores intermedios de λ (sobre todo para valores próximos a $1/2$), el parámetro x puede considerarse independiente de λ; siendo entonces cuando adquiere pleno sentido la denominación atribuida de "*preferencia entre riesgo y seguridad*".

4 ORDENACIÓN DE NÚMEROS GRADUALES

Con objeto de establecer un orden total en $\mathcal{G}_Z(\mathbb{R})$, haciendo uso de las Definiciones 3.1 y 3.2, interesa resumir en un solo intervalo la información aportada por cada número gradual $\psi(\alpha) = [a(\alpha), b(\alpha)]$. Como las funciones $a(\alpha)$ y $b(\alpha)$ son monótonas, se pueden integrar. Por tanto, podemos usar el Teorema de la media del Cálculo Integral, considerando las constantes que representan a esas dos funciones; de la siguiente forma:

DEFINICIÓN 4.1
Consideremos la aplicación de $\mathcal{G}_Z(\mathbb{R})$ en $\mathcal{C}$ que asigna a cada número gradual $\psi(\alpha) = [a(\alpha), b(\alpha)]$ el intervalo real $[\mathcal{A}(\psi), \mathcal{B}(\psi)]$ definido por:

$$\mathcal{A}(\psi) := \int_0^1 a(\alpha)\, d\alpha, \qquad \mathcal{B}(\psi) := \int_0^1 b(\alpha)\, d\alpha.$$

El uso directo de los órdenes definidos en $\mathcal{C}$ sugiere considerar los correspondientes órdenes en el conjunto cociente obtenido en $\mathcal{G}_Z(\mathbf{R})$ mediante la relación de equivalencia asociada a la aplicación dada en la definición anterior . De hecho, el orden parcial así establecido a través de la relación $\leq_\lambda$, consistente en la comparación de dos números graduales mediante el valor asociado $\mathcal{M}_\lambda(\psi) := \lambda\mathcal{B}(\psi) + (1-\lambda)\mathcal{A}(\psi)$, es totalmente análogo al método propuesto por González para la comparación de números difusos [11, 12]; método que generaliza al dado por Yager [17], correspondiente al caso particular $\lambda = 1/2$. Observemos que González usa la integración de Stieljes respecto a cualquier medida aditiva definida sobre $[0,1]$ (la cual puede anularse fuera de cierto $Y \subset [0,1]$ formado por aquellos niveles α a los que el decisor concede importancia); mientras que aquí nos hemos limitado al caso más usual de la medida de Lebesgue (como también hace Yager).

Pero la comparación de números graduales mediante los intervalos $[\mathcal{A}(\psi), \mathcal{B}(\psi)]$ sólo puede ofrecer un orden parcial en $\mathcal{G}_Z(\mathbf{R})$, aun cuando usemos los órdenes totales ya definidos en $\mathcal{C}$. En efecto, cualquiera sea el orden usado en $\mathcal{C}$, no podremos comparar cualesquiera dos números graduales que, aun siendo diferentes, originen el mismo intervalo $[\mathcal{A}, \mathcal{B}]$. Ahora bien, para obtener el orden total buscado en $\mathcal{G}_Z(\mathbf{R})$, bastará con extender el orden así dado (basado en cualquiera de los órdenes totales de $\mathcal{C}$) mediante algún criterio que permita comparar los distintos números graduales que originan el mismo intervalo $[\mathcal{A}, \mathcal{B}]$. Para definir tal criterio podemos usar de nuevo el parámetro "preferencia por el riesgo o seguridad" en el siguiente sentido: sean ψ y ϕ dos números graduales distintos entre sí pero tales que $\mathcal{A}(\psi) = \mathcal{A}(\phi)$ y $\mathcal{B}(\psi) = \mathcal{B}(\phi)$. Al ser $\psi \neq \phi$, tenemos que el conjunto $D := \{\alpha \in [0,1] : \psi(\alpha) \neq \phi(\alpha)\}$ no es vacío. Por tanto, cualquier valor de $\alpha \in D$ permite distinguir ψ de ϕ; distinción que puede usarse para comparar ambos números graduales mediante el orden total establecido para los intervalos $\psi(\alpha)$ y $\phi(\alpha)$. Ahora bien, entre todos estos posibles valores de α, un decisor "cauto" considerará el mayor, atendiendo así a los intervalos de menor amplitud. Por el contrario, un decisor "arriesgado" elegirá el menor valor posible de α, correspondiente a los intervalos de mayor amplitud (que son los que ofrecen mayor "riesgo").

Si intentamos usar este criterio para todo el conjunto $\mathcal{G}_Z(\mathbf{R})$, nos encontramos el siguiente problema: D no tiene por qué ser cerrado, de manera que los valores $\max D$ y $\min D$ pueden no existir. Puede que nisiquiera existan otros valores a tomar como "sustitutos razonables" de $\max D$ y $\min D$. Eso es lo que ocurre en el siguiente ejemplo: consideremos los números graduales $\psi = \tau_{(0,1,2)}$ y ϕ determinado por las funciones siguientes (donde aparece el símbolo $\mathbf{N}$ para representar al conjunto de números enteros positivos):

$$a(\alpha) := \begin{cases} 0, & \alpha = 0 \\ \frac{1}{2n+1}, & \frac{1}{2n+2} < \alpha \leq \frac{1}{2n}, n \in \mathbf{N} \\ 1 - \frac{1}{2n+1}, & 1 - \frac{1}{2n} < \alpha \leq 1 - \frac{1}{2n+2}, n \in \mathbf{N} \\ 1, & \alpha = 1 \end{cases}$$

$$b(\alpha) := 2 - a(\alpha), \quad \forall \alpha \in [0,1].$$

Para estos dos números graduales se cumple que $\mathcal{A}(\psi) = \mathcal{A}(\phi) = 1/2$ y $\mathcal{B}(\psi) = \mathcal{B}(\phi) = 3/2$. Además ambos coinciden sobre el conjunto $J = \{0,1\} \cup \{\frac{1}{2n+1} : n \in \mathbf{N}\} \cup \{1 - \frac{1}{2n+1} : n \in \mathbf{N}\}$, pero son distintos sobre el complementario $D = [0,1] \backslash J$; conjunto éste que no posee mínimo ni máximo (ni valores que los puedan sustituir razonablemente como los adecuados para discriminar ambos números graduales).

Es evidente que en las aplicaciones prácticas no aparecerá este "contraejemplo teórico" ni otros similares. En efecto, en las aplicaciones aparecerán siempre números graduales que pueden determinarse por un número finito de parámetros (como los *números graduales triangulares* dados por 3 parámetros, los *números L-R-graduales* dados por 4 parámetros, o los *números n-graduales* determinados por $2n$ parámetros). En tales casos sí podremos encontrar el mínimo y el máximo del conjunto D, u otros valores apropiados para discriminar (y, en consecuencia, ordenar) dos números graduales distintos cuyos valores medios originan el mismo intervalo. En definitiva, para que resulte válido el método antes propuesto, nos restringimos al subconjunto de $\mathcal{G}_Z(\mathbf{R})$ formado por aquellos números graduales que pueden determinarse por un número finito de parámetros. O bien, para especificar con más detalle las hipótesis que necesitamos, se trata de considerar aquellos números graduales para los cuales, siempre que tengamos $\psi \neq \phi$ tales que $\mathcal{A}(\psi) = \mathcal{A}(\phi)$ y $\mathcal{B}(\psi) = \mathcal{B}(\phi)$, puede determinarse el nivel $\alpha_x \in [0,1]$ definido como el mínimo, para $x = r$, y como el máximo, para $x = s$, de entre todos los niveles $\alpha \in [0,1]$ tales que $\psi(\alpha) \neq \phi(\alpha)$.

Para los tipos concretos de números graduales antes definidos basta efectuar los cálculos de los distintos valores que intervienen en la definición de α_x para ver que se pueden tomar los siguientes:

PROPOSICIÓN 4.2 *1. Para comparar dos números graduales triangulares cualesquiera, podemos tomar:* $\alpha_r = 0$, $\alpha_s = 1$.

2. *Para comparar los números 3-graduales* $\xi_1 = \xi_{(A_1, A_1', A_1'', B_1'', B_1', B_1)}$ *y* $\xi_2 = \xi_{(A_2, A_2', A_2'', B_2'', B_2', B_2)}$, *podemos tomar:*

$$\alpha_r = \begin{cases} \frac{1}{3}, & si\ [A_1, B_1] \neq [A_2, B_2] \\ \frac{2}{3}, & si\ [A_1, B_1] = [A_2, B_2] \end{cases}$$

$$\alpha_s = \begin{cases} 1, & si\ [A_1'', B_1''] \neq [A_2'', B_2''] \\ \frac{2}{3}, & si\ [A_1'', B_1''] = [A_2'', B_2''] \end{cases}$$

Concretando, el criterio antes propuesto origina los siguientes órdenes totales:

DEFINICIÓN 4.3 *Para cada $\lambda \in [0,1]$ y cada $x \in \{r,s\}$ (y supuesto que manejamos números graduales que permiten obtener el correspondiente valor α_x), se define el orden total $\leq_{\lambda x}$ como sigue:*

- *Si los intervalos $[\mathcal{A}(\psi), \mathcal{B}(\psi)]$ y $[\mathcal{A}(\phi), \mathcal{B}(\phi)]$ son diferentes, entonces se usan para ordenar los números graduales ψ y ϕ de acuerdo al orden total $\leq_\lambda$ si $\lambda \in \{0,1\}$ ó al orden total $\leq_{\lambda x}$ si $\lambda \in (0,1)$. (Ver las Definiciones 3.1 y 3.2).*

- *Si $[\mathcal{A}(\psi), \mathcal{B}(\psi)] = [\mathcal{A}(\phi), \mathcal{B}(\phi)]$, entonces usamos los intervalos $\psi(\alpha_x)$ y $\phi(\alpha_x)$ para ordenar ψ y ϕ (mediante los mismos órdenes totales considerados en el caso anterior).*

Para finalizar veamos la compatibilidad del orden $\leq_{\lambda x}$ (que es obvio que extiende al orden usual de $\mathbb{R}$) con las operaciones producto por reales (1) y suma (2) definidas para números graduales:

PROPOSICIÓN 4.4 *Para cualquier valor de los parámetros $\lambda \in [0,1]$ y $x \in \{r,s\}$ y cualesquiera números graduales ψ, ϕ, ξ (en los que esté definido el orden $\leq_{\lambda x}$), se verifica:*

1. $\forall P \in (0, +\infty), \quad [\psi \leq_{\lambda x} \phi \Leftrightarrow P\psi \leq_{\lambda x} P\phi]$.
2. $\psi \leq_{\lambda x} \phi \Rightarrow \psi + \xi \leq_{\lambda x} \phi + \xi$.

DEMOSTRACIÓN: Se obtiene sin más que considerar los siguientes hechos: en primer lugar, la linealidad de la integración, que da lugar a las igualdades $\mathcal{A}(P\psi) = P\mathcal{A}(\psi)$ y $\mathcal{A}(\psi + \phi) = \mathcal{A}(\psi) + \mathcal{A}(\phi)$, así como las igualdades análogas correspondientes a los valores $\mathcal{B}$ y $\mathcal{M}_\lambda$. En segundo lugar, que el producto por cualquier número real positivo P mantiene el orden entre los extremos y las amplitudes de los intervalos compactos.

NOTAS: Observemos que, como el producto por un número real negativo P invierte el orden entre los extremos de cualesquiera dos intervalos compactos, pero mantiene el orden entre sus amplitudes, entonces resulta la inversión del orden entre dos números graduales ψ y ϕ, al ser multiplicados por el número negativo P en todos los casos excepto cuando tales números deben compararse atendiendo a la amplitud de los intervalos $[\mathcal{A}(\psi), \mathcal{B}(\psi)]$ y $[\mathcal{A}(\phi), \mathcal{B}(\phi)]$ o la amplitud de los intervalos $\psi(\alpha_x)$ y $\phi(\alpha_x)$.

Asimismo, notemos la no compatibilidad del orden $\leq_{\lambda x}$ con las operaciones mínimo (3) y máximo (4). Por ejemplo, $\tau_{(0,6,6)} \leq_{\lambda x} \tau_{(5,5,11)}$, $\forall \lambda \in [0,1]$, $\forall x \in \{r,s\}$, mientras que ni $\min\{\tau_{(0,6,6)}, \tau_{(5,5,11)}\}$ ni $\max\{\tau_{(0,6,6)}, \tau_{(5,5,11)}\}$ son números triangulares, no coincidiendo por tanto con $\tau_{(0,6,6)}$ ni con $\tau_{(5,5,11)}$.

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] J.F. Baldwin and N.C.F. Guild, Comparison of fuzzy sets on the same decision space. *Fuzzy Sets and Systems* **2** (1979), 213-233.

[4] R.E. Bellman and L.A. Zadeh, Decision Making in a Fuzzy Environment. *Man. Sci.* **17** (1970), 141-164.

[5] G. Bortolan and R. Degani, A review of some methods for ranking fuzzy subsets. *Fuzzy Sets and Systems* **15** (1985), 1-19.

[6] B. Bouchon-Meunier, O. Kosheleva, V. Kreinovich and H.T. Nguyen, Fuzzy numbers are the only fuzzy sets that keep invertible operations invertible. *Fuzzy Sets and Systems* **91** (1997), 155-163.

[7] L.M. de Campos and A. González, A subjective approach for ranking fuzzy numbers. *Fuzzy Sets and Systems* **29** (1989), 145-153.

[8] M. Delgado, J.L. Verdegay and M.A. Vila, A procedure for ranking fuzzy numbers using fuzzy relations. *Fuzzy Sets and Systems* **26** (1988), 49-62.

[9] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications* (1980). New York: Academic Press.

[10] D. Dubois and H. Prade, Ranking fuzzy numbers in the setting of possibility theory. *Inform. Sci.* **30** (1983), 183-224.

[11] A. González, *Métodos subjetivos para la comparación de números difusos.* Tesis Doctoral (Universidad de Granada, 1987).

[12] A. González, A study of the ranking function approach through mean values. *Fuzzy Sets and Systems* **35** (1990), 29-41.

[13] J.A. Herencia, Conjuntos, puntos y números graduales. *Actas del VI Congreso Español Sobre Tecnologías y Lógica Fuzzy* (Asturias, 1996), 137-142.

[14] J.A. Herencia, Graded numbers and graded convergence of fuzzy numbers. *Fuzzy Sets and Systems* **88** (1997), 135-264.

[15] J.A. Herencia, M.T. Lamata and J.L. Verdegay, The graded numbers as a tool in decision problems. *Proceedings of the I.P.M.U.'96* (Granada, 1996), 371-376.

[16] C.V. Negoita and D. Ralescu, *Applications of Fuzzy Sets to Systems Analysis* (1975). Basel: Birkhäuser Verlag.

[17] R.R. Yager, A procedure for ordering fuzzy subsets of the unit interval. *Inform. Sci.* **24** (1981), 143-161.

Un Algoritmo Para Reconstruir Matrices Reciprocas

LAMATA M.T.
Depto.Ciencias de la Computación e Inteligencia Artificial.
E.T.S. Ingeniería Informática.
Universidad de Granada. 18071.
Granada. España.
E-mail: Mlamata@goliat.ugr.es.

PELÁEZ J.I.
Depto. Lenguajes y Ciencias de la Computación.
Complejo Tecnológico. E.T.S.I. Informática.
Campus de Teatinos.
Universidad de Málaga.29071.
Málaga. España.
E-mail: jignacio@lcc.uma.es.

Resumen

Este trabajo presenta un estudio de los principales factores que influyen en la inconsistencia de las matrices de preferencia reciproca del método AHP (Analytic Hierarchy Process). Definimos una nueva relación de preferencia multiplicativa para evitar la inconsistencia debida a dichos factores, y presentamos un algoritmo para reconstruir matrices de preferencia reciproca inconsistentes.

Palabras Clave: Múltiple Criterio Decisión; Analytic Hierarchy Process (AHP).

1 INTRODUCCIÓN

Un problema de decisión multicriterio trata de seleccionar la mejor alternativa de entre un conjunto de alternativas A_1, A_2,..., A_n, las cuales son caracterizadas en términos de sus criterios C_1,C_2, ..., C_m . Una de las formas posibles de caracterización es por comparación entre pares, los cuales pueden representarse mediante matrices $n \times n$, cuyo elemento general a_{ij} indica la preferencia de la alternativa A_i sobre la alternativa A_j.

AHP (Analytic Hierarchy Process) [6], es un método de decisión multicriterio. Este utiliza una relación de preferencia multiplicativa, p.e. El elemento a_{ij} de la matriz de comparación expresa la preferencia de la alternativa A_i sobre la alternativa A_j. Si instanciamos el valor $a_{ij} = 3$, la alternativa A_i es tres veces mas preferida que la alternativa A_j. Como nuestra matriz de preferencias $n \times n$ es reciproca y positiva, entonces: $a_{ij} > 0$, $a_{ji} = 1/a_{ij}$ para todo i,j = 1,..,n, por lo que $a_{ii} = 1$.

Dado que las matrices de preferencias son positivas, es bien conocido que el teorema de Perron-Frobenius garantiza que cada matriz de preferencias tiene un valor propio (real) dominante ($\lambda_{max} > 0$), monótono creciente con respecto a los valores de la matriz de preferencias, siendo $\lambda_{max} \geq n$ para las matrices de preferencia reciproca de AHP. Ademas el vector propio (v) asociado al λ_{max} en el método AHP, establece las prioridades de las alternativas.

Una de las características de AHP es la posibilidad de medir la consistencia de los juicios emitidos por el decisor. Esto se realiza mediante el λ_{max} , de manera que si los juicios son perfectos $\lambda_{max} = n$ p.e. Si $a_{ij} = 2$ (la alternativa A_i es dos veces mas preferida que la alternativa A_j), $a_{jk} = 3$ (la alternativa A_j es tres veces mas preferida que la alternativa A_k), y $a_{ik}=6$, (la alternativa A_i es seis veces mas preferida que la alternativa A_k), entonces el valor de λ_{max} es igual a 3.

Saaty [6] define un indice de consistencia (IC) para matrices reciprocas en función del λ_{max}:

$$IC = \frac{\lambda_{max} - n}{n-1}$$

y un indice aleatorio (IA). El IA es calculado como sigue: Para cada orden de matriz construimos un ejemplo de tamaño 500; rellenamos sus entradas con valores aleatorios para diferentes escalas: 1-5, 1-7, 1-9; calculamos los IC para cada uno de ellos; y finalmente calculamos la media de los IC para cada escala. Entonces IA es calculado como la desviación tipica de los IC de cada escala con respecto de su media. Comprobandose que dicha desviación es inferior al 10% para todas las escalas. En la Tabla 1, mostramos los valores de IA para diferentes ordenes de matrices.

Tabla 1. Valores del Indice Aleatorio

Orden	1-2	3	4	5	6
I.A	0.00	0.58	0.90	1.12	1.24

Con IC e IA, Saaty define el ratio de consistencia (RC) como:

$$RC = \frac{IC}{IA}$$

de manera que si RC ≤ 0.10, la matriz de preferencias es considerada consistente y por lo tanto aceptada, siendo rechazada en caso contrario.

Otra de las características de AHP es la utilización de una escala para que el decisor exprese sus preferencias, siendo la más utilizada la escala 1-9. En la tabla 2, mostramos la escala 1-9 utilizada en AHP.

Tabla 2. Escala de Juicios 1-9. AHP

Intensidad	Definición	Explicación
1	Igual importancia	Las dos actividades contribuyen igual al objetivo.
3	Débilmente mas preferido	Expresa un juicio a un poco a favor de una actividad.
5	Mas importante	Expresa un juicio fuerte a favor de una actividad
7	demostrado que es mas importante	Una actividad es preferida fuertemente a la otra.
9	Absolutamente mas preferida	Afirma que una actividad es preferida a la otra.
2,4,6,8	Valores intermedios	Valores de compromiso.

La intención de este trabajo es la de estudiar los principales factores que influyen en la inconsistencia de las matrices de preferencia reciproca. Para ello el trabajo esta estructurado como sigue: en la sección 2 se presentan los principales factores que influyen en la inconsistencia; en la sección 3 proponemos un algoritmo de recuperación de matrices de preferencias inconsistentes, una relajación de la consistencia para matrices de preferencia multiplicativa para AHP y finalmente un ejemplo del algoritmo.

2 FACTORES QUE INFLUYEN EN LA INCONSISTENCIA

Cuando el decisor expresa sus preferencias, este puede estar cometiendo errores debido a que sus expectativas son erróneas o irracionales. Pero este error, muchas otras veces puede ser debido a la escala que el decisor utiliza para expresar sus preferencias. p.e. Sean tres alternativas A_i, A_j, A_k, las cuales son comparadas por el decisor utilizando la escala 1-9. (Tabla 2), siendo las preferencias del decisor las siguientes: $a_{ij} = 5$ (la alternativa A_i es cinco veces mas preferida que la alternativa A_j), $a_{jk} = 7$ (la alternativa A_j es siete veces mas preferida que la alternativa A_k), y $a_{ik} = 9$, (la alternativa A_i es nueve veces mas preferida que la alternativa A_k). El valor correcto de a_{ik} seria 35, pero el decisor ha expresado $a_{ik} = 9$, por lo que la preferencia es inconsistente. Esto podría ser considerado un error del decisor al expresar sus preferencias, pero también podría ser un error atribuible a la escala de juicios utilizada. El decisor expresa una preferencia errónea (a_{ik}=9), pero dicha preferencia es la mayor que le permite la escala 1-9. La utilización por parte del decisor del mayor valor de preferencia, hace suponer que su intención es la de expresar el valor correcto ($a_{ik} = 35$), pero no puede realizarlo al no estar permitido en la escala. utilizada. Por lo que el error en este caso podría ser atribuido a la escala y no al decisor.

Otro factor que afecta a la inconsistencia en las matrices de preferencias reciprocas es RC. AHP define el RC para aceptar/rechazar las matrices reciprocas de preferencias. Este valor esta definido en función del λ_{max}, y establece que si RC ≤ 0.1, la matriz es aceptada, siendo rechazada en caso contrario.

Lamata-Peláez [4] definen una medida de inconsistencia alternativa al IC de Saaty. Esta medida es denominada indice de inconsistencia y es denotado por IC*. A continuación mostramos la deficición de IC*.

Definición 1. Definimos el indice de inconsistencia de una matriz reciproca de preferencias $M_{3\times3}$, y lo denotamos por IC*(M), como el determinante de M.

$$IC^*(M_{3\times3}) = \det(M_{3\times3})$$

Definición 2. Sea $M_{n\times n}$ una matriz reciproca de preferencias, y sea Γ el conjunto de matrices 3×3 distintas que hay en M/ $card(\Gamma) = C_n^3$.

$$\Gamma = \{\Gamma_i \text{ ;conjunto de matrices } 3\times3 \neq \text{de M / } card(\Gamma) = C_n^3\}$$

Definimos el coeficiente de inconsistencia para una

matriz $M_{nxn,}$ y lo denotamos por IC*, como la media de las inconsistencias de Γ.

$$IC^* = \frac{1}{NT(M)} \sum_{i=1}^{NT(M)} IC^*(\Gamma_i)$$

En la tabla 3, mostramos la relación entre IC, λ_{max}, e IC*, y los valores alternativos para RC ≤ 0.1.

Tabla 3. Relación entre IC*, λ_{max} e IC, para el calculo de la inconsistencia en matrices de importancia reciproca. Escala 1-9.

Orden \ valores	IC *	λ_{maximo}	IC
3×3	1,127	3,116	0,058
4×4	1,243	4,270	0,090
5×5	1,317	5,448	0,112
6×6	1,364	6,655	0,131
7×7	1,422	7,846	0,141
8×8	1,426	9,015	0,145
9×9	1,442	10,192	0,149

En la tabla 4. mostramos los valores de preferencia mínimos (matrices 3×3) para RC≤0.10. Podemos ver que cuando el producto de $a_{ij} \times a_{jk} \geq 25$, cualquier valor a_{ik} expresado por el decisor con la escala 1-9 seria inconsistente, ya que el valor mínimo permitido de a_{ik} para RC≤0.10, seria mayor que 9.

Tabla 4. Valores de preferencia mínimos de a_{ik} para que RC ≤ 0.10.Valores obtenidos mediante IC*.

$a_{ij} \backslash a_{jk}$	1	2	3	4	5
1	0.36	*			
2	0.72	1.44	*		
3	1.08	2.17	3.26	*	
4	1.44	2.89	4.34	5.79	*
5	1.81	3.62	5.43	7.24	9.05*
6	2.17	4.34	6.52	8.69	10.86 *
7	2.53	5.07	7.60	10.14 *	
8	2.89	5.79	8.69	*	
9	3.26	6.52	9.78*	*	

(Continuación tabla 4)

RC es un factor que produce un error de intuición en los juicios emitidos por el decisor. p.e. Supongamos que utilizamos la escala 1-9. (Tabla 2). Si $a_{ij} = 1$ (la alternativa A_i es indiferente frente a la alternativa A_j), $a_{jk} = 9$ (la alternativa A_j es nueve veces mas preferida que la alternativa A_k), y $a_{ik} = 9$, (la alternativa A_i es nueve veces mas preferida que la alternativa A_k). El valor correcto de a_{ik} seria 9. Utilizando los valores de la tabla 4. serian aceptados valores para a_{ik} que oscilarían desde 4 (⌈3.26⌉) hasta 9, que seria el limite de la escala 1-9. Si calculamos el valor de prioridad de las alternativas A_i y A_j para los valores a_{ik} = 4, 5, 6, 7, 8, y 9, la alternativa A_i es siempre igual o más preferida que la alternativa A_j. Este es un resultado incorrecto ya que el decisor expresó que las alternativas A_i y A_j tenían igual importancia.

Como hemos mostrado tanto la escala como RC son factores de inconsistencia en las matrices de preferencias reciprocas.

3 ALGORITMO PARA RECONSTRUIR MATRICES

Como hemos mostrado en la sección anterior tanto la escala de juicios como el RC son factores que influyen en la inconsistencia de las matrices. Para intentar eliminar dichos errores proponemos una relajación de la consistencia para AHP, junto con un algoritmo basado [3, 4] para reconstruir matrices de preferencias reciprocas.

Definición 3. Definimos el limite inferior para a_{ik} como:

$$\text{máx. } \{ \text{máx.}(a_{ij}, a_{jk}), (a_{ij} \times a_{jk} \times 0.36)\}$$

Algoritmo Recuperación

Paso 0. Obtener del decisor el orden de las alternativas. Junto con *n-1* preferencias.

Paso 1. Obtener del decisor la preferencia de a_{ik} cumpliendo la definición 3.

Paso 2. Calcular el vector propio asociado a la matriz 3×3 del paso 1.

Paso 3. Calcular el valor de las alternativas en función de la alternativa más importante de la matriz 3×3 del paso 1. Aplicando la definición 3 en el calculo de cada preferencia.

Paso 4. Fin

A continuación mostramos un ejemplo donde aplicamos el algoritmo de reconstrucción de matrices.

Ejemplo 1 Sean las alternativa A_1, A_2, A_3 y A_4, y sea la matriz de la tabla 5, la cual refleja el orden de las alternativas para el decisor, y las preferencias de dicho orden.

Paso 0.

Tabla 5. Matriz de preferencias que muestra el orden de las alternativas.

	A_1	A_2	A_3	A_4
A_1	1	5		
A_2	1/5	1	4	
A_3		1/4	1	4
A_4		1/6	1/4	1

Paso 1.

$a_{24} = \max \{\max (a_{23}, a_{34}), \max (a_{23} \cdot a_{34} \cdot 0.36)\}$

$$a_{24} = 6.$$

Tabla 6. Matriz de preferencias que muestra el orden de las alternativas. Contiene la preferencia a_{24} con la que se reconstruira el resto de la matriz.

	A_1	A_2	A_3	A_4
A_1	1	5		
A_2	1/5	1	4	6
A_3		1/4	1	4
A_4		1/6	1/4	1

Paso 2.

Calculamos el autovector asociado a la matriz 3×3 (A_2, A_3, A_4). V = [W_2= 0 .682, W_3=0.236, W_4= 0.0811].

Paso 3.

Calculamos el valor de la primera alternativa en función de la alternativa más importante A_2= 0,682. El peso de la alternativa A_1 lo calculamos basandonos en el valor de preferencia de la alternativa A_1 respecto de A_2 y la prioridad de la alternativa A_2, la cual dice que A_1 es cinco veces mas preferida que A_2. Esto hace que W_1= 5 · 0.682= 0.773 (normalizada con el resto de prioridades). El valor de A_1 con respecto a las alternativas A_3 y A_4 se calcula mediante este valor W_1.

La matriz reconstruida se muestra en la tabla 7.

Tabla 7. Matriz de preferencia reconstruida. IC*= 0.341

	A_1	A_2	A_3	A_4
A_1	1	5	14.5	42.9
A_2	1/5	1	4	6
A_3	1/14.5	1/4	1	4
A_4	1/42.9	1/6	1/4	1

4 CONCLUSIONES

Hemos mostrado como la escala utilizada en AHP como el RC son factores que influyen en la inconsistencia de las matrices. La construcción de matrices donde el decisor expresa todas sus preferencias , muestra importantes deficiencias debido principalmente al problema de escala. Para solucionar este problema hemos propuesto un algoritmo para reconstruir matrices de preferencias. Con este método, el decisor solamente tiene que expresar *n* preferencias para poder reconstruir la matriz, en lugar de las 2n-1. Ademas hemos propuesto una relajación de la relación de preferencia multiplicativa dada en AHP para evitar la inconsistencia de las matrices.

REFERENCIAS

[1] Bryson, Noel. Mobolurin, ayodele. " An approach to using the analytic hierarchy process for solving multiple criteria decision making " European Journal of Operational Research. 76. 1994

[2] Harker,P.T. and Vargas L.C. "The theory of ratio scale estimation: Saaty's analytic hierarchy process". Management Science 33. 1987.

[3] Herman Monsuur. "An intrinsic consistency threshold for reciprocal matrices". European Journal of Operational Research 96. 1996.

[4] Lamata M.T. Peláez J.I. "A new consistency measure for reciprocal matrices", IPMU'98.

[5] Murphy, C.K. "Limits on the analytic hierarchy process from its consistency index". European Journal of Operational Research. 65. 1993.

[6] Saaty, Thomas L. 1980. *The Analytic Hierarchy Process.* Macgraw Hill.

[7] Saaty, Thomas L. 1997. *"Toma de decisiones para lideres".* RWS.

[8] Saaty Thomas L. 1986. *"Axiomatic Foundation of the Analytic Hierarchy Process." Management Science.* 32:841-55.

[9] Saaty, Thomas L. 1990. "An Exposition of the AHP in Reply to the Paper "Remarks on the Analytic Hierarchy Process"." *Management Science.* 36.:259-68.

[10] Salo. A.A. "Inconsistency analysis by approximately specified priorities". Mathematical and Computer Modelling. 17. 1993.

Semánticas de Lenguajes con Datos Borrosos *

Daniel Sánchez Alvarez y Antonio F. Gómez Skarmeta
Departamento de Informática, Inteligencia Artificial y Electrónica
Universidad de Murcia
E-mail: daniel@dif.um.es

Resumen

Al diseñar, con la ayuda de la semántica denotacional, un lenguaje de programación que maneje datos borrosos, encontramos que tanto la semántica denotacional como el paradigma borroso manejan el concepto de información parcial y aproximación de la información. Pero desde dos puntos de vista bien diferentes. Necesitamos integrar ambos puntos de vista y para ello hemos de revisar tanto las limitaciones de la teoría de dominios como las exigencias del cálculo con datos borrosos.

1 Introducción

Al diseñar, con la ayuda de la semántica denotacional, un lenguaje de programación que maneje datos borrosos, encontramos que tanto la semántica denotacional como el paradigma borroso manejan el concepto de información parcial y aproximación de la información. Pero desde dos puntos de vista bien diferentes.

En semántica denotacional: supongamos que tenemos el cpo de las funciones parciales $[\mathbb{N} \rightsquigarrow \mathbb{N}]$. En este cpo, la noción de aproximación viene dada por la inclusión entre los grafos de funciones. La función totalmente indefinida es el conjunto vacio, y una función totalmente definida tiene un grafo tal que, para cada $m \in \mathbb{N}$, existe algún $n \in \mathbb{N}$ tal que $(m,n) \in grafo(f)$ De forma que dadas dos dunciones f_1 y f_2 decimos que f_1 es una aproximación de f_2 si $grafo(f_1) \subseteq grafo(f_2)$.

En el cálculo borroso: Un conjunto borroso F queda definido a partir de un referencial Ω y una aplicación, escrita μ_F, de Ω en $[0,1]$. De forma que $\forall \omega \in \Omega$ $\mu_F(\omega)$ se interpreta como el grado de pertenencia de ω al conjunto borroso F. Esto permite modelizar categorias vagas del lenguaje natural definidas sobre un soporte objetivo con el que clasificarlo con la ayuda de dichas categorias. $\mu_F(\omega)$ expresará entonces hasta que punto el valor ω es compatible con el concepto F.[1]

Necesitamos integrar ambos puntos de vista y para ello hemos de revisar tanto las limitaciones de la teoría de dominios como las exigencias del cálculo con datos borrosos.

La teoría de dominios se estableció para tener espacios apropiados en los que definir funciones semánticas para el enfoque denotacional de la semántica de los lenguajes de programación. Con este fin se deben cubrir dos necesidades: en primer lugar deben existir espacios de varios tipos distintos que permitan las diversas clases de construcciones semánticas y en segundo lugar la teoría ha de dar cuenta de de las propiedades computacionales de las funciones. Por ello como se señala en [2] "... los dominios son generalmente *planos* (y numerables). Notar en partícular que las aproximaciones finitas numericas a los números reales hechas por el ordenador no son representadas por valores relacionados con el orden de aproximación computacional de los dominios...". Todo ello nos lleva, lo que para nosotros es especialmente importante, a que, como se muestra en [3], el intervalo $[0,1]$ no es un dominio. Proponemos sustituirlo por un conjunto numerable pero no *plano*.

Por otro lado en [4] podemos leer: " ... la sentencia condicional if P then S_1 else S_2 debe ejecutarse en paralelo. Esto significa que las sentencias S_1 y S_2 deben ejecutarse independientemente y a los resultados de S_1 denotados como $r(S_1)$ se les asigna el grado de pertenencia (de verdad) igual a $v(P)$...De forma similar, a los resultados de S_2 se les asigna el grado igual a $\neg v(P)$. En esta forma, el resultado obtenido por medio de la ejecución

* Trabajo parcialmente soportado por el proyecto CICYT-TIC97-1343-C002-02

de las sentencias dadas puede interpretarse como un subconjunto borroso del conjunto de todos los resultados posibles". Algo similar podria decirse de la sentencia iterativa. Esta caracterítica de la programación borrosa es la que vamos a estudiar en este trabajo.

2 El lenguaje

El lenguaje que vamos a presentar es una variante de los "comandos guardados" de Dijkstra [5]. El resultado final, debe llenar algunos de los requisitos exijidos en [6] y [7]. Para ello proponemos las siguientes modificaciones:

Introducir un nuevo elemento al que llamaremos índice perteneciente a $\mathbb{D}$. Este será un conjunto totalmente ordenado a partir del cual construiremos los dominios no planos, $\mathbb{ND}$ y $\mathbb{BD}$, los producto cartesianos de los naturales $\mathbb{N}$ y los booleanos $\mathbb{B}$ por $\mathbb{D}$.

Donde Dijkstra habla de "predicados" y "condiciones", nosotros hemos de hablar en términos de conjuntos, de tal forma que hemos de sustituir nociones proposicionales (**and**, **or**, $\Rightarrow$, etc) por sus correspondientes nociones conjuntistas ($\cap, \cup, \subseteq$, etc).

Postulamos un conjunto numerable, Σ, de almacenes dado por el espacio de funciones $\Sigma = Var \to \mathbb{ND}$. Es decir un almacén, σ, es una función que para variable en Var produce un elemento de $\mathbb{ND}$.

Para dar significado semántico a las construcciones sintácticas utilizaremos los dominios potencia y como las especificaciones denotacionales son muy parecidas a los programas escritos en en lenguaje de programación funcional [8],[9], utilizaremos "el λ-cálculo_B", ambos los presentamos en los apendices A y B.

2.1 Sintaxis abstracta

Dado el lenguaje por medio de la siguiente la sintaxis abstracta:

P ::= S

S ::= v := E | skip | abort | (S;S) | case G esac | do G od

G ::= B → S | (G □ G)

E ::= (n,d) | x | E op E

B ::= (true,d) | (false,d) | B Bop B | E rel E

con las siguientes categorias sintácticas

P ∈ Programas

S ∈ Sentencias

G ∈ Comandos guardados

E ∈ Expresiones

B ∈ Expresiones booleanas

v ∈ Variables

2.2 Semántica Operacional

La semántica operacional del lenguaje viene dada en términos de cuatro relaciones: $\Rightarrow_A, \Rightarrow_B, \Rightarrow_S$ y $\Rightarrow_{CG}$, para las expresiones, las expresiones booleanas, las sentencias y los comandos guardados:
La funcionalidad de $\Rightarrow_A$ es

$$\Rightarrow_A : (\text{Expr} \times \Sigma) \to \mathbb{ND}$$

y su definición es la siguiente:

1. $\dfrac{}{\langle (n,d), \sigma \rangle \Rightarrow_A \langle (n,d) \rangle}$
2. $\dfrac{}{\langle x, \sigma \rangle \Rightarrow_A \sigma(x)}$
3. $\dfrac{\langle e_1, \sigma \rangle \Rightarrow_A \langle (n_1, d_1) \rangle, \langle e_2, \sigma \rangle \Rightarrow_A \langle (n_2, d_2) \rangle}{\langle e_1 \textbf{ op } e_2, \sigma \rangle \Rightarrow_A \langle (n_1 \textbf{ op } n_2, s(d_1, d_2)) \rangle}$
   donde s es la función definida en [11]

Similarmente, el tipo para $\Rightarrow_B$ viene dado por:

$$\Rightarrow_B : (\text{BExpr} \times \Sigma) \to \mathbb{BD}$$

y su definición es la siguiente:

1. a) $\dfrac{}{\langle (true, d), \sigma \rangle \Rightarrow_B \langle (true, d) \rangle}$
   b) $\dfrac{}{\langle (false, d), \sigma \rangle \Rightarrow_B \langle (false, d) \rangle}$
2. $\dfrac{\langle be_1, \sigma \rangle \Rightarrow_B \langle (vb_1, d_1) \rangle, \langle be_2, \sigma \rangle \Rightarrow_B \langle (vb_2, d_2) \rangle}{\langle be_1 \textbf{ Bop } be_2, \sigma \rangle \Rightarrow_B \langle (vb_1 \textbf{ Bop } vb_2, s(d_1, d_2)) \rangle}$
3. $\dfrac{\langle e_1, \sigma \rangle \Rightarrow_A \langle (n_1, d_1) \rangle, \langle e_2, \sigma \rangle \Rightarrow_A \langle (n_2, d_2) \rangle}{\langle e_1 \textbf{ rel } e_2, \sigma \rangle \Rightarrow_B \langle (n_1 \textbf{ rel } n_2, s(d_1, d_2)) \rangle}$

La relación para la evaluación de los sentencias $\Rightarrow_S$, toma una tripla formada por una sentencia, un índice y un almacén y devuelve un índice y un almacén; intuitivamente un comando solo modifica el almacén, el índice se utilizará dentro del comando. Su tipo es

$$\Rightarrow_S : (S \times I \times \Sigma) \to \Sigma$$

1. $a)$ $\dfrac{\langle e, \sigma \rangle \Rightarrow_A \langle (n,d) \rangle}{\langle v := e, i, \sigma \rangle \Rightarrow_S \langle [v \mapsto (n, s(i,d))] \sigma \rangle}$
2. $a)$ $\dfrac{}{\langle \underline{skip}, i, \sigma \rangle \Rightarrow_S \langle \sigma \rangle}$

3. *a*) $\dfrac{\langle S_1, i, \sigma\rangle \Rightarrow_S \langle \sigma'\rangle, \langle S_2, i, \sigma'\rangle \Rightarrow_C \langle \sigma''\rangle}{\langle S_1; S_2, i, \sigma\rangle \Rightarrow_S \langle \sigma''\rangle}$

4. *a*) $\dfrac{\langle G, i, \sigma\rangle \Rightarrow_{CG} \langle \sigma'\rangle}{\langle \underline{\text{case}}\ G\ \underline{\text{esac}}, i, \sigma\rangle \Rightarrow_S \langle \sigma'\rangle}$

5. *a*) $\dfrac{\langle G, i, \sigma\rangle \Rightarrow_{CG} \langle \sigma''\rangle, \langle \underline{\text{do}}\ G\ \underline{\text{od}}, i, \sigma''\rangle \Rightarrow_{CG} \langle \sigma'\rangle}{\langle \underline{\text{do}}\ G\ \underline{\text{od}}, i, \sigma\rangle \Rightarrow_S \langle \sigma'\rangle}$

b) $\dfrac{\langle G, i, \sigma\rangle \rightarrow \langle \delta\rangle}{\langle \underline{\text{do}}\ G\ \underline{\text{od}}, \sigma\rangle \Rightarrow_S \langle \delta\rangle}$

donde δ se utliza para señalar la no terminación.

La relación para la evaluación de los comandos guardados $\Rightarrow_{CG}$, toma una tripla formada por un comando guardado, un índice y un almacén y devuelve un índice y un almacén; el índice junto con grado de la guarda produciran un nuevo índice en cuyo seno se evaluará el comando guardado. Su tipo es

$$\Rightarrow_{CG}: (CG \times I \times \Sigma) \rightarrow \mathcal{P}(\Sigma)$$

1. *a*) $\dfrac{\langle be, \sigma\rangle \Rightarrow_B \langle true, d\rangle, \langle S, s(i,d), \sigma\rangle \Rightarrow_S \langle \sigma'\rangle}{\langle be \rightarrow S, i, \sigma\rangle \Rightarrow_{CG} \langle \sigma'\rangle}$

2. *a*) $\dfrac{\langle be, \sigma\rangle \Rightarrow_B \langle (false, d)\rangle}{\langle be \rightarrow S, i, \sigma\rangle \Rightarrow_{CG} \langle \sigma\rangle}$

3. *a*) $\dfrac{\langle G_1, i, \sigma\rangle \Rightarrow_{CG} \langle \sigma_1\rangle, \langle G_2, i, \sigma\rangle \Rightarrow_{CG} \langle \sigma_2\rangle}{\langle (G_1 \square G_2), i, \sigma\rangle \Rightarrow_{CG} \langle \{\sigma_1, \sigma_2\}\rangle}$

b) $\dfrac{\langle G_1, i, \sigma\rangle \Rightarrow_{CG} \langle \delta\rangle, \langle G_2, i, \sigma\rangle \Rightarrow_{CG} \langle \delta\rangle}{\langle (G_1 \square G_2), i, \sigma\rangle \Rightarrow_{CG} \langle \delta\rangle}$

2.3 Semántica Axiomática

Siguiendo la linea de la citada obra de Dijkstra discutiremos los mecanismos S, y su semántica considerando lo que llamaremos la "precondición débil_b". Estas precondiciones, el adjetivo de débil se aclarará más adelante, son un subconjuntos de almacenes representados por $wp(S, i, R)$, lo son de una postcondición R con respecto a un mecanismo S y a un índice i, diciendo que "la condición que caracteriza al conjunto de todos los almacenes iniciales tales que la activación cuanto termina adecuadamente tiene como resultado dejar al sistema en un almacén final que satisface a una postcondición dada es llamada "la precondición débil_b correpondiente a tal postcondición".". Nuestra tarea ahora es especificar las condiciones y los mecanismos.

2.3.1 Las condiciones

Dado el espacio de funciones $\Sigma = Var \rightarrow \mathbb{ND}$ podemos definir las condiciones bien desde un punto de vista intensional o extensional. Siguiendo a Dijkstra, tomaremos el conjunto *Pred*, de predicados o condiciones como: $Pred = \mathcal{P}(\Sigma)$.
Por tanto *Pred* es un conjunto parcialmente ordenado con las siguientes propiedades:

1. $\bot_{Pred} = \emptyset$
2. Si c_i es una cadena entonces $\bigsqcup c_i$ existe y viene definido por $\bigcup c_i$

2.3.2 Los mecanismos

Partiendo de la anterior descripción de $wp(S, i, R)$ todo lo que necesitamos saber acerca de S es que para cualquier almacén inicial σ dado y cualquier i, si va a terminar adecuadamente o no, y en el caso afirmativo, si todos los posibles almacenes finales satisfacen a R. Tomando este punto de vista consideramos a S como algún tipo de función transformadora de almacenes. Afinaremos esta idea considerando a los mecanismos como realizadores de transiciones de un almacén a otro almacén. Para ello definimos el conjunto de *configuraciones* como $Conf = (S \times \mathbb{D} \times \Sigma) \cup \Sigma$. Seguidamente postulamos una *relación de transición*, $\rightarrow \subseteq Conf \times Conf$ y suponemos que no existen transiones válidas de la forma $\sigma \rightarrow C$ donde $C \in Conf$. La idea es que las configuraciones, C, representan estadios de un cálculo: las que son de la forma $\langle S, i, \sigma\rangle$ son posibles estadios iniciales o intermedios y las que son de la forma $\langle \sigma\rangle$ son estadios finales. Las transiciones $C \rightarrow C'$, representarán un paso del cálculo, las de la forma $\langle S, i, \sigma\rangle \rightarrow \langle S', i, \sigma'\rangle$ son pasos intermedios y los de la forma $\langle S, i, \sigma\rangle \rightarrow \langle \sigma'\rangle$ serán los pasos finales.

Definición.- Un sistema S, está *bloquedado* en un almacén σ con índice i, si y solo si no existe una configuración C tal que $\langle S, i, \sigma\rangle \rightarrow C$.

Un sistema S, puede *no terminar* desde un almacén σ con índice i bien porque exista S', i, σ' tal que $\langle S, i, \sigma\rangle \rightarrow^* \langle S', i, \sigma'\rangle$ y S' está bloqueado en el almacén σ' con índice i o bien porque exista una secuencia infinita de la forma $\langle S, i, \sigma\rangle = C_0 \rightarrow \ldots \rightarrow C_n \rightarrow \ldots$.

La precondición débil_b, $wp(S, i, R)$, de S, i y R puede definirse formalmente como:

$$wp(S, i, R) = \{\sigma \mid S \text{ con índice } i \text{ debe terminar desde } \sigma \text{ y si } \langle S, i, \sigma\rangle \rightarrow^* \langle \sigma'\rangle \text{ entonces } \sigma' \in R\}$$

La correspondiente noción de precondición se define para P, S, i y R por

$$P\ [S, i]\ R \text{ si y solo si } \forall \sigma \in P(S \text{ con índice } i \text{ debe terminar partiendo de } \sigma \text{ y si } \langle S, i, \sigma\rangle \rightarrow^* \langle \sigma'\rangle \text{ entonces } \sigma' \in R)$$

Evidentemente $P\ [S, i]\ R$ es válida si y solo si se verifica que $P \subseteq wp(S, i, R)$, lo cual justifica el nombre de "precondición débil"

En particular las propiedades impuestas en [5] de ser estricta. monótoma, multiplicativa y continua para cualquier S, P y Q se reescribirian como:

1. $wp(S,i,\emptyset) = \emptyset$ *(exclusión de milagros)*
2. Si $P \subseteq Q$ entonces $wp(S,i,P) \sqsubseteq wp(S,i,Q)$ *(monotonía)*
3. $(wp(S,i,P) \cap wp(S,i,Q)) = wp(S,i,(P \cap Q))$ *(multiplicatividad)*
4. Si $P_0 \subseteq P_1 \subseteq \ldots$ es una secuencia creciente entonces
   $wp(S,i,\bigsqcup P_n) = \bigsqcup wp(S,i,P_n)$ *(continuidad)*

   Esta propiedad es el correlato de la dada por Dijstra en el capítulo titulado "On Nondeteterminacy Being Bounded". Para establecer la propiedad en el presente contexto notar que:

   - Como $\bigsqcup P_n \supseteq P_n$, y teniendo en cuenta la monotonía se tiene que $wp(S,i,\bigsqcup P_n) \supseteq \bigsqcup wp(S,i,P_n)$.
   - Supongamos que $\sigma \in wp(S,i,\bigsqcup P_n)$ entonces S con el índice i debe terminar desde σ. Es decir que el conjunto finito $\langle S,i,\sigma\rangle$ estará contenido en $\bigsqcup P_n$, esto permite afirmar que existe algún q tal que $\langle S,i,\sigma\rangle \subseteq P_q$, mostrando que $\sigma \in wp(S,i,P_q)$. Por tanto $wp(S,i,\bigsqcup P_n) \subseteq \bigsqcup wp(S,i,P_n)$.

2.3.3 Transformadores de predicados

Si en la precondición débil $wp(S,i,R)$ fijamos S se obtiene una función $\pi^i_S : Pred \longrightarrow Pred$ de forma que

$$\pi^i_S(R) = wp(S,i,R)$$

Teniendo en cuenta las propiedades de las precondiciones débiles en lugar de considerar la colección de funciones continuas π^i_S de $Pred$ en $Pred$ consideramos solo aquellas que además de ser continuas sean estrictas y multiplicativas. Llamaremos a dicha colección PT. A continuación introducimos un orden parcial en PT de la siguiente forma:

$\pi^i_{S_1} \subseteq \pi^i_{S_2}$ si y solo si $\forall R \in Pred, \pi^i_{S_1}(R) \subseteq \pi^i_{S_2}(R)$

Con lo cual PT es un conjunto parcialmente ordenado con las siguinetes propiedades:

2.3.4 Propiedades

1) Su menor elemento es $\bot_{Pred,Pred}$
2) Si $\{\pi^i_{S_0}, \pi^i_{S_1}, \ldots\}$ es una cadena entonces existe la menor de sus cotas superiores π^i_S y viene dada por
   $(\sqcup \pi^i_{S_n})(R) = \sqcup \pi^i_{S_n}(R)$
   Demostración.- Teniendo en cuenta la multiplicatividad para cualesquiera Q y R se tiene:

$$\begin{aligned}
\pi_{S,i}(Q \cap R) &= (\bigsqcup \pi^i_{S_n})(Q \cap R) \\
&= \bigsqcup (\pi^i_{S_n}(Q \cap R)) \\
&= \bigsqcup (\pi^i_{S_n}(Q) \cap \pi^i_{S_n}(R)) \\
&= \bigcup_n \pi^i_{S_n}(Q) \cap \bigcup_n \pi^i_{S_n}(R) \\
&= \pi^i_S(Q) \cap \pi^i_S(R)
\end{aligned}$$

3) Si $\pi^i_{S_1}, \pi^i_{S_2} \in PT$ entonces $\pi^i_{S_1} \circ \pi^i_{S_2} \in PT$ y lo representaremos por $\pi^i_{S_1;S_2}$
4) Si $\pi^i_{S_1}, \pi^i_{S_2} \in PT$ y P, Q dos predicados disjuntos entonces h definido por $h(R) = (P \cap \pi^i_{S_1}(R)) \cup (Q \cap \pi^i_{S_2}(R))$ es un transformador de predicados
   Demostración.- Es evidente que h es estricta y continua. Probemos que es multiplicativa

$$\begin{aligned}
h(R \cap S) &= (P \cap \pi^i_{S_1}(R \cap S) \cap \pi^i_{S_2}(R \cap S)) \cup \\
&\quad (Q \cap \pi^i_{S_2}(R \cap S) \cap \pi^i_{S_1}(R \cap S)) \\
&= (P \cap \pi^i_{S_1}(R) \cap P \cap \pi^i_{S_2}(S)) \cup \\
&\quad (P \cap \pi^i_{S_1}(R) \cap Q \cap \pi^i_{S_2}(S)) \cup \\
&\quad (Q \cap \pi^i_{S_2}(R) \cap P \cap \pi^i_{S_1}(S)) \cup \\
&\quad (Q \cap \pi^i_{S_2}(R) \cap Q \cap \pi^i_{S_2}(S)) \\
&= ((P \cap \pi^i_{S_1}(R)) \cup (Q \cap \pi^i_{S_2}(R))) \cap \\
&\quad ((P \cap \pi^i_{S_1}(S)) \cup (Q \cap \pi^i_{S_2}(S)) \\
&= h(R) \cap h(S)
\end{aligned}$$

2.3.5 Transformadores del lenguaje

Suponemos que tenemos definidos dos conjuntos:

1. El conjunto de funciones que con un índice i transforma un almacén en otro, lo llamaremos Det
2. El conjunto de funciones de los almacenés en $\mathbb{BD}$, lo llamaremos Test. Necesitamos generalizar el método por el que representar un predicado parcial por medio de dos predicados totales, para ello proponemos las siguientes definiciones:

   Definición.- Sea r un predicado parcial sobre $\mathbb{BD}$ entonces $\forall n \in \mathbb{D}$ definimos:

$$r^{+_n}(x) = \begin{cases} \underline{\text{true}} \text{ si y solo si } r(x) = (\text{true},n) \\ \\ \underline{\text{false}} \text{ en otro caso} \end{cases}$$

$$r^{-_n}(x) = \begin{cases} \underline{\text{true}} \text{ si y solo si } r(x) = (\text{false},n) \\ \underline{\text{false}} \text{ en otro caso} \end{cases}$$

Por tanto, $\forall n \in \mathbb{D}$, podemos considerar los conjuntos $r^{+_n} = (r^{+_n})^{-1}(\underline{true})$ y $r^{-_n} = (r^{-_n})^{-1}(\underline{true})$. De esta forma: $r^{+_n} \cap r^{-_n} = \emptyset$ $\forall n \in \mathbb{D}$.

Definición.- Sea r un predicado parcial sobre $\mathbb{BD}$ entonces r es representado por el par de predicados totales (r^+, r^-), donde $r^+ = \bigcup_{n\in\mathbb{D}} r^{+_n}$ y $r^- = \bigcup_{n\in\mathbb{D}} r^{-_n}$ de forma que $r^+ \cap r^- = \emptyset$

Con su ayuda construiremos los transformadores de predicados necesarios para el lenguaje:

Conversión .- Definimos $Conv : Deter \longrightarrow TP$ de la siguiente forma:

$$Conv(\pi_S^i)(R) = (\pi_S^i)^{-1}(R)$$

con lo cual $Conv(m) \in TP$

Composición .- Definimos $Comp : TP^2 \longrightarrow TP$ por:

$$Comp(\pi_{S_1}^i, \pi_{S_2}^i) = \pi_{S_1 \circ S_2}^i$$

por lo que $Comp(\pi_{S_1}^i, \pi_{S_2}^i) \in TP$

Condicional .- Definimos $Cond : Test \times TP \longrightarrow TP$ por:

$$Cond(p, \pi_S^i)(R) = \bigcup_{n\in\mathbb{D}} (p^{+_n} \cap \pi_S^{s(i,n)}(R))$$

o lo que es lo mismo

$$Cond(p, \pi_S^i)(R) = (p^+ \cap (\bigcup_{n\in\mathbb{D}} (p^{+_n} \cap \pi_S^{s(i,n)}(R))) \cup$$

$$(p^- \cap \bot_{TP}(R))$$

por lo que la propiedad 4) nos asegura que $\in TP$

Iteración .- Definimos $Do : Test \times TP \longrightarrow TP$ por:

$$Do(p, \pi_S^i)(R) = fix(\lambda Q.(p^- \cap R) \cup$$

$$(p^+ \cap (\bigcup_{n\in\mathbb{D}} (p^{+_n} \cap \pi_S^{s(i,n)}(Q)))))$$

donde $Q \in Pred$. Su significado es el siguiente: para p, π_S^i y cualquier R se define la función Φ_R sobre los predicados por

$$\Phi_R(Q) = (p^- \cap R) \cup (p^+ \cap (\bigcup_{n\in\mathbb{D}} (p^{+_n} \cap \pi_S^{s(i,n)}(Q))))$$

con lo cual Φ_R es continua y por tanto tiene un menor punto fijo $Y(\Phi_R) = \cup\Phi_R^k(\emptyset)$ el cual queremos que sea el valor de $Do(p, \pi)$ en R. A partir de ella construimos ls función $Do_k(p, \pi_S^i) : Pred \longrightarrow Pred$ por;

$$Do_k(p, \pi_S^i)(R) = \Phi_R^k(\emptyset)$$

entonces $Do(p, \pi_S^i)(R) = \cup Do_k(p, \pi_S^i)(R)$. Ahora por inducción sobre k es fácil ver que cada $Do_k(p, \pi_S^i) \in TP$. Esto es claro para para $k = 0$ y para $k + 1$ se observa que;

$$Do_{k+1}(p, \pi_S^i)(R) = \Phi_R(Do_k(p, \pi_S^i)(R)) =$$

$$(p^- \cap R) \cup (p^+ \cap (\bigcup_{n\in\mathbb{D}} (p^{+_n} \cap \pi_S^{s(i,n)}(Do_k(p, \pi_S^i)(R)))))$$

y como tal, por la hipótesis de inducción y la propiedad 4) $Do_{k+1}(p, \pi_S^i) \in TP$. Por tanto $Do(p, \pi_S^i)$ es tambien un transformador de predicados.

Guarda .- Para dar significado semántico al símbolo de Dijkstra $\Box$, se define una función $Guarda : (Test \times TP)^2 \longrightarrow (Test \times TP)$

$$Guarda(< p, \pi_{S_1}^i >, < q, \pi_{S_2}^i >)) =$$

$$< p \vee q, \lambda R.(\bigcup_{n\in\mathbb{D}} (p^{+_n} \cap \pi_{S_1}^{s(i,n)}(R))) \cup$$

$$(\bigcup_{n\in\mathbb{D}} (q^{+_n} \cap \pi_{S_2}^{s(i,n)}(R)) >$$

y teniendo encuenta que $(p^+ \cup q^+) \cap (p^- \cap q^-) = \emptyset$ su cuerpo tambien puede escribirse como:

$$((p^+ \cup q^+) \cap [\bigcup_{n\in\mathbb{D}} (p^{+_n} \cap \pi_{S_1}^i(R)) \cup (q^{+_n} \cap \pi_{S_2}^i(R))]) \cup$$

$$((p^- \cap q^-) \cap \bot_{TP}(R))$$

lo que demuestra que *Guarda* es un transformador de predicados. Por tanto *Guarda* es claramente conmutativa; tambien se puede mostrar que es asociativa, y además, usando una notación obvia:

$$Guarda(< p_1, \pi_{S_1}^i >, \ldots, < p_k, \pi_{S_k}^i >) =$$

$$< \bigvee_{1\le m\le k} p_m, \lambda R. \bigcup_{\substack{1\le m\le k \\ n\in\mathbb{D}}} (p_m^{+_n} \cap \pi_{S_m}^{s(i,n)}(R)) >$$

Por otro lado, *Guarda* no es absorbente: ya que

$$Guarda(< p, \pi_S^i >, < p, \pi_S^i >) =$$

$$< p, \lambda R. \bigcup_{n\in\mathbb{D}} (p^{+n} \cap \pi_S^{s(i,n)}(R)) >$$

la absorción es válida para el par $< p, \pi_S^i >$ si y solo si $\bigcup_{n\in\mathbb{D}} \pi_S^{s(i,n)}(R) \subseteq p^+$ para cualquier

R. Igualmente, el que podriamos considerar como elemento anulador

$$< \lambda\rho.false, \bot_{TP} >$$

no opera como tal, pues

$$Guarda(< \lambda\rho.(false, n), \bot_{TP} >, < p, \pi_S^i >) =$$

$$< p, \lambda R. \bigcup_{n \in \mathbb{D}} (p^{+_n} \cap \pi_S^{s(i,n)}(R)) >$$

2.4 Semántica denotacional

2.4.1 Algebras semánticas

Ya hemos señalado los distintos dominios semánticos sobre los que vamos ha realizar los cálculos, $\mathbb{ND}$, $\mathbb{BD}$, varibles, Σ, etc y como los posibles resultados pertenecera a $\mathcal{P}(\Sigma)$, utilizaremos los operadores $\{|\cdot|\}$ y $\uplus$(Ver apéndice A). Como hacemos uso del "λ-cálculo_b" (Ver apéndice B) hemos de considerar que cada vez que hablemos de funciones estas deberán llevar siempre un grado, y en el caso que dicho grado sea el neutro N de s lo omitiremos, por ejemplo en los elementos $\rho \in \Sigma$, que debemos considerarlo como (ρ,N). El índice que hemos introducido para que refleje el grado con el que estamos trabajando, ya que no es un elemento del "λ-cálculo_b", lo sustituimos por una familia de funciones $I = (\lambda x.x, q)$, donde $q \in \mathbb{D}$.
De las operaciones que se relizan en los dominios semánticos solo necesitamos definir *mod_alm* y $\otimes$:

$$mod\_alm : Var \to \mathbb{ND} \to \Sigma \to \Sigma$$

$$mod\_alm \equiv (\lambda j \lambda n \lambda \rho.[j \mapsto n]\rho, N)$$

donde N es el neutro de s, y además admitimos que: $([j \mapsto n]\rho, q))$ se escriba como $([j \mapsto (n, q)]\rho, N))$

$$\otimes : ((I \times \Sigma) \to \mathcal{P}(\Sigma)) \times ((I \times \Sigma) \to \mathcal{P}(\Sigma)) \to (I \times \Sigma) \to \mathcal{P}(\Sigma)$$

$f_1 \otimes f_2 = (\underline{\lambda}\sigma$ cases σ of
$\sigma \in \Sigma \to f_2(\sigma)$
[] $\sigma = \delta \to \{|\delta|\}$
end)$^+(f_1)$

2.4.2 Funciones de valuación

$$\mathcal{S} : S \to (I \times \Sigma) \to \mathcal{P}(\Sigma)$$

$\mathcal{S}$[[S_1;S_2]] = $\mathcal{S}$[[S_1]] $\otimes$ $\mathcal{S}$[[S_2]]
$\mathcal{S}$[[skip]] = $\lambda(\iota, \rho).\{|\rho|\}$
$\mathcal{S}$[[abort]] = $\lambda(\iota, \rho).\{|\delta|\}$
$\mathcal{S}$[[v:=E]] = $\lambda(\iota, \rho)$.
{|(mod_alm [[v]] ($\mathcal{E}$[[E]]ρ) (ι ρ))|}
$\mathcal{S}$[[case G esac]] = $\lambda(\iota, \rho)$.
$\mathcal{T}$[[G]]$\rho \to \mathcal{G}$[[G]](ι, ρ) [] {|δ|}
$\mathcal{S}$[[do G od]] = $fix(\lambda f.\lambda(\iota, \rho)$.
$\mathcal{T}$[[G]]$\rho \to (\mathcal{G}$[[G]] $\otimes$ $f)(\iota, \rho)$ [] {|ρ|})

$$\mathcal{T} : G \to \Sigma \to \mathbb{BD}$$

$\mathcal{T}$[[G_1 □ G_2]] = $\lambda\rho$. ($\mathcal{T}$[[G_1]]$\rho \wedge \mathcal{T}$[[G_2]]ρ)
$\mathcal{T}$[[B → S]] = $\mathcal{B}$[[B]]

$$\mathcal{G} : G \to (I \times \Sigma) \to \mathcal{P}(\Sigma)$$

$\mathcal{G}$[[G_1 □ G_2]] = $\lambda(\iota, \rho)$.
$\mathcal{G}$[[G_1]]$(\iota, \rho) \uplus \mathcal{G}$[[$G_2$]]$(\iota, \rho)$
$\mathcal{G}$[[B → S]] = $\lambda(\iota, \rho)$.
let $(\lambda x.x, g) = \iota$ in
let (b,h) = $\mathcal{B}$[[B]]ρ in
cases b of
$true \to \mathcal{S}$[[S]]$((\lambda x.x, s(g, h)), \rho)$
[] $false \to \emptyset$
end

$$\mathcal{E} : Expr \to \Sigma \to \mathbb{ND}$$

$\mathcal{E}$[[E_1 **op** E_2]] = $\lambda\rho$.
let $(n_1, d_1) = \mathcal{E}$[[E_1]]ρ in
let $(n_2, d_2) = \mathcal{E}$[[E_2]]ρ in
$(n_1$ **op** $n_2, s(d_1, d_2))$
$\mathcal{E}$[[I]] = $\lambda\rho$. acces [[I]] ρ
$\mathcal{E}$[[(n,d)]] = $\lambda\rho$. (n,d)

$$\mathcal{B} : BExpr \to \Sigma \to \mathbb{BD}$$

$\mathcal{B}$[[(true,d)]] = $\lambda\rho$. (true,d)
$\mathcal{B}$[[(false,d)]] = $\lambda\rho$. (false,d)
$\mathcal{B}$[[E_1 **rel** E_2]] = $\lambda\rho$.
let $(n_1, d_1) = \mathcal{E}$[[E_1]]ρ in
let $(n_2, d_2) = \mathcal{E}$[[E_2]]ρ in
$(n_1$ **rel** $n_2, s(d_1, d_2))$
$\mathcal{B}$[[B_1 **Bop** B_2]] = $\lambda\rho$.
let $(b_1, d_1) = \mathcal{B}$[[B_1]]ρ in
let $(b_2, d_2) = \mathcal{B}$[[B_2]]ρ in
$(b_1$ **Bop** $b_2, s(d_1, d_2))$

3 Conclusiones

Partienedo del conjunto totalmente ordenado $\mathbb{D}$, hemos construido los dominios no planos, $\mathbb{ND}$ y $\mathbb{BD}$, y para manejarlos hemos diseñado un lenguaje del cual se dan las semánticas operanional, axiomática y denotacional. Nos quedan por realizar, entre otras, las siguientes tareas:

Ampliar los tipos de datos incluyendo variables linguisticas

Introducir funciones y procedimientos

Permitir la definición de tipos

Permitir la definición de la lógica que se ha de utilizar

Apéndices

A Los Dominios Potencia

Necesitamos una nueva herramienta para dar cuenta del paralelismo del lenguaje, ya que la respuesta de un programa no será un único elemento de un dominio determinado, sino un conjunto de elementos del dominio, por ello hemos de considerar los dominio potencia y las operaciones definidas sobre él plot.
Para el dominio A el constructor $\mathcal{P}(\_)$ crea el dominio $\mathcal{P}(A)$, una colección cuyos miembros son conjuntos $X \in A$. Las operaciones asociadas, en las que estamos interesados en su continuidad son:

1. $\emptyset : \mathcal{P}(A)$, una constante que denota el menor elemento de $\mathcal{P}(A)$
2. $\{\!|x|\!\} : A \to \mathcal{P}(A)$ que aplica su argumento $a \in A$ en el conjunto $\{a\}$
3. $\_ \uplus \_ : \mathcal{P}(A) \times \mathcal{P}(A) \to \mathcal{P}(A)$ la operación unión binaria que combina sus dos argumentos $M = \{a_0, a_1, \ldots\}$ y $N = \{b_0, b_1, \ldots\}$ en el conjunto $M \cup N = \{a_0, a_1, \ldots, b_0, b_1, \ldots\}$
4. Para $g : A \to \mathcal{P}(B)$ existe una única operación $g^+ : \mathcal{P}(A) \to \mathcal{P}(B)$ tal que para cualquier $M \in \mathcal{P}(A)$, $g^+(M) = \bigcup\{f(m) | m \in M\}$

Para cualquier conjunto S, sea $\mathcal{P}^*_f(S)$ el conjunto de los subconjuntos finitos no vacios de S.

Definición.- Dado un conjunto parcialmente ordenado $(A, \sqsubseteq)$, definimos un preorden sobre $\mathcal{P}^*_f(A)$ como sigue:

1. $u \sqsubseteq_1 v$ si y solo si $(\forall x \in u)(\exists y \in v)$ tal que $x \sqsubseteq y$
2. $u \sqsubseteq_2 v$ si y solo si $(\forall y \in v)(\exists x \in u)$ tal que $x \sqsubseteq y$
3. $u \sqsubseteq_3 v$ si y solo si $u \sqsubseteq_1 v$ y $u \sqsubseteq_2 v$

Si D es un dominio, entonces sea $\mathsf{C}(D)$ es el dominio de ideales sobre $(\mathcal{P}^*_f(\mathsf{K}(D)), \sqsubseteq_2)$. Llamaremos a $\mathsf{C}(D)$ el dominio potencia convexo de D. Similarmente se define el dominio potencia superior $\mathsf{U}(D)$ y el dominio potencia inferior $\mathsf{L}(D)$ los dominios de ideales sobre $(\mathcal{P}^*_f(\mathsf{K}(D)), \sqsubseteq_1)$ y $(\mathcal{P}^*_f(\mathsf{K}(D)), \sqsubseteq_3)$ respectivamente
Las operaciones anteriormente enumeradas se definen, tomando como referencia al dominio potencia convexo, de la siguiente forma:

1. Si $x \in D$, definimos
$$\{\!|x|\!\} = \{u \in \mathcal{P}^*_f(\mathsf{K}(D)) \mid u \sqsubseteq_2 \{a\}\}$$
para algún compacto $a \sqsubseteq x$. Esto forma un ideal, y $\{\!|\cdot|\!\} : D \to \mathsf{C}(D)$ es una función continua.
2. Si $s, t \in \mathsf{C}(D)$ se define una operación binaria
$$s \uplus t = \{w \mid w \sqsubseteq_2 u \cup v \text{ para algún } u \in s \text{ y } v \in t\}$$
Este conjunto es un ideal, y la función $\uplus : \mathsf{C}(D) \times \mathsf{C}(D) \to \mathsf{C}(D)$ es continua.
3. Si $g : D \to E$ es continua se puede definir una función continua g^+ definiendo una función monótona de los elementos compactos de $\mathsf{C}(D)$ en $\mathsf{C}(E)$. Sea $s = \downarrow u$ el ideal principal generado por un subconjunto finito no vacio $u \subseteq \mathsf{K}(D)$. Sea $g(u) = \{g(a) \mid a \in u\}$. Entonces
$$g^+(s) = \{v \in \mathcal{P}^*_f(\mathsf{K}(E)) \mid v \sqsubseteq_2 g(u)\}$$
para algún $u \in s$

Definiciones similares pueden hacerse para los dominios potencia superior e inferior.
Dados los operadores $\{\!|\cdot|\!\}$ y $\uplus$, se puede decir que un punto $x \in D$ es un 'elemento' de un conjunto s en el dominio potencia de D si $\{\!|x|\!\} \uplus s = s$. Si s y t pertenecen al dominio potencia de D, entonces s es un 'subconjunto' de t si $s \uplus = t$. Como es natural estas relaciones 'elemento' y 'subconjunto' tienen diferentes propiedades en los tres dominios potencia. Incluso puede ocurrir que s sea un subconjunto de t sin darse el caso que $s \subseteq t$.

B El λ-cálculo_B

A fin de introducir el concepto de conjunto borroso en el lambda cálculo en [11] se intrroduce el concepto de λ-término_b, formado por un par, cuya primera componente es un λ-término clásico y cuya segunda componente es un elemento de un conjunto D. Como consecuencia se reformula la sustitución y la β-reducción, para dar cuenta de como se manipulan adecuadamente las etiquetas. Si por ejemplo (λx.(A,c), a) y (B,b) son dos de tales λ-términos_b entonces (λx.(A,c), a)(B,b) = ([B/x]A, s(a,s(b,c))) donde s es una operación binaria con las siguientes propiedades:

- Debe tener un neutro
- Debe ser asociativa
- Debe ser conmutativa

en particular si $D = [0, 1]$ s puede ser cualquier t-norma o t-conorma.

Referencias

[1] D. Dubois and H. Prade. *Fuzzy Set and System: Theory and applications.* Academic Press, New York, 1980.

[2] P.D.Moses. Denotational semantics. *In Formals Models and Semantics. (Ed Jan van Leewen). Elsewier. pag.575-629*, 1990.

[3] C.A.Gunter and D.S.Scott. Ssemantics domains. *In Formals Models and Semantics. (Ed Jan van Leewen). Elsewier. pag.633-674*, 1990.

[4] W. Ostasiewicz. A new approach to fuzzy programming. *Fuzzy Sets and Sistems*, 7:139–152, 1982.

[5] E.W.Dijkstra. *A Discipline of Programming.* Prentice-Hall, New Jersey, 1976.

[6] E.G. Manes. A classes of fuzzy theories. *Journal of Mathematical Analysis and Applications*, 85:409–451, 1982.

[7] M. Delgado Calvo-Flores y M.A. Vila Miranda. Software difuso. *En Fundamentos e Introducción a la Ingenieria "Fuzzy". (Ed Enric Trillas). Omron. pag.115-130*, 1994.

[8] J.E.Stoy. *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory.* MIT Press, Cambridge.Mass., 1977.

[9] R.D. Tennent. *Principles of Programming Languages.* Prentice-Hall, Englewood Cliffs,N.J., 1981.

[10] G.D.Plotkin. Dijkstra's predicate transformer and smyth's powerdomains. *Lecture Notes in Computer Science*, 86:527–553, 1978.

[11] D.Sanchez Alvarez. El lambda cálculo_b. *En Actas del VII Congreso Español sobre Tecnología y Lógica Fuzzy. pag.311-316*, 1997.

Función de distribución y mediana de variables aleatorias difusas

Inés Couso Susana Montes Pedro Gil
Departamento de Estadística e I.O. y D.M.
Universidad de Oviedo
e-mail:couso,smr,pedro@pinon.ccu.uniovi.es

Resumen

En este trabajo se extienden los conceptos de función de distribución y de mediana a los casos de conjuntos aleatorios y de variables aleatorias difusas. Se mostrarán importantes propiedades, así como la relación existente entre ambos conceptos.

1 Introducción

Frecuentemente se obtienen mediciones imprecisas en la percepción de características de los individuos resultantes de un experimento aleatorio.

En general, un experimento aleatorio será descrito por un espacio probabilístico, $(\Omega, \mathcal{A}, P)$, donde Ω es el conjunto de todos los posibles resultados o individuos de la población, $\mathcal{A}$ es una σ-álgebra definida en Ω y P es una distribución de probabilidad. Representaremos por medio de una variable aleatoria, $U_0 : \Omega \longrightarrow \Omega'$, cierta característica de los elementos del espacio muestral, donde Ω' es el conjunto de posibles valores o modalidades de la característica. En el caso en que la medición obtenida no sea precisa, no conocemos el valor exacto, $U_0(\omega)$, de la característica para el individuo ω. Lo único que sabemos es que dicho valor pertenece a un subconjunto de Ω' que denotaremos por $\Gamma(\omega)$. Así, podemos definir una aplicación multivaluada, $\Gamma : \Omega \longrightarrow \mathcal{P}(\Omega')$, que representará la percepción imprecisa de la variable aleatoria $U_0 : \Omega \longrightarrow \Omega'$, a la que, siguiendo la notación establecida por los autores Kruse y Meyer[9], llamaremos *variable aleatoria original.* Diremos que Γ es un conjunto aleatorio [10],[1] si, además, cumple la condición de *medibilidad fuerte*[11]:

$$\Gamma^*(A) = \{\omega \in \Omega \mid \Gamma(\omega) \cap A \neq \emptyset\} \in \mathcal{A}, \ \forall A \in \mathcal{A}'$$

Tal como señalan los autores Kruse y Meyer[9], en algunas ocasiones es conveniente modelar la percepción imprecisa de una variable aleatoria mediante una variable aleatoria difusa, en lugar de utilizar un conjunto aleatorio. Así, representaremos nuestro conocimiento acerca de la variable "original" U_0 mediante una aplicación $\widetilde{X} : \Omega \longrightarrow \widetilde{\mathcal{P}}(\Omega')$. El grado de pertenencia de un valor $\omega' \in \Omega'$ al conjunto difuso $\widetilde{X}(\omega)$ indicará el grado de aceptabilidad de la proposición:

$$\text{"}U_0(\omega) = \omega'\text{"}$$

Una variable aleatoria difusa[12],[14] es una aplicación que toma como valores subconjuntos difusos del espacio final y cumple determinada condición de medibilidad. Las definiciones dadas por los distintos autores difieren en la naturaleza del espacio de llegada y en la condicición de medibilidad exigida. En este trabajo, consideraremos la condición de medibilidad fuerte para las aplicaciones α-corte.

Se han realizado multitud de estudios acerca de conjuntos aleatorios y variables aleatorias difusas. Nosotros, en trabajos anteriores [4],[3] hemos generalizado el concepto de probabilidad inducida por una variable aleatoria a los casos de conjuntos aleatorios y de variables aleatorias difusas. Una primera solución del problema para el caso de conjuntos aleatorios podría haber sido dotar al conjunto $\mathcal{P}(\Omega')$, o bien, a una subclase $\mathcal{C} \subseteq \mathcal{P}(\Omega')$, de estructura de espacio medible y definir la probabilidad inducida por la aplicación medible $\Gamma : \Omega \longrightarrow \mathcal{P}(\Omega')$ en el espacio medible $(\mathcal{C}, \mathcal{A}_\mathcal{C})$ de la forma $P \circ \Gamma^{-1} : \mathcal{A}_\mathcal{C} \rightarrow [0,1]$. Lo mismo se podría decir para el caso de variables aleatorias difusas. La resolución del problema de esta forma sería bastante sencilla. Pero, como se comprueba en [3], esta generalización del concepto de probabilidad inducida no es adecuado para nuestros propósitos. En aquellos trabajos [4],[3], definíamos la "envol-

vente probabilística" inducida por una aplicación multivaluada, $\Gamma : \Omega \longrightarrow \mathcal{P}(\Omega')$, de un conjunto medible, $A \in \mathcal{A}'$, como el menor conjunto al cual pertenece con seguridad el valor $P_{U_0}(A)$. Este concepto está estrechamente relacionado con las probabilidades imprecisas [15], las capacidades de Choquet [2], las medidas difusas, los conjuntos convexos y no convexos de probabilidades, las probabilidades superiores e inferiores [5], y la teoría de la evidencia [13], tal y como se muestra en [4],[3]. Después extendíamos el concepto al caso de las variables aleatorias difusas. Para una variable aleatoria difusa, $\widetilde{X} : \Omega \longrightarrow \widetilde{\mathcal{P}}(\Omega')$, la envolvente probabilística de un conjunto medible, $A \in \mathcal{A}'$, está asociada a un subconjunto difuso del intervalo $[0,1]$, que representa nuestro conocimiento acerca del valor $P_{U_0}(A)$. El grado de pertenencia de un determinado valor $p \in [0,1]$ a dicho conjunto difuso representa el grado de aceptabilidad de la proposición:

" p coincide con el valor $P_{U_0}(A)$"

Con el esquema que acabamos de presentar, nos resultará sencillo definir la "envolvente de la función de distribución" tanto para el caso de conjuntos aleatorios como para el caso de variables aleatorias difusas. Existen algunos estudios acerca de la definición de algunos parámetros de variables aleatorias difusas, como la esperanza [1],[12] y la varianza de conjuntos aleatorios y variables aleatorias difusas. En la última sección del trabajo extenderemos el concepto de mediana al caso de conjuntos aleatorios y, después, al caso de variables aleatorias difusas. Definiremos la mediana de una aplicación multivaluada como el menor conjunto que incluye con seguridad al intervalo mediano de la variable aleatoria original, U_0, teniendo en cuenta la información disponible. Además comprobaremos que, en ciertos casos, también se puede definir a partir de la envolvente de la función de distribución. Por último extenderemos el concepto de mediana al caso de variables aleatorias difusas y mostraremos algunas propiedades.

2 Conceptos preliminares

2.1 Envolvente probabilística de conjuntos aleatorios

Sea $(\Omega, \mathcal{A}, P)$ un espacio probabilístico asociado a un determinado experimento aleatorio. Sea $(\Omega', \mathcal{A}')$ un espacio medible y sea $\Gamma : \Omega \longrightarrow \mathcal{P}(\Omega')$ una aplicación multivaluada que representa la observación imprecisa de una aplicación medible, $U_0 : \Omega \to \Omega'$. Llamamos [3],[4] *envolvente probabilística inducida por* Γ a la aplicación $P_\Gamma : \mathcal{A}' \to \mathcal{P}[0,1]$ dada por:

$$P_\Gamma(A) := \{P_U(A) \mid U \in C(\Gamma)\},\ \forall A \in \mathcal{A}',$$

donde $C(\Gamma) = \{U : \Omega \longrightarrow \Omega' \text{ medible } U(\omega) \in \Gamma(\omega)\ c.s.(P)\}$.

Dicho de otra forma, la envolvente probabilística de un conjunto medible, $A \in \mathcal{A}'$ es el menor subconjunto del intervalo $[0,1]$ al cual sabemos con seguridad que pertenece el valor $P_{U_0}(A)$, teniendo en cuenta la información disponible. A cada variable aleatoria, $U \in C(\Gamma)$ la llamaremos variable aleatoria *contenida* en Γ o bien *selección medible de* Γ, siguiendo la notación establecida por otros autores[1],[9].

La probabilidad inducida por U_0 de un conjunto medible $A \in \mathcal{A}'$, $P_{U_0}(A)$ está comprendida entre dos cantidades denominadas por Dempster [5] probabilidad inferior y probabilidad superior, cuya definición solamente tiene sentido si la aplicación Γ es fuertemente medible.

En el caso en que se cumpla la condición $\Gamma(\omega) \neq \emptyset\ \forall \omega \in \Omega$, las probabilidades superior e inferior vendrán dadas por las cantidades $P^*(A) = P(\Gamma^*(A))$ y $P_*(A) = P(\Gamma_*(A))$, donde $\Gamma^*(A) = \{\omega \in \Omega \mid \Gamma(\omega) \cap A \neq \emptyset\}$ y $\Gamma_*(A) = \{\omega \in \Omega \mid \Gamma(\omega) \subseteq A\}$, $\forall A \in \mathcal{A}'$ respectivamente. En ese caso, es evidente que el conjunto $P_\Gamma(A)$ está incluido en el intervalo $[P_*(A), P^*(A)]$. La otra inclusión no es, en general, cierta. Algunos autores representan el conocimiento acerca de la verdadera probabilidad $P_{U_0}(A)$ por medio del intervalo citado. Esto resulta claramente más operativo que el trabajo con el conjunto $P_\Gamma(A)$. Pero puede ocurrir que dicho intervalo no sea suficientemente preciso. Por una parte, sabemos que los valores $P^*(A)$ y $P_*(A)$ acotan el conjunto $P_\Gamma(A)$, pero, en principio, no sabemos si coinciden con el supremo y el ínfimo. En [3] se comprueba que, bajo condiciones bastante generales, las cotas dadas por Dempster son precisamente el supremo y el ínfimo del conjunto $P_\Gamma(A)$. Por lo tanto, el primero de los problemas queda solucionado en un caso bastante general. Por otra parte, el conjunto $P_\Gamma(A)$ no es necesariamente convexo. Dado que lo único que nos interesa

conocer en realidad es una cota superior y otra inferior para el valor $P_{U_0}(A)$, podríamos pensar que la sustitución del conjunto $P_\Gamma(A)$ por su envolvente convexa, $co[P_\Gamma(A)]$ no produce ningún cambio importante. Nosotros[3] hemos comprobado que, en ciertos casos, la sustitución del conjunto $P_\Gamma(A)$ por $co[P_\Gamma(A)]$ lleva a resultados no deseados. No obstante, en [3] se comprueba que, cuando el espacio inicial, $(\Omega, \mathcal{A}, P)$, es no atómico, la envolvente probabilística de cualquier conjunto medible, $P_\Gamma(A)$ es un subconjunto convexo del intervalo $[0,1]$.

2.2 Envolvente probabilística de variables aleatorias difusas

Sea $(\Omega, \mathcal{A}, P)$ un espacio probabilístico asociado a un determinado experimento aleatorio. Sea $(\Omega', \mathcal{A}')$ un espacio medible y sea $U_0 : \Omega \longrightarrow \Omega'$ una variable aleatoria que representa cierta característica de los elementos del espacio muestral. Supongamos ahora que la medición imprecisa de dicha característica viene representada por medio de una aplicación definida en Ω y que toma valores difusos, $\widetilde{X} : \Omega \longrightarrow \widetilde{\mathcal{P}}(\Omega')$. Dicha aplicación define, para cada elemento $\omega \in \Omega$, una función $\mu_{\widetilde{X}(\omega)}$, que asocia a cada elemento $\omega' \in \Omega'$ un valor entre 0 y 1. Este valor puede interpretarse como el grado de confianza o aceptabilidad de que la variable aleatoria desconocida tome el valor ω'.

Definición 2.1 *Sea $(\Omega, \mathcal{A}, P)$ un espacio probabilístico. Sea $(\Omega', \mathcal{A}')$ un espacio medible. Consideremos una aplicación $\widetilde{X} : \Omega \longrightarrow \widetilde{\mathcal{P}}(\Omega')$ que toma valores difusos. Llamamos envolvente probabilística de $\widetilde{X}$ a la aplicación $P_{\widetilde{X}} : \mathcal{A}' \longrightarrow \mathcal{G}[0,1]$ dada por:*

$$P_{\widetilde{X}}(A)(\alpha) := P_{\widetilde{X}_\alpha}(A) \quad \forall A \in \mathcal{A}',\ \forall \alpha \in [0,1].$$

La justificación de esta definición se puede encontrar en [3]. En situaciones bastante generales, el conjunto difuso asociado al conjunto gradual $P_{\widetilde{X}}(A)$ viene dado por la función de pertenencia:

$$\mu_{\widetilde{P}_{\widetilde{X}}(A)}(p) = \sup\{\mu_{\widetilde{X}}(U) \mid U : \Omega \longrightarrow \Omega' \quad \exists V = U \text{ c.s.}(P) \text{ v.a. } P_V(A) = p\}.$$

donde $\mu_{\widetilde{X}}(U) := \inf_{\omega \in \Omega} \mu_{\widetilde{X}(\omega)} U(\omega)$ y se puede interpretar como el grado de aceptabilidad de que U sea la verdadera variable aleatoria que rige el experimento.

De esta forma, se puede entender el grado de pertenencia $\mu_{\widetilde{P}_{\widetilde{X}}(A)}(p)$ como el grado de aceptabilidad de que p sea el verdadero valor de probabililidad del suceso A, $P_{U_0}(A)$.

Algunos autores señalan que, para que un conjunto difuso definido en $\mathbb{R}$ sea la representación imprecisa de un número real debe cumplir algunas propiedades adicionales. De un resultado mencionado en la sección anterior deducimos fácilmente que, cuando el espacio inicial es no atómico, los conjuntos difusos $\widetilde{P_{\widetilde{X}}(A)}$ son *números difusos* [8].

En [3] se recogen algunas otras propiedades importantes de la envolvente probabilística de variables aleatorias difusas.

3 La función de distribución

Como señalábamos en la introducción, en este contexto resulta sencillo definir la envolvente de la función de distribución de aplicaciones multivaluadas y de aplicaciones que toman valores difusos.

3.1 La función de distribución de conjuntos aleatorios

Definición 3.1 *Sea $(\Omega, \mathcal{A}, P)$ un espacio probabilístico. Sea $\Gamma : \Omega \longrightarrow \mathcal{P}(\mathbb{R})$ una aplicación multivaluada que representa la observación imprecisa de una aplicación medible, $U_0 : \Omega \to \mathbb{R}$. Llamaremos envolvente de la función de distribución de Γ a la aplicación $F_\Gamma : \mathbb{R} \to \mathcal{P}([0,1])$ definida de la forma:*

$$F_\Gamma(x) := P_\Gamma((\infty, x]) = \{F_U(x) \mid U \in C(\Gamma)\},\ \forall x \in \mathbb{R}$$

Con otras palabras, $F_\Gamma(x)$ es el menor conjunto que contiene al valor $F_{U_0}(x)$, teniendo en cuenta solamente la información de la que se dispone.

Ejemplo: Sea $\Gamma : \Omega \longrightarrow \mathcal{P}(\mathbb{R})$ una aplicación multivaluada fuertemente medible, de manera que $\Gamma(\omega)$ es un intervalo cerrado, para todo elemento del conjunto Ω. Se puede comprobar que las aplicaciones $f^*, f_* : \Omega \longrightarrow \mathcal{P}(\mathbb{R})$ definidas como $f_*(\omega) = \inf \Gamma(\omega)$, $f^*(\omega) = \sup \Gamma(\omega)$, $\forall \omega \in \Omega$ son variables aleatorias y que la envolvente de la función de distribución de Γ en un punto $x \in \mathbb{R}$ es el conjunto compacto:

$$F_\Gamma(x) = \{P(C) \mid C \in \mathcal{A}, (f^*)^{-1}(-\infty, x] \subseteq C \subseteq (f_*)^{-1}(-\infty, x]\},\ \forall x \in \mathbb{R}$$

Si, además el espacio inicial, $(\Omega, \mathcal{A}, P)$, es no atómico, la envolvente de la función de distribución en un punto queda determinada por sus valores extremos:

$$F_\Gamma(x) = [F_{f^*}(x), F_{f_*}(x)], \ \forall x \in I\!R,$$

lo cual es mucho más operativo.

En esta sección mostraremos algunos resultados relacionados con el concepto que acabamos de definir. La mayor parte de ellos se pueden extender a $I\!R^n$, si bien las demostraciones son más largas.

A continuación expondremos algunas propiedades de la envolvente de la función de distribución que generalizan las propiedades de la función de distribución de una medida de probabilidad en $I\!R$.

Proposición 3.1 *Sea $(\Omega, \mathcal{A}, P)$ un espacio probabilístico. Sea $\Gamma : \Omega \longrightarrow \mathcal{P}(I\!R)$ un conjunto aleatorio (aplicación multivaluada fuertemente medible). Se cumplen las siguientes propiedades:*

1. *Dados dos números $x_1, x_2 \in I\!R$ tales que $x_1 < x_2$ se cumple:*
   - *Si $a \in F_\Gamma(x_1)$, entonces existe $b \geq a$ tal que $b \in F_\Gamma(x_2)$.*
   - *Si $b \in F_\Gamma(x_2)$, entonces existe $b \geq a$ tal que $a \in F_\Gamma(x_1)$.*
2. *Si Γ está acotado superiormente c.s.(P) entonces $\lim_{x \to -\infty} d_H(F_\Gamma(x), \{0\}) = 0$*
3. *Si Γ está acotado inferiormente c.s.(P) entonces $\lim_{x \to \infty} d_H(F_\Gamma(x), \{1\}) = 0$*
4. *Si $\Gamma(\omega)$ es un conjunto compacto para todo $\omega \in \Omega$, entonces: $\lim_{x \downarrow a} d_H(F_\Gamma(x), F_\Gamma(a)) = 0$*

Un resultado bien conocido en el cálculo de probabilidades es que la función de distribución de una variable aleatoria es, a su vez, una aplicación medible. En el siguiente resultado veremos que, bajo condiciones bastante generales, la envolvente de la función de distribución de un conjunto aleatorio es, también, una aplicación multivaluada fuertemente medible.

Proposición 3.2 *Sea $(\Omega, \mathcal{A}, P)$ un espacio de probabilidad. Sea $\Gamma : \Omega \longrightarrow \mathcal{P}(I\!R)$ un conjunto aleatorio de forma que $\Gamma(\omega)$ tiene máximo y mínimo para todo $\omega \in \Omega$. Entonces $F_\Gamma : I\!R \to \mathcal{P}([0,1])$ es una aplicación multivaluada fuertemente medible en el espacio completado, $(I\!R, \overline{\beta_{I\!R}}, \overline{\lambda_{I\!R}})$ y además $F_\Gamma(x)$ es un conjunto compacto, para todo valor real x.*

3.2 La función de distribución de variables aleatorias difusas

La envolvente de la función de distribución se puede generalizar de forma sencilla al caso de variables aleatorias difusas.

Definición 3.2 *Sea $(\Omega, \mathcal{A}, P)$ un espacio probabilístico. Sea $\widetilde{X} : \Omega \longrightarrow \widetilde{\mathcal{P}}(I\!R)$ una aplicación que toma valores difusos y que representa la observación imprecisa de una aplicación medible, $U_0 : \Omega \to I\!R$. Llamaremos envolvente de la función de distribución de $\widetilde{X}$ a la aplicación $F_{\widetilde{X}} : I\!R \to \mathcal{G}([0,1])$ definida de la forma:*

$$F_{\widetilde{X}}(x)(\alpha) := F_{\widetilde{X}_\alpha}(x), \ \forall x \in I\!R, \ \forall \alpha \in [0,1]$$

donde $\mathcal{G}([0,1]$ representa la clase de los subconjunto graduales del intervalo $[0,1]$.

Un *conjunto gradual* [6] es, simplemente, una familia decreciente de conjuntos con índices en el intervalo $[0,1]$. ($\{A(\alpha)\}_{\alpha \in [0,1]}$ tal que $\alpha \geq \beta \Rightarrow A(\alpha) \subseteq A(\beta)$). Todo conjunto gradual está asociado a un único conjunto difuso [6]. La interpretación de esta definición es la siguiente: si, para cada variable aleatoria, $U : \Omega \to I\!R$, definimos el "grado de aceptabilidad de que U sea la variable aleatoria original U_0" de la forma:

$$\mu_{\widetilde{X}}(U) := \inf_{\omega \in \Omega} \mu_{\widetilde{X}(\omega)} U(\omega),$$

tenemos que $U \in C(\widetilde{X}_\alpha)$ si el grado de aceptabilidad de que U coincide con la variable aleatoria original es mayor o igual que α. Así, podremos decir que el grado de aceptabilidad de que $F_U(x)$ coincida con $F_{U_0}(x)$ es mayor o igual que α y, por tanto, pertenece al conjunto $F_{\widetilde{X}}(x)(\alpha)$.

Según un resultado análogo mostrado en la sección 2, la función de pertenencia del conjunto difuso asociado al conjunto gradual $F_{\widetilde{X}}(x)$ tiene la expresión:

$$\mu_{\widetilde{F_{\widetilde{X}}(x)}}(p) = \sup\{\mu_{\widetilde{X}}(U) \mid \exists V : \Omega \to I\!R \text{ v.a.} \\ V = U \text{ c.s.(P) } F_V(x) = p\}$$

Según esta construcción, si denotamos por $\widetilde{F}_{\widetilde{X}}(x)$ el conjunto difuso asociado al conjunto gradual $F_{\widetilde{X}}(x)$, el grado de pertenencia de un número $p \in [0,1]$ a dicho conjunto difuso, $\mu_{\widetilde{F}_{\widetilde{X}}(x)}(p)$ representará el grado de aceptabilidad de la proposición:

"$p = F_{U_0}(x)$"

Los autores Kruse y Meyer[9] definen la función de distribución de una variable aleatoria difusa de forma parecida, aunque existen algunas diferencias notables. En primer lugar, ellos suponen que, para cualquier característica de los individuos de una población, existen dos espacios probabilísticos, $(\Omega, \mathcal{A}, P)$ y $(\Omega', \mathcal{A}', P')$ de forma que, para todo valor $\lambda \in [0,1]$, existe algún conjunto $A' \in \mathcal{A}'$ de manera que $P'(A') = \lambda$ y la variable aleatoria que representa la característica citada, U_0, está definida en el espacio producto, $(\Omega \times \Omega', \mathcal{A} \otimes \mathcal{A}', P \times P')$. También exigen que la variable aleatoria difusa que representa la medición imprecisa de la característica tenga α-cortes convexos, es decir, $\widetilde{X}_\alpha(\omega) = [\widetilde{X}(\omega)]^{[\alpha]}$ ha de ser un conjunto convexo, para todo elemento $\omega \in \Omega$ y para todo $\alpha \in [0,1]$. Los autores exigen todas estas condiciones para que la función de distribución aplicada a un punto sea un subconjunto difuso convexo de $\mathbb{R}$ (número difuso [8]). Nosotros creemos que, en muchas ocasiones, no tiene sentido considerar un espacio inicial de la forma citada. En esos casos, la sustitución de $\widetilde{F}_{\widetilde{X}}(x)$ por el conjunto difuso convexo asociado puede llevar a resultados muy diferentes. Por otra parte, los autores definen la función de disribución directamente, sin extender previamente el concepto de probabilidad inducida al caso de variables aleatorias difusas. Además, teniendo en cuenta un resultado citado en la sección 2, se puede comprobar que, cuando el espacio inicial es no atómico, los valores que toma la aplicación $\widetilde{F}_{\widetilde{X}}$ son números difusos, es decir, que sus α- cortes son conjuntos convexos. Con este resultado vemos que no es necesario que los conjuntos $\widetilde{X}_\alpha(\omega)$ sean convexos para que los conjuntos difusos $\widetilde{F}_\Gamma(x)$ sean convexos, tal como exigen los autores Kruse y Meyer [9].

En el resultado siguiente, veremos que la envolvente de la función de distribución de una variable aleatoria difusa con α-cortes compactos es también una variable aleatoria difusa con α-cortes compactos.

Proposición 3.3 *Sea $(\Omega, \mathcal{A}, P)$ un espacio probabilístico no atómico. Sea $\widetilde{X} : \Omega \longrightarrow \widetilde{\mathcal{P}}(\mathbb{R})$ una variable aleatoria difusa de forma que $\widetilde{X}_\alpha(\omega)$ es un conjunto compacto, para todo $\omega \in \Omega$, para todo $\alpha \in (0,1]$. Sea $F_{\widetilde{X}} : \mathbb{R} \to \mathcal{G}([0,1])$ la envolvente de las función de distribución de $\widetilde{X}$. Para cada valor $x \in \mathbb{R}$, denotemos por $\widetilde{F}_{\widetilde{X}}(x)$ el conjunto difuso asociado al conjunto gradual $F_{\widetilde{X}}(x)$. Entonces la aplicación $\widetilde{F}_{\widetilde{X}}$ es una variable aleatoria difusa. Además $[\widetilde{F}_{\widetilde{X}}(x)]^{[\alpha]}$ es un conjunto compacto para todo $x \in \mathbb{R}$, para todo $\alpha \in (0,1]$.*

4 La mediana

4.1 La mediana de conjuntos aleatorios

Dado un espacio probabilístico , $(\Omega, \mathcal{A}, P)$, y una variable aleatoria real definida en él, $U_0 : \Omega \to \mathbb{R}$, se define el intervalo mediano de U_0 de la forma:

$$Me(U_0) = \{x \in \mathbb{R} \mid F_{U_0}(x^-) \leq 1/2 \leq F_{U_0}(x)\}$$

Definiremos la mediana de una aplicación multivaluada como la unión de los intervalos medianos de todas las selecciones medibles de Γ:

Definición 4.1 *Sea $(\Omega, \mathcal{A}, P)$ un espacio de probabilidad. Sea $\Gamma : \Omega \longrightarrow \mathcal{P}(\mathbb{R})$ una aplicación multivaluada definida en Ω. Se llama envolvente de la mediana de Γ al conjunto:*

$$Me(\Gamma) := \cup_{U \in C(\Gamma)} Me(U) =$$
$$\{x \in \mathbb{R} \mid \exists\, U \in C(\Gamma),\ x \in Me(U)\}$$

Según esta definición, la envolvente de la mediana de Γ es el menor conjunto que incluye con seguridad al intervalo mediano, teniendo en cuenta la información disponible.

Tal como señalábamos en la introducción, veremos que, bajo ciertas condiciones, la envolvente de la mediana se puede expresar a partir de la envolvente de la función de distribución. Tal como hemos definido la envolvente de la mediana, resulta sencillo comprobar que está incluida en el conjunto:

$$\{x \in \mathbb{R} \mid \exists\, a, b \in P_\Gamma(-\infty, x],\ a \leq 1/2 \leq b\}.$$

A continuación se muestra que, bajo ciertas condiciones, ambos conjuntos coinciden. Además se puede comprobar que el conjunto anterior es convexo. Por tanto, en esas condiciones, la envolvente de la mediana quedará determinada por sus valores extremos.

Proposición 4.1 *Sea $(\Omega, \mathcal{A}, P)$ un espacio de probabilidad. Sea $\Gamma : \Omega \longrightarrow \mathcal{P}(\mathbb{R})$ una aplicación multivaluada definida en Ω, de manera que $\Gamma(\omega)$ es un conjunto convexo para todo $\omega \in \Omega$, o bien, $(\Omega, \mathcal{A}, P)$ es un espacio no atómico. Entonces:*

$$Me(\Gamma) = \{x \in \mathbb{R} \mid \exists\, a, b \in P_\Gamma(-\infty, x],\ a \leq \frac{1}{2} \leq b\}$$

En ese caso, $Me(\Gamma)$ es un intervalo real.

4.2 La mediana de variables aleatorias difusas

Podemos generalizar el concepto de mediana al caso de variables aleatorias difusas.

Definición 4.2 *Sea $(\Omega, \mathcal{A}, P)$ un espacio de probabilidad. Sea $\widetilde{X} : \Omega \longrightarrow \widetilde{\mathcal{P}}(\mathbb{R})$ una aplicación multivaluada definida en Ω. Se llama envolvente de la mediana de $\widetilde{X}$ al conjunto gradual $Me(\widetilde{X})$ definido de la forma:*

$$Me(\widetilde{X})(\alpha) := Me(\widetilde{X}_\alpha),\ \forall \alpha \in [0,1]$$

La interpretación de esta definición es similar a la de la función de distribución. Si, para cada variable aleatoria, $U : \Omega \to \mathbb{R}$, tenemos que el "grado de aceptabilidad de que U sea la variable aleatoria original U_0" de la forma:

$$\mu_{\widetilde{X}}(U) := \inf_{\omega \in \Omega} \mu_{\widetilde{X}(\omega)} U(\omega),$$

tenemos que $U \in C(\widetilde{X}_\alpha)$ si el grado de aceptabilidad de que U coincide con la variable aleatoria original es mayor o igual que α. Así, podremos decir que el grado de aceptabilidad de que un valor perteneciente al conjunto $Me(U)$ pertenezca al conjunto $Me(U_0)$ es mayor o igual que α y, por tanto, pertenece al conjunto $Me(\widetilde{X})(\alpha)$.

La expresión de la función de pertenencia del conjunto difuso asociado al conjunto gradual $Me(\widetilde{X})$ es la siguiente:

$$\mu_{\widetilde{Me}(\widetilde{X})}(x) = \sup\{\mu_{\widetilde{X}}(U) \text{ t.q. } \exists V = U \text{ c.s. } (P), x \in Me(V)\}$$

Así, $\mu_{\widetilde{Me}(\widetilde{X})}(x)$ se puede entender como el grado de aceptabilidad de la proposición

$$Me(U_0) = x$$

El conjunto difuso $\widetilde{Me}(\widetilde{X})$ será un número difuso bajo condiciones análogas a las de la proposición 4.1.

Referencias

[1] J. Aumann(1965) Integral of set valued functions, *J. Math. Anal. Appl.* **12** 1-12.

[2] G. Choquet (1954) Theory of Capacities. *Ann. Inst. Fourier (Grenoble)* **V** 131-295.

[3] I. Couso (1997) *La envolvente probabilística: definición y propiedades* Trabajo de Investigación. Departamento de Estadística e I.O. y D.M. Universidad de Oviedo.

[4] I. Couso, P.Gil (1997) *La envolvente probabilística de variables aleatorias difusas* ESTYLF'97. Tarragona.

[5] A.P. Dempster (1967) Upper and Lower Probabilities Induced by a Multi-valued Mapping, *Ann. Math. Statistics* **38** 325-339.

[6] J.A. Herencia (1995) *Origen y Uso de los conjuntos Graduales.* Tesis Doctoral. Departamento de Matemáticas. Universidad de Córdoba.

[7] P.J. Huber (1981) *Robust Statistics*, John Wiley and sons.

[8] A. Kaufmann, M. Gupta (1985) *Introduction to fuzzy arithmetic* Van Nostrand Reinhold.

[9] R. Kruse, K.D. Meyer (1987) *Statistics with vague data* D. Reidel Publishing Company.

[10] G. Matheron (1975) *Random sets and integral geometry* Wiley, New York.

[11] H.T. Nguyen (1978) On random sets and belief functions, *J. Math. Analysis and Applications* **63** 531-542

[12] M.L. Puri y D. Ralescu (1986) Fuzzy Random Variables, *J. Math. Anal. Appl.* **114** 409-422.

[13] G. Shafer (1976) *A mathematical theory of evidence* Princeton University Press.

[14] M. Stojaković (1992) Fuzzy conditional expectation *Fuzzy Sets and Systems* **52** 53-60

[15] P. Walley (1991) *Statistical Reasoning with Imprecise Probabilities*, Chapman and Hall.

Análisis estructural de la base de reglas de un controlador difuso

P. Carmona
Dpto. Informática. *Facultad de Ciencias*
Universidad de Extremadura
Avda. Elvas, s/n. 06017-Badajoz (Spain)
pablo@unex.es

J.M. Zurita*
Dpto. Ciencias de la Computación e I.A.
E.T.S.I. Informática. Universidad de Granada
Avenida de Andalucía, 38 18071-Granada (Spain)
zurita@decsai.ugr.es

Resumen

La base de reglas que representa el conocimiento en un controlador difuso puede contener deficiencias estructurales que se traduzcan en un comportamiento inadecuado o ineficiente. En el presente trabajo se propone un método para detectar contradicciones en dicha base de reglas, integrando en el concepto de contradicción una componente gradual. El método permite aprovechar la simetría presente en la base de conocimiento de un gran número de controladores difusos (como los PD y PI), al hacer uso de ella para conducir al experto en la búsqueda de posibles errores tanto en la definición de las reglas como en la definición de los dominios difusos asociados a las variables de estado y control. Se ilustra el método con un ejemplo.

1 Introducción

La verificación y validación de sistemas es una etapa clave para asegurar la utilidad y fiabilidad de los mismos. Esto hace que la existencia de herramientas que ayuden a esa tarea sea indispensable para el desarrollo de cualquier sistema, para poder así contrastar la solución obtenida con una serie de requerimientos que establezcan su validez. Las herramientas de verificación y validación de software tradicionales han disminuido su eficacia a medida que los sistemas sobre los que actúan han ido aumentando su complejidad.

Éste es el caso de los sistemas basados en el conocimiento [3]. Los sistemas expertos, sistemas de apoyo para la toma de decisiones, sistemas de control automático y cualquier otro que explote un conocimiento que le ha sido transferido mediante algún mecanismo, posee unas características que hace que las herramientas clásicas sean insuficientes para asegurar su correción. Una de las principales es la existencia de una importante dosis de subjetividad e imprecisión inherente al propio conocimiento experto.

Los sistemas basados en el conocimiento pueden encerrar distintos tipos de anomalías estructurales que los hagan inconsistentes o incompletos, tales como redundancia, circularidad, deficiencia o ambivalencia [8][6][7][10]. La tarea de verificación debe ser capaz de detectar tales anomalías y proporcionar los medios apropiados para que el ingeniero del conocimiento o el experto puedan subsanarlas. Aún más, una herramienta de verificación del conocimiento puede proporcionar al experto que lo aportó información sobre su estructura o el modo en que hace uso de él.

La siguiente sección introduce someramente el concepto de controlador difuso y sus ventajas respecto a los controladores tradicionales. En la sección tercera se describe una medida que permite verificar la existencia de un importante tipo de anomalías en cualquier base de conocimiento: las contradicciones. En la sección cuarta se expone un mecanismo para el análisis de las contradiciones sobre la totalidad de la base de conocimiento empleando la medida anterior y para la transferencia al experto de los resultados del análisis.

2 Controladores difusos

La teoría del control automático es una disciplina encaminada a diseñar herramientas capaces de gobernar el comportamiento de un proceso mediante el análisis de su estado y la aplicación como resultado de una acción sobre el mismo para que alcance el comportamiento deseado. Con ese fin, los controladores obtienen la información a través de una o más variables de estado o de entrada y, en base a éstas, establecen el valor de la variable de control o de salida, como puede verse en la figura 1.

Tradicionalmente, la teoría de control se ha venido abordando desde el punto de vista de la ingeniería

*Correspondencia a Dr. J.M. Zurita

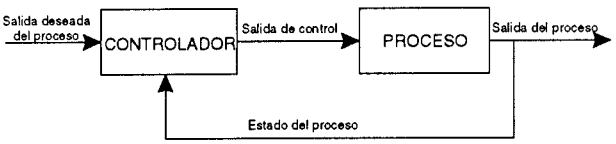


Figura 1: Esquema general de control

de sistemas. Sin embargo, este enfoque es incapaz de tratar de forma directa sistemas que integren algunas características a menudo presentes en el mundo real, tales como sistemas dinámicos, sistemas sometidos a perturbaciones, sistemas flexibles y, en general, sistemas no lineales [4]. Además, requieren la obtención de un modelo matemático que describa el sistema a controlar, lo cual, en muchos casos, es inviable.

Los sistemas basados en el conocimiento han surgido como una alternativa al enfoque tradicional que a menudo permite superar las deficiencias de las que adolece éste. Surgen así los *controladores basados en el conocimiento*, dentro de los cuales los controladores difusos están cobrando cada día mayor importancia tanto desde el punto de vista científico como desde el tecnólogico e industrial.

Un controlador difuso emplea la lógica difusa [12] para el tratamiento del conocimento. Para ello, hace uso de una base de conocimiento que representa a través de un conjunto de reglas difusas. Cada regla posee un antecedente que indica la situación en la que dicha regla debe aplicarse mediante la asignación de una restricción difusa a cada una de las variables de estado y un consecuente donde se expresa la acción a realizar mediante la aplicación de una restricción difusa a la variable de control. Es decir, cada regla será de la forma:

R^i : if X_1 is $A^i_{1,j}$ and ... and X_n is $A^i_{n,k}$ then Y is B^i_l

donde X_j son las variables de estado, Y es la variable de control y toman sus valores difusos de los conjuntos $\{A_{j,1}, ..., A_{j,p_j}\}$ y $\{B_1, ..., B_q\}$ definidos sobre los dominios U_j y V, respectivamente.

El tratamiento del conocimiento expresado en una base de reglas requiere la utilización de un mecanismo de inferencia. En control, habitualmente se emplea la composición sup-min en combinación con la implicación de Mamdani [5]. Una mayor profundización en el campo de los controladores difusos puede encontrarse en [2] y [11].

3 Grado de contradicción entre reglas

La posibilidad de generar contradicciones ha constituido desde siempre un tipo de anomalía inaceptable en cualquier base de reglas clásica. En la lógica de predicados, dos reglas son contradictorias si sus antecedentes son idénticos y sus consecuentes contrarios. En el caso de la lógica difusa, la contradicción no puede medirse en términos absolutos, sino que, en consonancia con la naturaleza difusa inherente a esta lógica, está sujeta a una graduación.

En un reciente trabajo [1] se presenta una medida que permite establecer el grado de contradicción entre pares de reglas difusas. Para ello, modifica el concepto clásico de contradicción integrando en él las nociones de similitud y disimilitud entre valores difusos.

La idea principal consiste en considerar en qué medida los antecedentes de las reglas en estudio son idénticos y los consecuentes contrarios. Dos reglas con los mismos antecedentes serán tanto más contradictorias cuanto más distintos sean sus consecuentes. Por otro lado, dados dos pares de reglas con el mismo grado de disimilitud entre sus consecuentes, será más contradictorio aquél formado por reglas con antecedentes más similares.

Para la obtención del grado de disimilitud entre los consecuentes difusos B^i y B^j se hace uso de la distancia media [9] que, tras ser normalizada, proporciona un valor d_N entre 0 y 1. Es decir,

$$d_N(B^i, B^j) = \frac{d(B^i, B^j)}{d_{\max}}$$

donde

$$d(B^i, B^j) = \frac{\sum\sum_{v_1, v_2 \in V} |v_1 - v_2| \inf[\mu_{B^i}(v_1), \mu_{B^j}(v_2)]}{\sum\sum_{v_1, v_2 \in V} \inf[\mu_{B^i}(v_1), \mu_{B^j}(v_2)]}$$

y como distancia máxima ($d_{\max}$) se considera la longitud del dominio sobre el que se definen los conjuntos difusos, que equivale a la distancia máxima posible entre dos conjuntos difusos definidos en dicho dominio.

El grado de similitud entre los antecedentes A^i y A^j se define como la inversa de las distancias medias parciales entre cada par de premisas difusas (A^i_k, A^j_k) asociado a la k-ésima variable de estado:

$$s_N(A^i, A^j) = 1 - \frac{\sum_{k=1}^{n} d_N(A^i_k, A^j_k)}{n}$$

Finalmente, la medida de contradicción entre las reglas R^i y R^j vendrá expresada en función de ambos grados como

$$C(R^i, R^j) = d_N(B^i, B^j) \times s_N(A^i, A^j)$$

4 Análisis de la base de reglas

En esta sección se mostrará cómo puede emplearse la medida de contradicción descrita en la sección anterior para proporcionar al experto (o al ingeniero

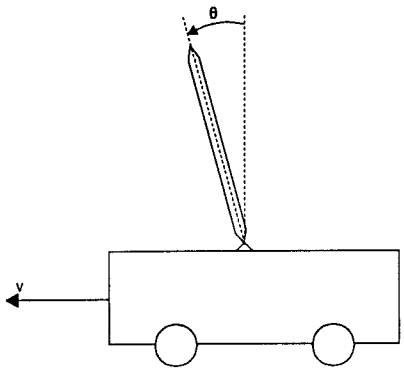


Figura 2: El problema del péndulo invertido.

del conocimiento) información que permita poner de manifiesto posibles defectos en la base de reglas difusa que representa el conocimiento.

Centraremos el análisis en controladores difusos del tipo PD o PI, cuya base de conocimiento está formada por n reglas de la forma

$$R^i : \text{if } X_1 \text{ is } A^i_{1,j} \text{ and } X_2 \text{ is } A^i_{2,k} \text{ then } Y \text{ is } B^i_l$$

donde $i \in \{1, ..., n\}$. Es decir, reglas que contemplan dos variables de estado y una de control. Las variables de estado X_1 y X_2 tomarán, respectivamente, valores difusos de los conjuntos $\{A_{1,1}, ..., A_{1,p_1}\}$ y $\{A_{2,1}, ..., A_{2,p_2}\}$ definidos sobre los dominios U_1 y U_2, mientras que la variable de control Y tomará valores difusos del conjunto $\{B_1, ..., B_q\}$ definido sobre el dominio V. La elección de controladores difusos PD o PI se debe a que, en general, poseen una base de reglas con una topología en su representación tabular que mantiene una simetría entre los elementos de coordenadas (i, j) y $(F - i + 1, C - j + 1)$, donde F y C son el número de filas y número de columnas de la tabla, respectivamente. Por lo tanto, cualquier controlador de esas características podría ser enteramente aplicado al método que a continuación se propone.

A modo de ilustración, consideraremos el problema del péndulo invertido. En él, una barra o péndulo situado sobre un vehículo debe ser mantenido en posición vertical mediante el movimiento del vehículo en uno de dos posibles sentidos y a una determinada velocidad (fig. 2). Para ello, se aplica la acción de control sobre el vehículo en el instante t en base al ángulo $\theta(t)$ que forma el péndulo con la vertical y a la velocidad angular $\dot{\theta}(t)$. Este problema se ha resuelto satisfactoriamente mediante la aplicación de la teoría del control difuso.

En nuestro ejemplo, el sistema estará gobernado por un controlador difuso del tipo PD cuya base de conocimiento está formada por el siguiente conjunto

		$\dot{\theta}$				
		NM	NS	Z	PS	PM
θ	NM			PM		
	NS			PS	Z	
	Z	PM	PS	Z	NS	NM
	PS		Z	NS		
	PM			NM		

Tabla 1: Base de reglas expresada en forma tabular

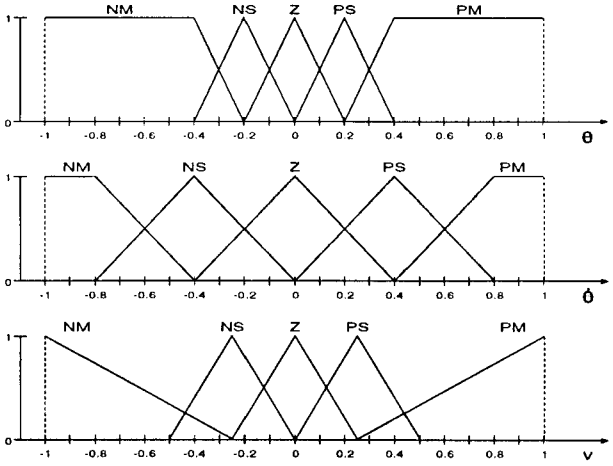


Figura 3: Dominios difusos de las variables de control y estado.

de reglas difusas

R^1 : if θ is PM and $\dot{\theta}$ is Z then v is NM
R^2 : if θ is Z and $\dot{\theta}$ is NM then v is PM
R^3 : if θ is Z and $\dot{\theta}$ is PM then v is NM
R^4 : if θ is NM and $\dot{\theta}$ is Z then v is PM
R^5 : if θ is PS and $\dot{\theta}$ is Z then v is NS
R^6 : if θ is Z and $\dot{\theta}$ is NS then v is PS
R^7 : if θ is NS and $\dot{\theta}$ is PS then v is Z
R^8 : if θ is Z and $\dot{\theta}$ is PS then v is NS
R^9 : if θ is NS and $\dot{\theta}$ is Z then v is PS
R^{10} : if θ is PS and $\dot{\theta}$ is NS then v is Z
R^{11} : if θ is Z and $\dot{\theta}$ is Z then v is Z

representado en forma tabular en la tabla 1. La etiqueta Z hace referencia al cero y en las etiquetas NM, NS, PS y PM, la primera letra significa *negativo/positivo* y la segunda *medio/pequeño*.

Cada una de las reglas contempla en el antecedente las variables de estado θ y $\dot{\theta}$, y en el consecuente la variable de control v, que toman sus valores difusos de los dominios de la figura 3.

Una primera aproximación al análisis podría consistir

en obtener el grado de contradicción de cada regla con respecto conjunto de la base de reglas. De este modo, el experto puede tener una idea de qué porción del conocimiento expresado puede estar en conflicto con la globalidad y centrar su atención en dicha parte.

El grado de contradicción de la regla R^i vendrá expresado como la media aritmética de los grados de contradicción parciales de R^i con cada una de las reglas, es decir:

$$C(R^i) = \frac{\sum_{j=1}^{n} C(R^i, R^j)}{n}$$

Para proporcionar esta información al experto puede aprovecharse la forma tabular empleada en la representación de la base de reglas y, de este modo, permitir, no sólo el análisis individual del grado de contradicción obtenido para cada regla, sino el estudio de las posibles relaciones entre ellos a tenor de la topología de la base de reglas. De este modo, la información puede proporcionarse como una matriz de $p \times q$ nodos, de forma que cada nodo (j, k) contendrá el grado de contradicción de la regla

$$R^i : \text{if } X_1 \text{ is } A^i_{1,j} \text{ and } X_2 \text{ is } A^i_{2,k} \text{ then } ...$$

En la figura 4 se muestran los grados de contradicción globales para las reglas del ejemplo. Puede observarse cómo las reglas R^1, R^2, R^3 y R^4 son las que presentan un grado de contradicción más alto, posiblemente debido a que deben controlar situaciones donde una de las variables de estado toma valores extremos y en consecuencia la acción de control requerida también lo es en cierto grado. Obsérvese también que en el ejemplo la base de reglas posee una topología fácilmente identificable, ya que tanto la acción del controlador (tabla 1) como los dominios difusos de las variables (figura 3) guardan una determinada simetría. Esa información puede utilizarse para el análisis de los grados de contradicción que, como puede verse en la figura 4, mantienen la misma simetría que la tabla de reglas. Así, en este ejemplo, la obtención de un grado de contradicción para una regla distinto al de su simétrica alertaría al experto sobre la existencia de algún error, bien en los antecedentes de las reglas, bien en los consecuentes o bien en la definición de los dominios difusos asociados.

Los grados de contradicción globales de las reglas a veces no son suficientes para conocer exactamente qué porción de la base de reglas contiene la contradicción. Por ello, en la mayoría de los casos será necesario también apoyarse en el grado de contradicción entre cada par de reglas. Esto puede hacerse conectando cada par de nodos de la matriz mediante un arco etiquetado con el grado de contradicción entre dichos nodos. Ahora bien, dado que, en general, los grados de contradicción más altos son los de mayor interés, puede

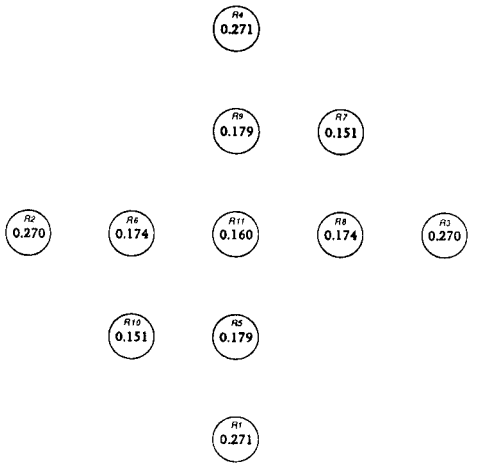


Figura 4: Matriz de grados de contradicción globales entre reglas.

establecerse un umbral δ que determine los arcos que van a mostrarse finalmente en el grafo de contradicciones y centrar así la atención del experto en la información relevante. Es decir, dadas dos reglas

$$R^{i_1} : \text{if } X_1 \text{ is } A^{i_1}_{1,j_1} \text{ and } X_2 \text{ is } A^{i_1}_{2,k_1} \text{ then } ...$$
$$R^{i_2} : \text{if } X_1 \text{ is } A^{i_2}_{1,j_2} \text{ and } X_2 \text{ is } A^{i_2}_{2,k_2} \text{ then } ...$$

existirá un arco que una los nodos (j_1, k_1) y (j_2, k_2) etiquetado con el valor $C(R^{i_1}, R^{i_2})$ si $C(R^{i_1}, R^{i_2}) \geq \delta$.

Por otro lado, al igual que ocurría con los grados de contradicción globales, a veces puede ser conveniente utilizar información sobre la topología de la tabla de reglas para contrastarla con los grados de contradicción obtenidos entre pares de reglas y orientar al experto en la búsqueda de fallos.

En la figura 5 se muestra el grafo de contradicciones del ejemplo considerando un umbral igual a 0.35 y donde puede observarse que los grados de contradicción más altos siguen perteneciendo a los pares de reglas que controlan situaciones extremas ($C(R^1, R^2) = C(R^3, R^4) = 0.470$). Ahora, supóngase que la regla R^8 es sustituida erróneamente por

$$R^8 : \text{if } \theta \text{ is } Z \text{ and } \dot{\theta} \text{ is } PS \text{ then } v \text{ is } PS$$

El grado de contradicción global de cada regla nos indica que existe algún tipo de conflicto en la regla R^3, pero un análisis más profundo revela que la contradicción viene provocada, en su mayor parte, por la inconsistencia con la regla R^8. Esto puede verse gráficamente en la figura 6, donde se considera un umbral de 0.45 y se muestran también los arcos que contienen grados de contradicción en desacuerdo con la simetría reflejada en la base de reglas. Obsérvese cómo

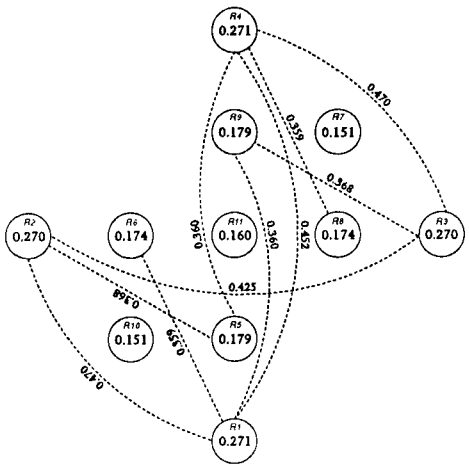


Figura 5: Grafo de contradicciones de la base de reglas original ($\delta = 0.35$).

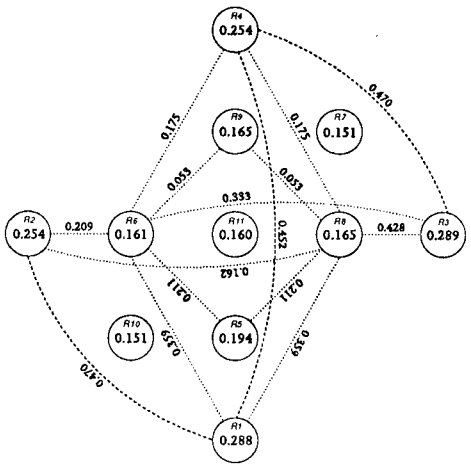


Figura 6: Grafo de contradicciones de la base de reglas modificada ($\delta = 0.45$).

el hecho de que todos los arcos que parten de la regla R^8 rompen con dicha simetría refuerza la hipótesis de que es ésa la regla incorrecta.

Por lo tanto, es posible ver cómo el grafo de contradicciones puede ser una herramienta útil para detectar posibles conflictos en el conocimiento expresado mediante una base de reglas difusa. Ahora bien, debe advertirse que el método mostrado en ningún modo asegura la existencia real de errores, sino que trata de orientar al experto hacia la búsqueda de posibles mejoras en la base de reglas y, a la vez, proporcionarle información sobre su propio conocimiento, el cual puede contener inconsistencias no advertidas en principio por el experto.

5 Conclusión

La verificación de cualquier sistema basado en el conocimiento es una etapa indispensable para asegurar su fiabilidad y, por tanto, su utilidad. En el campo del control automático, los controladores difusos representan el conocimiento mediante un conjunto de reglas difusas. Se presenta en este trabajo un método para proporcionar al experto información a distintos niveles sobre el grado de contradicción de la base de reglas. Además, el método utiliza características topológicas de la misma para facilitar la localización de la porción de conocimiento que presenta cierto grado de contradicción.

El método, por tanto, es aplicable en su totalidad a aquellos controladores donde la base de reglas y los dominios difusos asociados a las variables de entrada y de control mantengan una determinada simetría. Esta característica se encuentra presente en gran número de controladores, como los del tipo PD, PI y otros. Por otro lado, aquellos controladores que no presenten la topología descrita pueden igualmente hacer uso de la información proporcionada por el grafo de contradicciones, si bien la detección de las contradicciones requerirá un mayor esfuerzo.

Conviene reseñar que la información proporcionada debe interpretarse como una orientación hacia posibles modificaciones en la base de reglas que eliminen o mitiguen efectos indeseados que pudiera provocar la inclusión de conocimiento contradictorio de forma involuntaria y no como una señal inequívoca de la existencia de errores. El experto deberá plantearse si la modificación de la base de reglas es compatible con el comportamiento deseado del controlador.

El método de detección de contradicciones mostrado en este trabajo requiere un grado de participación importante por parte del experto. La detección de forma autónoma de divergencias entre el grafo y la topología de la tabla de reglas supondría una mejora del método expuesto. Por otro lado, la capacidad de sugerir al experto las modificaciones apropiadas en la base de reglas que subsanen los conflictos detectados (y la determinación de su existencia real) requeriría la consideración de aspectos tales como el efecto de esas variaciones sobre el comportamiento del controlador o el modelo subyacente del sistema a controlar. Éstas mejoras quedan abiertas a futuros estudios.

Referencias

[1] P. Carmona y J.M. Zurita. "Fuzzy Knowledge Based Controllers Verification". Por aparecer en *Proc. of the Int. Conf. on Information Processing*

and Management of Uncertainty in Knowledge-Based Systems (IPMU). Paris, (1998).

[2] D. Driankov, H. Hellendoorn y M. Reinfrank. *An Introduction to Fuzzy Control.* New York, Springer-Verlag (1996).

[3] M. Stefik. *Introduction to Knowledge Systems.* San Francisco, CA, Morgan Kauffman Publishers (1995).

[4] B.C. Kuo. *Sistemas de Control Automático.* México, Prentice-Hall Hispanoamericana (1996).

[5] E.H. Mamdani. "Applications of fuzzy logic to approximate reasoning using linguistic synthesis". *IEEE Transactions on Computers*, **26** (12) (1977), 1182-1191.

[6] T.A. Nguyen, W.A. Perkins, T.J. Laffey y D. Pecora. "Knowledge Base Verification". *AI Magazine*, **8** (2) (1987), 69-75.

[7] R.M. O'Keefe y D.E. O'Leary. "Expert System Verification and Validation: A Survey and Tutorial". *Artificial Intelligence Review*, **8** (1993), 3-42.

[8] A.D. Preece y R. Shinghal. "Foundation and Application of Knowledge Base Verification". *International Journal of Intelligent Systems*, **9** (8) (1994), 683-701.

[9] A. Rosenfeld. "Distances Between Fuzzy Sets". Pattern Recognition Letters, **3** (1985), 229-233.

[10] M. Rousset. "On the consistency of knowledge bases: The COVADIS system". *En Proceedings of the 8th European Conference on Artificial Intelligence (ECAI'88),* (1988), 79-84.

[11] M. Sugeno. "An Introductory Survey of Fuzzy Control", *Information Sciences*, **36** (1985), 59-83.

[12] L.A. Zadeh, "Fuzzy Sets", *Information and Control*, **8** (3) (1965), 338-353.

AN INTERCONNECTED FUZZY LOGIC CONTROLLER FOR SYNCHRONOUS GENERATOR AND TURBINE

Zbigniew Lubośny, Hocine Tiliouine

Annotation: *The paper presents interconnected fuzzy logic based controller of synchronous generator and turbine. There is presented general structure of the controller consisting of: voltage regulator, speed (and real power) regulator and stabilising element influencing synchronous generator and turbine controllers. The controller utilises one stabilising element which directly influences on governor and indirectly (throughout filter) influences on synchronous generator controller. The filter allows changing the magnitude and phase of the stabilising signal. Its parameters are chosen appropriate to achieve high damping of electromechanical oscillations and are modified during generating unit operation according to kind of disturbance.*
The paper presents way of implementation of such a controller to synchronous generator and turbine control. The results of tests, carried out throughout simulations on mathematical model of generating unit working parallel with power system, are presented.

1. Introduction

The problems related to synthesis and constructing synchronous generator and turbine controllers appear due to:

- Large variety of possible operating condition and disturbances which can occur in power system.
- Variation of plant parameters as a result of power network configuration change.
- Complexity and non-linearity of electric power system.
- Difficulty with working out mathematical models capable of adequately describing generating unit in its various operating conditions.
- The state-of-the-art classical methods for designing above control which usually turn out impractical and inefficient.
- Opposite influence of voltage regulator on the process of voltage control and damping of electromechanical oscillations. The fast voltage regulator ensures fast control and high damping of generator voltage but causes simultaneously high swings of active power, rotor angle, speed etc. And on the contrary, the voltage regulator, which allows achieving high damping of electromechanical oscillations (swings of active power, speed, rotor angle) can not ensure fast control of generator voltage. The voltage error is then minimised slowly and with high oscillations.

The conventional controllers are unable to perform optimally over the full range of operating conditions and disturbances. Then today to overcome the above difficulties and fulfil requirements related to generating unit controller performance the following three broad categories of control methods are being employed:

- New non-adaptive linear control - Linear Quadratic Gaussian (LQG) and H-infinity polynominal theories [1, 2]
- Non-adaptive non-linear control - artificial neural networks and fuzzy logic theories that allow to use heuristics to synthesise of efficient controllers [3, 4, 5, 6, 7].
- Adaptive non-linear control - because of drawbacks of classical adaptive schemes (especially related to problems with parameters estimator which is susceptible to numerical difficulties [8, 9]) the artificial neural networks and fuzzy logic theories are used to synthesise of efficient adaptive controllers [3, 10, 11].

The carried out investigations [5, 6, 7] show high effectiveness of fuzzy control schemes applied to generating unit control and specially effectiveness of integrated control of synchronous generator and turbine

Dr inż. Zbigniew Lubośny
Division of Electric Power Systems
Technical University of Gdańsk
G. Narutowicza 11/12 Str.
80-952 Gdańsk, PL
Phone: 058-3472034, Fax: 058-3471898
E-mail: zlub@ely.pg.gda.pl

Dr inż. Hocine Tiliouine
Division of Theoretical Electrotechnics
Technical University of Gdańsk
G. Narutowicza 11/12 Str.
80-952 Gdańsk, PL.
Phone: 058-3471335, Fax: 058-3471898
E-mail: tiliouin@sparc10.ely.pg.gda.pl

schemes [12,13,14]. Therefore an integrated fuzzy logic controller of synchronous generator and turbine (as interesting and prospective proposition for synchronous generator and turbine control) is considered and presented in the paper.

2. Fuzzy logic control

In systems using the fuzzy logic, the portion of the controller that accomplishes the *control rules* (control algorithm) is preceded by the *fuzzification*, and followed by the *defuzzification* (Fig. 1). The three stages of a fuzzy controller are shortly described below.

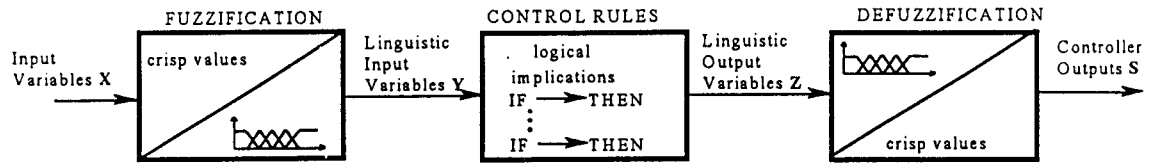


Fig. 1. Fuzzy logic control system

Fuzzification is a process **F** which transfers input value $x_k \in X_k$ to a set of fuzzy membership values $\mu_i \in <0, 1>$ according to the membership function defined in the universe of discourse of X_k (Fig. 2):

$$\mathbf{F}: x_k \rightarrow \{(m_A^k, \mu_A^k(x_k)),...,(m_i^k, \mu_i^k(x_k))\} \tag{1}$$

where: m_i^k - membership of *k*th input x_k in the *i*th fuzzy set (centre value of the *i*th fuzzy set),

μ_i^k - membership value of x_k in the *i*th fuzzy set,

i=1,...,N - index, where N is the number of fuzzy sets applicable to x_k.

When the membership function is trapezoidal shape, the membership consists of two values: $m_{i,L}^k$, $m_{i,R}^k$ left and right membership.

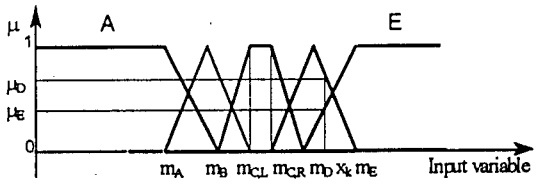


Fig. 2. Fuzzification process example

Control rules define function **R** that determines fuzzy control actions by matching them with fuzzy information:

$$\mathbf{R}: \{(m_A^k, \mu_A^k(x_k)),...,(m_i^k, \mu_i^k(x_k))\} \otimes \{(m_A^l, \mu_A^l(x_l)),...,(m_j^l, \mu_j^l(x_l))\} \rightarrow \{(m_A^s, \mu_A^s(s)),...,(m_p^s, \mu_p^s(s))\} \tag{2}$$

where: m_p^s - membership of the *p*th fuzzy action set,

μ_p^s - strength of the *p*th action (i.e. membership value of output s in the *p*th fuzzy set),

j=1,...,M, p=1,...,P - indexes, where M and P - numbers of fuzzy sets applicable to x_l and *s*,

$\otimes$ - operator which means logic operation on Cartesian product.

As a practical matter, the logic rules that make up the controller algorithm can usually be expressed in the form $\mathbf{A}^k \wedge (\vee)\mathbf{B}^l \Rightarrow \mathbf{C}^s$. Thus, in addition to the inference function, the rules chiefly rely on the logic product and logic sum functions. The truth values of these functions can be determined using the T-operators. The latter can be constructed in a variety of ways. Here the formulations given by Zadech have been adopted, where the truth value of logic sum and logic product are defined as (x_k and x_l denotes different input signals):

$$T(\mu_A^k(x_k), \mu_B^l(x_l)) = \max(\mu_A^k(x_k), \mu_B^l(x_l)), \qquad G(\mu_A^k(x_k), \mu_B^l(x_l)) = \min(\mu_A^k(x_k), \mu_B^l(x_l)) \tag{3}$$

The defuzzification involves generation of the ultimate output (control) signal by appropriately converting the results of logic manipulations on fuzzy sets. This can be symbolically represented as:

$$\mathbf{D}: \{(m_A^s, \mu_A^s(s)),...,(m_p^s, \mu_p^s(s))\} \rightarrow s \tag{4}$$

The basic defuzzification method, the centre of area (COA) method, from point of view of implementation in real system (i.e. digital) is impractical. The faster and practical method for computing the control signal is realised on using the so-called *centroid defuzzification function*:

$$s = \sum_{p=1}^{P} m_p^s \mu_p \Big/ \sum_{p=1}^{P} \mu_p \tag{5}$$

In the paper a following simplified defuzzification procedure has been used. For this procedure, if P is the number of action fuzzy sets, the value of control signal s is computed from:

$$s = f(\sum_{p=1}^{P} m_p^s \mu_p(s)) \tag{6}$$

where: m_p^s - weight coefficient equal to its corresponding fuzzy membership,

μ_p - membership value of s in the pth action (output) fuzzy set,

$f(\bullet)$ - slope function with limitations 0 and 1.

The functions (5) and (6) will yield the same results when $\Sigma\mu_p=1$. As regards the considered synchronous generator and turbine controller, the above condition is satisfied in the steady state and during low-swing transients. When the swing is high, the above total may periodically be violated, what leads to an increase in the control signal s. However, this does not adversely affect either the controller effectiveness or the system stability.

3. Structure of the controller

Structure of the proposed controller in block diagram form is given in Fig. 3. The fuzzy controller is composed of four following parts:

- Voltage controller, with inputs: voltage error (V_g - V_{ref}) and derivative of the generator voltage dV_g/dt, and output s_E which is introduced to static exciter.
- Speed (and active power) controller, with inputs: speed error (ω_{ref} - ω_g), speed derivative $d\omega_g/dt$ and output signal s_G. The feedback loop from output of speed controller into its input is added. The feedback loop allows to introduce the permanent speed drop χ and then allow to operate of generating unit parallel with power system. The speed reference value ω_{ref} directly depends on active power reference value P_{ref}.
- Power system stabiliser with inputs: derivative of the generator voltage dV_g/dt, derivative of active power dP_g/dt and the speed derivative $d\omega_g/dt$ and output s_{PSS} added directly to speed controller output s_G and indirectly (throughout) to voltage controller output s_E.
- Low pass, first order filter, which modifies magnitude and phase of the output signal of power system stabiliser s_{PSS} and influences excitation system (signal s_F).

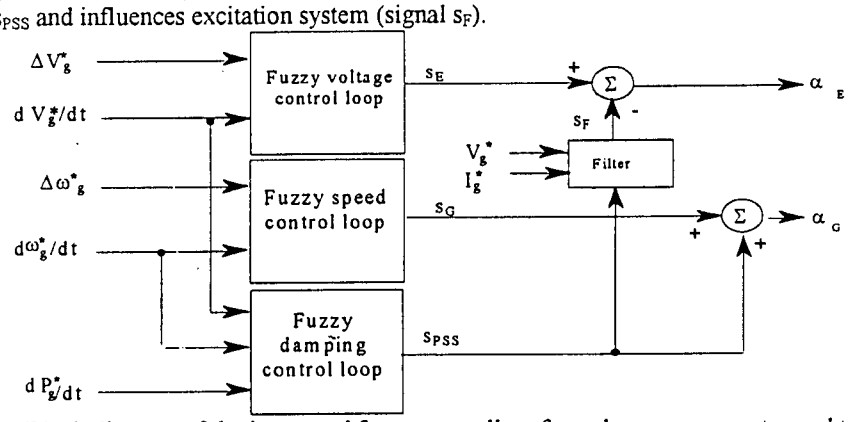


Fig. 3. Block diagram of the integrated fuzzy controller of synchronous generator and turbine

The linguistic input variables as well as the linguistic output variables of the voltage controller and the speed controller are defined as five fuzzy sets. However for the power system stabiliser, to every linguistic input variable and to every linguistic output variable correspond three fuzzy sets. To describe the fuzzy sets of the linguistic variables for error inputs and outputs have been considered the membership functions of triangular shape and for the derivative inputs, have been considered the membership functions of triangular and trapezoidal shape as shown in Fig. 2.

The output signal $\alpha_E = s_E - s_F$ (Fig. 3) of the voltage controller and power system stabiliser (after filter) is applied to the excitation system. The output signal $\alpha_G = s_G + s_{PSS}$ of speed (and real power) controller and power system stabiliser is introduced to the electro-hydraulic converter of hydraulic governor (labelled HG in Fig. 4) controlling position of turbine valves.

The action of the voltage controller and the action of the speed controller are defined in each case by 25 rules. Eighteen other rules are created to define the action of the power system stabiliser.

As a filter there has been considered low pass, first order one described by transfer function $K_F/(1+sT_F)$. Parameters of the filter the gain K_F and the time constant T_F were established during optimisation process. Their values depend on type of disturbance, which can occur in the power system. When the disturbance takes place the proper (for given disturbance type) values of filter coefficients are settled. The values of the coefficients change itself during transient process and converge to their values appropriate to steady state of generating unit. The change of filter coefficients is realised by exponential function with high time constant (approximately 20 s). The type of disturbance is identified on the basis of terminal voltage V_g and armature current I_g.

4. Implementation of the controller

The fuzzy logic based controller of synchronous generator and turbine controller was tested in single machine power system (Fig. 4) as digital one with sampling constant T_p=10ms.

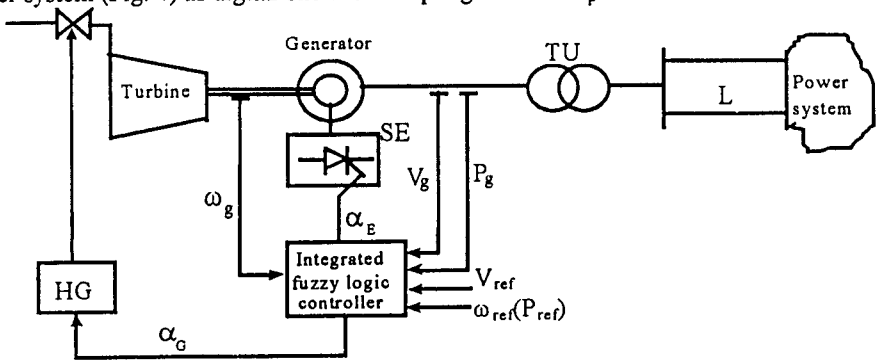


Fig. 4. Block diagram of investigated large-power generating unit. (SE - static exciter, TU - step-up transformer, L - transmission line, HG – hydraulic part of governor, V_g, V_{ref} - synchronous generator voltage and reference voltage, ω_g, ω_{ref} - shaft speed and reference speed, P_g, P_{ref} - active power and reference power, α_E, α_G - control signals)

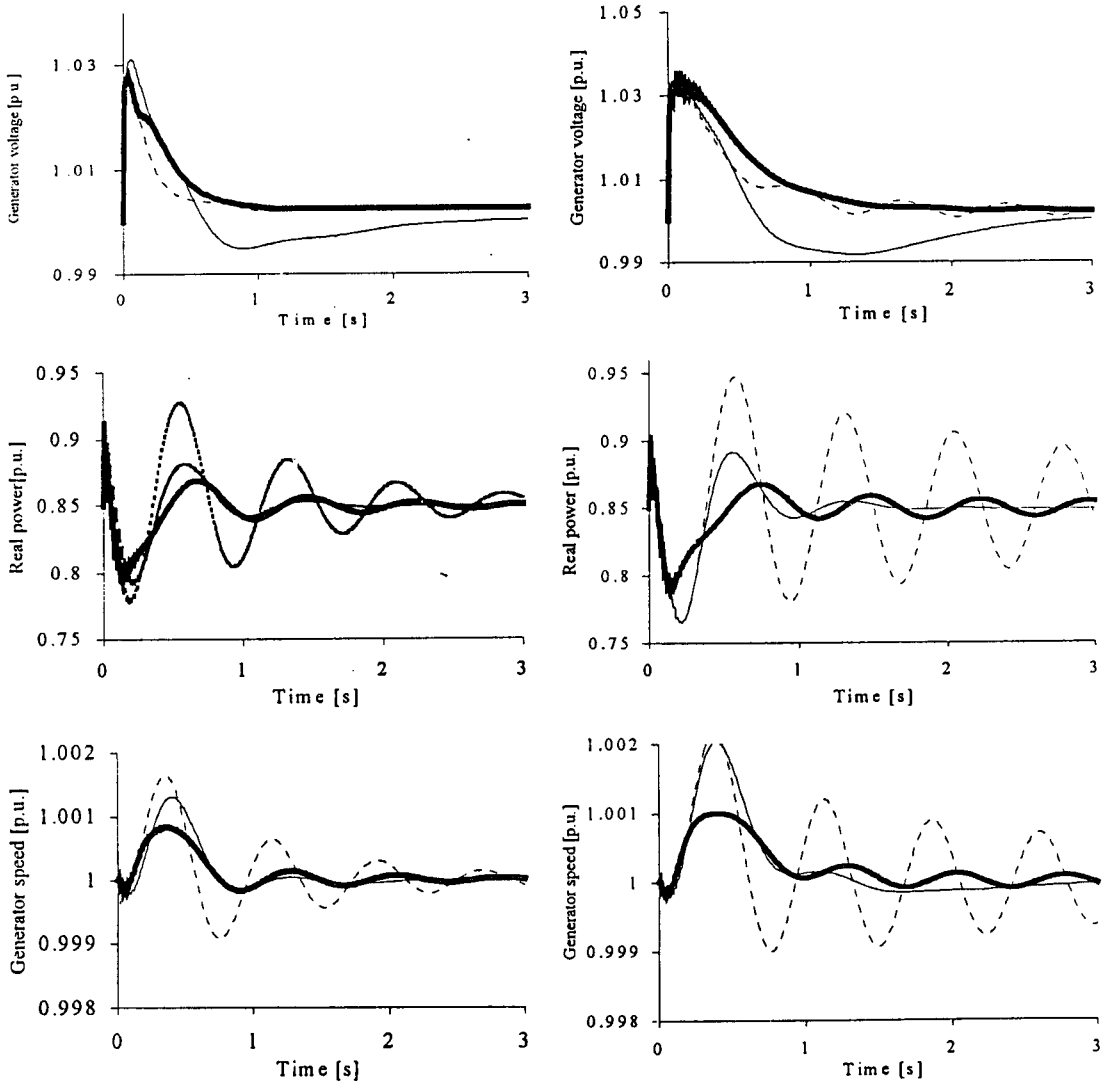


Fig. 5. Generator voltage, real power and speed swings after 5% step increase in infinite bus voltage, **a** - $P_g=P_{gn}$, $Q_g=Q_{gn}$; **b** - $P_g=P_{gn}$, $Q_g=0$, K_F=0.72, T_F=0.2s,. (thin line – conventional controller with PSS, bold line – interconnected fuzzy controller with PSS, dotted line - interconnected fuzzy controller without PSS)

The performance of the proposed fuzzy logic controller has been investigated in a number of simulation tests for a variety of operating conditions and disturbances. Performance comparison has been made with three

variants of controllers: conventional controller equipped with a PSS, proposed in the paper interconnected fuzzy logic controller, and presented in the paper fuzzy logic controller but without fuzzy PSS.

Selected results of carried out tests are presented in the following Figure. In Fig. 5.a there is presented comparison of the mentioned systems as a response on a 5% step change in infinite bus voltage in case when initial load of synchronous generator is nominal and equal to Pg=0.85, Qg=-0.53 (inductive).
Presented curves show that implemented fuzzy controller allows to achieve high level of damping of electromechanical oscillations. Such comparison lets to appreciate not only the high damping effect of the fuzzy PSS but to show also the superiority of the proposed controller over the conventional controller.

In Fig. 5.b there is presented comparison of the mentioned systems as a response on a 5% step change in infinite bus voltage in case when initial load of synchronous generator is equal to Pg=0.85, Qg=0. In this case the operating point of synchronous generator is close to stability border. The increase of infinite bus voltage additionally reduces the stability margin and then synchronous generator operates closer to the stability border. The simulation results show that even in considered case proposed controller is more effective than classical one (even equipped with PSS).

In both cases we can notice that the presence of the PSS causes a light deterioration in the properties of the terminal voltage control process. But we can observe also that the terminal voltage is less affected with the proposed controller than with the conventional controller. The active power responses and the speed responses in both cases show clearly the high damping effect of the PSS utilisation. This effect is higher, especially for the first swing, for the proposed interconnected fuzzy controller.

6. Conclusions

An integrated fuzzy logic controller for a generator excitation and governing systems was developed in this paper. To evaluate its performance simulation tests, under different operating conditions, were carried out. The test results demonstrate high damping effect of the PSS and the superiority of the proposed controller in comparison with a conventional controller. The presented interconnected fuzzy logic controller of synchronous generator and turbine is characterised by following primary advantages:

- High dynamic efficiency with high damping of electromechanical oscillations.
- Low sensitivity to the initial operating state of generating unit. In particular, the controller enjoys low sensitivity to the initial generator load, while also being capable of accommodating largely different parameters of the power network.
- Easiness of the robust controller developing by heuristics utilisation.
- Simplicity of the algorithm, what allow to easy implementation of the controller in real systems.

Literature

1. Eker I., et al: „Robust cascade control design for process control" AMST'94 Application of Multivariable System Techniques, 29-30 March, Bradford, U.K. pp. 119-131.
2. Hogg B.W.: „Representation and Control of Turbogenerators in Electric Power Systems" in Nicholson, H. 1981.
3. Fukuda T., Shibata T.: „Theory and Applications of Neural Networks for Industrial Control Systems" IEEE Transactions on Industrial Electronics, Vol. 39, No 6, December 1992. pp. 472-489.
4. Furuhashi T., Hasegawa T., Horikawa S., Uchikawa Y.: „An adaptive fuzzy controller using fuzzy neural networks" Fifth IFSA World Congress (1993), pp. 769-772.
5. Lubośny Z.: „Neural network based fuzzy controller of synchronous generator", Stockholm Power Tech Conference, 18-22.06.1995.
6. Lubośny Z.: „Improvement of the synchronous generator performance by using knowledge base in ANN controller" 31st Universities Power Engineering Conference 1996 - UPEC'96, Iraklio, Crete, 18-20,09,1996.
7. Lubośny Z.: „Fuzzy logic based microcontroller for the synchronous generator" International Symposium on Modern Electric Power Systems MEPS'96, Wroc³aw 09.96.
8. Wu Q., Hogg B. W.: „Adaptive control for a turbogenerator system" IEE Proceedings Pt. D, Vol. 135, 1988, pp 35-41.
9. Lubośny Z., Pałkowski J., Szczerba Z.: „Adaptacyjny regulator generatora synchronicznego GTHW-360". Międzynarodowa Konferencja APE-93 16-17.09.1993, Gliwice-Kozubnik.
10. Lubośny Z.: „An adaptive controller for the synchronous generator" 2nd International Conference on Control of Power Systems, Bratys³awa, 14-15.05.1996.
11. Lown M., Swindebank E., Hogg B. W.: „Adaptive fuzzy logic control of a turbine generator system" Proceedings of the 31st Universities Power Engineering Conference UPEC'96, 18-20 September 1996, pp. 543-546.
12. Wu Q., Hogg B. W., Irwin G. W.: „A neural network regulator for turbogenerators", IEEE Transation on Neural Networks, Vol. 3, No 1, January 1992, pp. 95-100.
13. Lubośny Z., Tiliouine H.: "An integrated neural-network-based adaptive controller of generator and turbine unit", Międzynarodowa Konferencja SYS 95, Brno, 3-5.07.1995.
14. Lubośny Z.: „Powiązania układów regulacji generatora synchronicznego i turbiny poprawiające własności dynamiczne turbozespołu w systemie elektroenergetycznym". Archiwum Energetyki, Tom XXI, 1992, Nr 3-4.

PROTOTIPADO RÁPIDO DE CONTROLADORES BORROSOS DEDICADOS ORIENTADOS A CÓMPUTO *

José Ignacio Martínez Torre
Arquitectura de Computadores y Automática
Facultad de Ciencias Físicas
Univ Complutense de Madrid
tel: + 34 1 394 51 86
fax: + 34 1 394 46 87
e-mail: jimt@eucmax.sim.ucm.es

Jean-Pierre Deschamps
Dpto. Ingeniería Informática
Univ Autónoma de Madrid
tel: +34 1 394 51 86
fax: + 34 1 394 46 87
e-mail:
jimt@eucmax.sim.ucm.es

Milagros Fernandez Centeno
Arquitectura de Computadores y Automática
Facultad de Ciencias Físicas
Univ Complutense de Madrid
tel: + 34 1 394 43 86
fax: + 34 1 394 46 87
e-mail: mila45@eucmax.sim.ucm.es

Resumen

En este artículo se propone la exploración de distintas alternativas de materialización de controladores borrosos dedicados (CBD) orientados a cómputo empleando técnicas de Síntesis de alto nivel (SAN).
Las principales ventajas de esta aproximación son el uso controlado de los recursos físicos para adecuarnos a las restricciones, la capacidad de selección de la materialización más adecuada para cada aplicación específica sin limitarnos a arquitecturas predefinidas, y la reducción del tiempo de diseño que transcurre desde la especificación del CBD hasta la verificación de la funcionalidad del diseño final.
Palabras clave: Prototipado rápido, hardware dedicado, síntesis de alto nivel, reuso de hardware, planificación.

1 INTRODUCCIÓN

Cada vez existe un mayor número de aplicaciones (comerciales, científicas, etc.) que ante la dificultad o imposibilidad de crear modelos que simulen razonablemente bien su funcionamiento utilizan las técnicas que proporcionan los sistemas basados en lógica borrosa [1].
En caso de que los requisitos de procesamiento no sean excesivamente exigentes, muchas de estas soluciones se realizan en software que se ejecuta sobre procesadores de propósito general o sobre controladores convencionales. En caso contrario, se recurre a soluciones hardware con dedicación exclusiva [2,3].
Dentro de este tipo de aproximación, la mayoría de los autores de la literatura se restringen a materializar las soluciones sobre una serie de módulos hardware previamente definidos sin más que adecuar sus parámetros al problema concreto, pero sin comprobar si el número de módulos empleado es óptimo o si está siendo aprovechada razonablemente su funcionalidad.
Por ejemplo, se pueden estar realizando sumas en el algoritmo sobre variables distintas o con distinta vida activa sobre módulos sumadores distintos, cuando puede haber módulos sumadores inactivos pertenecientes a una etapa anterior que podríamos volver a usar salvando tiempo y posiblemente área.
Con frecuencia, las restricciones en la velocidad de operación de una aplicación vienen dadas como un rango de valores. Por tanto, es muy útil poder explorar cuáles son las mejores arquitecturas posibles para ese intervalo.
Esta exploración de arquitecturas válidas no sólo pretende tener en cuenta restricciones de área y tiempo sino también restricciones de interfaz, que se concretan en el tipo de almacenamiento de datos, anchura de las líneas de comunicaciones, etc.
La sencillez de los algoritmos de lógica borrosa hace que esa exploración de alternativas pueda ir fácilmente desde la colocación de tantos módulos como se necesiten (explotación del paralelismo, incremento de área frente a reducción de tiempo de ejecución) hasta la resolución secuencial del problema (explotación de los recursos, incremento el tiempo de ejecución frente a la reducción del número de módulos), pasando por cualquier alternativa intermedia.
El principal objetivo de nuestro trabajo es utilizar al máximo el margen que exista para cada aplicación para poder seleccionar la mejor alternativa de materialización hardware sin tener que estar mediatizados por organizaciones hardware previamente definidas.

2 DESCRIPCIÓN DEL CBD

Los fundamentos de esta metodología global para el desarrollo de hardware borroso dedicado fueron presentados en una publicación anterior [4]. En este artículo nos centraremos en su aplicación sobre un ejemplo concreto [5].
Si queremos obtener el máximo rendimiento de un sistema fotovoltaico (panel solar y convertidor conmutado dc-dc) con acumulación por batería - normalmente empleados en aplicaciones terrestres - es imprescindible su polarización en el punto de máxima potencia en todo momento y bajo

Este trabajo ha sido subvencionado por los programas TIC96-1071 y HCM CHRX-CT94-0459.

cualquier condición, es decir realizar el seguimiento del punto de máxima potencia (SPMP).
Sin entrar en los detalles técnicos, el control se realiza a través de un controlador borroso dedicado, que recibe como entrada dos variables E y CE (error y cambio en el error del sistema) y que genera una única variable de salida dD (variación del ciclo de trabajo del convertidor), cuya interfaz y base de reglas se muestran en la Figura 1.
En este caso tanto las variables de entrada como la de salida están especificadas mediante cinco funciones de pertenencia denominadas NB, NS, ZO, PS y PB.

dD	**CE**				
	NB	*NS*	*ZO*	*PS*	*PB*
E					
NB	**ZO**	**ZO**	**NB**	**NB**	**NB**
NS	**NS**	**ZO**	**NS**	**NS**	**NS**
ZO	**NB**	**ZO**	**ZO**	**ZO**	**PS**
PS	**ZO**	**PS**	**PS**	**ZO**	**PS**
PB	**PB**	**PB**	**PB**	**ZO**	**ZO**

Figura 1: Base de reglas

A la hora de definir completamente la especificación del controlador borroso hay que fijar un gran número de parámetros. Algunos de éstos se fijan durante la etapa de verificación de la funcionalidad del controlador borroso (por ejemplo, el método de inferencia), mientras que otros tienen que ver con la materialización deseada o con el entorno donde se va a integrar dicho controlador (por ejemplo, el estilo de difuminado: orientado a memoria o a cómputo).
La elección de los parámetros guiará los siguientes pasos en el flujo de diseño. Por ejemplo, con un estilo de difuminado orientado a memoria es imprescindible conocer cuál es número de memorias, la organización de sus datos, etc. Mientras que para un estilo de difuminado orientado a cómputo es crucial cuál es el formato de los datos empleados en la reconstrucción de las funciones de pertenencia y el algoritmo empleado para llevarlo a cabo.
En la Figura 2 se muestra la descripción de alto nivel del CBD que define su interfaz, la base de reglas y el resto de parámetros del ejemplo.

3 ALGORITMO DEL CBD VÁLIDO PARA SAN

Desde esta especificación se puede generar un algoritmo que represente al CBD con un nivel de abstracción lo suficientemente alto como para que sea factible aplicar sobre él técnicas de SAN.

```
SPECIFICATION SPMP

 INPUTS
 E  NB NS ZO PS PB UOFD –1 +1  PREC 8 MBFV  0 1 PREC 4
 PATH Epath
 CE  NB NS ZO PS PB UOFD –1 +1  PREC 8 MBFV  0 1 PREC 4
 PATH CEpath

 OUTPUTS
 dD  NB NS ZO PS PB UOFD –1 +1  PREC 8 MBFV  0 1 PREC 4
 PATH CEpath

 RULES
 IF  E is NB and CE is NB THEN dD is ZO
 IF  E is NB and CE is NS THEN dD is ZO
 IF  E is NB and CE is ZO THEN dD is NB
 ...
 IF  E is PB and CE is PB THEN dD is ZO

 FUZZIFICATION          Singleton
 INFERENCE              Min-Max
 DEFUZZIFICATION        WACoG
```

Figura 2: Especificación del problema

Este proceso lo lleva a cabo un módulo software que analiza la corrección léxica y sintáctica de la especificación, detectando inconsistencias de varios tipos como duplicación de entradas, salidas, antecedentes o consecuentes, incoherencias en las reglas, etc.
En un primer momento la herramienta de SAN escogida fue la construida y diseñada por nuestro grupo de trabajo, denominadas Fidias [6], por su funcionalidad y por la carencia en el ámbito comercial de herramientas válidas. Sin embargo, ante la aparición de herramientas comerciales [7] el algoritmo para SAN se puede crear sin más que utilizar un nuevo módulo generador de código.
En la Figura 3 se muestra un código genérico para SAN de un CBD orientado a cómputo.

```
PROGRAM SPMP

VAR [DECLARACIÓN DE VARIABLES];
PORT  [DECLARACIÓN DE PUERTOS];
BGN

[ASIGNACIÓN DE PUERTOS A VARIABLES];
[ACCESO A LOS PTOS CRÍTICOS DE LAS MBFS];
[ALGORITMO DE RECONSTRUCCIÓN DE LAS MBFS DE ENT];
[ALGORITMO RETICULAR];
[ALGORITMO DE RECONSTRUCCIÓN DE LAS MBFS DE SAL];
[ALGORITMO DE CÓMPUTO DE VALORES DE SALIDA];
[ASIGNACIÓN DE VARIBLES A PUERTOS];

END
```

Figura 3: Algoritmo para SAN

4 NÚMERO DE OPERADORES RETICULARES

Se puede comprobar que una traducción directa del método de inferencia puede implicar un número excesivo de operadores reticulares que no puede ser reducido mediante técnicas de SAN. Para solventar este problema se ha propuesto la solución siguiente.

Toda función reticular (1) se representa directamente como un conjunto de operadores min y max con dos operandos (2).

$$w = x_1 \cdot x_2 \cdot x_5 \vee x_1 \cdot x_2 \cdot x_4 \vee x_3 \cdot x_5 \vee x_3 \cdot x_4 \quad (1)$$

$$\begin{array}{ll} r_1 = x_1 \cdot x_2 & \downarrow \\ r_2 = r_1 \cdot x_5 & r_6 = x_3 \cdot x_4 \\ r_3 = x_1 \cdot x_2 & r_7 = r_2 \vee r_4 \\ r_4 = r_3 \cdot x_4 & r_8 = r_5 \vee r_6 \\ r_5 = x_3 \cdot x_5 & w = r_7 \vee r_8 \end{array} \quad (2)$$

Parece claro que (2) no es una solución óptima ya que (1) puede reducirse aún más, si se omiten las expresiones reticulares duplicadas, por ejemplo $x_1 \cdot x_2$.

Existen proposiciones del cálculo reticular que permiten reducir la longitud original de (3), y por tanto el algoritmo de cómputo reticular (4) ahora carece de expresiones intermedias redundantes.

$$w = (x_1 \cdot x_2 \vee x_3) \cdot (x_4 \vee x_5) \quad (3)$$

$$\begin{array}{l} r_1 = x_1 \cdot x_2 \\ r_2 = r_1 \vee x_3 \\ r_3 = x_4 \vee x_5 \\ w = r_2 \cdot r_3 \end{array} \quad (4)$$

Esta reducción del número de operadores va a influir en el número de módulos hardware, interconexiones, retraso y consumo.

Los fundamentos de este algoritmo de minimización – un algoritmo heurístico basado en la existencia de dos expresiones canónicas para cada función reticular – fueron presentados en una publicación anterior [8].

5 TAREAS DE SAN: PLANIFICACIÓN Y ASIGNACIÓN

Cualquier sistema de SAN debe admitir como entrada una descripción del comportamiento de la aplicación y debe generar como salida un hardware (procesador y controlador formados por unidades aritmético-lógicas, registros, multiplexores e interconexiones) que materialicen dicho comportamiento.

Aunque las técnicas de SAN son aplicables a una gran variedad de algoritmos, son especialmente adecuadas cuando en éstos es más fácilmente explotable el paralelismo implícito, como ocurre en los algoritmos que materializan CBD.

El primer paso en estas tareas consiste en planificar todas las operaciones del algoritmo, no sólo en función de los requisitos del usuario, sino también según el número y el tipo de unidades aritmético-lógicas disponibles (librerías con distintas operaciones, áreas, consumos, rendimientos, etc.). Ante esto, podemos seleccionar materializaciones que varíen desde las que posean un alto grado de paralelismo (alto rendimiento a expensas de un alto consumo de área) hasta las que realizan todo el calculo secuencialmente con una sola unidad aritmético-lógica (mínimo consumo de área a expensas de un bajo rendimiento).

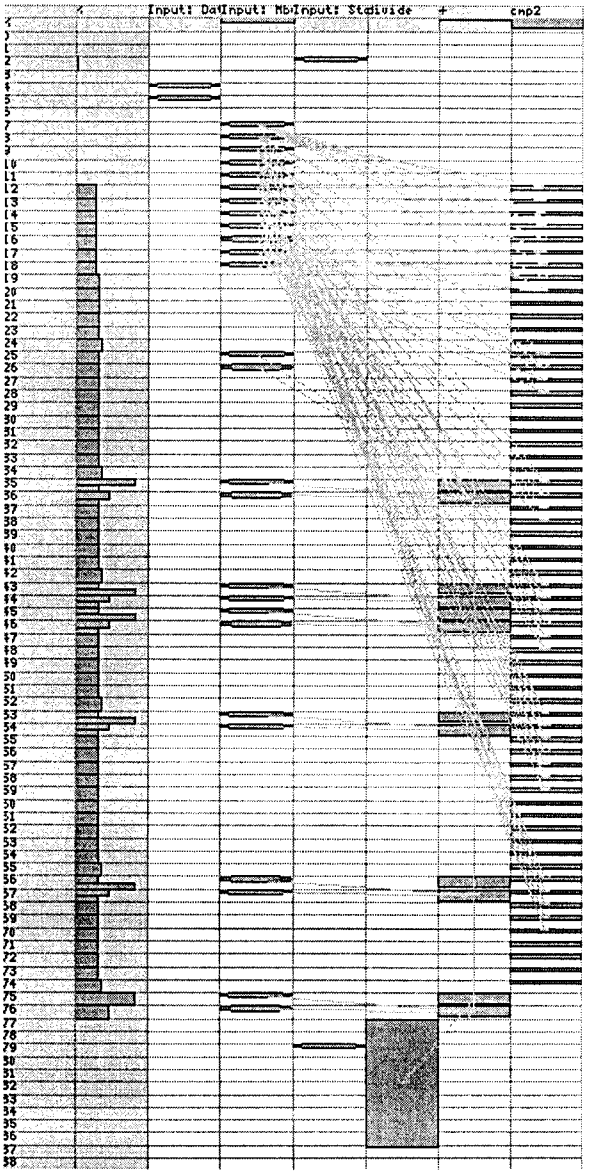


Figura 4: Planificación intermedia (módulo/operador)

En la Figura 4 aparece una planificación intermedia donde se emplea un sólo operador por cada tipo de operación para realizar todas aquellas que lo requieran (un comparador para el cálculo reticular, y un sumador y un divisor para el difuminado y aclarado).
En este paso, el planificador calcula para cada operación cuál es el instante temporal (ciclo de reloj) donde va a ser ejecutada y con qué tipo de unidad aritmético-lógica.
A partir de la planificación escogida como apropiada, el asignador de hardware se encarga de: hacer corresponder a cada operador un módulo hardware concreto y perfectamente definido, de colocar registros para el almacenamiento de las variables intermedias, y de abrir caminos de datos entre módulos con la colocación de multiplexores.

6 MATERIALIZACIÓN DEL CBD

Todo esto concluye con el ensamblado de las distintas partes formando el procesador o ruta de datos que realiza los cómputos (Figura 5). Los datos pueden fluir en el procesador desde los puertos de entrada hacia los registros, desde los registros hacia las unidades aritmético-lógicas a través de los multiplexores, etc.
Finalmente, para poder regir el comportamiento de todos los elementos de la ruta de datos es preciso contar con un dispositivo (controlador) que reciba señales informando del estado del procesador y que genere señales que informen al procesador de cuál es el siguiente paso que debe realizar. En la Figura 5 también se muestra el esquema general de conexión de la ruta de datos y del controlador, junto a las señales de estado y de control (procesador hacia controlador y viceversa, respectivamente).
Como se ha dicho, estos diseños están orientados hacia una materialización específica para una aplicación concreta, de modo que la tecnología objetivo puede ser un ASIC o un dispositivo lógico programable.
Para poder acelerar al máximo el ciclo de diseño, incluyendo la verificación *on-line* de la funcionalidad del circuito, la tecnología objetivo elegida son las FPGAs (en nuestro caso de la casa Xilinx).

7 EXPLORANDO CBD ORIENTADOS A CÓMPUTO

En el ejemplo que se presenta, el universo de discurso tanto de las variables de entrada (E y CE) como de la variable de salida (dD) se representa con 8 bits. Los valores de pertenencia de cada una de las entradas respecto a sus etiquetas lingüísticas se generan con 5 bits.
A diferencia de las materializaciones orientadas a memoria donde el tipo y número de bloques de memoria con que se cuenta, su ancho de banda, etc. limita el rendimiento del CBD, en las materializaciones orientadas a cómputo la complejidad de reconstrucción de las etiquetas lingüísticas de entrada (cómputo de la pertenencia) y de salida (cómputo de la salida) son el cuello de botella del diseño.

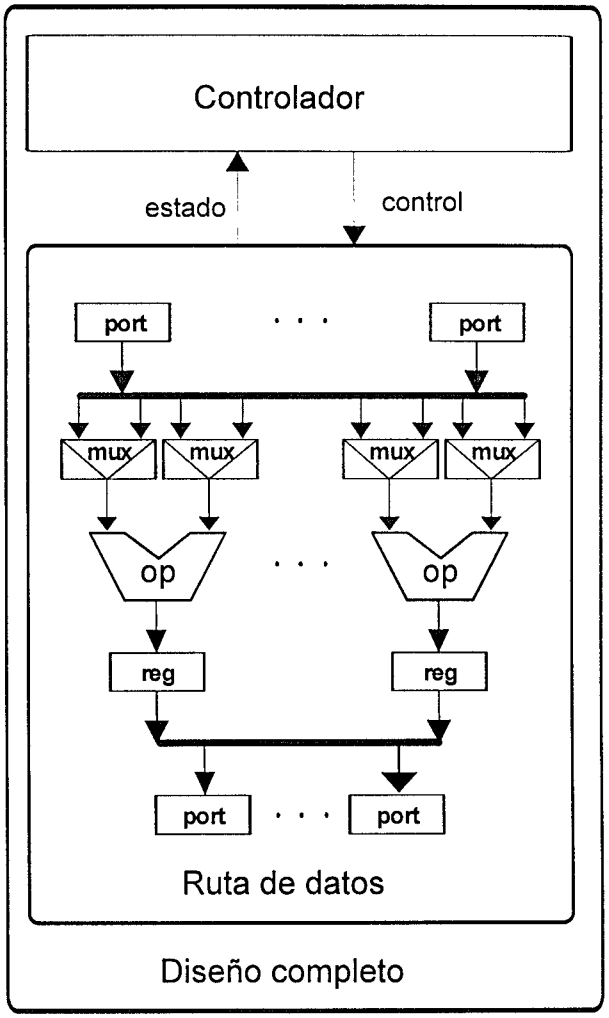


Figura 5: Controlador Borroso Dedicado

En este caso, la sencillez de las etiquetas lingüísticas (todas ellas definidas como triángulos) simplifica notablemente el cómputo de ambos valores, y por tanto reduciendo el tiempo de respuesta del CBD frente a un cambio en sus entradas. En la Figura 6 se muestra como el cómputo del centro de gravedad de una etiqueta lingüística de salida simplemente consiste en acceder al valor de su vértice, al tratarse de triángulos isósceles.
El número de bytes necesario para almacenar los puntos críticos que definen a las etiquetas lingüísticas tanto de entradas como de salida, y por tanto poder reconstruirlas, es significativamente inferior al elevado (y a veces prohibitivo) número de bytes que necesitan los CBD orientados a memoria.

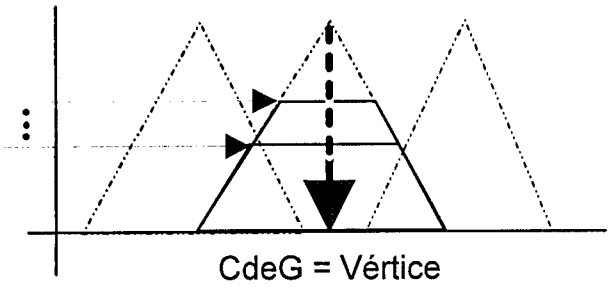


Figura 6: Cálculo del CdeG

Los parámetros más ampliamente utilizados en la literatura se corresponden con el método de difuminado *Singleton* (opuesto a borroso), el método de inferencia Max-min, y el método de aclarado Media ponderada de centros de gravedad.
En la Figura 7 se muestran propuestas de materialización para distintos tiempos de ciclo (desde 20 ns hasta 500 ns) con restricciones que orientan al planificador hacia soluciones que ahorran área. Lógicamente, para un mismo diseño cuanto más rápida sea la materialización mayor será el área consumida.
Todos los ejemplos han sido materializados sobre las celdas estándar de 0.7 micras de ES2.

Este tipo de materializaciones orientadas a cómputo tienen ventajas y desventajas. En cuanto a las primeras podemos citar: el no tener que acceder a una o varias memorias externas mejora el rendimiento del sistema, o que el CBD puede trabajar sin apoyo de otros módulos. En cuanto a las segundas, la poca adaptabilidad a los cambios en la forma o el número de las etiquetas lingüísticas.

CONCLUSIONES

La mayoría de las materializaciones hardware de controladores borrosos dedicados suelen asignar las operaciones del algoritmo a esquemas de cómputo predeterminados que han sido diseñados sacando partido a alguna restricción. Sin embargo, ese esquema con interconexiones de módulos, podría reutilizarse – cuanto menos parcialmente – en otras etapas de la ejecución del control borroso.
Gracias al uso de la metodología de SAN, no estamos restringidos al uso de determinadas arquitecturas, sino que por el contrario se pueden explorar distintas soluciones sólo con los requisitos de SAN sin tener que retocar el algoritmo.

BIBLIOGRAFÍA

[1] Thomas D.E., Armstrong-hélovry B, *Fuzzy logic lontrol - a taxonomy of demonstrated benefits*, Proceedings of the IEEE, vol. 83, nº 3, pp. 407 - 421, 1995.
[2] Ungering A. P., Goser K., *Architecture of a PDM VLSI fuzzy logic controller with pipeline and optimized chip area*, Second IEEE international conference on fuzzy systems, pp. 447 - 452, 1993.

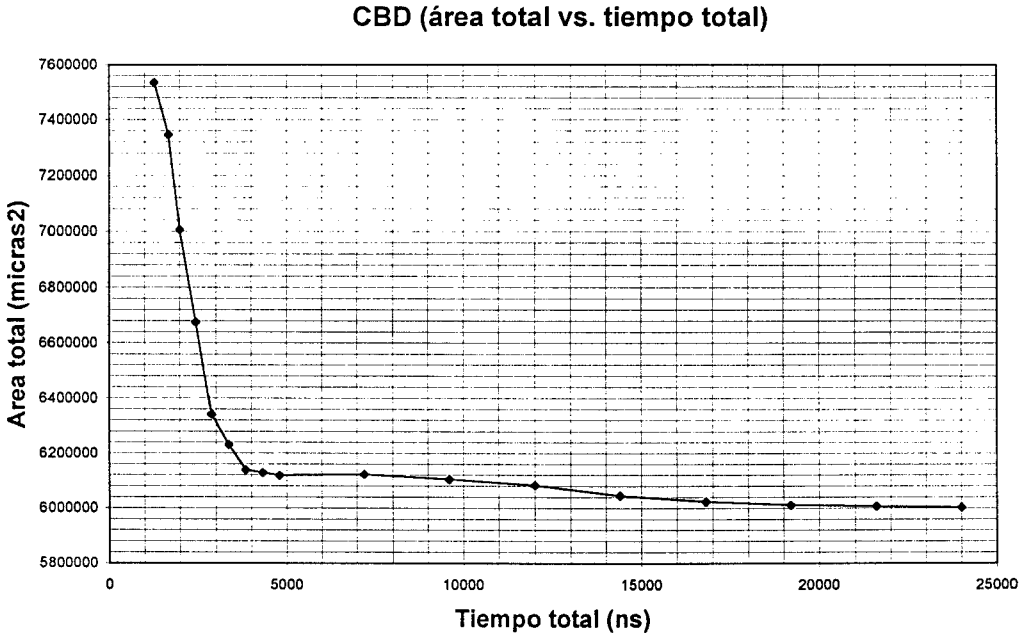


Figura 7: Distintas planificaciones con ahorro de área para distintos ciclos de reloj.

[3] Masetti M., *4 input VLSI fuzzy chip design able to process an input data set every 320 ns*, Second joint conference on information science, 1995.
[4] Martínez Torre J.I., Deschamps, J-P, *Design flow and tools for dedicated fuzzy hardware,* Proceedings of IIZUKA'96, pp 255-258, 1996.
[5] Gomariz S., Guerrero J.A., Guinjoan F., *Seguimiento del punto de máxima potencia de un sistema fotovoltaico mediante control difuso,* Actas de ESTYLF'97, pp 251-256, 1997.
[6] Septien, J., Mozos, D., Tirado, F., Hermida, R., Fernández, M., Mecha, H., *An Integral Approach to High Level Synthesis*, IEE Transactions on Circuits and Systems, in press, 1995.e
[7] Behavioral Compiler, Synopsys inc.
[8] Deschamps, J-P, Martínez Torre J.I., *Optimización de operadores reticulares en un entorno de síntesis automática de controladores borrosos*, Proceedings of ESTYLF'96, pp 107-112, 1996.

ESTABILIDAD DEL MODELO GENERAL DE TAKAGI-SUGENO PARA SISTEMAS CONTINUOS

Fernando Matía, Basil M. Al-Hadithi y Agustín Jiménez
Departamento de Automática, Ingeniería
Electrónica e Informática Industrial
Universidad Politécnica de Madrid
J.Gutiérrez Abascal, 2. E-28006 Madrid, Spain
matia@disam.upm.es

Resumen

El modelo de Takagi-Sugeno es ampliamente utilizado en aplicaciones de control borroso, siendo uno de los temas más difíciles de abordar, en el análisis de sistemas no lineales, el de la estabilidad. Por este motivo, Tanaka-Sugeno presentaron un criterio de estabilidad para sistemas borrosos basados en un caso particular del modelo anterior, pero que prácticamente ningún sistema físico cumple. Este trabajo, analiza este criterio, y lo extiende al caso general.

Palabras Clave: Control Fuzzy, Análisis Dinámico, Estabilidad.

1 INTRODUCCIÓN

El modelo de Takagi-Sugeno sigue siendo uno de los más utilizados en control borroso [5]. Existen muchos trabajos en la actualidad que abordan el problema de la estabilidad de los sistemas borrosos, como [1], [2], [4], [6], [7], [8] y [9].

Uno de los más destacados es el presentado por Tanaka y Sugeno [8], en el cual se analizan las condiciones bajo las cuales un sistema borroso discreto es asintóticamente estable. Para el análisis, se supone que el consecuente lineal de cada regla ha sido obtenido linealizando el sistema original no lineal, en torno al origen. Kosko presenta la versión continua de este criterio de estabilidad [2], mientras que Tanaka continúa todos sus desarrollos posteriores sobre control robusto sobre el modelo anterior [6].

En este trabajo, se muestra cómo en el modelo anterior es incorrecto linealizar en torno al origen, y se presenta una variante del criterio de estabilidad para el caso general en el cual la linealización de cada subsistema se hace en torno a un punto diferente. Finalmente se muestran ejemplos de aplicación.

2 EL MODELO DE TAKAGI-SUGENO

El modelo propuesto por Takagi y Sugeno consiste en un conjunto de reglas en las que el consecuente es una función lineal de las entradas. Su principal característica es que permite modelar con mucha precisión la dinámica del sistema original alrededor de los puntos de linelización. Y entre cada dos de estos puntos, la salida del sistema es un promedio de los consecuentes de cada regla, con lo que la dinámica también se aproxima mejor que linealizando sólo en un punto como ocurre con los modelos lineales. Cada regla $R^{(i_1 \cdots i_n)}$ es de la forma:

$$\textit{Si } x \textit{ es } M_1^{i_1} \textit{ y } \dot{x} \textit{ es } M_2^{i_2} \ldots \textit{ y } x^{(n-1)} \textit{ es } M_n^{i_n}$$
$$\textit{entonces } \dot{\mathbf{x}} = \mathbf{a}_o^{(i_1 \ldots i_n)} + \mathbf{A}^{(i_1 \ldots i_n)}\mathbf{x} + \mathbf{B}^{(i_1 \ldots i_n)}\mathbf{u} \quad (1)$$

donde $M_1^{i_1}$ $(i_1 = 1, 2, \ldots, r_1)$ son los conjuntos borrosos para x, $M_2^{i_2}$ $(i_2 = 1, 2, \ldots, r_2)$ los conjuntos borrosos para $\dot{x}$ y $M_n^{i_n}$ $(i_n = 1, 2, \ldots, r_n)$ los conjuntos borrosos para x^{n-1} (figura 1).

$\mathbf{x}$ es el vector de estado y $\mathbf{u}$ es el vector de entrada, con lo que el sistema se compone de $r_1 r_2 \ldots r_n$ reglas. Suponiendo que la matriz $\mathbf{A}^{(i_1 \ldots i_n)}$ corresponde a la forma canónica controlable en la cual $x = f(\dot{x}, \ldots, x^{(n)})$, ésta vendrá dada por:

$$\begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \ldots & 0 & 1 \\ -a_1^{(i_1 \ldots i_n)} & -a_2^{(i_1 \ldots i_n)} & \ldots & -a_{n-1}^{(i_1 \ldots i_n)} & -a_n^{(i_1 \ldots i_n)} \end{bmatrix} \quad (2)$$

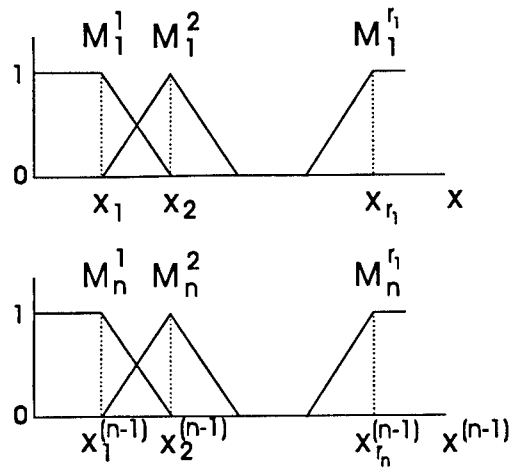


Figura 1: Funciones de Pertenencia Genéricas

mientras que

$$\mathbf{a}_0^{(i_1 \ldots i_n)} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ a_0^{(i_1 \ldots i_n)} \end{bmatrix}, \tag{3}$$

con

$$\mathbf{x}^T = [\ x \quad \dot{x} \quad \ldots \quad x^{n-1}\ ] \tag{4}$$

Si la entrada es nula, la salida final del sistema borroso se obtiene como:

$$\begin{aligned} \dot{\mathbf{x}} &= \frac{\sum_{i_1=1}^{r_1} \cdots \sum_{i_n=1}^{r_n} w^{(i_1 \ldots i_n)} (\mathbf{a}_o^{(i_1 \ldots i_n)} + \mathbf{A}^{(i_1 \ldots i_n)} \mathbf{x})}{\sum_{i_1=1}^{r_1} \cdots \sum_{i_n=1}^{r_n} w^{(i_1 \ldots i_n)}(\mathbf{x})} \\ &= \mathbf{a}_0(\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{x} \end{aligned} \tag{5}$$

siendo

$$\mathbf{A}(\mathbf{x}) = \frac{\sum_{i_1=1}^{r_1} \cdots \sum_{i_n=1}^{r_n} w^{(i_1 \ldots i_n)}(\mathbf{x}) \mathbf{A}^{(i_1 \ldots i_n)}(\mathbf{x})}{\sum_{i_1=1}^{r_1} \cdots \sum_{i_n=1}^{r_n} w^{(i_1 \ldots i_n)}(\mathbf{x})} \tag{6}$$

$$\mathbf{a}_0(\mathbf{x}) = \frac{\sum_{i_1=1}^{r_1} \cdots \sum_{i_n=1}^{r_n} w^{(i_1 \ldots i_n)}(\mathbf{x}) \mathbf{a}_0^{(i_1 \ldots i_n)}(\mathbf{x})}{\sum_{i_1=1}^{r_1} \cdots \sum_{i_n=1}^{r_n} w^{(i_1 \ldots i_n)}(\mathbf{x})} \tag{7}$$

Cada ecuación del consecuente es un subsistema lineal que viene dado por la expresión $\mathbf{a}_o^{(i_1 \ldots i_n)} + \mathbf{A}^{(i_1 \ldots i_n)}\mathbf{x}$.

Teorema 1 *Si* $\mathbf{a}_0^{(i_1 \ldots i_n)} = 0, \quad \forall\ i_1 \ldots, i_n,$ *es decir,* $\dot{\mathbf{x}} = \mathbf{A}^{(i_1 \ldots i_n)}\mathbf{x}$, *entonces, el sistema borroso definido por la ecuación (5) es asintóticamente estable si existe una matriz* $\mathbf{P}$ *definida positiva, tal que:*

$$(\mathbf{A}^{(i_1 \ldots i_n)})^T \mathbf{P} + \mathbf{P}\mathbf{A}^{(i_1 \ldots i_n)} < 0 \tag{8}$$

$\forall\ i_1, \ldots, i_n$, *es decir, para todos los subsistemas.*

Véanse [2] y [6] para demostración. Se observa fácilmente además que el Teorema anterior queda reducido al Teorema de estabilidad por Lyapunov para sistemas lineales, cuando $r_1, \ldots, r_n = 1$.

También está tratada ampliamente en la literatura la estabilidad de sistemas borrosos discretos [2], [6], siendo el trabajo inicial el de Tanaka-Sugeno [8]. El caso discreto, más complejo que el continuo, conduce a resultados sorprendentes, como el hecho de que el sistema borroso puede no ser estable, aunque todos los subsistemas lineales lo sean. Esto no ocurre en el caso continuo, tal y como se desprende del Teorema anterior.

3 TEOREMA DE ESTABILIDAD GENERAL

Al suponer en el Teorema anterior que $\mathbf{a}_0^{(i_1 \ldots i_n)} = 0$, se está limitando su marco de aplicación casi a cero. Un sistema no lineal, linealizado en dos puntos diferentes (dos reglas) da lugar a dos sistemas lineales que, en general no pasan por el origen $\mathbf{x} = \mathbf{0}$, tal y como puede apreciarse en la figura 2.

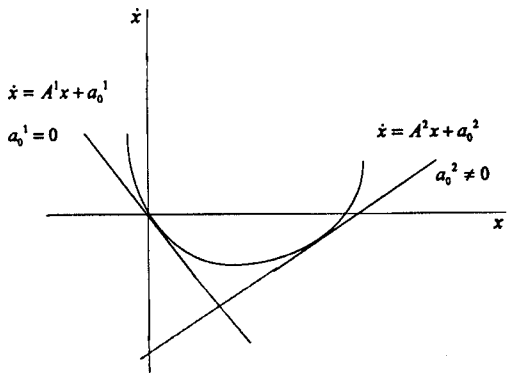


Figura 2: General Takagi-Sugeno Model

Si sólo se trabajase con una regla (sistema puramente lineal, no borroso), se podrían tomar variables incrementales en torno al punto de linealización, con lo cual desaparecería el término independiente a_0. Sin embrago, esto no puede hacerse en un sistema borroso con varios subsistemas lineales, pues las variables incrementales resultantes serían también diferentes (al serlo los puntos de linealización), con lo que no se podrían promediar posteriormente, para obtener la salida final del sistema.

Se presenta a continuación un Teorema para el caso general en el que $\mathbf{a}_0^{(i_1 \ldots i_n)} \neq 0$. En otras palabras, se

analiza la estabilidad del modelo general de Takagi-Sugeno.

Teorema 2 *El sistema borroso representado por:*

$$\mathbf{B}(\mathbf{x})\dot{\mathbf{x}} + \mathbf{x} = \mathbf{b}_0(\mathbf{x}) \tag{9}$$

donde

$$\mathbf{B}^{(i_1 \ldots i_n)} = -\left(\mathbf{A}^{(i_1 \ldots i_n)}\right)^{-1} \tag{10}$$

$$\mathbf{b}_0^{(i_1 \ldots i_n)} = \mathbf{B}^{(i_1 \ldots i_n)} \mathbf{a}_0^{(i_1 \ldots i_n)} \tag{11}$$

y en el que todos los subsistemas lineales son estables, es asintóticamente estable si:

$$\left[I - \frac{d\mathbf{b}_0(\mathbf{x})}{d\mathbf{x}}\right] \mathbf{B}^{-1}(\mathbf{x}) > 0, \quad \forall \mathbf{x} \neq 0 \tag{12}$$

Demostración. En primer lugar, al corresponder la matriz $\mathbf{A}^{i_1 \ldots i_n}$ a la forma canónica controlable $x = f(\dot{x}, \ldots, x^{(n)})$, y ser ésta estable, resulta sencillo probar que existe la inversa $\mathbf{B}^{i_1 \ldots i_n}$ y, por tanto, $\mathbf{B}(\mathbf{x})$.

Tomando como función de Lyapunov,

$$V(\mathbf{x}) = [\mathbf{x} - \mathbf{b}_0(\mathbf{x})]^T [\mathbf{x} - \mathbf{b}_0(\mathbf{x})] \tag{13}$$

ésta verifica que:

$$V(\mathbf{0}) = 0$$

$$V(\mathbf{x}) > 0, \ \forall \mathbf{x} \neq \mathbf{0}$$

y debe satisfacer que:

$$\dot{V}(\mathbf{0}) = 0$$

$$\dot{V}(\mathbf{x}) < 0, \ \forall \mathbf{x} \neq \mathbf{0}$$

Derivando $V(\mathbf{x})$:

$$\begin{aligned} \dot{V}(\mathbf{x}) &= 2[\mathbf{x} - \mathbf{b}_0(\mathbf{x})]^T \left[\dot{\mathbf{x}} - \frac{d\mathbf{b}_0}{d\mathbf{x}}\dot{\mathbf{x}}\right] = \\ &= -2[\mathbf{x} - \mathbf{b}_0(\mathbf{x})]^T \left[I - \frac{d\mathbf{b}_0}{d\mathbf{x}}\right] \\ &\quad \mathbf{B}^{-1}(\mathbf{x}) [\mathbf{x} - \mathbf{b}_0(\mathbf{x})] < 0, \quad \forall \mathbf{x} \neq 0 \end{aligned} \tag{14}$$

Y esto es cierto si $\left[I - \frac{d\mathbf{b}_0(\mathbf{x})}{d\mathbf{x}}\right] \mathbf{B}^{-1}(\mathbf{x}) > 0, \ \forall \mathbf{x} \neq 0.$

□.

Otro Teorema más concreto es el siguiente:

Teorema 3 *El sistema borroso representado por el modelo general de Takagi-Sugeno (9), es asintóticamente estable si* $\forall x_{i_l}^{(l-1)} \leq x^{(l-1)} \leq x_{i_l+1}^{(l-1)}$, $\forall i_l = \{1, \ldots, r_l - 1\}$, $\forall l = \{1, \ldots n\}$, *los siguientes polinomios corresponden a sistemas lineales estables:*

$$\begin{aligned} &\left(1 - \frac{b_0^{i_1+1 j_2 \ldots j_n} - b_0^{i_1 j_2 \ldots j_n}}{x_{i_1+1} - x_{i_1}}\right) \lambda^n + \\ &+ \left(b_1^{j_1 \ldots j_n} - \frac{b_0^{j_1 i_2+1 j_3 \ldots j_n} - b_0^{j_1 i_2 j_3 \ldots j_n}}{\dot{x}_{i_2+1} - \dot{x}_{i_2}}\right) \lambda^{n-1} + \\ &+ \ldots + \\ &+ \left(b_{n-1}^{j_1 \ldots j_n} - \frac{b_0^{j_1 \ldots j_{n-1} i_n+1} - b_0^{j_1 \ldots j_{n-1} i_n}}{x_{i_n+1}^{(n-1)} - x_{i_n}^{(n-1)}}\right) \lambda + \\ &+ b_n^{j_1 \ldots j_n} = 0 \qquad \forall j_l = \{i_l, i_l + 1\} \end{aligned} \tag{15}$$

Demostración. Se partirá para la demostración del Teorema 2. $\left[I - \frac{d\mathbf{b}_0(\mathbf{x})}{d\mathbf{x}}\right] \mathbf{B}^{-1}(\mathbf{x})$ es el producto de las matrices:

$$\begin{bmatrix} 1 - \frac{\partial b_0}{\partial x} & -\frac{\partial b_0}{\partial \dot{x}} & \ldots & \ldots & -\frac{\partial b_0}{\partial x^{(n-1)}} \\ 0 & 1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \ldots & \ldots & 0 & 1 \end{bmatrix} \tag{16}$$

y

$$\begin{bmatrix} 0 & -1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \ldots & \ldots & 0 & -1 \\ \frac{1}{b_n(\mathbf{x})} & \frac{b_1(\mathbf{x})}{b_n(\mathbf{x})} & \ldots & \ldots & \frac{b_{n-1}(\mathbf{x})}{b_n(\mathbf{x})} \end{bmatrix} \tag{17}$$

y, por lo tanto, $\left[I - \frac{d\mathbf{b}_0(\mathbf{x})}{d\mathbf{x}}\right] \mathbf{B}^{-1}(\mathbf{x}) = E(\mathbf{x})$ se puede expresar como:

$$\begin{bmatrix} e_1(\mathbf{x}) & e_2(\mathbf{x}) & \ldots & \ldots & \ldots & e_n(\mathbf{x}) \\ 0 & 0 & -1 & 0 & \ldots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & 0 \\ 0 & \ldots & \ldots & \ldots & 0 & -1 \\ \frac{1}{b_n(\mathbf{x})} & \frac{b_1(\mathbf{x})}{b_n(\mathbf{x})} & \ldots & \ldots & \ldots & \frac{b_{n-1}(\mathbf{x})}{b_n(\mathbf{x})} \end{bmatrix} \tag{18}$$

con

$$\begin{cases} e_1(\mathbf{x}) &= \frac{-1}{b_n(\mathbf{x})}\frac{\partial b_0(\mathbf{x})}{\partial x^{(n-1)}} \\ e_2(\mathbf{x}) &= (\frac{\partial b_0(\mathbf{x})}{\partial x} - 1) - \frac{b_1(\mathbf{x})}{b_n(\mathbf{x})}\frac{\partial b_0(\mathbf{x})}{\partial x^{(n-1)}} \\ \cdots & \\ e_n(\mathbf{x}) &= \frac{\partial b_0(\mathbf{x})}{\partial x^{(n-2)}} - \frac{b_{n-1}(\mathbf{x})}{b_n(\mathbf{x})}\frac{\partial b_0(\mathbf{x})}{\partial x^{(n-1)}} \end{cases} \quad (19)$$

Para que la matriz $E(\mathbf{x})$ sea definida positiva, la parte real de las raíces de $|\lambda I - (-E^{-1}(\mathbf{x}))| = 0$, que son las mismas que las de $|\lambda E(\mathbf{x}) + \mathbf{I}| = 0$, deben ser estrictamente negativas $\forall \mathbf{x}$. El determinante $|\lambda E(\mathbf{x}) + \mathbf{I}|$ se puede calcular como:

$$\begin{vmatrix} 1+\lambda e_1(\mathbf{x}) & \lambda e_2(\mathbf{x}) & \dots & \dots & \dots & \lambda e_n(\mathbf{x}) \\ 0 & 1 & -\lambda & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & 0 \\ 0 & \dots & \dots & \dots & 1 & -\lambda \\ \frac{\lambda}{b_n(\mathbf{x})} & \lambda\frac{b_1(\mathbf{x})}{b_n(\mathbf{x})} & \dots & \dots & \dots & 1+\lambda\frac{b_{n-1}(\mathbf{x})}{b_n(\mathbf{x})} \end{vmatrix} \quad (20)$$

que es igual a

$$\begin{aligned} &\left(1 - \tfrac{\partial b_0}{\partial x}\right)\lambda^n + \left(b_1(\mathbf{x}) - \tfrac{\partial b_0}{\partial x}\right)\lambda^{n-1} + \ldots + \\ + \;&\left(b_{n-1}(\mathbf{x}) - \tfrac{\partial b_0}{\partial x^{(n-1)}}\right)\lambda + b_n(\mathbf{x}) \end{aligned} \quad (21)$$

polinomio que debe tener raíces con parte real negativa.

Ahora bien, dado que las funciones de pertenencia se han supuesto de forma triangular y verifican que $\sum_{i_l=1}^{r_l} \mu_{M_l^{i_l}}(x^{(l-1)}) = 1$, en [3] se demuestra que:

$$\begin{aligned} &\sum_{i_1=1}^{r_1} \cdots \sum_{i_n=1}^{r_n} w^{i_1 \ldots i_n}(\mathbf{x}) = \\ = \;&\sum_{i_1=1}^{r_1} \cdots \sum_{i_n=1}^{r_n} \prod_{l=1}^{n} \mu_{M_l^{i_l}}(x^{(l-1)}) = 1 \end{aligned} \quad (22)$$

por lo que

$$\mathbf{b_0}(\mathbf{x}) = \sum_{i_1=1}^{r_1} \cdots \sum_{i_n=1}^{r_n} \prod_{l=1}^{n} \mu_{M_l^{i_l}}(x^{(l-1)}) \mathbf{b_0}^{i_1 \ldots i_n} \quad (23)$$

o incluso:

$$b_0(\mathbf{x}) = \sum_{i_1=1}^{r_1} \cdots \sum_{i_n=1}^{r_n} \prod_{l=1}^{n} \mu_{M_l^{i_l}}(x^{(l-1)}) b_0^{i_1 \ldots i_n} \quad (24)$$

y, sabiendo que las funciones de pertenencia se superponen por parejas:

$$b_0(\mathbf{x}) = \sum_{j_1=i_1, i_1+1} \cdots \sum_{j_n=i_n, i_n+1} \prod_{l=1}^{n} \mu_{M_l^{j_l}}(x^{(l-1)}) b_0^{j_1 \ldots j_n} \quad (25)$$

$$\forall x_{i_l}^{(l-1)} \leq x^{(l-1)} \leq x_{i_l+1}^{(l-1)}$$
$$\forall i_l = \{1, \ldots, r_1\}\,, \quad \forall l = \{1, \ldots, n\}$$

Finalmente, derivando la expresión anterior, se puede calcular $\frac{\partial b_0(\mathbf{x})}{\partial x^{(m-1)}}$ como:

$$\begin{aligned} &\sum_{j_1=i_1, i_1+1} \cdots \sum_{j_n=i_n, i_n+1} \\ &\frac{\partial \mu_{M_m^{j_m}}(x^{(m-1)})}{\partial x^{(m-1)}} \prod_{l=1}^{n} \mu_{M_l^{j_l}}(x^{(l-1)}) b_0^{j_1 \ldots j_n} = \\ = \;&\sum_{j_1=i_1, i_1+1} \cdots \sum_{j_{m-1}=i_{m-1}, i_{m-1}+1} \sum_{j_{m+1}=i_{m+1}, i_{m+1}+1} \\ &\cdots \sum_{j_n=i_n, i_n+1} \left[\frac{\partial \mu_{M_m^{i_m}}(x^{(m-1)}) b_0^{j_1 \ldots j_{m-1} i_m j_m \ldots j_n}}{\partial x^{(m-1)}} \right. \\ &\left. + \frac{\partial \mu_{M_m^{i_m+1}}(x^{(m-1)}) b_0^{j_1 \ldots j_{m-1} i_m+1 j_{m+1} \ldots j_n}}{\partial x^{(m-1)}} \right] \\ &\prod_{l=1}^{n} \mu_{M_l^{j_l}}(x^{(l-1)}) = \\ = \;&\sum_{j_1=i_1, i_1+1} \cdots \sum_{j_{m-1}=i_{m-1}, i_{m-1}+1} \\ &\sum_{j_{m+1}=i_{m+1}, i_{m+1}+1} \cdots \sum_{j_n=i_n, i_n+1} \\ &\frac{b_0^{j_1 \ldots j_{m-1} i_m+1 j_{m+1} \ldots j_n} - b_0^{j_1 \ldots j_{m-1} i_m j_{m+1} \ldots j_n}}{x_{i_m+1}^{(m-1)} - x_{i_m}^{(m-1)}} \\ &\prod_{l=1}^{n} \mu_{M_l^{j_l}}(x^{(l-1)}) \end{aligned} \quad (26)$$

Tomando en consideración (25) y (26), la ecuación (21) resulta $\forall x_{i_l}^{(l-1)} \leq x^{(l-1)} \leq x_{i_l+1}^{(l-1)}$, $\forall i_l = \{1, \ldots, r_1\}$, $\forall l = \{1, \ldots, n\}$

$$\begin{aligned} &\sum_{j_1=i_1}^{i_1+1} \cdots \sum_{j_n=i_n}^{i_n+1} \prod_{l=1}^{n} \mu_{M_l^{j_l}}(x^{(l-1)}) \\ &\left[\left(1 - \frac{b_0^{i_1+1 j_2 \ldots j_n} - b_0^{i_1 j_2 \ldots j_n}}{x_{i_1+1} - x_{i_1}} \right) \lambda^n + \right. \end{aligned}$$

$$+ \left(b_1^{j_1 \cdots j_n} - \frac{b_0^{j_1 i_2+1 j_3 \cdots j_n} - b_0^{j_1 i_2 j_3 \cdots j_n}}{\dot{x}_{i_2+1} - \dot{x}_{i_2}} \right) \lambda^{n-1} + \ldots$$
$$+ \left(b_{n-1}^{j_1 \cdots j_n} - \frac{b_0^{j_1 \cdots j_{n-1} i_n+1} - b_0^{j_1 \cdots j_{n-1} i_n}}{x_{i_n+1}^{(n-1)} - x_{i_n}^{(n-1)}} \right) \lambda$$
$$+ \quad b_n^{j_1 \cdots j_n}] = 0 \qquad (27)$$

$\forall j_l = \{i_l, i_l + 1\}$.

De manera que se debe verificar si la parte real de las raíces de

$$\left(1 - \frac{b_0^{i_1+1 j_2 \cdots j_n} - b_0^{i_1 j_2 \cdots j_n}}{x_{i_1+1} - x_{i_1}} \right) \lambda^n +$$
$$\left(b_1^{j_1 \cdots j_n} - \frac{b_0^{j_1 i_2+1 j_3 \cdots j_n} - b_0^{j_1 i_2 j_3 \cdots j_n}}{\dot{x}_{i_2+1} - \dot{x}_{i_2}} \right) \lambda^{n-1} + \ldots$$
$$\ldots + \left(b_{n-1}^{j_1 \cdots j_n} - \frac{b_0^{j_1 \cdots j_{n-1} i_n+1} - b_0^{j_1 \cdots j_{n-1} i_n}}{x_{i_n+1}^{(n-1)} - x_{i_n}^{(n-1)}} \right) \lambda$$
$$+ b_n^{j_1 \cdots j_n} = 0 \qquad (28)$$

es estrictamente negativa. Obsérvese finalmente que si $x^{(l-1)} \notin \left[x_1^{(l-1)}, x_{r_l}^{(l-1)} \right]$ para algún $l = \{1, \ldots, n\}$ (es decir, que x se encuentra fuera del universo de discurso), el sistema es estable, siempre y cuando todos los subsistemas $\mathbf{B}^{j_1 \cdots j_{l-1} 1 j_{l+1} \cdots j_n}$ y $\mathbf{B}^{j_1 \cdots j_{l-1} r_l j_{l+1} \cdots j_n}$ lo sean.

□.

4 EJEMPLO

Sea el sistema borroso definido por:

$$\begin{aligned} R^1 &: Si\ (x\ es\ M_1^1)\ entonces\ \dot{x} + x = -2 \\ R^2 &: Si\ (x\ es\ M_1^2)\ entonces\ \dot{x} + x = 2 \end{aligned} \qquad (29)$$

Es fácil de identificar que $\mathbf{B}^1 = \mathbf{B}^2 = 1$ y que $\mathbf{b_0}^1 = -2$, $\mathbf{b_0}^2 = 2$. La figura 3 muestra las funciones de pertenecia para las premisas de las reglas.

Los parámetros del sistema borroso pueden obtenerse como sigue:

$$\begin{aligned} \mathbf{B}(\mathbf{x}) &= \frac{\sum_{i_1=1}^{2} w^{i_1}(\mathbf{x}) \mathbf{B}^{i_1}(\mathbf{x})}{\sum_{i_1=1}^{2} w^{i_1}(\mathbf{x})} \\ &= \frac{0.5(1-x)(1) + 0.5(1+x)(1)}{0.5(1-x) + 0.5(1+x)} \\ &= 1 \end{aligned}$$

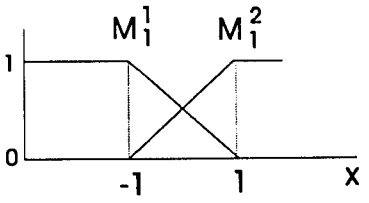


Figura 3: Funciones de Pertenencia

$$\begin{aligned} \mathbf{b_0}(\mathbf{x}) &= \frac{\sum_{i_1=1}^{2} w^{i_1}(\mathbf{x}) \mathbf{b_0}^{i_1}(\mathbf{x})}{\sum_{i_1=1}^{2} w^{i_1}(\mathbf{x})} \\ &= \frac{0.5(1-x)(-2) + 0.5(1+x)(2)}{0.5(1-x) + 0.5(1+x)} \\ &= 2x \end{aligned} \qquad (30)$$

Nótese que $\mathbf{b_0}(0) = 0$.

Los dos subsistemas lineales son estables, dado que $\mathbf{B_1}^1$ y $\mathbf{B_1}^2$ son definidas positivas. De acuerdo con los resultados obtenidos en [2] y [6] (Teorema 1), el sistema borroso debería ser asintóticamente estable, puesto que existe una matriz $\mathbf{P}$ definida positiva, tal que $\forall i_1$, $(\mathbf{A}^{i_1})^T \mathbf{P} + \mathbf{P}\mathbf{A}^{i_1} < 0$. Si se elige $\mathbf{P} = \lambda = 1$,

$$\begin{aligned} (\mathbf{A}^{i_1})^T \mathbf{P} + \mathbf{P}\mathbf{A}^{i_1} &= \left[(\mathbf{B}^{i_1})^{-1} \right]^T \mathbf{P} + \mathbf{P} (\mathbf{B}^{i_1})^{-1} \\ &= \frac{1}{(-1)}(1) + \frac{1}{(1)}(-1) \\ &= -1 < 0 \end{aligned} \qquad (31)$$

Sin embargo, el sistema borroso resultante es inestable. De hecho, se puede verificar que no puede garantizarse la estabilidad asintótica del sistema, aplicando el Teorema 2:

$$\left[I - \frac{d\mathbf{b_0}(\mathbf{x})}{d\mathbf{x}} \right] \mathbf{B}^{-1}(\mathbf{x}) = (1-2)\frac{1}{1} = -1 \qquad (32)$$

que no es estrictamente positivo $\forall x \neq 0$.

Sin embargo, para aplicar este Teorema, es preciso obtener una expresión explícita para $\mathbf{B}(\mathbf{x})$. Visto así, el sistema borroso podría reescribirse como:

$$\mathbf{B}(\mathbf{x})\dot{\mathbf{x}} + \mathbf{x} = \mathbf{b_0}(\mathbf{x}) \Leftrightarrow \dot{x} + x = 2x \Leftrightarrow \dot{x} - x = 0 \qquad (33)$$

que es claramente un sistema inestable.

Resulta de mayor utilidad el Teorema 3. Debe verificarse que:

$\forall x_{i_1} \leq x \leq x_{i_1+1}$, $\forall i_1 = \{1, \ldots, r_1 - 1\}$, los siguientes polinomios correspondan a sistemas lineales estables:

$$\left(1 - \frac{b_0^{i_1+1} - b_0^{i_1}}{x_{i_1+1} - x_{i_1}}\right)\lambda + b_1^{j_1} = 0 \qquad \forall j_1 = \{i_1, i_1 + 1\} \tag{34}$$

Es decir, debe examinarse la estabilidad de los polinomios:

$$\left(1 - \frac{b_0^2 - b_0^1}{x_2 - x_1}\right)\lambda + b_1^1 = 0 \tag{35}$$

$$\left(1 - \frac{b_0^2 - b_0^1}{x_2 - x_1}\right)\lambda + b_1^2 = 0 \tag{36}$$

que resultan ser:

$$-\lambda + 1 = 0 \tag{37}$$

$$-\lambda + 1 = 0 \tag{38}$$

Ambos son inestables, luego no puede concluirse la estabilidad asintótica del sistema borroso.

5 CONCLUSIÓN

En este trabajo se han presentado pasos adicionales para complementar el análisis de estabilidad de sistemas borrosos continuos basados en el modelo de Takagi-Sugeno. La solución aquí presentada corresponde a un modelo realista, en el que existe término independiente en el consecuente de las reglas. Además, la aplicación de los Teoremas presentados resulta de fácil aplicación, tal y como ha quedado probado en los ejemplos.

Agradecimientos

Queremos agradecer la financiación recibida de la CICYT, proyecto TAP96-0600 EVS (Plataforma Virtual para Ingeniería de Sistemas Autónomos Distribuídos), así como del programa TMR (Training and Mobility of Researchers), proyecto FMRX CT96-0070 MobiNet (Mobile Robotics Technology for Health Care Services).

Referencias

[1] J. Concha y A. Cipriano (1997). A new design method of fuzzy controllers based on stability analysis. En *Seventh IFSA World Congress*, Pág. 312–317.

[2] B. Kosko (1997). *Fuzzy Engineering.* Prentice Hall.

[3] F. Matía y A. Jiménez (1996). On optimal implementation of fuzzy controllers. *Intenational Journal of Intelligent Control and Systems*, 1(3) Pág. 407–415.

[4] F. Sheikholeslam y V. Tahani (1997). Stability analysis of fuzzy systems and design of fuzzy controllers for linear plants. En *Seventh IFSA World Congress*, Pág. 404–409.

[5] T. Takagi y M. Sugeno (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-15(1) Pág. 116–132.

[6] K. Tanaka, T. Ikeda y H. O. Wang (1996). Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: Quadratic stabilizability, H_∞ control theory, and linear matrix inequalities. *IEEE Transactions on Fuzzy Systems*, 4(1) Pág. 1–13.

[7] K. Tanaka y M. Sano (1994). A robust stabilization problem of fuzzy control systems and its application to backing up control of a truck-trailer. *IEEE Transactions on Fuzzy Systems*, 2(2) Pág. 119–134.

[8] K. Tanaka y M. Sugeno (1992). Stability analysis and design of fuzzy control systems. *Fuzzy Sets and Systems*, 45 Pág. 135–156.

[9] H. O. Wang, K. Tanaka, and M. F. Griffin (1996). An approach to fuzzy control of nonlinear systems: Stability and design issues. *IEEE Transactions on Fuzzy Systems*, 4(1) Pág. 14–23.

Validación formal de métodos de control borroso. Perspectivas.

Antonio Sala, Pedro Albertos
Dept. de Ing. de Sistemas y Automática.
Universidad Politécnica de Valencia
Apdo. 22012, E-46071 Valencia
asala@isa.upv.es, pedro@aii.upv.es

Resumen

En esta ponencia se discuten el estado actual y las perspectivas de las dos líneas de investigación principales en control borroso: por un lado, la línea interpolativa-funcional, que estudia los sistemas borrosos como aproximadores funcionales universales y aplica resultados de control no-lineal, redes neuronales, etc. y, por otro lado, la línea formal que explora la validación lógica, el tratamiento de la incertidumbre y la supervisión del conocimiento. Se citan referencias en ambos aspectos y resultados resumidos en el segundo de ellos.

Palabras Clave: Control borroso, lógica borrosa, validación, control experto.

1 Introducción.

La mejora de calidad de producción de un gran número de procesos industriales pasa en gran medida por conseguir un control eficiente de los mismos. En un entorno industrial complejo, la aplicación práctica de técnicas de control inteligente basadas en redes neuronales con capacidades de aprendizaje [11] está en un estadio inicial. Sin embargo, los usuarios aceptan con facilidad y gran interés las aplicaciones basadas en lógica borrosa, por el paralelismo con su propio razonamiento.

Una de las principales metas en control inteligente de procesos industriales es la construcción de sistemas borrosos que controlen con garantía sistemas complejos de alta dimensionalidad, mediante implementaciones generales, robustas y fácilmente entendibles por el usuario. La capacidad de explicación de las conclusiones por parte de un sistema basado en el conocimiento es una ventaja de cara a la interacción con el usuario respecto a herramientas como las redes neuronales.

No obstante, pese a la gran fama y aceptación que los sistemas borrosos están ganando, muchas de las aplicaciones del diseño teórico actual están pensadas para procesos de baja dimensionalidad, o en procesos multivariables poco acoplados. El control borroso, originado a partir de una lógica de conceptos "vagos" e "imprecisos", se utiliza en la mayoría de los casos para aproximación de funciones precisas, deterministas, contradiciendo con ello parte de los pretextos argüidos para fundamentar la utilidad de la lógica borrosa en control.

Asimismo, un problema que las estructuras actualmente utilizadas plantean es la dificultad de aprendizaje y validación cuando se aplican a sistemas que requieren una estrategia compleja. La validación se lleva a cabo experimentalmente en la mayor parte de los casos realizando numerosas simulaciones y, en la práctica industrial, la modificación de las bases de reglas en los sistemas expertos aplicados a control se suele realizar de forma manual y tediosa, por ensayo y error. Un ejemplo de ello es la aplicación industrial a hornos de cemento [10].

La integración de técnicas borrosas y neuronales da lugar a estructuras denominadas neuroborrosas que, en bastantes casos, combinan la interpretación lingüística con una estructura regresiva lineal en los parámetros que permite ciertos resultados teóricos sobre el aprendizaje [20, 8, 2]. El conceptos base del enfoque es la *granularidad* o *localidad.* El aprendizaje realiza dos tipos de modificaciones:

- *Modificaciones paramétricas*: modificaciones en ganancias u otros parámetros, para mejorar gradualmente las acciones de control (adaptación).
- *Modificaciones estructurales*: Adición y eliminación de reglas, reconfiguración (cambio de estrategia de control o algoritmo de adaptación) producidas por detección de contradicciones, funcionamiento poco satisfactorio, detección de fallos, etc.

Las primeras modificaciones se pueden llevar a cabo por algoritmos de gradiente o/y con derivadas de orden superior. En sistemas sencillos, de estructura matemática conocida, pueden derivarse modificaciones cuya estabilidad esté garantizada, mediante técnicas basadas en funciones de Lyapunov o pasividad[13]. En sistemas complejos, de modelo relativamente desconocido excepto determinadas características cualitativas, dichas técnicas no se pueden aplicar con total seguridad.

Todas estas ideas, y otras que se detallarán en las siguientes secciones, plantean cuestiones a resolver que faciliten la creación eficiente de sistemas complejos de

control basados en el conocimiento. Una posible solución sería el establecer un nivel supervisor de la adquisición de conocimiento en sistemas borrosos y neuroborrosos. Este nivel debería evaluar la cantidad y calidad del conocimiento disponible para decidir entre adaptación o reestructuración, para asegurar la ausencia de contradicciones internas, etc.

2 Cuestiones sobre control borroso.

En las aplicaciones actuales de la lógica borrosa al control de procesos existen una serie de cuestiones que muchas veces se dejan de lado o se resuelven de modo intuitivo. A continuación se exponen algunas de ellas.

Multiplicidad de operadores. La generalización de los operadores clave en lógica binaria presenta multitud de variantes (implicación [9], defuzzificación [20], encadenamiento de reglas), basadas en razones formales y/o intuitivas, algunas de las cuales adolecen de graves problemas, como conclusiones contradictorias si los consecuentes no son convexos, o la predominancia de conjuntos borrosos con elevada área. Refinamientos recurriendo a coeficientes adicionales de entropía o mediante operaciones de índole heurístico permiten resolver en ciertos casos esos problemas [22]. La elección de operadores lógicos, implicación y deborrosificación en aplicaciones prácticas industriales [1] se suele basar en la facilidad de implementación y comprensión por parte del usuario final.

Procesamiento numérico vs. conceptual. En bastantes referencias bibliográficas sobre control borroso se hace hincapié en el aspecto numérico y funcional (estabilidad, funciones de Lyapunov, parametrizaciones lineales) de los reguladores borrosos[20]. En ese contexto, pues, se ha perdido parte del espíritu inicial de la lógica borrosa como "computación con palabras y conceptos".

Interfaz sencilla para interpolación. Entroncando con la idea anterior, en muchas de las aplicaciones y desarrollos teóricos de control borroso la lógica borrosa se considera únicamente como una interfaz adecuada e intuitiva hacia unas rutinas de interpolación multidimensional, en muchos casos con modelos de aproximación funcional estáticos, lineales en parámetros, de tipo:

$$u = \frac{\sum \phi_i(\mathbf{x})\theta^i}{\sum \phi_i(\mathbf{x})} \qquad (1)$$

donde $\mathbf{x}$ es el estado del sistema y θ^i son los parámetros modificables en el aprendizaje[1].

Ambigüedad. La lógica borrosa, nacida como modelo de conceptos vagos e imprecisos se utiliza en la mayoría de las aplicaciones de control en ponderaciones del tipo antes propuesto, completamente *deterministas.*

[1]Si las funciones de validez ϕ_i incorporan parámetros modificables, se convierten en herramientas neuroborrosas análogas a redes neuronales multicapa.

Por supuesto, las ideas aquí mencionadas de una forma categórica pueden ser objeto de discusión. En esta discusión entrarían aspectos sobre la interpretación del propio significado de "borrosidad" [6], sobre la necesidad de que el control sea o no determinista, etc.

En resumen, el enfoque "interpolativo" actual de la investigación sobre control borroso ha conseguido una serie de logros que lo justifican:

- En ciertos contextos, existencia de algoritmos de identificación sencillos (derivados de mínimos cuadrados)[20].
- Algoritmos de desigualdades matriciales lineales [19] sobre combinaciones convexas de sistemas lineales.
- Generalización de resultados sobre control no lineal: pequeña ganancia, función descriptiva, identificación y control adaptativo de sistemas afines en control [13, 3].
- Aplicación de algoritmos neuronales: backpropagation, Levenberg-Marquardt[5].

Si a ello le unimos, como razones para abandonar un enfoque lógico y formal junto a la ya referida multiplicidad de operadores, el hecho de que las capacidades de los reguladores borrosos son a veces exageradas, sin comparación adecuada con otros reguladores (por ej. PID), o resultado de un ajuste fino por métodos manuales de ensayo y error, ello explica que el énfasis en el área de control borroso se concentre principalmente en sistemas de aproximación funcional, en muchos casos lineales en los parámetros.

3 Papel de los métodos borrosos en control.

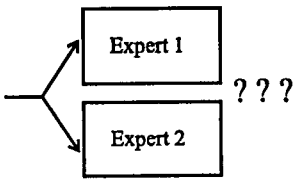
El éxito en control de la aplicación de la lógica borrosa se debe a la capacidad de la misma para utilizar modelos de conceptos ambiguos para *reducir la complejidad intuitiva* de un proceso, de manera que permite realizar operaciones de control, planificación y supervisión, aunque sea de un modo aproximado o heurístico, sobre plantas no lineales o/y variantes en el tiempo.

Una vez analizados los logros actuales en control borroso, cabe preguntarse por las posibilidades y límites del enfoque funcional. Respecto a las primeras, cabe argumentar que, dado que los controladores borrosos son aproximadores funcionales universales de funciones continuas suaves, la investigación se puede dirigir hacia el área de control no lineal geométrico, control adaptativo no lineal, etc. cuyos resultados y expectativas ya han sido esbozados.

El objetivo de esta sección es reflexionar sobre si existen o no posibilidades inexploradas en el área de la aplicación de herramientas de lógica formal al control borroso,

Si T es *Alta*, u es *Abrir*
Si T es *Alta*, u es *NoAbrir*
Si T es *Media*, u es *Notocar*
Si T es *Baja*, y es *Cerrar*
Si T es *Baja*, y es *CerrarPoco*
...
Si h es *Alto*, (T es *Baja* o Q es *Alto*)
Si h no es *MuyBajo* o T es *Alta*, Q es *Normal*

(a) Base de reglas compleja.



(b) Fusión de conocimiento.

Figura 1: Necesidad de métodos lógicos.

que no puedan ser abordadas fácilmente en el ámbito del mencionado enfoque funcional. Las cuestiones ya comentadas, junto con las siguientes, motivan la necesidad del enfoque formal:

Ambigüedad del conocimiento. En la ingeniería de control, existen dos fuentes de ambigüedad que deben ser tratadas por parte del control inteligente:

- Ambigüedad del *modelo* del proceso: la variabilidad o escaso conocimiento sobre el mismo hace que, en muchos casos, sólo se disponga de un modelo intuitivo que describe comportamientos de orden bajo, a escalas de tiempo grandes.
- Ambigüedad en las *especificaciones* de control: los índices de optimización relativos a calidad final y coste de producción son expresados de forma cualitativa ambigua. En la práctica, la experiencia del operador humano se usa para fijar referencias para reguladores jerárquicamente inferiores.

Bases de reglas complejas. Fusión. Si se dispone de una base de reglas compleja, introducida por un experto o por un algoritmo de aprendizaje, como por ejemplo la representada en la figura 1(a), sería útil detectar anomalías como contradicciones entre reglas, reglas redundantes, reglas que faltan para que la base esté completa, etc. Asimismo, en el caso de disponer de información de varias fuentes (humanas -fig. 1(b)- o/y algoritmos de aprendizaje) deberían establecerse mecanismos formales de verificación del grado de acuerdo entre ellas, mecanismos de fusión, etc.

Aprendizaje. En el caso de control con un algoritmo de aprendizaje (que incluye una evaluación del cumplimiento de determinadas especificaciones de control), la decisión entre modificaciones paramétricas o estructurales, la codificación eficiente del conocimiento disponible, la verificación de interpretabilidad de los resultados, etc. deberían ser realizadas por un nivel jerárquico supervisor de la adquisición de conocimiento.

Interpretabilidad de resultados. En el enfoque funcional, con modelos tipo (1) con un número suficiente de funciones ψ_i puede aproximarse uniformemente cualquier función continua suave [11]. El problema que se plantea es que la inclusión de un gran número de parámetros hace que los modelos (neuro-)borrosos pierdan el sentido original, intuitivamente orientado a captar las características principales del proceso y reducir complejidad. Por tanto, parece existir la necesidad de ponderar interpretabilidad con exactitud de aproximación. Esta cuestión requeriría ser discutida en más detalle: la introducción de índices de validación (regularización) en los algoritmos de aprendizaje puede posibilitar un aprendizaje dirigido con mejor convergencia de parámetros y mejor interpretabilidad final [12].

3.1 Perspectivas del control borroso avanzado.

Las conclusiones de las consideraciones hasta ahora expuestas son, en definitiva, que *sí* existe un papel a jugar en el control borroso para la la línea de investigación basada en métodos formales de lógica borrosa para el manejo del conocimiento:

- *Los métodos formales borrosos deben abordar el manejo de la incertidumbre en el conocimiento de los sistemas.*
- *La línea de investigación es distinta y complementaria al enfoque funcional interpolativo*: una lista de parámetros θ^i en (1) es un modelo en algunos casos útil, pero relativamente pobre, de la complejidad del conocimiento humano sobre un proceso.
- *El tratamiento del conocimiento debe abordar cuestiones relativas a la mejora de las capacidades de aprendizaje y a la validación de dicho conocimiento, utilizando nociones de*:
  - Consistencia (calidad del conocimiento).
  - Completitud (cantidad de conocimiento).
  - Eficiencia (recursos usados, detección de redundancia, simplicidad de la representación, conservación de la interpretabilidad por parte del usuario).

La conjunción de las dos líneas de investigación debe dar lugar en el futuro a sistemas de control con características como:

- Sistemas jerárquicos, que combinen varios modelos de distinta precisión sobre el proceso a controlar.

- Combinación de conocimiento funcional (modelos derivados de principios y leyes físicas), heurístico (proporcionado por humanos) y experimental (datos de planta) del proceso o de sus subsistemas.
- Detección de fallos, cuando el sistema *contradice* al conocimiento que se tiene de él.
- Supervisión de la adquisición del conocimiento.
- Descubrimiento autónomo de propiedades que permitan reducir la dimensionalidad del universo de entrada.
- Aprendizaje avanzado (adaptación paramétrica + modificación estructural).

3.2 Validación del conocimiento.

El conocimiento que en determinado instante tiene un regulador sobre el proceso al cual controla debería ser una compilación "inteligente" de la experiencia pasada. Bajo el término "validación" deberá abordarse si dicho conocimiento es adecuado y útil para conseguir unas determinadas especificaciones de control. La validación admite dos enfoques:

- *Interno*: verificación de la estructura lógica.
- *Externo*: verificación experimental.

Los métodos usuales de aprendizaje se centran en el segundo de los aspectos. Desde el punto de vista de mejora de los algoritmos de aprendizaje, cabe decir que la invalidez externa tiene diferentes consecuencias según su grado: ligeras discrepancias indicarían necesidad de adaptación paramétrica, mientras que discrepancias mayores necesitarían de modificación estructural (adición o eliminación de reglas), o bien serían indicativo de fallos en el sistema.

La validación interna podría incrementar la velocidad y calidad del aprendizaje [12, 14]. En la situación actual, dicha validación interna puede abordarse de dos formas:

- Verificación lógica de bases de reglas complejas, aplicado a situaciones de control experto.
- Verificación de la adecuación de determinados algoritmos de inferencia clásicos a una base de reglas (supuesta coherente): por ejemplo, aplicar métodos centroidales no está indicado cuando alguno de los conjuntos es de tamaño muy superior a otros o cuando no es convexo. Esta validación debería hacerse de forma automatizada para asegurar que determinado algoritmo interpolativo es capaz de recuperar de forma coherente la información conceptual contenida en la base de reglas [16].

4 Estado del arte en enfoque formal.

La validación lógica se originó en la validación de sistemas binarios [18, 4], investigándose recientemente su generalización al caso borroso [21, 7, 16] a partir, en ciertos casos, de primeras definiciones de consistencia basadas en el máximo de la intersección [23]. En [14] se analiza en profundidad el significado de la validación orientado hacia la aproximación de funciones en control. El enfoque citado se resume a continuación.

Se plantea una equivalencia entre una regla y una ecuación o inecuación del siguiente modo:

Definición 1 *Sean u e y dos variables numéricas pertenecientes a dos universos de discurso (de entrada U y salida Y, respectivamente), y sean $A \subset U$ y $B \subset Y$ dos conjuntos borrosos definidos por las funciones de pertenencia $\mu_A : U \to [0,1]$ y $\mu_B : Y \to [0,1]$.*

Entonces, la regla "Si u es A, entonces y es B" se define equivalente a la inecuación:

$$\mu_A(u) \leq \mu_B(y) \tag{2}$$

y la regla "Si y sólo si u es A, entonces y es B" se define equivalente a la ecuación:

$$\mu_A(u) = \mu_B(y) \tag{3}$$

Definición 2 *Dada una base de reglas, se define como conclusión ideal de la misma al conjunto de soluciones del sistema de ecuaciones equivalente, supuestas conocidas el valor de ciertas variables (premisas).*

Ejemplo. La base de reglas:

1- u es Bajo $\Rightarrow$ y es No Bajo Y z es Bajo
2- u es Medio $\Leftrightarrow$ y es Bajo
3- u es Alto$\Rightarrow$ y es Alto O z es Alto

se define equivalente al sistema de ecuaciones:

$$Bajo(u) \leq \min(1 - Bajo(y), Bajo(z))$$
$$Medio(u) = Bajo(y)$$
$$Alto(u) \leq \max(Alto(y), Alto(z))$$

La inferencia consiste en, para un particular u, calcular qué valores de y y z verifican las ecuaciones e inecuaciones, dadas las funciones de pertenencia. En el caso, por ejemplo, en el que Bajo(u)=0.5 y Medio(u)=0.6, no existiría solución: en ese caso, se dice que existe contradicción (entre las reglas 1 y 2).

Nótese que la deborrosificación *no existe* en este esquema como ente teórico básico. La presencia de deborrosificadores en un esquema práctico supone, simplemente, un algoritmo rápido de obtener, en determinadas circunstancias, una solución a los sistemas de ecuaciones.

La expresión de un modelo de sistema como un conjunto de ecuaciones abre la puerta a investigar en soluciones donde se combinen dichas ecuaciones con otras

ecuaciones obtenidas por primeros principios o experimentación sobre el proceso.

Las propiedades lógicas de la base de reglas tienen un paralelo en las propiedades algebraicas de existencia y unicidad de solución en las ecuaciones equivalentes:

Definición 3 *Si el sistema de ecuaciones equivalente a una base de reglas no tiene solución se dirá que la base es* contradictoria. *Si la tiene se dirá que es* coherente. *Si tiene más de una solución, se dirá que es* incompleta *y si la solución es la misma tras eliminar una ecuación se dirá que la regla correspondiente es* redundante.

Para posibilitar la inferencia ante bases de reglas contradictorias se definirá un índice de contradicción asociado a cada regla denominado error de inferencia. Este error de inferencia $\epsilon : U \times Y \rightarrow [0,1]$ tiene la expresión:

$$\epsilon_{IF}(u,y) = \begin{cases} 0 & \mu_A(u) \leq \mu_B(y) \\ \mu_A(u) - \mu_B(y) & \mu_A(u) > \mu_B(y) \end{cases} \quad (4)$$

$$\epsilon_{IIF}(u,y) = |\mu_A(u) - \mu_B(y)| \quad (5)$$

El error de inferencia es nulo si y es una conclusión ideal de la regla ante u. El error es 1 si y contradice totalmente a la regla ante la premisa u.

Con el error de inferencia de cada regla se construye una función de error de inferencia acumulada de una base de N reglas:

$$\epsilon(u,y) = \left(\sum_{i=1}^{N} \phi_i(u,y) \epsilon_i(u,y)^p \right)^{\frac{1}{p}} \quad (6)$$

donde p es un parámetro decidible por el usuario con valor de referencia unidad. Las funciones ϕ_i son ponderaciones interpretables como coeficientes de confianza, definidos por el diseñador de la base de reglas.

Así, la inferencia ideal (def. 2) se generaliza definiendo la *conclusión ideal* como los valores de y que minimizan $\epsilon(u,y)$. De este modo, la inferencia se convierte en la minimización de un índice de *coste de contradicción*.

En cuanto a la validación de **algoritmos convencionales** de inferencia, si $y = f(u)$ es la salida ante u de un algoritmo cualquiera (por ejemplo, uno de los muchos similares a (1)) entonces el error de inferencia de las reglas $\epsilon(u, f(u))$ provee de un índice de invalidez de dicho algoritmo. Este índice depende del algoritmo utilizado y de la disposición y forma de las funciones de pertenencia. Para que tenga significado, la base de reglas sobre la que se aplica debe ser coherente.

Si, por el contrario, $f(u)$ es una **función** genérica que se pretende modelar con la base de reglas (usada como aproximador funcional), entonces se dirá que la base de reglas modela coherentemente a la función si $\epsilon(u, f(u)) = 0$ para todo $u \in U$.

En ese caso, puede probarse que los antecedentes (A_i) y consecuentes (B_i) de las reglas verifican $A_i \subseteq f^{-1}(B_i)$, donde f^{-1} se define según el principio de extensión:

$$\mu_{f^{-1}(B_i)}(u) = \mu_{B_i}(f(u))$$

En reglas "si y sólo si", se verifica $A_i = f^{-1}(B_i)$.

Esta interpretación funcional basada en el principio de extensión permite explicar determinados procedimientos intuitivos para la construcción de bases de reglas que aseguren la coherencia, así como algoritmos de **aprendizaje** de mínima contradicción [15].

La estructura de coherencia y redundancia interna en una base de reglas pueden ser evaluadas, por ejemplo, mediante los coeficientes de coherencia y redundancia que, particularizados a dos reglas i,j son:

$$c_{ij}(u) = \min_{y|\epsilon_j(u,y)=0} \epsilon_i(u,y) \quad \rho_{ij}(u) = \max_{y|\epsilon_j(u,y)=0} \epsilon_i(u,y) \quad (7)$$

Estas definiciones posibilitan derivar herramientas formales para poder modificar, de forma automática, la forma de las funciones de pertenencia para mejorar la consistencia interna. Asimismo, dadas dos reglas con antecedentes y consecuentes similares, pueden establecerse cotas para la contradicción si una de ellas es eliminada, en función del coeficiente de redundancia.

En cuanto al **control**, los métodos formales esbozados facilitan el diseño de bases de reglas coherentes y la selección de algoritmos capaces de extraer la información de ellas con fidelidad.

Una forma sencilla de abordar el problema de control es la inversión de modelos borrosos del proceso basándose en las ecuaciones equivalentes (si la dinámica interna no observable en bucle cerrado es estable): las ecuaciones del regulador son las mismas que las del proceso, pero se trata de inferir u suponiendo conocida una referencia y_{ref}. Si el modelo es ambiguo (base de reglas incompleta) este resultado produce las acciones de control que *posiblemente* dan y_{ref}. Para manejar la ambigüedad en el modelo, se deberá calcular el conjunto de acciones de control que *necesariamente* produzcan una salida en $[y_{\text{ref}} - \delta, y_{\text{ref}} + \delta]$ como el complementario de las que produzcan posiblemente una salida *fuera* del intervalo objetivo. En [15, 14, 17] se detallan ejemplos de aplicación a aprendizaje, control por linealización por realimentación y control deslizante, que no se describen aquí por limitación de espacio.

5 Conclusiones

En este trabajo se han presentado y discutido un buen número de reflexiones sobre la situación actual de la aplicación de la lógica borrosa al control y sobre sus posibilidades futuras.

Se ha insistido en la necesidad de combinar el enfoque funcional con el enfoque formal para poder llegar a resultados con mayor "inteligencia" que integren diversos modelos del proceso y que supervisen la adquisición del conocimiento.

Una vez motivada la necesidad y utilidad de consideraciones sobre contradicción y redundancia en bases de reglas para control, se han expuesto de forma resumida algunos resultados, remitiendo a las referencias bibliográficas para su consulta en detalle y ejemplos.

Referencias

[1] P. Albertos, M. Martinez, J.L. Navarro, and F. Morant. Fuzzy controller design: a methodology. In *Proc. IEEE Conference on Control Applications*, Vancouver, 1993.

[2] P. Albertos, J. Picó, and A. Sala. Learning control systems: An overview. In *Procs. Workshop on Control of Complex Systems (COSY)*, pages 38–45, La Sapienza, Rome, Sep. 1995. European Science Foundation.

[3] J. Aracil, A. Ollero, and A. García-Cerezo. Stability indices for the global analysis of expert control systems. *IEEE Trans. on Systems, Man & Cybernetics*, 19:988–1007, 1989.

[4] B. Cragun and H. Steudel. A decision-table-based processor for checking completeness and consistency in rule-based expert systems. *Int. J. Man-Machine Studies*, 26(5):633–648, 1987.

[5] M. di Martino, S. Fanelli, and M. Protasi. Exploring and comparing the best direct methods for the efficient training of MLP-networks. *IEEE Trans. on Neural Networks*, 7:1497–1502, 1996.

[6] D. Dubois and H. Prade. The three semantics of fuzzy sets. *Fuzzy Sets and Systems*, 90(2):142–150, 1997.

[7] D. Dubois, H. Prade, and L. Ughetto. Checking the coherence and redundancy of fuzzy knowledge bases. *IEEE Trans. on Fuzzy Systems*, 5(3), August 1997.

[8] J.-S.R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing.* Prentice Hall, 1997.

[9] E. Kerre. A comparative study of the behaviour of some popular fuzzy implication operators on the generalized modus ponens. In L.A. Zadeh and J. Kacprzyk, editors, *Fuzzy Logic for the Management of Uncertainty*, pages 281–295. John Wiley & Sons, New York, 1992.

[10] F. Morant, P. Albertos, M. Martinez, A. Crespo, and J.L. Navarro. RIGAS: An intelligent controller for cement kiln control. In *Proc. IFAC Symp. Artif. Intelligence in Real Time Control*, Delft, 1992.

[11] K.Ñarendra and K. Parthasarathy. Gradient methods for the optimization of dynamical systems containing neural networks. *IEEE Trans. on Neural Networks*, 2(2):252–262, March 1991.

[12] W. Pedrycz and J. Valente de Oliveira. Optimization of fuzzy models. *IEEE Trans. on Systems, Man & Cybernetics*, 26(4):627–636, August 1996.

[13] J. Picó. *Identificación y Control Adaptativo Neuroborroso de Sistemas Discretos no Lineales.* PhD thesis, Depto. Ingeniería de Sistemas, Comp. y Automática, Univ. Politécnica Valencia, 1996.

[14] A. Sala. *Validación y aproximación funcional en sistemas de control basados en lógica borrosa.* PhD thesis, DISA, Univ. Politécnica Valencia, 1998.

[15] A. Sala and P. Albertos. Inference error minimisation: Fuzzy modeling of ambiguous functions. In *Proc. of Seventh IFSA World Congress*, volume 3, pages 440–445, Prague, 1997. Academia.

[16] A. Sala and P. Albertos. Fuzzy systems evaluation: The inference error approach. *IEEE Trans. on Syst. Man & Cybernetics*, 28B(2):268–275, 1998.

[17] A. Sala and P. Albertos. Variable-structure control with fuzzy plant model. In *Proc. IFAC Workshop on System Structure and Control*, Nantes (F), 1998.

[18] M. Suwa, A. Scott, and E. Shortliffe. An approach to verifying completeness and consistency in a rule-based expert system. *AI Magazine*, 3(4):16–21, 1982.

[19] K. Tanaka, T. Ikeda, and H.O. Wang. Fuzzy regulators and fuzzy observers: Relaxed stability conditions and LMI-based designs. *IEEE Trans. on Fuzzy Systems*, 6(2):250–265, 1998.

[20] L.-X. Wang. *Adaptive Fuzzy Systems and Control.* Prentice-Hall, Englewood Cliffs, NJ, 1994.

[21] R.R. Yager and H.L. Larsen. On discovering potential inconsistencies in validating uncertain knowledge bases by reflecting on the input. *IEEE Trans. on Systems, Man & Cybernetics*, 21(4):790–801, 1991.

[22] W. Yu and Z. Bien. Design of fuzzy logic controller with inconsistent rule base. *Journal of Intelligent and Fuzzy Systems*, 2:147–159, 1994.

[23] L.A. Zadeh. A theory of approximate reasoning. *Machine Intelligence*, 9:149–194, 1979.

Construcción de un Clasificador Borroso mediante Programación Genética basada en Registros

Luis Junco Navascués Luciano Sánchez Ramos
Departamento de Informática. Universidad de Oviedo
E.S.M.C. Campus de Viesques
Gijón. Asturias
junco,luciano@lsi.uniovi.es

Resumen

Un sistema de clasificación borroso puede inducirse automáticamente a partir de una muestra de ejemplos clasificados, mediante la combinación de un algoritmo genético y de un lenguaje basado en reglas borrosas. En este trabajo definiremos una metodología en la que se realiza una estimación no paramétrica de las funciones de pertenencia que definen la semántica del lenguaje mencionado, mediante una estrategia de tipo Pittsburg combinada con técnicas propias de la programación genética. La programación genética se emplea para inducir las expresiones de las funciones de pertenencia de los conjuntos borrosos que definen la semántica del clasificador. Estas expresiones consisten en una secuencia de instrucciones interpretables en una máquina de registros.

Palabras clave: Programación Genética Basada en una Máquina de Registros, Sistemas Basados en Reglas Borrosas Evolutivos, Algoritmos GA-P

1 Introducción

Los sistemas basados en reglas borrosas son una de las aplicaciones más importantes de la teoría de conjuntos borrosos de Zadeh. Estos sistemas extienden los sistemas clásicos basados en reglas y se han aplicado a una gran cantidad de problemas, entre los cuales está la generalización borrosa del problema del análisis discriminante. Llamaremos a este problema *clasificación borrosa.*

La clasificación borrosa admite que la frontera entre dos clases vecinas no es una superficie sino una región borrosa, de forma que un objeto puede tener pertenencia parcial a varias clases. Este enfoque no sólo refleja la realidad de que muchas categorías reales tienen fronteras difusas, sino que también proporciona una representación simple de una partición potencialmente complicada del espacio de características. Usando este esquema podemos aplicar reglas *si-entonces* para describir un clasificador [12][13]. Una regla típica de clasificación podría ser

$$\text{si } X_1 \text{ es } A_1 \text{ y } X_2 \text{ es } A_2 \text{ y } \ldots \text{ y } X_n \text{ es } A_n \text{ entonces } Z \text{ es } C$$

donde $X_1, X_2, \cdots, X_n$ son las variables de entrada y $A_1, \cdots, A_n$ son variables lingüísticas, caracterizadas por funciones de pertenencia apropiadas, que describen los valores de las propiedades del objeto Z. La fuerza con que dispara una regla, o el grado de validez de ésta, es el grado con el que el objeto pertenece a la clase C. Llamaremos *descriptivo* a un banco de reglas definido de este modo.

Cuando cada regla define una región borrosa arbitraria (es decir, que no necesariamente sea el resultado de componer etiquetas lingüísticas mediante conectivas lógicas) la interpretabilidad del sistema es menor, pero cada regla puede cubrir regiones de forma más compleja y en general el banco será más sencillo. Los sistemas *aproximativos* constan de reglas

$$\text{si } (X_1, X_2, \ldots, X_n) \text{ es } A \text{ entonces } Z \text{ es } C$$

donde A es un conjunto borroso no necesariamente asociado a un término lingüístico.

El objetivo de este trabajo es encontrar expresiones adecuadas para las funciones de pertenencia de las variables A_i en los sistemas descriptivos o para los conjuntos A en los sistemas aproximativos. Estas pertenencias han de dar lugar a reglas compatibles con una muestra de objetos clasificados previamente. Este estudio ya ha sido realizado por otros autores cuando las funciones de pertenencia se definen paramétricamente. Entre otros métodos, se pueden buscar los valores de los parámetros desconocidos mediante un algoritmo genético [9][1]. También hay sistemas que ajustan modificadores más o menos complejos [7], lo que permite mayor riqueza en la definición. Nosotros no definiremos paramétricamente los conjuntos borrosos, sino

que buscaremos un algoritmo que computará la forma de esa función. Las únicas limitaciones impuestas al algoritmo son su longitud máxima (definida como el número de instrucciones que emplea) y la ausencia de composiciones iterativas y llamadas recursivas en su definición. Llamaremos a esta representación *funcional.* En la representación funcional, cada uno de los conjuntos borrosos se asociará a una secuencia de instrucciones en una máquina de registros, cuya ejecución (tomando como dato el valor de las variables de entrada) producirá como salida el valor de la función de pertenencia. Estas secuencias de instrucciones se inducirán mediante Programación Genética.

2 Derivación de una base de reglas borrosas mediante Fuzzy GP y Fuzzy GA

La "programación genética borrosa" (*Fuzzy GP*) es un nuevo método de aprendizaje de máquina que se basa en la combinación de un algoritmo genético con un lenguaje de reglas borrosas. En contraste con los algoritmos genéticos borrosos (*Fuzzy GA*) [1] la programación genética borrosa no optimiza los parámetros de funciones de pertenencia trapezoidales o con forma de campana de una base de reglas con estructura fija, sino que explora diferentes estructuras de clasificadores definidos mediante un lenguaje borroso a partir de un diccionario en el que se definen las variables lingüísticas y varios modificadores [3]. La programación genética borrosa no cambia la semántica del lenguaje de reglas borrosas, pero en cambio es capaz de derivar la estructura de la base de reglas, que no necesita definirse por el programador. De este modo, las variables que no contribuyen a producir buenas soluciones son eliminadas durante el proceso de evolución. Se hace notar que ésta no es la única aplicación de la programación genética a los sistemas de clasificación. Por ejemplo, un clasificador basado en una idea similar, pero empleando funciones discriminantes y un lenguaje no borroso, fue aplicada también a problemas de visión artificial, en una metodología orientada a reducir al mínimo el tiempo de clasificación en problemas con gran número de entradas [11] y pueden encontrarse otras aplicaciones en [6].

Por otra parte, es bien sabido que los algoritmos genéticos se combinan con los sistemas borrosos de tres formas fundamentales, conocidas como enfoques Michigan, Pittsburg e Iterativo [1] y que con estos métodos se pueden inducir tanto la estructura del banco como su semántica. En este estudio emplearemos programación genética en combinacion con el segundo de estos métodos (y por tanto nuestro método, aun siendo un algoritmo de programación genética aplicado al aprendizaje de reglas borrosas, no está relacionado directamente con la programación genética borrosa tal y como ha sido definida en [3]). Plantearemos el empleo de la programación genética en la determinación de la semántica del lenguaje y no en la determinación de la estructura del banco, lo que es un problema abierto, según este mismo autor:

> However, a lot of work remains to be done. The most important problems to solve are to increase the speed of convergence while maintaining stability, [...], including the definition of the membership function in the genetic coding, ...

La solución que proponemos supone una mejoría respecto a los métodos basados solamente en algoritmos genéticos ya que la codificación de las funciones de pertenencia no es paramétrica. Cada función de pertenencia se codifica mediante un fragmento de programa con lo que la gama de funciones representables es muy amplia. Nuestra propuesta está basada en la combinación de dos metodologías: la Programación Genética Basada en una Máquina de Registros y los Algoritmos GA-P.

La primera de las técnicas logra una codificación muy eficiente de un programa en una cadena de símbolos similar a las empleadas en algoritmos genéticos. Esta representación tiene dos ventajas importantes con respecto a la programación genética canónica [6], en la que el código se representa mediante árboles: la evaluación de los individuos requiere menos cálculos, lo que permite simular un número mayor de generaciones, y la evolución de la población hacia soluciones adecuadas requiere menor número de generaciones, debido a la forma en que se definen los operadores de cruce y mutación. Por otro lado, los algoritmos GA-P mejoran a la programación genética clásica en la forma en que se manejan las constantes, ya que el valor de éstas se hace evolucionar, al contrario que en el algoritmo de Koza, y permiten el empleo de técnicas de optimización local [2]. La programación genética basada en registros y los algoritmos GA-P se describen, respectivamente, en [10] y [5].

3 Metodología propuesta

El método de diseño de clasificadores que proponemos es una variante del enfoque Pittsburg [9]. Hemos asociado las pertenencias de los conjuntos borrosos que definen las etiquetas lingüísticas con secuencias de instrucciones de longitud fija. El lenguaje que hemos empleado se interpreta en una máquina elemental basada en registros, que describiremos en el siguiente apartado.

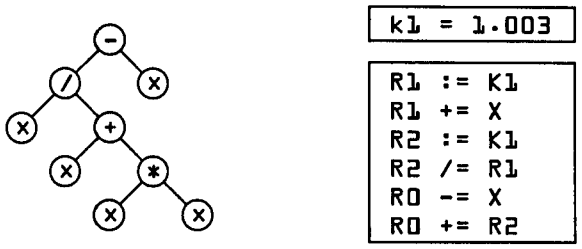


Figura 1: Codificación de una función: GP canónica y GP registros

Al igual que en el algoritmo GA-P de Howard [5] cada individuo consta de dos partes: la parte GA y la parte GP. En la parte GA se almacenan valores de constantes, con codificación real, y en nuestra variante del GA-P en la parte GP se concatenan tantas cadenas de instrucciones como valores pueden tomar las variables lingüísticas de las que hace uso el banco de reglas borrosas.

En los bancos de reglas descriptivos se completarán dos fases: en primer lugar se define una sintaxis del problema (se asignan funciones de pertenencia trapezoidales a las variables $A_1, \ldots, A_n$) y se determina la estructura del banco de reglas que optimice un índice de funcionamiento. En esta etapa se puede incorporar el conocimiento lingüístico del que se disponga acerca del sistema de clasificación (se mostrará un ejemplo práctico de esta incorporación de conocimiento en una sección posterior). En la segunda fase se optimizan las funciones de pertenencia reemplazando sus definiciones por secuencias de instrucciones que se hacen evolucionar mediante el esquema de la programación genética basada en registros. La definición de los sistemas aproximativos es aun más sencilla; la representación funcional que proponemos permite que cada clase sea discriminada por una única regla sin que sea preciso abordar la primera de las fases.

Las partes GA y GP pueden tratarse de forma homogénea, al contrario que en el GA-P de Howard. Emplearemos las operaciones de cruce y mutación definidas en [10] sobre ambas partes.

3.1 La máquina virtual de registros

Como hemos visto, cada función de pertenencia se define mediante una función, implementada en el lenguaje de una máquina de registros que definimos a continuación.

En nuestra máquina todas las instrucciones realizan operaciones aritméticas sobre los registros. Cada programa lleva asociado un vector de entrada, un número determinado de constantes, una serie de registros y diversos procesos de inicialización y término de la ejecución. La longitud máxima de un programa está limitada a un número dado de instrucciones aunque, de hecho, su longitud efectiva va a ser siempre menor, ya que gran parte del código no interviene en el resultado final. Estos fragmentos de código no utilizado o *introns* son necesarios para el proceso de búsqueda.

El proceso de inicialización asigna el valor cero a cada uno de los registros. El proceso terminal calcula la media aritmética de una serie de registros seleccionados (normalmente el registro R0) y aplica al resultado una función sigmoide con un rango de salida entre 0.0 y 1.0 para normalizar la salida del programa.

3.1.1 Conjunto de instrucciones

El conjunto de instrucciones de la máquina virtual de registros define la base del comportamiento de las funciones de pertenencia que vamos a codificar con los mismos.

Para los casos más generales se han implementado las siguientes operaciones (llamamos R_i al registro i-ésimo, K_j a la constante j-ésima y X_k a la entrada k-ésima). Todas las operaciones están protegidas para evitar desbordamientos.

$R_i \leftarrow 0$	$R_i \leftarrow R_l$	$R_i \leftarrow K_j$
$R_i \leftarrow X_k$	$R_i \leftarrow R_i + K_j$	$R_i \leftarrow R_i + R_l$
$R_i \leftarrow R_i + X_k$	$R_i \leftarrow R_i - K_j$	$R_i \leftarrow R_i - R_l$
$R_i \leftarrow R_i - X_k$	$R_i \leftarrow R_i * K_j$	$R_i \leftarrow R_i * R_l$
$R_i \leftarrow R_i * X_k$	$R_i \leftarrow R_i / K_j$	$R_i \leftarrow R_i / R_l$
$R_i \leftarrow R_i / X_k$	$R_i \leftarrow \sin(R_l)$	$R_i \leftarrow \sin(X_k)$
$R_i \leftarrow \cos(R_l)$	$R_i \leftarrow \cos(X_k)$	

4 Resultados Experimentales

En orden a analizar y comparar las características de nuestro clasificador lo hemos aplicado a dos problemas clásicos, bien conocidos: Iris y Pima. Hemos empleado bancos de reglas aproximativos, con una única regla por clase (o lo que es lo mismo, los conjuntos borrosos obtenidos son una partición del espacio de características).

En la próxima sección aplicaremos la metodología expuesta anteriormente para solucionar un problema práctico de clasificación con un banco de reglas descriptivo.

4.1 Iris

El problema inicialmente planteado por Fisher [14] cuenta con 150 ejemplos en los que se han medido cuatro atributos con los que tratamos de diferenciar tres tipos diferentes de flores (setosa, versicolor y virginica).

Hemos aplicado el método de validación cruzada con diez particiones aleatorias para medir la capacidad de aprendizaje en cada uno de los conjuntos de entrenamiento y calcular una estimación del error a través de los subsiguientes tests. En la función de fitness, que mide la adecuación del clasificador durante la etapa de entrenamiento, intervienen dos factores: el porcentaje de aciertos por un lado, y la longitud del programa generado por otro. Este último está relacionado con la capacidad de generalización del clasificador. A menos instrucciones se corresponden funciones más sencillas, menos tendentes a sobreentrenar el sistema.

Para cada una de las series hemos lanzado 10 veces el programa de generación del clasificador, eligiendo aquel con mejor ajuste para realizar el test. Los resultados para cada una de las series se muestran a continuación:

Serie	Entrenamiento	N. Instr.	Test
1	100.00%	48	93.34%
2	100.00%	24	100.00%
3	100.00%	29	100.00%
4	100.00%	43	93.34%
5	100.00%	42	100.00%
6	100.00%	23	100.00%
7	100.00%	29	93.34%
8	100.00%	34	100.00%
9	99.26%	54	93.34%
10	99.26%	30	93.34%
Global	99.85%		96.67%

Los resultados obtenidos con varios algoritmos clásicos (C4.5 [15], CN2 [16] y LVQ [17]) pueden apreciarse en la siguiente tabla.

Algoritmo	Entrenamiento	Test
C4.5	98.38%	92.7%
CN2	98.92%	94.16%
LVQ	98.55%	95.72%

Observando los resultados podemos notar que nuestro sistema posee una gran capacidad de generalización y un gran porcentaje de aciertos en Test, al mismo tiempo. Los programas generados poseen una media de 36 instrucciones, lo que da una idea de la sencillez de las soluciones encontradas.

4.2 Pima

Esta base de datos contiene 768 casos de diagnóstico de diabetes donde se toman 8 variables para diferenciar los dos posibles estados: enfermo o sano.

A continuación se muestran los resultados globales para todas y cada una de las series así como los porcentajes de clasificación de otros sistemas.

Serie	N. Instr.	Entrenamiento	Test
1	62	80.20%	76.65%
2	82	81.08%	77.08%
3	34	79.65%	75.13%
4	36	79.65%	79.29%
5	68	79.90%	74.35%
Global	57	80.10%	76.50%

Algoritmo	Entrenamiento	Test
C4.5	96.06%	71.4%
CN2	85.4%	74.5%
LVQ	83.68%	67.71%

Es interesante reseñar que el clasificador obtenido para la serie 4 únicamente posee 36 instrucciones y depende de cuatro de las 8 variables de entrada. En otro articulo [11], como ya hemos mencionado, mostrábamos la capacidad de un método de similares características (basado en GA-P tradicional y no en registros) para actuar como selector de características en problemas de alta dimensionalidad.

5 Aplicación práctica: Clasificación de defectos de fabricación en planchas de vidrio

Este es el problema original que nos impulsó a desarrollar el método de clasificacion. En este caso vamos a abordar el entrenamiento de las funciones de pertenencia correspondientes a un juego de etiquetas lingísticas que forman parte de la base de conocimiento. Contamos con un conjunto de 360 ejemplos, cada uno de los cuales posee 5 atributos tomados de medidas realizadas sobre las imágenes de los defectos. Cada tipo de defecto viene caracterizado por una regla definida a priori por uno de los operarios que anteriormente trabajaba en la identificación visual de los mismos. No entraremos a detallar la base del funcionamiento del sistema automático de localización y extracción de características de cada defecto. La descripción completa del problema está en [8].

Para el entrenamiento del clasificador hemos codificado cada una de las 5 funciones de pertenencia a cada etiqueta conjuntamente en cada cromosoma de la población. Cada programa posee una sola variable de entrada (la correspondiente a su etiqueta) y cuatro constantes. Se ha impuesto un número máximo de instrucciones reducido (20) con el fin de forzar al sistema la busqueda de funciones sencillas. Los parámetros que definen el algoritmo de entrenamiento son:

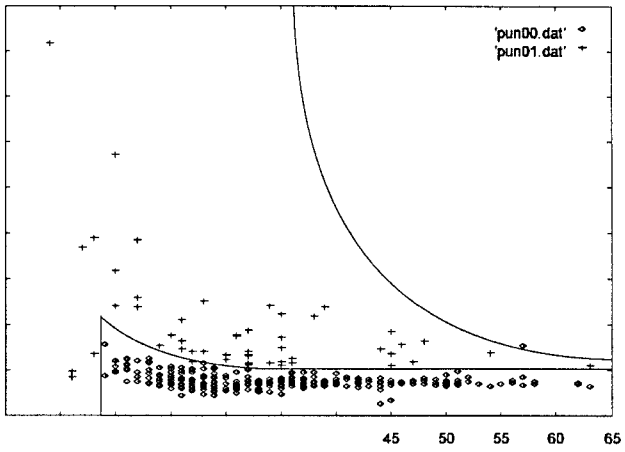


Figura 2: Relaciones entre los descriptores "forma de anillo" y "contraste"

Población	300 individuos
N. Subpoblaciones	3
N. funciones de pertenencia	5
Long. programa por cada f. de pert.	20 instrucciones
N. constantes por cada f. de pert.	4
N. registros	16
Prob. cruce	80%
Prob. mutación	20%
Prob. intercambio subpoblaciones	0.1%
Generaciones	350

En la tabla que sigue, se comparan los resultados obtenidos mediante este método, un clasificador estadístico y un clasificador borroso diseñado mediante el método Pittsburg pero con funciones de pertenencia paramétricas trapezoidales.

Algoritmo	Entr.	Test
K-Vecinos (K=3)		87.40%
Parametr. trapezoidal	94.2%	92.2%
Codificación funcional	97.38%	98.83%

También se ha utilizado el método aproximativo para resolver el mismo problema. Se aprecia una nueva mejora en los resultados debido a que el sistema es capaz de relacionar aspectos de los descriptores que el operador habría pasado por alto por constituir relaciones demasiado complejas entre las variables.

En la gráfica 2 se relacionan los descriptores "Forma de anillo" y "Contraste". El clasificador logró encontrar una función que haciendo únicamente uso de estos dos parámetros discriminaba en un 99.3% todos los ejemplos del entrenamiento. Nótese que haciendo uso de un clasificador descriptivo no es posible llegar a discriminar del mismo modo.

6 Conclusiones

En este trabajo se ha definido un método que permite inducir bancos de reglas borrosas aproximativos y descriptivos en problemas de clasificación. Los conjuntos borrosos que definen la semántica de los bancos de reglas se han obtenido mediante programación genética y no se basan en una parametrización de las funciones de pertenencia de los mismos, por lo que pueden definirse pertenencias de forma compleja que reducen el número de reglas necesario para describir el clasificador. Los resultados experimentales muestran que el método es capaz de competir con ventaja con otros clasificadores clásicos y borrosos.

Agradecimientos

Agradecemos a O. Cordón, M. José del Jesús y a F. Herrera la colaboración prestada asi como sus valiosos comentarios acerca de este trabajo.

A Apéndice

A.1 Resultados obtenidos en la aplicación práctica

A continuación se incluye como ejemplo la decodificación de uno de los términos del individuo ganador en el problema de la clasificacion de defectos de fabricacion en planchas de vidrio, así cono la representación gráfica de la función de pertenencia correspondiente (figura 3).

```
[Función 5 (término: Forma de Anillo)]

K00 = 63.498774;  K02 =  10.411939;
K01 = 30.524333;  K03 = -88.616320;
```

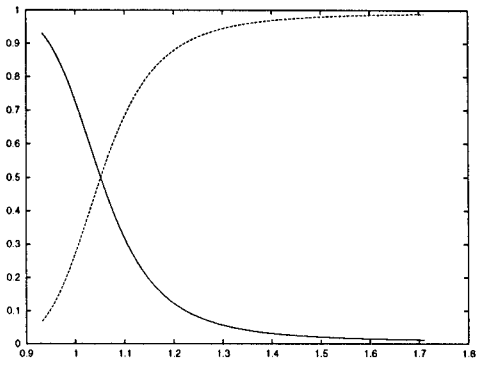


Figura 3: F. de pertenencia término "Anillo"

```
00 R05 += K02;     08 R06 /= X;
01 R10 -= R05;     09 R05 /= K02;
02 R06 *= X;       10 R05 += K01;
03 R06 += R10;     11 R05 += X;
04 R06 *= X;       12 R00 := cos R05;
05 R06 += K02;     13 R00 += R06;
06 R06 /= X;       14 R00 += R06;
07 R05 /= K02;
```

Referencias

[1] Cordón, O., Herrera, F. "A three stage evolutionary process for learning descriptive and approximative fuzzy logic controller knowledge bases from examples" IJAR 17(4), 1997, pp 369-407

[2] Cordon, O., Herrera, F. y Sanchez L. (1997) "Evolutionary learning processes for data analysis in electrical engineering applications". En Quaglirelli D., Periaux J., Poloni C., y Winter G. (Eds.) *Genetic Algorithms in Engineering and Computer Science*. John Wiley and Sons.

[3] Geyer-Schulz, A. (1997) *Fuzzy Rule Based Expert Systems and Genetic Machine Learning*. Physica-Verlag.

[4] Holland, J. (1987) "Genetic Algorithms and classifier systems: foundations and future directions" in *Genetic Algorithms and Their Applications*, pp 82-89.

[5] L. Howard, D. D'Angelo, "The GA-P: a genetic algorithm and genetic programming hybrid," *IEEE Expert*, pp. 11-15, 1995.

[6] Koza, J. R. *Genetic Programming: on the programming of computers by means of natural selection*. MIT Press.

[7] Velasco, J. López, S. , Magdalena L. (1997) "Genetic Fuzzy Clustering for the definition of Fuzzy Sets". proc. FUZZ-IEEE'97 pp. 1665-1670

[8] Junco Navascués, L., Sánchez L. (1998) "Localization and fuzzy clasification of defects in sheets of glass" Mathware and Soft Computing. Por aparecer.

[9] Michaliewicz, Z. *Genetic Algorithms + Data Structures=Evolution Programs*. Springer-Verlag, 1996.

[10] Nordin, P. (1997) *Evolutionary program induction of binary machine code and its applications*. Krehl Verlag.

[11] Sánchez, L., Junco, L., Cano, F., Couso, I. (1998) "Clasificación y selección de características mediante algoritmos GA-P". Revista Iberoamericana de Inteligencia Artificial. Por aparecer.

[12] Sun, C.T, Jang, J.S. (1997) "Fuzzy Classification based on adaptive networks and genetic algorithms". In *Genetic Algorithms and Fuzzy Logic Systems. Soft Computing Perpspectives*. E. Sanchez, T. Shibata, L. Zadeh, eds. pp 113-131

[13] Zadeh, L. (1978) "Fuzzy Sets and their application to pattern classification and clustering analysis" in *Classification and Clustering*, ed. J. van Ryzin , pp 251-299

[14] Fisher,R.A. (1936) "The use of multiple measurements in taxonomic problems".In *Annual Eugenics*, 7, Part II, 179-188; also in *Contributions to Mathematical Statistics*, John Wiley, NY, 1950.

[15] J.R. Quinlan (1993) "C4.5: Programs for Machine Learning", *Morgan Kaufmann Publishers*, San Francisdo.

[16] P. Clark, T. Niblett. (1986) "Learning If Then Rules in noisy domains", TIRM 86-019 The Turing Institute, Glasgow.

[17] T. Kohonen. (1995) "Self-Organizing Maps", *Springer-Verlag Series in Information Sciences*, 30.

SELECCIÓN DE VARIABLES CON TÉCNICAS DIFUSO-EVOLUTIVAS EN MINERÍA DE DATOS

Antonio F. Gómez-Skarmeta **Fernando Jiménez** **Jesús Ibáñez**
Departamento de Informática y Sistemas,
Facultad de Informática, Universidad de Murcia,
30071-Espinardo, Murcia
skarmeta@dif.um.es, fernan@dif.um.es, jesus@dif.um.es

Resumen

Se pueden encontrar muchas similaridades entre los procesos de Minería de Datos y de Modelado Difuso, especialmente cuando tenemos que tratar con grandes conjuntos de datos con ruido e imprecisión. Una de ellas corresponde al preprocesamiento de los datos para seleccionar las variables más adecuadas a usar en el proceso de descubrimiento de conocimiento por medio de técnicas de la minería de datos o del modelado difuso. En este documento se investiga un sistema de aprendizaje híbrido, que, como un paso de preprocesamiento, combina técnicas difuso-evolutivas para encontrar características o variables útiles para representar los datos.

Palabras Clave: algoritmos evolutivos, preprocesamiento de datos, sistemas híbridos.

1 INTRODUCCIÓN

La propuesta de Zadeh de modelar el mecanismo del pensamiento humano con valores lingüísticos difusos en vez de con números, condujo a la introducción de la borrosidad dentro de la teoría de sistemas y al desarrollo de una nueva clase de sistemas llamados sistemas difusos. En el modelado difuso, el problema más importante es la identificación de un modelo difuso usando datos del tipo entrada-salida. De un modo similar, la Minería de Datos tiene como objetivo la criba de datos para revelar información útil por medio de una representación adecuada al usuario, comprimiendo enormes registros de datos. En este contexto, un problema común a ambas técnicas es cómo tratar con mecanismos de selección de variables a partir de la posible colección de variables que pueden ser consideradas de los datos disponibles.

Aunque hemos usado el término Minería de Datos, usamos, como en [6], el término Descubrimiento de Conocimiento en Bases de Datos (DCBD) o sólo Descubrimiento de Conocimiento (DC) para denotar el proceso completo de extraer conocimiento de alto nivel a partir de datos de bajo nivel, y el término Minería de Datos como el acto concreto de extraer patrones o modelos de los datos. Por tanto, muchos pasos preceden al de minería de datos, y uno de ellos es el preprocesamiento de los datos para seleccionar las variables adecuadas para ser usadas en la construcción o extracción del modelo. Este paso es similar a la identificación de variables dentro del paso de identificación de la estructura, en el proceso de modelado difuso.

Dado un conjunto de datos para el que presuminos alguna dependencia funcional, surge la cuestión de si hay alguna metodología adecuada para derivar reglas (difusas) a partir de los datos que caracterizan la función desconocida, de forma tan precisa como sea posible. Recientemente se han propuesto numerosas aproximaciones para generar automáticamente reglas if-then a partir de datos numéricos sin expertos en el dominio [9]. Este tipo de problema es similar al que podemos encontrar en el área de Descubrimiento de Conocimiento, y en este sentido presentamos aquí una técnica Soft Computing que intenta tener en cuenta uno de los problemas fundamentales como es el preprocesamiento de los datos para seleccionar las características o variables adecuadas para que el proceso de Minería de Datos pueda realizarse con más información.

A medida que intentamos resolver problemas del mundo real, nos damos cuenta de que son normalmente sistemas mal definidos, difíciles de modelar y con espacios de solución de gran escala. En estos casos, los modelos precisos son poco prácticos, demasiado caros o inexistentes. La información relevante disponible está normalmente en la forma de conocimiento empírico previo y datos del tipo entrada-salida representando instancias del comportamiento

del sistema. Así pues, necesitamos sistemas de razonamiento aproximado capaces de manejar esa información imperfecta. Sof Computing es un término acuñado recientemente que describe el uso simbiótico de muchas disciplinas emergentes de computación que intentan manejar la información imperfecta. De acuerdo con Zadeh (1994): "... en contraste con lo tradicional, hard computing, soft computing es tolerante a la imprecisión, a la incertidumbre y a la verdad parcial". Dentro de estas técnicas tenemos la Lógica Difusa, el Razonamiento Probabilístico, las Redes Neuronales y los Algoritmos Evolutivos. Durante los últimos años hemos visto un número creciente de algoritmos híbridos, en los que dos o más tecnologías Soft Computing se han integrado para mejorar el rendimiento global del algoritmo [2].

Como sucede en la minería de datos, si intentamos revelar información útil de los datos, un paso fundamental es el preprocesamiento de los datos para seleccionar las variables más adecuadas. En el contexto del modelado y más concretamente del modelado difuso, el objetivo es encontrar un conjunto de relaciones que describan el comportamiento presente en los datos por medio de una colección de patrones o reglas if-then [11]. En este proceso, la identificación de la estructura de un sistema tiene que encontrar las variables que representan los datos de un modo más preciso, de entre una colección de posibles variables. En este contexto tenemos que seleccionar un número finito de variables entre una colección finita de posibles candidatos. Ésto es un problema combinatorio. Una posible aproximación a este problema es intentar asignar cierto grado a cada variable dependiendo de su importancia en la consecución del objetivo final. Ésto es similar a una fusión multisensor, que consiste en una combinación de diferentes fuentes de información en un formato de representación. Cuando cada variable representa características diferentes, tratamos con información complementaria. Su fusión nos permite resolver ambigüedad en la información [3].

En este proceso de fusión de las diferentes variables de entrada posibles, tenemos dos objetivos. Primero queremos determinar la importancia de las entradas a fusionar; segundo queremos determinar los parámetros exactos de las funciones de agregación usadas en la fusión. Encontrar los mejores parámetros de la función de agregación es un proceso de optimización. Los algoritmos evolutivos han demostrado ser muy útiles para este proceso porque aportan una búsqueda robusta en espacios de búsqueda complejos y no quedan atrapados en mínimos locales como les sucede a las técnicas de gradiente descendente.

Una vez que las variables son identificadas, el siguiente paso en la identificación de la estructura tiene que ver con la identificación de relaciones o, en otras palabras, con el descubrimiento de los patrones o el modelo a partir de los datos, y su representación por medio de diferentes alternativas como por ejemplo reglas difusas. En este paso se pueden usar diferentes técnicas soft computing en el contexto del DC, como las propuestas en [3][8].

2 UN ALGORITMO EVOLUTIVO PARA IDENTIFICACIÓN DE VARIABLES

Los *Algoritmos Evolutivos* (AE) [1] son procedimientos adaptativos de optimización y búsqueda que encuentran soluciones a problemas, inspirados en los mecanismos de la evolución natural. Imitan, a un nivel abstracto, en una aproximación basada en poblaciones, principios biológicos tales como la herencia de información, la variación de información mediante cruce/mutación, y la selección de individuos en base a la aptitud. Los AE comienzan con un conjunto inicial (población) de soluciones alternativas (inividuos) para el problema dado, que son evaluadas en términos de la calidad de la solución (aptitud). Entonces, se aplican los operadores de selección, replicación y variación para obtener nuevos individuos (descendencia) que constituyen una nueva población. La interacción de selección, replicación y variación de los más aptos lleva a soluciones cuya calidad va en aumento a lo largo de muchas iteraciones. Cuando finalmente se alcanza un criterio de terminación, tal como un número máximo de generaciones, el proceso de búsqueda termina, y la solución final se muestra como salida.

Básicamente, podemos decir que un AE se basa en los siguientes componentes, para cualquier tipo de aplicación: una *representación* de soluciones del problema, un modo de crear una *población inicial* de soluciones; una *función de evaluación* para medir la aptitud de cualquier solución, que juega además el papel de "entorno", en el que las mejores soluciones (es decir, aquellas que tienen un mejor valor objetivo) pueden tener mayor probabilidad de supervivencia; *operadores de variación* (reglas de transición probabilística) que realizan la composición de los hijos durante la reproducción; y valores para los *parámetros* que el algoritmo utiliza para guiar su evolución: tamaño de la población, número de generaciones, probabilidades de cruce y mutación, etc.

La clase de AE mejor conocida es la de los *Algoritmos Genéticos* (AG) [7], que han recibido mucha atención en los últimos años. Otras variantes de AE, tales como la *Programación Genética*, *Estrategias de Evolución*, o *Programación Evolutiva* son menos populares, aunque muy poderosas también. Los AG clásicos usan cadenas

binarias de longitud fija para representar individuos y dos operadores genéticos básicos (mutación binaria y cruce binario). No obstante, el campo de los AG se caracteriza por un gran dinamismo, y continuamente se desarrollan modificaciones y extensiones de la tecnología.

En este documento describimos un AE para identificación de variables en Minería de Datos y DC. Expresado en términos de programación matemática, el problema puede ser formulado como sigue:

$$Min \quad E = \sqrt{\frac{\sum_{t=1}^{n}\left(y_t - \left(\sum_{j=1}^{p} w_j x_j^{t\,f}\right)^{1/f}\right)^2}{n}} \qquad (1)$$

$$s.\,a: \quad \sum_{j=1}^{p} w_j = 1,\ 0 \le w_j \le 1,\ j = 1,\ldots,p$$

donde n es el número de datos, p es el número de variables, y_t es el valor esperado de salida para el vector de variables de entrada $X_t = \{x_1^t,\ldots,x_p^t\}$, y $\left(\sum_{j=1}^{p} w_j x_j^{t\,f}\right)^{1/f}$ es el valor del modelo *media generalizada* (Dyckhoff and Pedrycz [5]) con grado de borrosidad f. Este tipo de operador de agregación es similar al operador OWA (*Ordered Weighted Averaging*) introducido por Yager [13]. Los pesos w_j, $j = 1,\ldots,p$, han de ser averiguados.

Aunque pueden usarse numerosas conectivas de conjuntos difusos con la finalidad de agregación [14], el operador media generalizada satisface las propiedades deseables. Por lo tanto, el operador puede usarse como unión o como intersección en los casos extremos, y el ratio de compensación puede controlarse variando el grado de borrosidad f.

Buczak y Uhrig [3] describen un AG para resolver el mismo problema con aplicación a la fusión de datos, siendo la representación de soluciones al problema y el mecanismo de satisfación de restricciones las principales diferencias con respecto al AE que aquí describimos.

Más adelante se describen las principales características del AE propuesto. Estas características son una representación de soluciones al problema, satisfacción de restricciones, mecanismos para crear una población inicial de soluciones, la función de evaluación, operadores de variación y parámetros usados. Además, se hacen algunos comentarios al diseño.

Para diseñar el AE hemos tenido en cuenta la siguiente idea: parece (Michalevicz [10]) que una representación natural de una solución potencial para un problema dado junto con una familia de operadores de variación aplicables podría ser bastante útil en la aproximación de soluciones a muchos problemas. Además, las restricciones no triviales pueden implementarse incorporando conocimiento del problema específico en el AE.

2.1 REPRESENTACIÓN

Un individuo W de la población se representa como una colección de p componentes, es decir, $W = \{w_1,\ldots,w_p\}$, donde $w_j \in [0,1]$, $j = 1,\ldots,p$, es el peso asociado con la variable de entrada x_j.

2.2 SATISFACCIÓN DE RESTRICCIONES

Todos los mecanismos para crear un nuevo individuo $W = \{w_1,\ldots,w_p\}$ en el proceso evolutivo, es decir los procedimientos de inicialización y los operadores de variación, aseguran que se satisfacen las restricciones $\sum_{j=1}^{p} w_j = 1, 0 \le w_j \le 1, j = 1,\ldots,p.$

2.3 POBLACIÓN INICIAL

El siguiente procedimiento *población_inicial* obtiene una población completa de *tampob* individuos que satisfacen las restricciones impuestas. Consideramos dos procedimientos de inicialización. El procedimiento *pesos1* genera una colección de pesos $W = \{w_1,\ldots,w_q\}$ tales que $w_l \in [0,1]$, $l = 1,\ldots,q$, y $\sum_{l=1}^{q} w_l = val$, con $0 \le val \le 1$. El procedimiento *pesos2* es una modificación del procedimiento pesos1 para fijar un ratio aleatorio de pesos iguales a cero. Nótese que para $q = p$ y $val = 1$, ambos procedimientos pesos1 y pesos2 generan una solución factible para el problema. Estos procedimientos se usan con igual probabilidad para obtener la población inicial.

procedimiento población_inicial;

entrada: tamaño población *tampob*; número de variables p;

salida: población $POB = \{W_1 = \{w_1^1,\ldots,w_p^1\},\ldots,W_{tampob} = \{w_1^{tampob},\ldots,w_p^{tampob}\}\}$ de *tampob* individuos tales que $w_j^i \in [0,1]$, $j = 1,\ldots,p$, $i = 1,\ldots,tampob$, y $\sum_{j=1}^{p} w_j^i = 1$, $i = 1,\ldots,tampob$;

1. $i \leftarrow 1$;

2. *aleat* ← valor real aleatorio ∈ [0, 1];

3. Si *aleat* ≤ 0.5 entonces *pesos1*(*entrada* : $p, 1$; *salida* : W)
   sino *pesos2*(*entrada* : $p, 1$; *salida* : W)

4. $W_i \leftarrow W$;

5. Si $i < tampob$ entonces
   $i \leftarrow i + 1$;
   Ir al paso 2;

procedimiento pesos1;

entrada: número entero q, con $1 \leq q \leq p$; valor real val, con $0 \leq val \leq 1$;

salida: colección $W = \{w_1, \ldots, w_q\}$ tal que $w_l \in [0,1]$, $l = 1, \ldots, q$, y $\sum_{l=1}^{q} w_l = val$;

1. $w_l \leftarrow$ valor real aleatorio $\in [0,1]$, para $l = 1, \ldots, q$;

2. $w_l \leftarrow val \cdot w_l / \sum_{i=1}^{q} w_i$, para $l = 1, \ldots, q$;

procedimiento pesos2;

entrada: número entero q, con $1 \leq q \leq p$; valor real val, con $0 \leq val \leq 1$;

salida: colección $W = \{w_1, \ldots, w_q\}$ tal que $w_l \in [0,1]$, $l = 1, \ldots, q$, y $\sum_{l=1}^{q} w_l = val$;

1. Seleccionar aleatoriamente $K = \{k_1, \ldots, k_r\} \subseteq \{1, \ldots, q\}$ tal que $1 \leq r \leq q$;

2. $M \leftarrow \{1, \ldots, q\} - K$;

3. $w_l \leftarrow 0, l \in M$;

4. *pesos1*(*entrada* : r, val; *salida* : V);

5. $w_{k_l} \leftarrow v_l, l = 1, \ldots, r$;

2.4 FUNCIÓN DE EVALUACIÓN

La función de evaluación de los individuos $W_i = \{w_1^i, \ldots, w_p^i\}$, $i = 1, \ldots, tampob$, es claramente determinada por la función objetivo del problema:

$$eval(W_i) = \sqrt{\frac{\sum_{t=1}^{n} \left(y_t - \left(\sum_{j=1}^{p} w_j^i x_j^{tf} \right)^{1/f} \right)^2}{n}}, \quad i = 1, \ldots, tampob \qquad (2)$$

2.5 MECANISMO DE SELECCIÓN Y REEMPLAZO GENERACIONAL

Proponemos usar la *selección por torneo* [7]. En este método, se prueban un grupo de n_{torneo} individuos de la población y se elige para la reproducción al individuo con mejor aptitud. Se aplican operadores de variación a los individuos seleccionados y la descendencia se copia a la siguiente población. Este proceso se repite hasta que se genera completamente la nueva población (*reemplazo generacional completo*). Además se usa la *estrategia elitista*, que siempre siempre copia los mejores miembros de una población a la siguiente.

2.6 OPERADORES DE VARIACIÓN

Hemos considerado cinco operadores de variación (tres mutaciones, cruce e inversión). El operador *mutación1* realiza un cambio mínimo, intercambiando dos elementos aleatorios de los padres, mientras que los operadores *mutación2* y *mutación3* usan los procedimientos *pesos1* y *pesos2* respectivamente para producir un cambio en una parte arbitraria de los padres. El operador *cruce_aritmético* produce dos descendientes mediante la combinación lineal convexa de los padres. Con el operador *inversión*, un individuo $W = \{w_1, \ldots, w_p\}$ es invertido produciendo otro $W' = \{w_p, \ldots, w_1\}$.

procedimiento mutación1;

entrada: Padre $W = \{w_1, \ldots, w_p\}$;

salida: Descendencia $W' = \{w'_1, \ldots, w'_p\}$;

1. Seleccionar aleatoriamente $K = \{k_1, k_2\} \subseteq \{1, \ldots, p\}$;

2. $M \leftarrow \{1, \ldots, p\} - K$;

3. $w'_l \leftarrow w_l, l \in M$;

4. $w'_{k_1} \leftarrow w_{k_2}$;
   $w'_{k_2} \leftarrow w_{k_1}$;

procedimiento mutación2;

entrada: Padre $W = \{w_1, \ldots, w_p\}$;

salida: Descendencia $W' = \{w'_1, \ldots, w'_p\}$;

1. Seleccionar aleatoriamente $K = \{k_1, \ldots, k_r\} \subseteq \{1, \ldots, p\}$ tal que $1 < r \leq p$;
2. $M \leftarrow \{1, \ldots, p\} - K$;
3. $w'_l \leftarrow w_l$, $l \in M$;
4. $val \leftarrow \sum_{l=1}^{r} w_{k_l}$;
5. $pesos1(entrada : r, val; salida : V)$;
6. $w'_{k_l} \leftarrow v_l$, $l = 1, \ldots, r$;

procedimiento mutación3;

entrada: Padre $W = \{w_1, \ldots, w_p\}$;

salida: Descendencia $W' = \{w'_1, \ldots, w'_p\}$;

1. Seleccionar aleatoriamente $K = \{k_1, \ldots, k_r\} \subseteq \{1, \ldots, p\}$ tal que $1 < r \leq p$;
2. Set $M = \{1, \ldots, p\} - K$;
3. $w'_l \leftarrow w_l$, $l \in M$;
4. $val \leftarrow \sum_{l=1}^{r} w_{k_l}$;
5. $pesos2(entrada : r, val; salida : V)$;
6. $w'_{k_l} \leftarrow v_l$, $l = 1, \ldots, r$;

procedimiento cruce_aritmético;

entrada: Padres $W_1 = \{w_1^1, \ldots, w_p^1\}$ y $W_2 = \{w_1^2, \ldots, w_p^2\}$;

salida: Descendencia $W'_1 = \{w'^1_1, \ldots, w'^1_p\}$ y $W'_2 = \{w'^2_1, \ldots, w'^2_p\}$;

1. $c_1 \leftarrow$ valor real aleatorio $\in [0, 1]$;
   $c_2 \leftarrow 1 - c_1$;
2. $w'^1_j \leftarrow c_1 \cdot w_j^1 + c_2 \cdot w_j^2$, $j = 1, \ldots, p$;
   $w'^2_j \leftarrow c_2 \cdot w_j^1 + c_1 \cdot w_j^2$, $j = 1, \ldots, p$;

procedimiento inversión;

entrada: Padre $W = \{w_1, \ldots, w_p\}$;

salida: Descendencia $W' = \{w'_1, \ldots, w'_p\}$;

1. $w'_{p-l+1} \leftarrow w_l$, $l = 1, \ldots, p$;

2.7 PARÁMETROS

Los parámetros a considerar por parte del AE son el número máximo de generaciones (*maxgen*), el tamaño de la población (*tampob*), el número de miembros usados en la selección por torneo (n_{torneo}), la probabilidad de cruce (p_c), la probabilidad de mutación (p_m), y la probabilidad de inversión (p_i). Nótese que la probabilidad p_m corresponde a la suma de las probabilidades de todos los operadores de mutación descritos. La tabla 1 muestra los valores de los parámetros para los cuales se han obtenido buenos resultados en el proceso de experimentación.

Tabla 1: Valores de los parámetros del AE.

PARÁMETRO	VALOR
maxgen	1000
tampob	20
n_{torneo}	6
p_c	0.6
p_m	0.6
p_i	0.1

3 EJEMPLOS NUMÉRICOS

Para validar el AE hemos considerado un test estándar propuesto en Dyckhoff y Pedrycz (1984) y utilizado también en Buczak y Uhrig (1996). El conjunto de datos consiste en una colección de 2 variables de entrada x_1 y x_2 y una variable de salida. Para comprobar la bondad del AE, hemos añadido a los datos de entrada 6 nuevas variables x_3, x_4, x_5, x_6, x_7 y x_8 formadas a partir las variables x_1 y x_2 añadiéndoles ruido uniforme en un 10% (a las variables x_3 y x_4), en un 40% (a las variables x_5 y x_6), y en un 100% (a las variables x_7 y x_8). La tabla 2 muestra los pesos y fitness obtenidos considerando solamente las 2 variables de entrada del problema, para 3 valores diferentes de f (con el objetivo de poder comparar con los ejemplos de la literatura). La tabla 3 muestra los pesos y fitness obtenidos teniendo en cuenta las 2 variables de entrada originales mas las 6 variables ficticias de ruido. Los resultados muestran como el AE propuesto detecta las variables que realmente influyen en la variable de salida, excluyendo aquellas ficticias introducidas artificialmente.

4 CONCLUSIONES

Las técnicas Soft Computing para Análisis de Datos como selección o identificación de variables y descubrimiento de modelo, son una poderosa alternativa a las técnicas clásicas como las usadas hasta hoy en

Tabla 2: Resultados considerando únicamente la muestra.

	$f = 0.39$	$f = 1.25$	$f = 2$
w_1	0.447	0.405	0.316
w_2	0.553	0.595	0.684
fitness	0.053	0.114	0.161

Tabla 3: Resultados considerando la muestra con ruido.

	$f = 0.39$	$f = 1.25$	$f = 2$
w_1	0.420	0.406	0.318
w_2	0.486	0.518	0.682
w_3	0.024	0.0	0.0
w_4	0.015	0.076	0.0
w_5	0.002	0.0	0.0
w_6	0.012	0.0	0.0
w_7	0.019	0.0	0.0
w_8	0.022	0.0	0.0
fitness	0.052	0.113	0.161

la comunidad de DC. En este documento intentamos mostrar cómo pueden ser usadas estas técnicas en el preprocesamiento de datos, aunque son relevantes a los diversos pasos del proceso de DC, como la minería de datos o la selección de datos [12].

aplicarse con éxito en el contexto del DC y la Minería de Datos, dado que tienen dos características que los hacen muy adecuados:

- la primera es la imprecisión subyacente de los modelos o patrones que queremos encontrar, teniendo en cuenta que hemos asumido tratar con datos incompletos y con incertidumbre.
- la segunda es el poder computacional que aportan los algoritmos difusos y evolutivos, que es especialmente importante en el contexto de grandes bases de datos.

Los resultados obtenidos hasta ahora con la aplicación de este algoritmo, aunque no definitivos, muestran un buen rendimiento y su uso puede ser de gran interés para los encargados de la toma de decisiones en un entorno de aprendizaje supervisado.

Agradecimientos

Los autores agradecen a la Comisión Interministerial de Ciencia y Tecnología (CICYT) por el apoyo dado a la realización de este trabajo a través del proyecto con referencia TIC97-1343-C02-02, así como al Instituto de Fomento de la Región de Murcia por su financiación a través del Programa Séneca.

Referencias

[1] Biethahn, J., Nissen, V. (eds), *Evolutionary Algorithms in Management Applications.* Springer-Verlag Berlin Heidelberg, 1995.

[2] Bonissone, P., "Soft Computing: The Convergence of Emerging Reasoning Techniques", Journal of Soft Computing, vol. 1, n. 1, 1997.

[3] Buczak, A.L., Uhrig, R.E., "Hybrid Fuzzy-Genetic Technique for Multisensor Fusion", Information Sciences, vol. 93, pp. 265-281, 1996.

[4] Delgado, M., Gómez Skarmeta, A., Martín, F., "Some Methods to Model Fuzzy Systems for Inference Purposes", Int. Journal of Approximate Reasoning, vol. 16, n. 3-4, pp. 377-391, 1997.

[5] Dyckhoff, H., Pedrycz, W., "Generalized means of model of compensative connectives", Fuzzy Sets and Systems, vol. 14, pp. 143-154, 1984.

[6] Fayyad, U., "Data Mining and Knowledge Discovery: Making Sense Out of Data", IEEE Expert, pp. 20-25, 1996.

[7] Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

[8] Gómez-Skarmeta, A., Jiménez, F., "Generating and Tuning Fuzzy Rules using Hybrid Systems", Proceeding of the FUZZ/IEEE97, pp. 247-252, Barcelona, Spain, 1997.

[9] Kroll, A., "Identification of functional models using multidimensional reference fuzzy sets", Fuzzy Sets and Systems, vol. 80, pp. 149-158, 1996.

[10] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs.* Springer Verlag, 1992.

[11] Sugeno, M., Yasukawa, T., "A Fuzzy-Logic-Based Approach to Qualitative Modeling", IEEE Trans. on Fuzzy Systems, vol.1, n. 1, pp. 7-31, 1993.

[12] Vila, M.A., Delgado, M., Gómez Skarmeta, A.F., "On the Use of Probability and Possibility Measures in Fuzzy Clustering", Proceeding of the FUZZ/IEEE97, pp. 143-148, Barcelona, Spain, 1997.

[13] Yager, R.R., "On Ordered Weighted Averaging Aggregation Operators in Multi-Criteria Decision Making", IEEE Transactions on Systems, Man and Cybernetics, vol. 18, pp. 183-190, 1988.

[14] Zimmermann, H.J., *Fuzzy Sets, Decision Making and Expert Systems.* Kluwer Academic, 1987.

Circuito Experimental de Inferencia de Lukasiewicz

Luis de Salvador, Pedro E. Bernad, Julio Gutiérrez
Lab. de Hw. y Control
Instituto Nacional de Técnica Aeroespacial
Ctra. Ajalvir km. 4 - Torrejón de Ardoz
28850 - Madrid - España

Resumen

El operador de Lukasiewicz es uno de los más interesante para implementar en un procesador borroso de propósito general. Su diseño hardware ofrece ciertas dificultades cuando se quiere obtener un circuito de elevado rendimiento y reducidas dimensiones. En esta comunicación se ofrece una solución basada en arquitecturas dígito-serie y notación signo-dígito.

Palabras Clave: Hardware, fuzzy, FPGA, Lukasiewicz, signo-dígito, dígito-serie.

1 INTRODUCCIÓN

Esta trabajo se enmarca dentro de un proyecto de la CICYT que tiene por objeto el desarrollo de un procesador de lógica borrosa lo más general posible. Para que un procesador borroso pueda ser de propósito general es necesario contemplar las operaciones Max-Min, sino también otras que sean de uso común en la inferencia sobre reglas borrosas [4] [6] [9].

En este marco de trabajo, se ha desarrollado un circuito para evaluar la inferencia de Lukasiewickz de forma eficiente. La implementación de este prototipo tiene como propósito investigar la adecuación de este tipo de inferencia para un sistema de procesamiento de elevado rendimiento de información borrosa. Por lo tanto, la arquitectura diseñada está orientada a conseguir un procesamiento muy rápido de la información.

Adicionalmente, se persigue construir un circuito de pequeñas dimensiones y flexible en alguno de sus parámetros, adecuado para su implementación en FPGA. La flexibilidad de este circuito se va a concretar en no fijar *a priori* el número de dígitos empleados en la codificación de los valores de pertenencia.

1 OPERADOR A IMPLEMENTAR

El algoritmo de Lukasiewicz para la realización de inferencias borrosas se basa en la aplicación de los operadores del mismo nombre:

La función T o t-norma: (1)

$$W(x, y) = \text{Max}(0, x + y - 1)$$

La función S o t-conorma: (2)

$$W^*(x, y) = \text{Min}(1, x + y)$$

El par (T, S) de Lukasiewicz es un conjunto de operaciones dual:

$$S(x, y) = 1 - T(1 - x, 1 - y) \quad (3)$$

(cumpliendo las leyes de Morgan)

1.1 EVALUACIÓN DE LA INFERENCIA

El sistema de reglas se considera formado por reglas del tipo MISO:

$$r_j: \text{SI } x_1 \text{ es } B_{j1} \wedge x_2 \text{ es } B_{j2} \wedge \ldots. x_n \text{ es } B_{jn} \quad (4)$$
$$\text{ENTONCES y es } D_j$$

Cada regla se evalúa para producir un conjunto borroso, cuya función de pertenencia es:

$$D_j(y) = T\,(B_{j1}(x_1), B_{j2}(x_2), \ldots., B_{jn}(x_n)) \quad (5)$$

Cada uno de los valores $B_{ji}(x_i)$ se representa por un valor de verdad sobre un conjunto borroso, no por un conjunto borroso.

El operador de agregación que combina todas las salidas es el operador de unión, implementado con la t-conorma:

$$D'(y) = S\,(D_j(y)); \text{ para } j = (1,m) \quad (6)$$

1.2 APLICACIÓN DE LA T-NORMA DE LUKASIEWICZ PARA UN CONJUNTO DE VALORES DE PERTENENCIA

Para la aplicación de la t-norma sobre un conjunto de valores de pertenencia es necesario realizar de forma iterativa la operación Max(0, x+y-1) sobre un conjunto de valores {a, b, c, ... n}:

$$\text{Max}\,(0, \text{Max}(\,\text{Max}(\,\ldots\,\text{Max}(0, m+n-1)\ldots.\,) + \quad (7)$$
$$b\text{-}1\,\,) + a\text{ -}1)$$

La ejecución de la la t-norma de L. implica la realización de tres operaciones:

- Sumar x e y
- Restar 1
- Comparar con 0

Para simplificar el proceso, se hace uso del principio de dualidad (3) de la t-norma y de la t-conorma de L. Para ello, en vez de computar la t-norma, se evalua la t-conorma con los valores complementados. Esto significa utilizar valores complementados en la evaluación del operador: Min(1, 1-x+1-y).

$$1\text{- Min}\,(0, \text{Min}(\,\text{Min}(\,\ldots\,\text{Min}(0, 1\text{-}m+1\text{-}n)\ldots.\,) \quad (8)$$
$$+\,1\text{-}b\,) + 1\text{-}a\,)$$

La ejecución de esta operación implica sólo dos pasos:

- Sumar x e y.
- Comparar con 1.

2 MODELO DE PROCESO

El objetivo del circuito es realizar la evaluación de la inferencia de la forma más rápida posible y utilizando el mínimo número de puertas. Una característica adicional es que el mismo circuito se pueda emplear para valores de pertenencia codificados con un número arbitrario de dígitos.

Para ello se va a plantear el desarrollo de un sistema segmentado o pipeline que realice el proceso empleando técnicas dígito-serie [2], [3], [7], [8], [10]. La unidad elemental que se emplea como modulo de proceso bit-slice de evaluación del operador W^* es la siguiente: (Figura 1).

Figura 1 Ejemplo de operador dígito a dígito

Donde A y B son palabras que codifican el valor de verdad complementado y R es el resultado de la t-norma complementada. Los dígitos de cada palabra se evalúan de forma sucesiva. Para la evaluación sobre un conjunto de palabras la estructura de proceso es la siguiente:

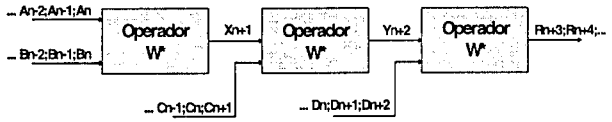


Figura 2 Utilización de operadores sobre dígito para procesar distintas palabras

Para ejecutar el operador W^* hay que ejecutar dos operaciones: suma y comparación. Pero hay que tener en cuenta que la comparación no sólo significa obtener un resultado de comparación, sino de selección de un valor u otro en función de la comparación de ambos. Es una comparación de dos pasos:

- Evaluación de la comparación.
- Selección del valor adecuado en función de la comparación.

Para comparar dígito a dígito dos palabras e ir proporcionando el resultando final sin esperar a la comparación de todos los dígitos es necesario realizar la comparación comenzando por el más significativo. En cambio, la operación de suma se puede realizar ofreciendo los resultados finales según se va operando dígito a dígito sólo si se empieza a procesar desde el dígito menos significativo. Para poder compatibilizar ambas operaciones en un sistema de proceso dígito-serie es necesario emplear una notación alternativa al sistema binario: notación signo-dígito [1]. Esta notación permitirá realizar el proceso de comparación y suma comenzando por el dígito más significativo.

2.1 REPRESENTACIÓN Y PROCESO DE VALORES DE PERTENENCIA

En notación signo-dígito [1], si se tiene una base de codificación r y dos valores de pertenencia (x,y) de n dígitos normalizados entre $[0, 2r^n-2]$, estos pueden representarse como $\{x- r^n+1, y- r^n+1\}$ para permanecer en los intervalos de codificación. La complementación se convierte en un cambio de signo (inversión del bit más significativo) de cada dígito del código (propiedad de la notación signo-dígito).

El proceso de suma de dos valores de pertenencia es el siguiente:

$$x + y \Rightarrow x- r^n+1+ y- r^n+1 = x + y - 2r^n-2 \equiv x + y - r^n+1 \quad (9)$$

Hay que retocar el resultado final para que el valor obtenido sea realmente el que se busca, ya que por defecto el resultado se encuentra modificado por un factor de (r^n+1). Por lo tanto, es necesario realizar la normalización de la salida del sumador. La normalización se realiza empleando otro sumador serie con una de sus entradas constante y con valor $r-1$ (para cada dígito de la palabra).

De esta forma el circuito que ejecuta el operador W^* se configura como una sumador en dos pasos o etapas, una etapa que realiza la suma en sí y otra etapa que realiza la normalización del resultado intermedio. Este sumador, tal y como aquí se plantea tiene los siguientes problemas:

- Después de cada suma, el resultado emplea más dígitos que los realmente necesarios para codificar la palabra. Exactamente un dígito más. Debido a que se emplean dos sumadores, el sistema intenta ocupar dos dígitos más.
- El operador incluye la comparación con el valor 1. Esta comparación se va a traducir en evitar el overflow del resultado cuando se realiza la operación de suma y mantener los valores saturados.
- Es necesario encadenar varios de estos operadores.

Los dos primeros problemas están relacionados: es necesario evitar la propagación por encima del número de dígitos necesarios para codificar la palabra y por encima del valor lógico 1, que es una sucesión de dígitos con el máximo valor.

Esto implica la necesidad el propagar hacia atrás el acarreo de los dígitos más significativos a los dígitos menos significativos cuando los primeros se encuentran saturados, en vez de añadir un nuevo dígito. Esta estrategia permite mantener el valor máximo de salida como el 1 lógico.

Evitar dicha propagación se puede conseguir en la segunda etapa del sumador (la de normalización) ya que la primera etapa precisa de un dígito adicional para codificar el resultado ya que este se encuentra en el intervalo $[-2r+2, 2r-2]$. La nueva configuración de los módulos de sumador serie es la siguiente:

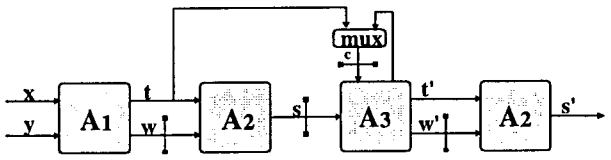


Figura 3 Bloque L-operador on-line

En esta figura aparece un nuevo módulo etiquetado como A_3. Tiene las señales { *t', w', s'* }, que tienen la misma definición que las señales no marcadas, y la señal c, que se emplea para propagar hacia atrás la saturación de los dígitos más significativos. Se define como sigue:

- c = 2; Si no hay saturación.
- c = 0; Si hay saturación pero no hay propagación hacia atrás.
- c = 1; Si hay saturación positiva y propagación hacia atrás de 1.
- c = -1; Si hay saturación negativa y propagación hacia atrás de -1.

2.2 MODELO DE PROCESO CON BASE 3

La implementación del L-operador en On-line [5], [11] depende de la base empleada para codificar los valores de pertenencia. En este caso se va a emplear una codificación en base 3. Las formulas que marcan dicha implementación se muestran a continuación.

Modulo A_1 t = 0 <= -2 < x+y < 2 t = 1 <= x+y > = 2 t = -1 <= x+y <= -2 w = x + y - 3 *t	(10)
Modulo A_2 s = w + t	(11)
Modulo A_3 If c = 2 -- sin saturación de los dígitos más significativos If -2 < x+2 < 2 ≡ 4 < x < 0 ≡ -2 ≤ x < 0 t = 0 c = 2 w = x + 2 If x ≥ 0 t = 1 c = 2 w = x – 1 If c = 0 -- saturación de los dígitos más significativos t = 0 If -4 < x < 0 ≡ -2 ≤ x < 0 c = 2 w = x + 2 else If x = 0 c = 0 w = 2 else c = 1 w = 2 If c = 1 -- saturación y propagación hacia atrás positiva t = 0 c = 1 w = 2 If c = -1 -- saturación y propagación hacia atrás negativa t = 0 If x ≤ -1 c = -1 w = -2 If -2 < x -1 < 2 ≡ -1 < x < 2 ≡ - 1 < x c = 2 w = x –1	(12)

2.3 EL VALOR INICIAL

Las fórmulas anteriores no reflejan el problema del valor inicial. Debido a que hay propagación hacia adelante de los resultados intermedios, es necesario guardar un ciclo de reloj de distancia entre la entrada de dos palabras consecutivas. Hay que tener en cuenta que, la suma de los dos dígitos más significativos de la entrada A_n y B_n producen dos resultados. El primero es R_n y el segundo es la posible propagación adelantada R_{n+1}.

Esto significa un dígito más en el resultado, lo que se pretende evitar. Por lo tanto, este dígito adicional ha de volver a sumarse al penúltimo dígito: $R_n + R_{n+1}$. Esta suma ha de realizarse únicamente en la operación del primer dígito. De ahí que en la Figura 3 se represente una alimentación de la señal **t** hacia la señal **c**. De esta forma, cualquier propagación que se produzca del dígito más significativo hacia adelante se anula en la segunda etapa del L-operador.

3 DISEÑO DE LOS MÓDULOS

El circuito que implementa el L-operador tiene la siguiente forma:

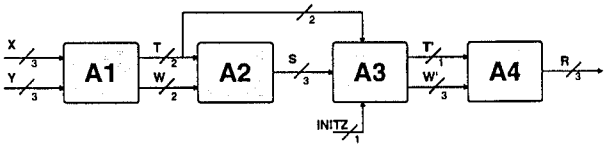


Figura 4 Módulos del circuito L-operador

Todos los módulos tienen entradas adicionales de **resetz** (reset a nivel bajo) y de **clk**. Las diferencias entre el circuito que se muestra en la Figura 4 con el circuito teórico de la Figura 3 son las siguientes:

- En esta última figura los registros se consideran incluidos en los módulos de donde procede la señal en las que estaban incluidos.
- Se ha sustituido en último módulo denominado A2 por un nuevo módulo denominado A4. La diferencia

funcional entre ambos no existe. El módulo A4 se diferencia del A2 en que se realiza en el diseño una optimización diferente para los casos especiales que no se contemplan en el A2.

Todos los módulos han sido diseñados en VHDL. Los siguientes apartados muestran los circuitos sintetizados empleando Aurora de Viewlogic.

3.1.1 Módulo A1

A continuación se muestra el esquemático sintetizado del módulo A1.

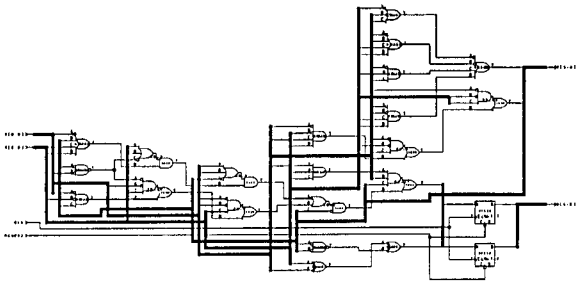


Figura 5 Esquemático módulo A1

3.1.2 Módulo A2

A continuación se muestra el esquemático sintetizado del módulo A2.

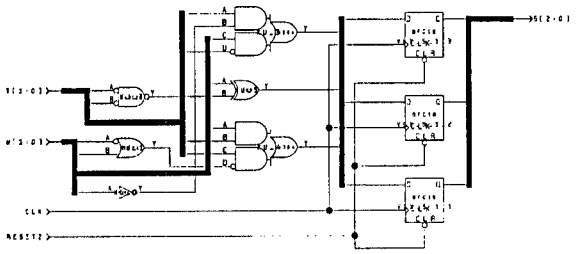


Figura 6 Esquemático módulo A2

3.1.3 Módulo A3

El esquemático sintetizado del módulo A3 es el módulo más complejo ya que es el que realiza la normalización y evita la propagación del resultado en un dígito adicional previniendo el overflow.

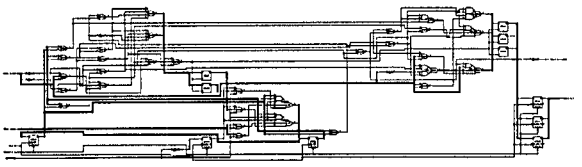


Figura 7 Esquemático módulo A3

3.1.4 Módulo A4

El módulo A4 ajusta el acarreo producido por la normalización.

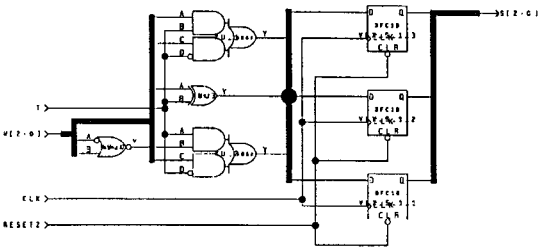


Figura 8 Esquemático módulo A4

3.2 PRUEBAS DE INTEGRACIÓN

Los módulos A1 al A4 se integran para formar una unidad de proceso que permite el proceso de cinco valores de inferencia codificados con un número de dígitos variable. El módulo que integra todos los circuitos necesarios para dicho proceso de inferencia sobre cinco variables de entrada se ha denominado GLMG. El plan de pruebas de integración del módulo GLMG se realiza introduciendo distintos juegos de 5 palabras (valores de inferencia). Las palabras están nombradas de la 'a' a la 'e'. En esta prueba tienen 6 dígitos más un dígito adicional de separación entre palabras introducidas. Este dígito ha de encontrarse al final de la palabra. Se introducen 3 juegos de test que son los siguientes:

Tabla 1 Juegos de Test

Juego 1	**a**	0	-2	0	-2	0	-2	-
	b	0	0	0	0	0	0	-
	c	-2	-2	-2	-2	-2	-2	-
	d	0	0	0	0	0	0	-
	e	-2	0	-2	0	-2	0	-
Juego 2	**a**	-2	-2	-2	-2	-2	-2	-
	b	-2	-2	-2	-2	-2	-2	-
	c	-1	-1	-1	-1	-1	-1	-
	d	-1	-1	-1	-1	-1	-1	-
	e	-2	-2	-2	-2	-2	-2	-
Juego 3	**a**	0	0	-2	-2	-2	-2	-
	b	-2	-2	-2	-2	0	0	-
	c	-1	-1	-1	-1	-1	-1	-
	d	-1	-1	-1	-1	-1	-1	-
	e	-2	-2	0	0	-2	-2	-

El resultado que se ha de obtener de los tres juegos introducidos es:

Tabla 2 Resultado previsto del juego de test

Juego 1	2	2	2	2	2	2	-
Juego 2	0	0	0	0	0	0	-
Juego 3	2	2	2	2	2	2	-

La pruebas de integración del módulo GLMG se muestran en los cronogramas que aparecen a continuación. Esta simulación prueba que el funcionamiento del módulo es el correcto:

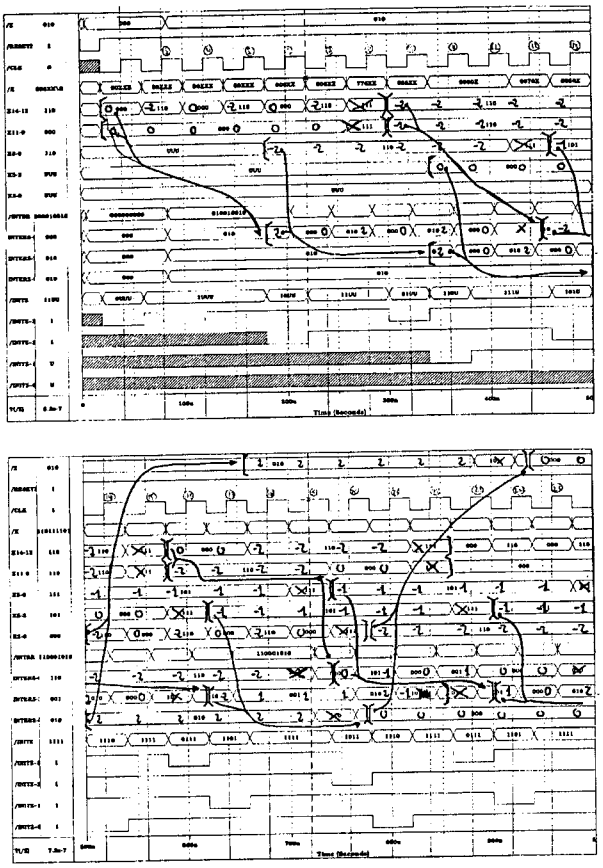


Figura 9 Cronogramas de prueba del módulo GLMG

4 SÍNTESIS DEL CIRCUITO

La síntesis del circuito se ha realizado sobre FPGA's de ACTEL familia ACT3. Concretamente sobre una A1460A de 6.000 puertas equivalentes.

El dispositivo programado no llega a ocupar un 20% de la capacidad de la pastilla. (80 registros y 200 módulos lógicos). De los retardos entre registros y la información de setup/hold entre registros se determina que el periodo mínimo de reloj será de 40 ns, lo que garantiza una velocidad de reloj de 25 MHz.

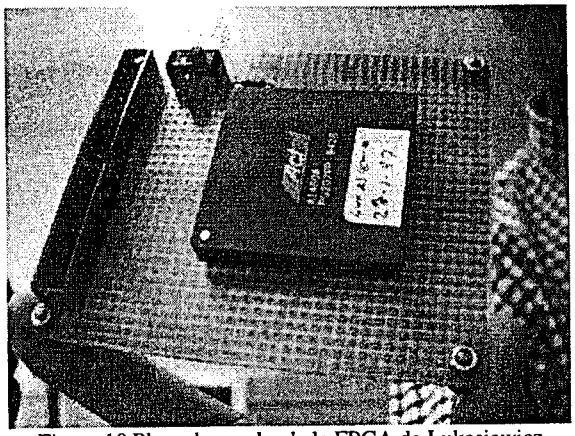


Figura 10 Placa de prueba de la FPGA de Lukasiewicz

5 CONCLUSIONES

El circuito etiquetado como Luka00 permite realizar la inferencia por el método de Lukasiewicz con cinco valores de inferencia de entrada. Los valores de inferencia pueden estar codificados con un número arbitrario de dígitos. La velocidad del circuito implementado en una FPGA de ACTEL 1460A estándar (velocidad lenta) es de 25 MHz y tiene un tamaño aproximado entre las 1000 y las 1500 puertas lógicas.

La velocidad de inferencia es igual a la velocidad del circuito dividida entre el número de dígitos empleados para codificar los valores de pertenencia más uno. Como los dígitos en base 3 notación sígno-dígito equivalen a 2,3 bits, la velocidad del circuito es aproximadamente igual a: $(F_{reloj} * 2.3) / (bits\_valores\_pertenencia+1)$.

En las simulaciones presentadas, se han empleado valores con 6 dígitos, lo que equivale a valores de 13 bits. En este caso la velocidad de inferencia es igual a 4 millones de inferencias por segundo.

La latencia depende del número de entradas y no del número de dígitos por entrada. Es igual a 4 * (número de variables-1). En este caso, la latencia es de 16 ciclos de reloj (640 ns).

El rendimiento se puede mejorar utilizando una arquitectura sistólica, en vez de utilizar un circuito serie. Esta es una posibilidad que deja abierta el diseño presentado. Se ganaría velocidad pero, por otro lado, el tamaño del circuito sería mucho mayor, para el caso que nos ocupa incluso hasta seis veces mayor.

6 AGRADECIMIENTOS

Este trabajo ha podido realizarse en el marco del proyecto de la CICYT 96-1393-C06-2 titulado "Microprocesador Borroso de Altas Prestaciones".

También deseamos agradecer la inestimable ayuda de nuestro compañero David Madroño Baeza en las tareas de laboratorio.

7 BIBLIOGRAFÍA

[1] Avizienis A., "Signed-digit number representations for fast parallel arithmetic", *IRE Transactions on Electronic Computers*, pp-389-400, 1961

[2] de Salvador L., Gutiérrez J., "A multilevel systolic approach for fuzzy inference hardware", *IEEE Micro Magazine*, October, pp-61-71, 1995

[3] de Salvador L., Gutiérrez J., "Serial Architectures for Efficient Fuzzy Processing", *Fuzzy Hardware Architectures and Applications,* Kluwer Academic Publishers , pp-117-157 ISBN 0-7923-8029-0 – 1998

[4] Driankov D., Hellendoorn H., Reinfrank M., *An Introduction to Fuzzy Control*, Springer-Verlag, 1993

[5] Ercegovac M.D., Lang T. "On-line arithmetic: a design methodology and applications in digital signal processing", *VLSI Signal Processing III*, pp-252-263, Broderson R.W. & Moscowitz H.S. eds. IEEE Press, 1988

[6] Gunn D.J., Maguire L.P., McGinnity T.M., Hashim A.A., "The implementation of fuzzy reasoning using FPGA's", *Proc. IIZUKA'96*, pp-243-246, Iizuka, 1996

[7] Hartley R.I., Parhi K.K., *Digit-Serial Computation*, Kluwer Academic Publishers, 1995

[8] Hartley, R., Corbett P. "Digit-serial processing techniques", *IEEE Trans. on Circuits and Systems,* vol. 37, n° 6, pp-707-719, 1990

[9] Manzoul M.A., "Fuzzy inference on a systolic array", *Proc. of 18th Modelling and Simulation Conference*, pp-1103-1108, 1987

[10] Parhi K.K., "A systematic approach for design of digit-serial signal processing architectures", *IEEE Trans. on Circuits and Systems*, vol. 38, n° 4, pp-358-375, April 1991

[11] Trivedi K.S., Ercegovac M.D., "On-line algorithms for division and multiplication", *IEEE Trans. Computing*, vol. C-26, July 1977

Un Modelo de Selección para Problemas de Decisión con Múltiples Expertos e Información Linguística Multi-Granular

Francisco Herrera Triguero
Dpto. de Ciencias de la Computación e I.A.
E.T.S. de Ingeniería Informática
Universidad de Granada
18071 - Granada
e-mail: herrera@decsai.ugr.es

Enrique Herrera Viedma
Dpto. de Ciencias de la Computación e I.A.
E.T.S. de Ingeniería Informática
Universidad de Granada
18071 - Granada
e-mail: viedma@decsai.ugr.es

Luis Martínez López
Dpto. de Informática
Escuela Politécnica Superior
Universidad de Jaén
23071 - Jaén
e-mail: martin@ujaen.es

Resumen

En este trabajo estudiamos los problemas de decisión con múltiples expertos, asumiendo que éstos proporcionan sus preferencias sobre las alternativas mediante valores lingüísticos evaluados en conjuntos de etiquetas con distinta granularidad y/o semántica, esto es, mediante información lingüística multi-granular. En este contexto de decisión, se presenta un modelo de selección compuesto por dos pasos con objeto de obtener el conjunto solución de alternativas. Primero, se realiza la fusión de la información lingüística multi-granular expresada por los expertos, obteniéndose las preferencias lingüísticas colectivas sobre las alternativas. Y segundo, se realiza la selección de las mejores alternativas a partir de los valores de preferencia colectivos.

Palabras clave: Decisión con multiples-expertos, información lingüística, multi-granularidad, grado de selección.

1 Introducción

En muchas actividades de decisión nos encontramos aspectos que no son fácilmente calificables mediante valores precisos, ya sea por su propia naturaleza (aspectos cualitativos : "belleza", "comfortabilidad", etc ...) o simplemente porque en ese momento no esta disponible o es muy costoso conseguir un valor exacto, por lo que un valor aproximado es suficiente. En tales circunstancias, es normal manejar y representar los aspectos cualitativos como términos lingüísticos mediante variables lingüísticas [12], es decir, variables cuyos valores no son números sino palabras o frases en un lenguaje natural o artificial. Este enfoque lingüístico ha sido utilizado por diversos autores para resolver problemas de decisión [4, 2, 5, 9, 11].

En el enfoque lingüístico es muy importante determinar la "*granularidad de la incertidumbre*", es decir, la cardinalidad del conjunto de términos lingüísticos utilizado para expresar la información. Según el grado de incertidumbre que un experto tiene al calificar un fenómeno, el conjunto de etiquetas elegido para expresar su opinión tendrá más o menos términos. Cuando existen distintos expertos con diferentes grados de incertidumbre sobre un fenómeno, pueden utilizar conjuntos de etiquetas con distinta granularidad y/o semántica para expresar sus preferencias.

En este trabajo, consideramos problemas de decisión con múltiples expertos que dan sus preferencias sobre un conjunto de alternativas usando valores lingüísticos evaluados sobre conjuntos de etiquetas con distinta granularidad y/o semántica, esto es, mediante *información lingüística multi-granular*. Presentamos un modelo de selección que obtiene el conjunto solución de alternativas siguiendo dos pasos:

1. *Fusión de la información lingüística multi-granular.* En esta fase, se obtiene una preferencia lingüística colectiva sobre cada alternativa mediante la fusión de las preferencias lingüísticas multi-granulares individuales dadas por los expertos sobre cada una de las alternativas. El esquema de fusión sigue las dos siguientes fases:
   (a) *Hacer uniforme la información lingüística multi-granular.* Consiste en expresar las preferencias lingüísticas multi-granulares individuales en un único conjunto básico de etiquetas (CBE). Cada valor lingüístico multi-granular suministrado por los expertos se representa como un conjunto difuso en el CBE.
   (b) *Cálculo de las preferencias lingüísticas colectivas.* Para cada alternativa se calcula la preferencia lingüística colectiva de acuerdo a las preferencias de todos los expertos.
2. *Selección de las mejores alternativas.* A partir de los valores colectivos calculamos una relación de preferencia difusa usando la Teoría de la Posibilidad aplicada a los conjuntos difusos definidos en CBE. Finalmente, sobre esta relación aplicamos una función de selección para obtener las mejores

alternativas.

El trabajo se estructura como sigue: en la Sección 2 presentamos el problema de decisión en contexto lingüístico; en la Sección 3 se muestra el modelo de selección; en la Sección 4 se da un ejemplo; y por último, apuntamos algunos comentarios finales.

2 El Problema de Decisión con Múltiples Expertos e Información Lingüística Multi-Granular

Consideramos un problema de decisión en el cuál tenemos un conjunto finito de alternativas $X = \{x_1, ..., x_n\}$ $(n \geq 2)$ que son calificadas según un conjunto finito de expertos $P = \{p_1, ..., p_m\}$ $(m \geq 2)$. Cada experto p_j da un vector de utilidad con un valor lingüístico de preferencia p^{ij} para cada alternativa x_i. Asumimos que cada experto, p_j, puede usar diferentes conjuntos de etiquetas $\{S_j\}$ para expresar sus preferencias. Por lo tanto, para cada experto p_j, el vector de utilidad se define como un subconjunto lingüístico de selección sobre el conjunto X de alternativas valorado lingüísticamente en S_j:

$$p_j \longrightarrow (p^{1j}, ..., p^{nj}) \;\; p^{ij} \in S_j$$

$$S_j = \{s_0^j, ..., s_{k_j}^j\} \; j \in \{1, ..., m\}$$

donde $k_j + 1$ es la granularidad de S_j.

Cada conjunto S_j es definido como un conjunto finito y totalmente ordenado de términos lingüísticos que representan un valor posible de una variable lingüística en el sentido usual [1].

La semántica de cada etiqueta está dada por números difusos definidos sobre el intervalo [0,1], descritos por funciones de pertenencia trapezoidales lineales, representadas por la tupla (x_0, x_1, x_2, x_3), donde los parámetros x_1, x_2 indican el intervalo en el que la función de pertenencia vale 1.0; y x_0, x_3 indican los límites izquierdo y derecho del soporte de la función de pertenencia. La etiqueta del centro representa una incertidumbre de "aproximadamente 0.5" y el resto de etiquetas está distribuído simétricamente a ambos lados de la misma [1].

3 Modelo de Selección

Aquí desarrollamos cada paso del modelo de selección para problemas de decisión con múltiples expertos e información lingüística multi-granular presentado en la introducción.

3.1 Fusión de la Información Lingüística Multi-granular

En este fase se obtienen las preferencias lingüísticas colectivas sobre las alternativas de acuerdo a las preferencias lingüísticas multi-granulares individuales expresadas por los expertos. La técnica de fusión de información lingüística multi-granular que nos permite hallar los valores colectivos se desarrolla en los dos siguientes pasos:

1. *Hacer uniforme la información lingüística multi-granular.*
2. *Cálculo de las preferencias lingüísticas colectivas.*

3.1.1 Hacer Uniforme la Información Lingüística Multi-Granular

Para poder manejar la información lingüística multi-granular la representamos uniformemente transformándola a un único conjunto de etiquetas, CBE, que notamos como S_T.

Antes de definir el proceso de conversión a S_T, hemos de seleccionar el CBE. Éste debe ser un conjunto de etiquetas lingüísticas que nos permita mantener el grado de incertidumbre asociado a cada experto y la capacidad de discriminación para expresar los valores de preferencia. Con vistas a cumplir este objetivo, buscamos los conjuntos de máxima granularidad en $\{S_j, \forall j\}$. Cuando existe un único conjunto de etiquetas con máxima granularidad, se elige como CBE. Si encontramos dos o más conjuntos de máxima granularidad , entonces el CBE es seleccionado dependiendo de la semántica de estos conjuntos de etiquetas:

1. Si todos los conjuntos de etiquetas de máxima granularidad tienen la misma semántica, entonces S_T es cualquiera de ellos.
2. Si existen algunos conjuntos de etiquetas con diferente semántica, entonces S_T será un conjunto de etiquetas especial con un número de términos superior al que una persona es capaz de discriminar o distinguir (como mucho 11 ó 13 [6]). Definimos un conjunto de etiquetas especial con 15 términos y la siguiente semántica (ver Figura 1).

s_0	(0, 0, .07)	s_1	(0, .07, .15)
s_2	(.07, .15, .22)	s_3	(.15, .22, .29)
s_4	(.22, .29, .36)	s_5	(.29, .36, .43)
s_6	(.36, .43, .5)	s_7	(.43, .5, .58)
s_8	(.5, .58, .65)	s_9	(.58, .65, .72)
s_{10}	(.65, .72, .79)	s_{11}	(.72, .79, .86)
s_{12}	(.79, .86, .93)	s_{13}	(.86, .93, 1)
s_{14}	(.93, 1, 1)		

Una vez seleccionado el CBE, S_T, definimos una función de transformación de información lingüística multi-granular, que expresa la información de cada valor lingüístico $p^{ij} \in S_j$ como un conjunto difuso sobre S_T.

Definición 1. *Sean* $A = \{l_0, \ldots, l_p\}$ *y* $S_T = \{c_0, \ldots, c_g\}$ *dos conjuntos de etiquetas, con* $g \geq$

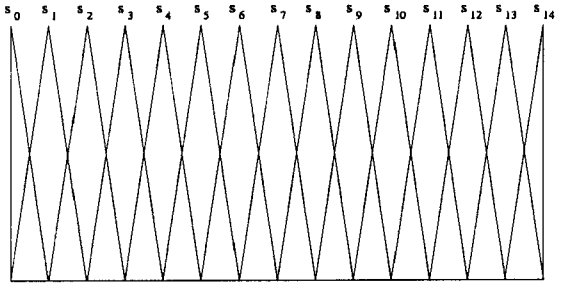


Figura 1: Conjunto de Etiquetas con 15 términos

p. Una función de transformación de información lingüística multigranular, τ_{AS_T}, se define como:

$$\tau_{AS_T} : A \longrightarrow F(S_T)$$

$$\tau_{AS_T}(l_i) = \{(c_k, \alpha_k^i) \ / k \in \{0, \ldots, g\}$$

$$\alpha_k^i = \max_y \min\{\mu_{l_i}(y), \mu_{c_k}(y)\}$$

donde $F(S_T)$ es el conjunto de conjuntos difusos definidos en S_T, $\mu_{l_i}(y)$ y $\mu_{c_k}(y)$ las funciones de pertenencia de las etiquetas l_i y c_k respectivamente.

El resultado de τ_{AS_T} para cualquier valor lingüístico de A es un conjunto difuso definido en términos del conjunto de etiquetas S_T.

Ejemplo 1. Sea $A = \{l_0, l_1, l_2, l_3, l_4\}$ y $S_T = \{c_0, c_1, c_2, c_3, c_4, c_5, c_6\}$ dos conjuntos de etiquetas, con 5 y 7 términos (Figura 2) y con las siguientes semánticas asociadas:

	A		S_T
l_0	(0, 0, .25)	c_0	(0, 0, .16)
l_1	(0, .25, .5)	c_1	(0, .16, .34)
l_2	(.25, .5, .75)	c_2	(.16, .34, .5)
l_3	(.5, .75, .1)	c_3	(.34, .5, .66)
l_4	(.75, 1, 1)	c_4	(.5, .66, .84)
		c_5	(.66, .84, 1)
		c_6	(.84, 1, 1)

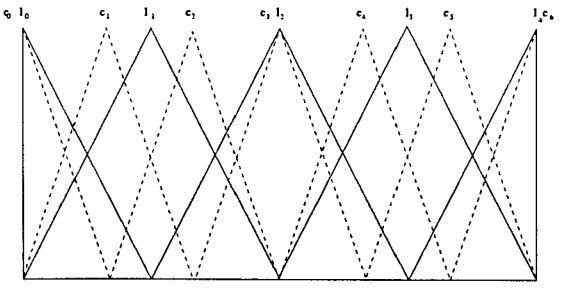


Figura 2: Conjuntos de etiquetas A y S_T

Veamos como expresar las etiquetas l_0 y l_1 en términos del conjunto S_T. Los conjuntos difusos asociados a l_0 y l_1 después de aplicar τ_{AS_T} son:

$$\begin{array}{lllll} \tau_{AS_T}(l_0) = & \{(c_0, 1), & (c_1, .58), & (c_2, .18), \\ (c_3, 0), & (c_4, 0), & (c_5, 0), & (c_6, 0)\} \end{array}$$

$$\begin{array}{lllll} \tau_{AS_T}(l_1) = & \{(c_0, .39), & (c_1, .85), & (c_2, .85), \\ (c_3, .39), & (c_4, 0), & (c_5, 0), & (c_6, 0)\} \end{array}$$

Por tanto, para hacer uniforme la información lingüística multi-granular seleccionamos S_T y aplicamos el conjunto de funciones de transformación de información lingüística multigranular $\{\tau_{S_j S_T}, \ j \in \{1, ..., m\}\}$ a los vectores de utilidad suministrados por los distintos expertos. Entonces cada valor lingüístico p^{ij} se representa mediante un conjunto difuso definido en $S_T = \{c_0, ..., c_g\}$, caracterizado por la siguiente expresión:

$$\tau_{S_j S_T}(p^{ij}) = \{(c_0, \alpha_0^{ij}), \ldots, (c_g, \alpha_g^{ij})\}$$

De este modo, el vector de utilidad de cada experto p_j se representa como un vector de conjuntos difusos en S_T:

$$(\tau_{S_j S_T}(p^{1j}), \ldots, \tau_{S_j S_T}(p^{nj})).$$

Para simplicar la notación, cada conjunto difuso $\tau_{S_j S_T}(p^{ij})$ lo representamos como r^{ij}, por lo que el vector de utilidad del experto p_j se representa como:

$$(r^{1j}, \ldots, r^{nj}) \text{ con } r^{ij} = \{\alpha_0^{ij}, \ldots, \alpha_g^{ij}\}.$$

3.1.2 Cálculo de las Preferencia Lingüísticas Colectivas

En este paso del proceso de decisión la información aportada por un experto p_j sobre una alternativa x_i, p^{ij}, está definida como un conjunto difuso r^{ij} sobre S_T. Por tanto, para obtener una preferencia lingüística colectiva sobre una alternativa, x_i, debemos agregar estos conjuntos difusos $\{r^{ij}, \forall j\}$. La preferencia lingüística colectiva sobre la alternativa x_i la notamos como r^i, y es un nuevo conjunto difuso definido sobre S_T de acuerdo a la siguiente expresión:

$$r^i = \{\alpha_0^i, ..., \alpha_g^i\}$$

con función de pertenencia

$$\alpha_k^i = f(\alpha_k^{i1}, ..., \alpha_k^{im}), \ k \in \{0, ..., g\}$$

donde f es un "operador de agregación".

Por tanto, el resultado de este paso del modelo de selección es un conjunto de preferencias lingüísticas colectivas, donde cada valor colectivo para cada alternativa es obtenido según las valoraciones individuales expresadas por todos los expertos sobre dicha alternativa,

$$(r^1, ..., r^n)$$

En la siguiente subsección, mostramos como encontrar el conjunto solución de alternativas a partir de las evaluaciones colectivas.

3.2 Selección de las Mejores Alternativas

El objetivo del modelo de selección es encontrar un conjunto de alternativas que contenga las mejores, de acuerdo a las preferencias de todos los expertos. En este caso las preferencias son conjuntos difusos sobre CBE, r^i. Entonces, tenemos que definir un método de selección, que aplicado directamente sobre las preferencias (conjuntos difusos), nos permita obtener la solución. Ésta no es una tarea fácil, pues hemos de comparar conjuntos difusos. Para resolverla cambiamos la representación de las preferencias lingüísticas colectivas basadas en conjuntos difusos por una representación basada en relaciones de preferencia difusas. Usamos el método de comparación de números difusos en contexto posibilístico descrito en [3]. Específicamente, aplicamos una modificación del *grado de posibilidad de dominancia* sobre números difusos propuesto en [3], para que actúe sobre conjuntos difusos r^i definidos en un universo discreto (el conjunto básico de etiquetas S_T). Este método de selección queda definido por los dos siguientes pasos:

1. Calcular una relación de preferencia difusa.
2. Aplicar un grado de selección a la relación de preferencia difusa para ordenar las alternativas y seleccionar la(s) mejor(es).

1. Obtener una Relación de Preferencia Difusa.

La siguiente definición se usa para comparar números difusos.

Definición 2 [3]. *Sean* **u** *y* **v** *dos números difusos, el grado de posibilidad de dominancia de u sobre v es:*

$$P(u \geq v) = \max_{x} \min_{y \leq x} \{\mu_u(x), \mu_v(y)\}$$

Sin embargo, nosotros tenemos que ordenar conjuntos difusos en un universo discreto, S_T. En la siguiente definición adaptamos el grado de posibilidad de dominancia para poder trabajar en S_T.

Definición 3. *Sean $x_i, x_j \in X (i \neq j)$ dos alternativas con sus respectivos conjuntos difusos de preferencia $r^i, r^j \in F(S_T)$, entonces el grado de preferencia de x_i sobre x_j, b_{ij}, se obtiene según la siguiente expresión:*

$$b_{ij} = \max_{c_l} \min_{c_h \leq c_l} \{\mu_{r^i}(c_l), \mu_{r^j}(c_h)\}$$

donde $\mu_{r_i}(c_l) = \alpha_l^i$ y $\mu_{r_j}(c_h) = \alpha_h^j$.

Aplicando esta definición sobre todos los posibles pares $(i \neq j)$ de las alternativas, obtenemos una relación de preferencia difusa $B = [b_{ij}]$.

2. Aplicación de un Grado de Selección.

Para finalizar, el modelo de selección calcula el conjunto solución de alternativas aplicando un grado de selección sobre la relación de preferencia difusa, B. En [8] se presenta una muestra de los distintos grados de selección que se pueden usar. Usando uno de ellos ordenamos las alternativas y seleccionamos aquella(s) con valor máximo en su grado de selección.

En la siguiente sección presentamos un ejemplo particular de aplicación de este modelo general de selección.

4 Ejemplo

Supongamos que tenemos una asesoría bursátil que recibe el encargo de hacer un estudio para invertir una cantidad de dinero de la forma más rentable. Existen cuatro posibles opciones de inversión:

- x_1 es una compañía de automóviles,
- x_2 es una compañía alimenticia,
- x_3 es una compañía de ordenadores,
- x_4 es una compañía armamentos.

La asesoría bursatil tiene un grupo de departamentos para consultar sus decisiones.

- p_1 es el departamento de análisis de riesgos,
- p_2 es el departamento de análisis de crecimiento,
- p_3 es el departamento de análisis medio-ambiental,
- p_4 es el departamento de análisis socio-político.

Cada uno es dirigido por un experto. Los expertos usan diferentes conjuntos de etiquetas para expresar sus preferencias lingüísticas sobre el conjunto de las alternativas. En particular:

- p_1 utiliza el conjunto de etiquetas A de 9 términos.
- p_2 utiliza el conjunto de etiquetas B de 7 términos.
- p_3 utiliza el conjunto de etiquetas C de 5 términos.
- p_4 utiliza el conjunto de etiquetas D de 9 términos.

	Conj. Etiquetas A		**Conj. Etiquetas B**
a_0	(0 0 .12)	b_0	(0 0 .16)
a_1	(0 .12 .25)	b_1	(0 .16 .33)
a_2	(.12 .25 .37)	b_2	(.16 .33 .5)
a_3	(.25 .37 .5)	b_3	(.33 .5 .66)
a_4	(.37 .5 .62)	b_4	(.5 .66 .83)
a_5	(.5 .62 .75)	b_5	(.66 .83.1)
a_6	(.62 .75 .87)	b_6	(.83 1 1)
a_7	(.75 .87 .1)		
a_8	(.87 1 1)		

	Conj. Etiquetas C		Conj. Etiquetas D
c_0	$(0, 0, .25)$	d_0	$(0, 0, 0, 0)$
c_1	$(0, .25, .5)$	d_1	$(0, .01, .02, .07)$
c_2	$(.25, .5, .75)$	d_2	$(.04, .1, .18, .23)$
c_3	$(.5, .75, 1)$	d_3	$(.17, .22, .36, .42)$
c_4	$(.75, 1, 1)$	d_4	$(.32, .41, .58, .65)$
		d_5	$(.58, .63, .80, .86)$
		d_6	$(.72, .78, .92, .97)$
		d_7	$(.93, .98, .99, 1)$
		d_8	$(1, 1, 1, 1)$

Después del estudio de las alternativas, los expertos suministran las siguientes preferencias:

		alternativas			
		x_1	x_2	x_3	x_4
expertos	p_1	a_4	a_6	a_3	a_5
	p_2	b_3	b_4	b_3	b_5
	p_3	c_2	c_3	c_2	c_1
	p_4	d_4	d_5	d_3	d_5

A continuación presentamos un modelo particular de selección el cuál nos permite resolver este ejemplo.

4.1 Un Modelo de Selección Basado en el Operador OWA y en el Grado de Selección de No Dominancia

Este modelo de selección específico sigue el esquema general presentado en la Sección 3, pero presenta los siguientes rasgos particulares:

1. El operador de agregación f utilizado para calcular los valores de preferencia colectivos, es el *operador OWA guiado por un cuantificador lingüístico* [10], representando el concepto de *mayoría difusa.*

2. El proceso de selección es realizado por el *"grado de selección de no dominancia"* definido por Orlovski [7].

1. Fusión de la Información Lingüística Multi-Granular

Se desarrolla en los dos siguientes pasos:

1.1 Hacer uniforme la información lingüística multigranular. Seleccionamos el CBE, siguiendo los pasos indicados en la Subsección 3.1.1. En este caso, como hay dos conjuntos de etiquetas con máxima granularidad y diferente semántica, elegimos como S_T el conjunto de etiquetas presentado en la Figura 1. Todos los valores de utilidad son convertidos a S_T a través de las funciones $\{\tau_{AS_T}, \tau_{BS_T}, \tau_{CS_T}, \tau_{DS_T}\}$, con lo que obtenemos los siguientes conjuntos difusos sobre S_T:

r^{11}	$(0, 0, 0, 0, .05, .45, .8, .82, .48, .23, 0, 0, 0, 0, 0)$
r^{12}	$(0, 0, 0, 0, .11, .45, .65, .95, .68, .39, .1, 0, 0, 0, 0)$
r^{13}	$(0, 0, 0, .23, .35, .6, .8, .98, .75, .5, .3, .1, 0, 0, 0)$
r^{14}	$(0, 0, 0, 0, .3, .77, 1, 1, 1, .51, 0, 0, 0, 0, 0)$
r^{21}	$(0, 0, 0, 0, 0, 0, 0, 0, .25, .99, .7, .31, .01, 0, 0)$
r^{22}	$(0, 0, 0, 0, 0, 0, 0, .35, .63, .94, .76, .46, .2, 0, 0)$
r^{23}	$(0, 0, 0, 0, 0, 0, .01, .25, .5, .7, .9, .9, .65, .45, .2)$
r^{24}	$(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, .55, 0, 0)$
r^{31}	$(0, 0, 0, .18, .55, .95, .7, .35, 0, 0, 0, 0, 0, 0, 0)$
r^{32}	$(0, 0, 0, 0, .1, .45, .65, .95, .68, .39, .1, 0, 0, 0, 0)$
r^{33}	$(0, 0, 0, .23, .35, .6, .8, .98, .75, .5, .3, .1, 0, 0, 0)$
r^{34}	$(0, 0, .41, 1, 1, .99, 0, 0, 0, 0, 0, 0, 0, 0, 0)$
r^{41}	$(0, 0, 0, 0, 0, 0, 0, .36, .71, .91, .56, .22, 0, 0, 0)$
r^{42}	$(0, 0, 0, 0, 0, 0, 0, 0, 0, .23, .54, .84, .86, .58, .3)$
r^{43}	$(.25, .4, .7, .9, .87, .65, .4, .2, 0, 0, 0, 0, 0, 0, 0)$
r^{44}	$(.25, .4, .7, .9, .87, .65, .4, .2, 0, 0, 0, 0, 0, 0, 0)$

1.2 Calcular las preferencias lingüísticas colectivas.

A partir de los r^{ij} calculamos la preferencias lingüística colectiva sobre cada alternativa, usando el operador OWA guiado por un cuantificador lingüístico.

Definición 4 [10]. *Sea $A = \{a_1, \ldots, a_n\}$ un conjunto de valores a ser agregados, el operador Ordered Weighted Averaging (OWA), F se define como,*

$$F(a_1, \ldots, a_n) = WB^T = \sum_{i=1}^{n} w_i b_i$$

donde $W = \{w_1, \ldots, w_n\}$ es un vector de pesos, tal que, $w_i \in [0, 1]$ y $\sum_i w_i = 1$. B es el vector ordenado asociado a A, donde $b_i \in B$ es el i-esimo mayor valor en A.

Nuestro interés es alcanzar soluciones que expresen la opinión de la mayoría de los expertos. En este sentido, los pesos w_i para la agregación se pueden calcular a partir de la función que describe a un cuantificador lingüístico proporcional creciente Q mediante la siguiente expresión [10]:

$$w_i = Q(i/m) - Q((i-1)/m), i = 1, \ldots, m,$$

con

$$Q(t) = \begin{cases} 0 & \text{si } t < a \\ \frac{r-a}{b-a} & \text{si } a \leq t \leq b \\ 1 & \text{si } t > b \end{cases}$$

con $a, b, t \in [0, 1]$. En este caso, notaremos al operador OWA guiado por el cuantificador lingüístico Q, como F_Q.

En este ejemplo utilizamos el cuantificador *"tantos como sea posible"* con parámetros $(a = 0.5, b = 1)$, y por tanto con el vector de pesos $W = \{0, 0, .5, .5\}$. Entonces, las preferencias lingüísticas colectivas que obtenemos son:

r^1	$(0, 0, 0, 0, .08, .45, .72, .88, .58, .31, 0, 0, 0, 0, 0)$
r^2	$(0, 0, 0, 0, 0, 0, 0, 0, .12, .82, .73, .38, .1, 0, 0)$
r^3	$(0, 0, 0, .05, .23, .52, .32, .17, 0, 0, 0, 0, 0, 0, 0)$
r^4	$(0, 0, 0, 0, 0, 0, 0, 0, 0, .12, .27, .11, 0, 0, 0)$

2. Selección de las Mejores Alternativas.

Se realiza también en dos pasos:

2.1 Cálculo de la relación de preferencia difusa. Obtenemos la siguiente relación de preferencia difusa a partir de los valores colectivos:

$$\begin{pmatrix} - & .31 & .52 & .12 \\ .82 & - & .52 & .27 \\ .45 & 0 & - & 0 \\ .27 & .27 & .27 & - \end{pmatrix}$$

2.2 Aplicación del grado de selección de no dominancia. A cada alternativa x_i, le calculamos su grado de selección de no dominacia NDD_i.

Definición 5 [7]. *Sea $B = [b_{ij}]$ una relación de preferencia difusa definida sobre el conjunto de alternativas X. Para la alternativa x_i, su grado de no dominancia NDD_i, es definido como sigue:*

$$NDD_i = \min_{x_j}[1 - b^s_{ji}, j \neq i]$$

donde b^s_{ji} se calcula mediante la siguiente expresión,

$$b^s_{ji} = \max\{b_{ji} - b_{ij}, 0\},$$

y representa el grado para el cuál x_i es estrictamente dominado por x_j.

Entonces, primero calculamos la relación de preferencia estricta B^s:

$$\begin{pmatrix} - & 0 & .07 & 0 \\ .51 & - & .52 & 0 \\ .0 & 0 & - & 0 \\ .15 & 0 & .27 & - \end{pmatrix}$$

y luego, los grados de no dominacia:

$$\{NDD_1 = .49, NDD_2 = 1, NDD_3 = .48,$$
$$NDD_4 = 1\}.$$

Por tanto, el conjunto solución de alternativas obtenido es $X^{ND} = \{x_2, x_4\}$, siendo las mejores opciones la compañía alimenticia y la armamentística.

5 Comentarios Finales

En este trabajo hemos presentado un modelo de selección para problemas de toma de decisión con múltiples expertos los cuales expresan sus preferencias usando distintos dominios lingüísticos.

Este modelo es útil en problemas de decisión donde los expertos provienen de distintas áreas de conocimiento o tienen distinto grado de conocimiento sobre el problema. La técnica de fusión de información lingüística multi-granular usada puede ser aplicada en muchas otras áreas como diagnosis, recuperación de información, etc.

Referencias

[1] P.P. Bonissone and K.S. Decker, Selecting Uncertainty Calculi and Granularity: An Experiment in Trading-off Precision and Complexity, en: L.H. Kanal and J.F. Lemmer, Eds., *Uncertainty in Artificial Intelligence* (North-Holland, 1986) 217-247.

[2] M. Delgado, F. Herrera, E. Herrera-Viedma and L. Martínez, Combining Numerical and Linguistic Information in Group Decision Making, *Information Sciences* **7** (1998) 177-194.

[3] D. Dubois and H. Prade, Ranking Fuzzy Numbers in the Setting of Possibility Theory, *Information Sciences* **30** (1983) 183-224.

[4] F. Herrera and E. Herrera-Viedma, Aggregation Operators for Linguistic Weighted Information, *IEEE Transactions on Systems, Man, and Cybernetics* **27** (1997) 646-656.

[5] 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** (1996) 175-190.

[6] G.A. Miller, The Magical Number Seven or Minus Two: Some limits On Our Capacity of Processing Information, *Psychological Rev.* **63** (1956) 81-97.

[7] S.A. Orlovski, Decision Making with a Fuzzy Preference Relation, *Fuzzy Sets and Systems* **1** (1978) 155-167.

[8] M. Roubens, Some Properties of Choice Functions Based on Valued Binary Relations, *European Journal of Operational Research* **40** (1989) 309-321.

[9] M. Tong and P. P. Bonissone, A Linguistic Approach to Decision Making with Fuzzy Sets, *IEEE Transactions on Systems, Man and Cybernetics* **10** (1980) 716-723.

[10] R.R. Yager, On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making, *IEEE Transactions on Systems, Man, and Cybernetics* **18** (1988) 183-190.

[11] R.R. Yager, Non-Numeric Multi-Criteria Multi-Person Decision Making, *Group Decision and Negotiation* **2** (1993) 81-93.

[12] 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.

PUTTING TOGETHER ŁUKASIEWICZ AND PRODUCT LOGICS

Francesc Esteva **Lluís Godo**

Institut d'Investigacions en Intel.ligència Artificial (IIIA)
Consejo Superior de Investigaciones Científicas (CSIC)
Campus Universitat Autònoma de Barcelona, s/n
08193 Bellaterra, Spain
{*esteva,godo*}*@iiia.csic.es*

Abstract

In this paper we investigate a propositional fuzzy logical system LΠ which contains the well-known Lukasiewicz, Product and Gödel fuzzy logics as sublogics. We define the corresponding algebraic structures, called LΠ-algebras and prove the following completeness result: a formula φ is provable in the LΠ logic iff it is a tautology for all linear LΠ-algebras. Moreover, linear LΠ-algebras are shown to be embeddable in linearly ordered abelian rings with an strong unit and cancellation law.

1 INTRODUCTION

The idea of this paper is to define a logical system extending both Lukasiewicz (L) and Product (Π) logics. Product logic, a fuzzy logic having product as conjunction, was introduced in [6]. In [5] Hájek defines the basic fuzzy logic BL having as main axiomatic extensions Lukasiewicz, Product and Gödel (G) logics. All these logics take the conjunction (&) and the implication ($\rightarrow$) as the only primitive connectives (besides the truth constant 0), and have the Lukasiewicz, product and minium t-norms and their corresponding residua as truth-functions respectively.

There has been a few attempts to extend these logics with more connectives. Baaz introduces in [2] a projection (boolean) connective Δ into Gödel logic: the truth value of $\Delta\varphi$ is 1 if the truth-value of φ is 1, 0 otherwise. After, this connective has been also introduced by Hájek [5] in the rest of the above logics, resulting in the extended logics, BL_Δ, L_Δ, Π_Δ and G_Δ. In a recent paper, Esteva et al. [4] extend SBL (an axiomatic extension of BL), Π and G logics with an involutive negation, since in all these logics the definable negation $\neg\varphi = \varphi \rightarrow 0$ is not involutive, in contradistinction to what happens in Luksiewicz logic. The resulting logics have been denoted $SBL_\sim$, $\Pi_\sim$ and $G_\sim$ respectively.

On the other hand, the strong system of predicate fuzzy logic of Takeuti-Titani [7] combine the above mentioned three conjunctions and three implications, together with truth-constants. In [5] Hájek also presents a different version of such logic, called TT∀, endowed with an infinitary deduction rule and having the predicate Lukasiewicz, product and Gödel logics as sublogics.

However, unless the latter logics (L, Π, G), the just mentioned Takeuti-Titani-like logics do not present a corresponding algebraic structure as the other ones. Actually, all the completeness results of the first group of logics are obtained via their associated algebraic structures (BL-algebras, MV-algebras, Π-algebras, etc.) by means of theorems of decomposition of such algebras into subdirect products of linear algebras and finally relating these with the t-norm based structures of the unit interval [0, 1].

With these results on mind, this paper aims at defining a logical system, and its corresponding algebras, having both Lukasiewicz and product logics as sublogics. As a consequence, we shall also have Gödel logic as sublogic for free. In the second section, after defining the logic LΠ and LΠ-algebras we prove their decomposition as subdirect product of linearly ordered (l.o.) ones and prove completeness of LΠ w.r.t. them. Finally, in Section 4 we prove that l.o. LΠ-algebras are emmbedable in a ring (they are the interval [0,1] of the ring). We end up with some discussion about open problems and future work. Annex 1 contains some necessary background on different systems of fuzzy logic while Annex 2 contains the proof of a Proposition 4 of Section 3.

2 THE LΠ LOGIC

Now we introduce the LΠ Logic, extending both Lukasiewicz and Product logics. We take three prim-

itive connectives $\to_L$,$\odot$,$\to_\Pi$ and the truth-constant 0. Other definable connectives are $\neg_L$, $\neg_\Pi$, Δ, $\&$, $\veebar$, $\wedge$, $\vee$ and $\to_G$, where:

$$\begin{array}{lll} \neg_L\varphi & \text{is} & \varphi \to_L 0 \\ \neg_\Pi\varphi & \text{is} & \varphi \to_\Pi 0 \\ \Delta\varphi & \text{is} & \neg_\Pi\neg_L\varphi \\ \varphi\&\psi & \text{is} & \neg_L(\varphi \to_L \neg_L\psi) \\ \varphi \veebar \psi & \text{is} & \neg_L\varphi \to_L \psi \\ \varphi \wedge \psi & \text{is} & \varphi\&(\varphi \to_L \psi) \\ \varphi \vee \psi & \text{is} & \neg_L(\neg_L\varphi \wedge \neg_L\psi) \\ \varphi \to_G \psi & \text{is} & \Delta(\varphi \to_L \psi) \vee \psi \end{array}$$

An standard truth-evaluation is any mapping e assigning to each propositional variable p a value of the unit interval [0, 1], and it extends to all formulas by means the Lukasiewicz and product truth-functions, that is:

$$\begin{array}{lcl} e(\varphi \to_L \psi) & = & \min(1, 1 - e(\varphi) + e(\psi)), \\ e(\varphi \odot \psi) & = & e(\varphi) \cdot e(\psi), \text{ and} \\ e(\varphi \to_\Pi \psi) & = & \begin{cases} 1, & \text{if } e(\varphi) \leq e(\psi) \\ e(\psi)/e(\varphi), & \text{otherwise} \end{cases} \end{array}$$

Notice that, with these definitions, we recover the usual truth functions for the above definable connectives: $e(\neg_L\varphi) = 1 - e(\varphi)$; $e(\neg_\Pi\varphi) = 1$ if $e(\varphi) = 0$, 0 otherwise (Gödel negation); $e(\Delta\varphi) = 1$ if $e(\varphi) = 1$, 0 otherwise; $e(\varphi\&\psi) = \max(0, e(\varphi) + e(\psi) - 1)$; $e(\varphi \veebar \psi) = \min(1, e(\varphi) + e(\psi))$; $e(\varphi \wedge \psi) = \min(e(\varphi), e(\psi))$; $e(\varphi \vee \psi) = \max(e(\varphi), e(\psi))$; and $e(\varphi \to_G \psi) = 1$ if $e(\varphi) \leq e(\psi)$, $e(\psi)$ otherwise (Gödel implication).

Definition 1 *Axioms of LΠ Logic are :*

1. *the axioms of Lukasiewicz logic with the projection, L_Δ, for the connectives $\&$, $\to_L$, and Δ,*
2. *the axioms of the product logic with involution, $\Pi_\sim$, for the connectives $\odot$, $\to_\Pi$, and $\neg_L$,*
3. *and the following additional axioms:*

$$\begin{array}{ll} \textit{(LP1)} & (\varphi\&\psi) \to (\varphi \odot \psi) \\ \textit{(LP2)} & (\varphi \to_\Pi \psi) \to (\varphi \to_L \psi) \\ \textit{(LP3)} & \Delta(\varphi \odot \psi) \to \Delta(\varphi\&\psi) \\ \textit{(LP4)} & \Delta(\varphi \to_L \psi) \to \Delta(\varphi \to_\Pi \psi) \\ \textit{(LP5)} & (\varphi \odot \psi)\veebar(\varphi \odot \neg_\Pi\psi) \equiv \varphi \\ \textit{(LP6)} & \neg_L\Delta(\psi\&\chi) \to \\ & (\varphi \odot (\psi\veebar\chi)) \equiv ((\varphi \odot \psi)\veebar(\varphi \odot \chi)) \\ \textit{(LP7)} & \Delta(\varphi\veebar\psi) \to \\ & (((\varphi\&\psi) \odot \chi)\veebar\chi) \equiv ((\varphi \odot \chi)\veebar(\psi \odot \chi)) \\ \textit{(LP8)} & \Delta(\varphi\veebar\psi) \to \\ & (((\varphi\&\psi) \odot \chi)\&\chi) \equiv ((\varphi \odot \chi)\&(\psi \odot \chi)) \end{array}$$

where $\to$ and $\equiv$ stand for any of the implications or their corresponding equivalences respectively.

Inference rules of LΠ are Modus Ponens for both implications and necessitation for Δ: from φ derive $\Delta\varphi$.

The notion of proof is as usual. We will write $\mathrm{L\Pi} \vdash \varphi$ and $T \vdash \varphi$ to denote that φ is provable in LΠ and provable from a theory T over LΠ, respectively.

It is easy to check that LΠ is sound w.r.t. the above semantics, that is, each axiom is a 1-tautology and the deduction rules preserve 1-tautologies. Next we introduce the algebraic structure corresponding to our logic.

Definition 2 *A LΠ-algebra is an algebra* $\mathbf{A} = \langle A, *, \Rightarrow_L, \odot, \Rightarrow_\Pi, \cup, \cap, 0, 1\rangle$ *such that:*

- $\langle A, \odot, \Rightarrow_\Pi, \cup, \cap, 0, 1, \neg_L\rangle$ *is a $\Pi_\sim$-algebra where $\neg_L(x) = x \Rightarrow_L 0$)*
- $\langle A, *, \Rightarrow_L, \cup, \cap, \Delta, 0, 1\rangle$ *is a MV_Δ-algebra where $\Delta(x) = (\neg_L x) \Rightarrow_\Pi 0$.*
- *For all $x, y, z \in A$ the following conditions hold:*
  (1) $x * y \leq x \odot y$
  (2) $\Delta(x \Rightarrow_L y) \leq \Delta(x \Rightarrow_\Pi y)$, $\Delta(x * y) \geq \Delta(x \odot y)$
  (3) If $x * y = 0$ then $z \odot (x \oplus y) = (z \odot x) \oplus (z \odot y)$
  (4) If $x \oplus y = 1$ then:
  $((x * y) \odot z) \oplus z) = (z \odot x) \oplus (z \odot y)$,
  $((x * y) \odot z) * z) = (z \odot x) * (z \odot y)$,
  where $x \oplus y = \neg_L(\neg_L(x) * \neg_L(y))$.

Lemma 1 *In a LΠ-algebra A the following conditions hold for all $x, y \in A$:*
(i) $x \Rightarrow_\Pi y \leq x \Rightarrow_L y$,
(ii) $\Delta(x \Rightarrow_L y) = \Delta(x \Rightarrow_\Pi y)$, $\Delta(x * y) = \Delta(x \odot y)$
(iii) $(x \odot y) \oplus (x \odot \neg_L y) = x$,
(iv) $\neg_L(x \odot y) = \neg_L x \oplus (x \odot \neg_L y)$.

Proof: Condition (i) is a consequence of the above property (1) and that $(*, \Rightarrow_L)$ and $(\odot, \Rightarrow_\Pi)$ are adjoint pairs. Equalities (ii) are easy consequences of the property (2) and that from (i) one can derive $\Delta(x \Rightarrow_\Pi y) \leq \Delta(x \Rightarrow_L y)$, taking into account usual properties of the Δ operator. The proof of (iii) is as follows: $x = x \odot 1 = x \odot (y \oplus \neg_L y)$, and since $y * \neg_L y = 0$ we can apply (3) and thus $x \odot (y \oplus \neg_L y) = (x \odot y) \oplus (x \odot \neg_L y)$. The proof of (iv) uses the following result from the theory of MV-algebras (see for instance [3]):

$$\text{if } x * y = 0, \text{ then } (x \oplus y) * \neg y = x.$$

From (iii) we have that $(x \odot y) * (x \odot \neg_L y) = 0$ and therefore,

$$x \odot y = ((x \odot y) \oplus (x \odot \neg_L y)) * \neg_L(x \odot \neg_L y) = x * \neg_L(x \odot \neg_L y).$$

Finally, using the De Morgan laws for negation w.r.t. $*$ and $\oplus$, we have $\neg_L(x \odot y) = \neg_L x \oplus (x \odot \neg_L y)$. □

It is easy to check that the LΠ logic is sound with respect to LΠ-algebras. This means that, for every LΠ-algebra $\mathbf{A}$, the axioms of LΠ are $\mathbf{A}$-tautologies and the

deduction rules preserve the **A**-tautologies. A formula φ is an **A**-tautology if it gets the value 1 (of A) in each evaluation over the algebra **A**, an evaluation being a mapping e assigning to each propositional variable p an element $e(p) \in A$ and extending to all formulas using the operations of **A** as truth-functions.

It is obvious that the real unit interval [0, 1] equipped with the Lukasiewicz and product related truth functions is a LΠ-algebra. We will prove that the algebra of classes of provable equivalent formulas is also an LΠ algebra. But we have to be cautious since in LΠ there are two different implications and equivalences. However one can show that if a theory proves one implication or equivalence, it has to prove the other one as well.

Proposition 1 *If T is a theory over LΠ, then the following conditions hold:*
(1) $T \vdash \varphi$ iff $T \vdash \Delta\varphi$.
(2) $T \vdash \varphi \to_\Pi \psi$ iff $T \vdash \varphi \to_L \psi$.
(3) $T \vdash \varphi \equiv_\Pi \psi$ iff $T \vdash \varphi \equiv_L \psi$.

Proof: (1) It comes from the rule of necessitation for Δ and axiom (Δ3).
(2) Axiom (LP2) proves one direction. On the other direction, if $T \vdash \varphi \to_L \psi$ then, applying the necessitation rule for Δ, $T \vdash \Delta(\varphi \to_L \psi)$ and so, by (LP4), $T \vdash \Delta(\varphi \to_\Pi \psi)$ and hence, by (1), $T \vdash \varphi \to_\Pi \psi$.
(3) It is an easy consequence of (2). □

As a result of this proposition we can define in the usual way the quotient set $\mathcal{L}/\equiv_T$ of equivalence classes of formulas w.r.t. a theory T, where $\varphi \equiv_T \psi$ iff $T \vdash \varphi \equiv \psi$, being $\equiv$ either $\equiv_L$ or $\equiv_\Pi$. Then in the quotient set, also as usual, connectives can be interpreted as operations and now the corresponding algebraic structure can be shown to be a LΠ algebra.

Lemma 2 *For any theory T over LΠ, $\mathcal{L}/\equiv_T$ is a LΠ-algebra.*

Proof: The analogous results for MV-algebras and Π-algebras prove that the equivalence relation $\equiv_T$ is a congruences w.r.t. the connectives $(\&, \to_L)$ and $(\odot, \to_\pi)$ respectively, so it remains to prove that the extra-conditions of the LΠ-algebras also hold in the quotient algebra. But these easily follow from the axioms (LP1)-(LP8) of the logical system. □

Next we have to prove that each LΠ-algebra is a subdirect product of linearly ordered LΠ-algebras. The proof of this fact is rather standard and the basic definitions and proofs are given below.

Definition 3 *A subset F of a LΠ-algebra A is a* filter *if it satisfies:*

(F1) *For all $x, y \in F$, $x * y \in F$*
(F2) *If $x \in F$ and $y \geq x$, then $y \in F$*
(F3) *If $x \to_\Pi y \in F$, then $\neg_L y \to_\Pi \neg_L x \in F$*

Moreover, F is said to be an ultrafilter *(or* prime filter*) iff it is a filter satisfying:*

(F4) *For all $x, y \in F$, either $x \Rightarrow_\Pi y \in F$ or $y \Rightarrow_\Pi x$*

Obviously, if F is a filter w.r.t. a LΠ-algebra, F is also a filter of the corresponding MV and $\Pi_\sim$ sub-algebras. Moreover, the following properties hold.

Lemma 3 *Let F be a filter in a LΠ-algebra A. Then the following equivalences hold:*
(i) $x \in F$ iff $\Delta x \in F$.
(ii) $x * y \in F$ iff $x \odot y \in F$
(ii) $x \Rightarrow_L y \in F$ iff $x \Rightarrow_\Pi y \in F$

Proof: (i) Since $\Delta x \leq x$, by (F2) if $\Delta x \in F$ then $x \in F$ too. On the other direction, $x = 1 \Rightarrow_\Pi x$ and thus, by (F3), $\Delta x = \neg_L x \Rightarrow_\Pi 0 \in F$ as well.
(ii) and (iii) easily come from (i) taking into account (ii) of Lemma 1. □

This lemma allows us to properly define the congruence relation in the next lemma.

Lemma 4 *Let F be a filter in a LΠ-algebra **A**. Define*

$$x \sim_F y \text{ iff both } x \Rightarrow y, y \Rightarrow x \in F,$$

where $\Rightarrow$ is any of the two implications. Then:
*(i) $\sim_F$ is a congruence on **A** and $\mathbf{A}/\sim_F$ is a LΠ-algebra.*
(ii) $A/\sim_F$ is linearly ordered iff F is an ultrafilter.

For the sake of space we omit the proof.

Lemma 5 *If F is a filter and $a \notin F$, then there exists an ultrafilter UF such that $F \subseteq UF$ and $a \notin UF$.*

Proof: Sketch. Suppose that F is a filter not containing a and that there exist two elements $x, y \in A$ such that $x \Rightarrow_\Pi y \notin F$ and $y \Rightarrow_\Pi x \notin F$. Then one can check that the least filters F_1 and F_2 such that contain F and $x \Rightarrow_\Pi y$ and $y \Rightarrow_\Pi x$ respectively are:

$$F_1 = \{u \in A \mid \exists v \in F \text{ and } u \geq (v * \Delta(x \Rightarrow_\Pi y))\}$$
$$F_2 = \{u \in A \mid \exists v \in F \text{ and } u \geq (v * \Delta(y \Rightarrow_\Pi x))\}.$$

Next one proves that either $a \notin F_1$ or $a \notin F_2$. Namely, if $a \in F_1 \cap F_2$ then $a \geq v_1 * \Delta(x \Rightarrow_\Pi y)$ and $a \geq v_2 * \Delta(y \Rightarrow_\Pi x)$ and so $a \geq (min(v_1, v_2) * \Delta(x \Rightarrow_\Pi y)) \cup (min(v_1, v_2) * \Delta(y \Rightarrow_\Pi x)) = min(v_1, v_2) * (\Delta(x \Rightarrow_\Pi y) \cup (\Delta(y \Rightarrow_\Pi x)) = min(v_1, v_2) * \Delta((x \Rightarrow_\Pi y) \cup (y \Rightarrow_\Pi x)) = min(v_1, v_2) * \Delta(1) = min(v_1, v_2) \in F$ in contradiction with the hypothesis. Finally, one builds an increasing sequence of filters $F_i \subseteq F_{i+1}$ such that $F_0 = \{1\}$ and $a \notin F_i$, for every i. The ultrafilter is then the big union of all the F_i's. □

This last lemma is used to show that the intersection of all ultrafilters of a LΠ-algebra is just the singleton $\{1\}$, and thus, using (ii) of Lemma 4 together with standard results about subdirect products, we get the following decomposition theorem.

Theorem 1 *Each LΠ-algebra is a subdirect product of linearly ordered LΠ-algebras.*

Moreover l.o. LΠ-algebras are subdirectly irreducible, which is not true for Lukasiewicz and product algebras.

Proposition 2 *A LΠ-algebra is subdirect irreducible iff it is linearly ordered.*

The proof is the same than in [4] for $SBL_\sim$ algebras. The only filters of a l.o. LΠ-algebra are the maximal element 1 and the total algebra. Finally, our completeness result is the following.

Theorem 2 (*Completeness.*) *The logic LΠ is complete w.r.t. the linearly ordered LΠ-algebras. That is, a formula φ is provable in LΠ if it is an* **A***-tautology for each linearly ordered LΠ-algebra* **A**.

Proof: One direction is soundness. For the other direction, if φ is an **A**-tautology for each linearly-ordered LΠ algebra **A** then, by the above decomposition theorem, it is also a tautology for each LΠ-algebra, in particular, for the LΠ-algebra of the classes of provable equivalent formulas, that is, the logic LΠ proves $\varphi \equiv \overline{1}$, that is, LΠ proves φ. □

It must be noticed however that this completeness result is only partial, in the sense that completeness is with respect to all l.o. LΠ-algebras, not only w.r.t. the standard LΠ-algebra of the unit interval [0, 1]. This remains as an open question. Nevertheless. in the next section we present results we have got so far towards that direction.

3 ON LINEARLY ORDERED LΠ-ALGEBRAS

It is known (see for example ([3]) that it is possible to embed a linearly ordered MV-algebra into a linearly ordered abelian group (l.o.a.g.). In particular, given a LΠ-algebra $\mathbf{A} = \langle A, *, \Rightarrow_L, \odot, \Rightarrow_\Pi, \cup, \cap, 0, 1\rangle$, consider its MV subalgebra $\mathbf{A}_{MV} = \langle A, *, \Rightarrow_L, \cup, \cap, 0, 1\rangle$ and the l.o.a.g $\mathbf{G_A} = \langle G_A, +, -, 0_G, \leq_G\rangle$ where $G_A = \{(n, x) \mid n \in Z, x \in A, x \neq 1\}$, $0_G = (0, 0)$ and

$$(n,x) + (m,y) = \begin{cases} (n+m, x \oplus y), & \text{if } x \oplus y < 1 \\ (n+m+1, x * y), & \text{if } x \oplus y = 1 \end{cases}$$

$$-(n,x) = \begin{cases} (-n, 0), & \text{if } x = 0 \\ (-(n+1), \neg_L x), & \text{if } 0 < x < 1 \end{cases}$$

$(n, x) \leq_G (m, y)$ if $n < m$ or $n = m$ and $x \leq y$.

It can be shown that $\mathbf{G_A}$ is a l.o.a.g. with neutral element (0,0) and strong unit (1,0) and that the MV-algebra $\mathbf{A}_{MV}$ is isomorphic to the interval $[(0,0),(1,0)] = \{(n,x) \in G_A \mid (0,0) \leq_G (n,x) \leq_G (1,0)\}$, identifying $(0, x)$ with x and (1,0) with 1.

On the other hand it is possible to define a product operation $\times$ on G_A, extension of the product $\odot$ of the algebra A. The product is defined as follows:

$$(n,x) \times (m,y) = (nm, x \odot y) + m(0,x) + n(0,y),$$

where $m(0, x)$ means the sum of $(0, x)$, m times. It is clear that this product is commutative, $(0, 0)$ is absorbent and $(1, 0)$ is the unit element. Moreover, it is possible to prove that $\times$, together with $+$ endows G_A with an structure of linearly ordered abelian ring.

Proposition 3 *With the above definitions,* $\mathbf{R_A} = \langle G_A, +, -, \times, (0,0), (1,0), \leq_G\rangle$ *is a linearly ordered abelian ring with unit* $(1, 0)$ *and with cancellation law for* $\times$.

The proof is given in Annex 2. Therefore, G_A can be embedded in a field of pairs of elements a/b of G_A in the same way the ring of integers can be embedded into the field of rational numbers. Of course the initial algebra is isomorphic to the subalgebra of elements of the type $p/1$, where $p = (0, x)$ for some $x \in A - \{1\}$ or $x = (1, 0)$. But, moreover, the converse of Proposition 3 is also true.

Proposition 4 *Let* $(A, +, \times, 0, 1)$ *be a commutative linearly ordered ring with unit 1 and satisfying the cancellation law for* $\times$. *If for each* $x \neq 0$ *the mapping* $f_x : [0, 1] \to [0, x]$, *defined as* $f_x(y) = x \times y$, *is onto, then* $[0, 1]_A = \{x \in A \mid 0 \leq x \leq 1\}$ *is a LΠ algebra with the operations*

$$x * y = \max(0, x + y + (-1))$$
$$x \odot y = x \times y$$

and the corresponding residuated implications.

The proof of this proposition is an easy checking. Condition of f_x being onto is needed to guarantee the existence of the residuum of the product operation. If we take a field instead of a ring this condition is always satisfied due to the existence of invers elements w.r.t. the product.

4 CONCLUSIONS

In this paper we have dealt with an axiomatic approach to a logical system containing Lukasiewicz, product and Gödel logics as sublogics and with its corresponding algebraic structure. We have also generalised existing results for linear MV and product algebras establishing that they can be identified with

the interval [0,1] and the negative part of a linear ordered abelian group respectively. In our case, linear ordered LΠ-algebras are the interval [0,1] of a linear ordered abelian ring satisfying the cancellation law. On the other hand the LΠ logic, axiomatically defined in section 2, has been shown to be sound and complete w.r.t. linearly ordered LΠ-algebras. However, to look for standard or semi-standard completeness results (i.e. in the real unit interval [0, 1]) remains as a matter of further work. The current problem is whether it is possible to extend the theorem of Gurevich and Kokorin for l.o.a. groups to l.o.a. rings. If this is possible, there is a nice result by Alsina [1] that would allow us to reach our goal. Alsina's result is the following: if S is a continuous t-conorm, T is a continuous t-norm and N is a strong negation, the general solution of the functional equation

$$S(T(x,y),T(x,N(y)) = x$$

is given by the following expressions:

$$\begin{aligned} S(x,y) &= g^{[-1]}(g(x)+g(y)) \\ T(x,y) &= g^{[-1]}(g(x)\cdot g(y)) \\ N(x) &= g^{[-1]}(1-g(x)), \end{aligned}$$

where g is an strictly increasing function $g : [0,1] \to R^+$ such that $g(0) = 0$ and $g(1) = 1$, and $g^{[-1]}$ is the pseudo-inverse function of g, that is, up to an isomorphism, S is the Lukasiewicz t-conorm, T is the product t-norm and N is the strong Lukasiewicz negation. And we have shown that the above functional equation is verified in any LΠ-algebra (see (iii) of Lemma 1).

Acknowledgments The authors acknowledge partial support of the project CYCIT project SMASH (TIC96-1138-C04-01/04).

References

[1] C. Alsina: On a Family of Connectives for Fuzzy Sets. Fuzzy Sets and Systems 16 (1985), pp.231-235.

[2] M. Baaz: Infinite-valued Gödel logics with 0-1 projections and relativizations. In GÖDEL'96 - Logical foundations of mathematics, computer science and physics. Lecture Notes in Logic 6 (1996), P. Hájek (Ed.), Springer Verlag, pp.23–33.

[3] R. Cignoli, I.M.L. D'Ottaviano and D. Mundici: Algebras Das Logicas de Lukasiewicz. Centro de Logica, Epistemologia e Historia da Ciencia-UNICAMP (1995), Campinas (Brasil).

[4] F. Esteva, L. Godo, P. Hájek and N. Navara: Residuated fuzzy logics with an involutive negation. Submitted. Also as IIIA Research Report 98/09, Bellaterra (Barcelona), Spain.

[5] P. Hájek: Metamathematics of fuzzy logic. (In press) Kluwer (1998).

[6] P. Hájek, L. Godo and F. Esteva: A complete many-valued logic with product conjunction. Archive for Mathematical Logic 35 (1996), 191–208.

[7] G. Takeuti and S. Titani: Fuzzy Logic and Fuzzy Set Theory. Archive for Mathematical Logic 32 (1992), 1–32.

ANNEX 1: Background on some systems of fuzzy logics

Here we summarize some important notions and facts from some systems of propositional fuzzy logics that are used in the paper.

The basic fuzzy logic BL and BL-algebras

The language of the basic logic BL [5] is built in the usual way from a set of propositional variables, a conjunction &, an implication $\to$ and the constant $\bar{0}$. Further connectives are defined as follows:

$$\begin{array}{lll} \varphi \wedge \psi & \text{is} & \varphi\&(\varphi \to \psi), \\ \varphi \vee \psi & \text{is} & ((\varphi \to \psi) \to \psi) \wedge ((\psi \to \varphi) \to \varphi), \\ \neg\varphi & \text{is} & \varphi \to \bar{0}, \\ \varphi \equiv \psi & \text{is} & (\varphi \to \psi)\&(\psi \to \varphi). \end{array}$$

The following formulas are the *axioms* of BL:

$$\begin{array}{ll} \text{(A1)} & (\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi)) \\ \text{(A2)} & (\varphi\&\psi) \to \varphi \\ \text{(A3)} & (\varphi\&\psi) \to (\psi\&\varphi) \\ \text{(A4)} & (\varphi\&(\varphi \to \psi) \to (\psi\&(\psi \to \varphi)) \\ \text{(A5a)} & (\varphi \to (\psi \to \chi)) \to ((\varphi\&\psi) \to \chi) \\ \text{(A5b)} & ((\varphi\&\psi) \to \chi) \to (\varphi \to (\psi \to \chi)) \\ \text{(A6)} & ((\varphi \to \psi) \to \chi) \to (((\psi \to \varphi) \to \chi) \to \chi) \\ \text{(A7)} & \bar{0} \to \varphi \end{array}$$

The *deduction rule* of BL is modus ponens.

If one takes a continuous t-norm $*$ for the truth function of & and the corresponding residuum[1] $\Rightarrow$ for the truth function of $\to$ (and evaluating $\bar{0}$ by 0) then all the axioms of BL become 1-tautologies (have identically the truth value 1). And since modus ponens preserves 1-tautologies all formulas provable in BL are 1-tautologies.

It has been shown [5] that the well-known Lukasiewicz logic, denoted L, is the extension of BL by the axiom

(L) $\neg\neg\varphi \to \varphi$,

and Gödel logic, denoted G, is the extension of BL by the axiom

[1] The residuum $\Rightarrow$ is the binary function on [0, 1] defined as $x \Rightarrow y = \sup\{z \in [0,1] \mid x * z \leq y\}$.

(G) $\varphi \to (\varphi \& \varphi)$.

Finally, product logic, denoted Π, is just the extension of BL by the following two axioms:

(Π1) $\neg\neg\chi \to (((\varphi \& \chi) \to (\psi \& \chi)) \to (\varphi \to \psi))$,

(Π2) $\varphi \wedge \neg\varphi \to \bar{0}$.

A *BL-algebra* is an algebra $\mathbf{L} = (L, \cap, \cup, *, \Rightarrow, 0, 1)$ with four binary operations and two constants such that

(i) $(L, \cap, \cup, 0, 1)$ is a lattice with the largest element 1 and the least element 0 (with respect to the lattice ordering $\leq$),

(ii) $(L, *, 1)$ is a commutative semigroup with the unit element 1, i.e. $*$ is commutative, associative and $1 * x = x$ for all x,

(iii) the following conditions hold:

(1) $z \leq (x \Rightarrow y)$ iff $x * z \leq y$ for all x, y, z.

(2) $x \cap y = x * (x \Rightarrow y)$

(3) $(x \Rightarrow y) \cup (y \Rightarrow x) = 1$.

Thus, in other words, a BL-algebra is a *residuated lattice* satisfying (2) and (3). The class of all BL-algebras is a variety. Moreover, each BL-algebra can be decomposed as a subdirect product of linearly ordered BL-algebras.

Defining $\neg x = x \Rightarrow 0$, it turns out that MV-*algebras* are BL-algebras satisfying $\neg\neg x = x$, G-*algebras* are BL-algebras satisfying $x * x = x$, and finally, *product algebras* are BL-algebras satisfying

$$x \cap \neg x = 0$$
$$\neg\neg z \Rightarrow ((x * z = y * z) \Rightarrow x = y) = 1.$$

The logic BL is sound with respect to **L**-tautologies: if φ is provable in BL then φ is an **L**-tautology for each BL-algebra **L**.

Theorem 3 *BL is complete, i.e. for each formula φ the following three conditions are equivalent:*

(i) φ is provable in BL,

(ii) for each BL-algebra **L**, *φ is an* **L**- *tautology,*

(iii) for each linearly ordered BL-algebra **L**, *φ is an* **L**-*tautology.*

This theorem also holds if we replace BL by a *schematic extension*[2] $\mathcal{C}$ of BL, and BL-algebras by the corresponding $\mathcal{C}$-algebras (BL-algebras in which all axioms of $\mathcal{C}$ are tautologies). There is also *strong completeness* for provability in theories over BL. For completeness theorems of the three main many-valued logics (Lukasiewicz, Gödel and product) see [5].

[2] A calculus which results from BL by adding some axiom schemata.

Extended basic fuzzy logics with Δ and Δ-algebras

Now we expand the language of BL by a new unary (projection) connective Δ whose truth function (denoted also by Δ) is defined as follows:

$$\Delta(x) = \begin{cases} 1, & \text{if } x = 1 \\ 0, & \text{otherwise} \end{cases}$$

The *axioms* of the extended basic logic BL_Δ (first formulated by Baaz in [2]) are those of BL plus:

(Δ1) $\Delta\varphi \vee \neg\Delta\varphi$

(Δ2) $\Delta(\varphi \vee \psi) \to (\Delta\varphi \vee \Delta\psi)$

(Δ3) $\Delta\varphi \to \varphi$

(Δ4) $\Delta\varphi \to \Delta\Delta\varphi$

(Δ5) $\Delta(\varphi \to \psi) \to (\Delta\varphi \to \Delta\psi)$

Deduction rules of BL_Δ are modus ponens and *generalization*: from φ derive $\Delta\varphi$.

A Δ-algebra is a structure $\mathbf{L} = (L, \cap, \cup, *, \Rightarrow, 0, 1, \Delta)$ which is a BL-algebra expanded by an unary operation Δ satisfying the following conditions:

$\Delta x \cup -\Delta x = 1$

$\Delta(x \cup y) \leq \Delta x \cup \Delta y$

$\Delta x \leq x$

$\Delta x \leq \Delta\Delta x$

$(\Delta x) * (\Delta(x \Rightarrow y)) \leq \Delta y$

$\Delta 1 = 1$

The notions of **L**-evaluation and **L**-tautology easily generalize to BL_Δ and Δ-algebras. The decomposition of any BL_Δ algebra as a subdiredt product of linearly ordered ones also holds. Notice that in linearly ordered Δ-algebras we have that $\Delta 1 = 1$ and $\Delta(a) = 0$, for $a \neq 1$. Then the above completeness theorem for BL extends to BL_Δ as follows.

Theorem 4 *BL_Δ is complete, i.e. for each formula φ the following three things are equivalent:*

(i) φ is provable in BL_Δ,

(ii) for each linearly ordered Δ-algebra **L**, *φ is an* **L**-*tautology;*

(iii) for each Δ-algebra **L**, *φ is an* **L**- *tautology.*

A *strong completeness* result for provability in theories over BL_Δ is also given in [5].

Moreover, each of the three dintinguished logics, Lukasiewicz, product and Gödel logics, can be added the Δ connective, together with its axioms (Δ1) – –(Δ5), leading to the so denoted logics L_Δ, Π_Δ and G_Δ, which are complete w.r.t. their corresponding algebras, i.e. MV_Δ-algebras, Π_Δ-algebras and G_Δ-algebras. See [5] for further details.

The Basic Strict Fuzzy Logic SBL and SBL-algebras

The strict basic logic SBL [4] is an extension of BL logic for which the linearly ordered BL-algebras that satisfy SBL axioms are those having Gödel negation. The axioms of the basic strict fuzzy logic SBL are those of BL plus the following axiom:

(STR) $(\varphi\&\psi \to 0) \to ((\varphi \to 0) \vee (\psi \to 0))$.

Notice that (STR) is a theorem in both Product and Gödel logics. Moreover, SBL proves $\varphi \wedge \neg\varphi \to 0$.

Definition 4 *A SBL-algebra is a BL-algebra* $(L, \cap, \cup, *, \Rightarrow, 0, 1)$ *verifying this further condition:*

$$(x * y) \Rightarrow 0 \;=\; (x \Rightarrow 0) \cup (y \Rightarrow 0).$$

Examples of SBL-algebras are the algebras in the real unit interval $([0,1], \max, \min, *, \Rightarrow, 0, 1)$, where $*$ is a t-norm without non-trivial zero divisors and $\Rightarrow$ its corresponding residuum, and the quotient algebra SBL/$\equiv$ of provable equivalent formulas.

In linearly ordered SBL-algebras, the above condition turns into

$$x * y = 0 \text{ iff } x = 0 \text{ or } y = 0.$$

Moreover, this condition identifies linear SBL-algebras with algebras which have Gödel negation.

Theorem 5 (Completeness.) *The logic SBL is complete w.r.t. the class of linearly ordered SBL-algebras.*

Strict basic fuzzy logics extended with an involutive negation

Now we extend SBL with an unary connective $\sim$ (See [4]). The *semantics* of $\sim$ is an arbitrary strong negation function $n : [0,1] \to [0,1]$, which is nothing but a decreasing involution, i.e. $n(n(x)) = x$ and $n(x) \leq n(y)$ whenever $x \geq y$. With both negations, $\neg$ and $\sim$, the projection connective Δ is now definable: $\Delta\varphi$ is $\neg\sim\varphi$.

Definition 5 *Axioms of* SBL$_\sim$ *are those of SBL plus*

($\sim$1) $(\sim\sim \varphi) \equiv \varphi$
($\sim$2) $\neg\varphi \to \sim \varphi$
($\sim$3) $\Delta(\varphi \to \psi) \to \Delta(\sim \psi \to \sim \varphi)$
(Δ1) $\Delta\varphi \vee \neg\Delta\varphi$
(Δ2) $\Delta(\varphi \vee \psi) \to (\Delta\varphi \vee \Delta\psi)$
(Δ5) $\Delta(\varphi \to \psi) \to (\Delta\varphi \to \Delta\psi)$

where $\Delta\varphi$ *is* $\neg\sim\varphi$. *Deduction rules of* SBL$_\sim$ *are those of* BL$_\Delta$, *that is, modus ponens and necessitation for* Δ.

Definition 6 *A* SBL$_\sim$*-algebra is a structure* $\mathbf{L} = (L, \cap, \cup, *, \Rightarrow, \sim, 0, 1)$ *which is a SBL-algebra expanded with an unary operation* $\sim$ *satisfying the following conditions:*

($A_\sim$1) $\sim\sim x = x$
($A_\sim$2) $\neg x \leq \sim x$
($A_\sim$3) $\Delta(x \Rightarrow y) = \Delta(\sim y \Rightarrow \sim x)$
($A_\sim$4) $\Delta x \cup \neg\Delta x = 1$
($A_\sim$5) $\Delta(x \cup y) \leq \Delta x \cup \Delta y$
($A_\sim$6) $\Delta x * (\Delta(x \Rightarrow y)) \leq \Delta y$

where $\neg x = x \Rightarrow 0$ *and* $\Delta x = (\sim x \Rightarrow 0)$.

The decomposition of any SBL$_\sim$-algebra as a subdirect product of linearly ordered ones also holds.

Theorem 6 SBL$_\sim$ *is complete w.r.t. the class of* SBL$_\sim$*-algebras.*

Remarkable extensions of SBL$_\sim$ are the product logic with involution, $\Pi_\sim$, and Gödel's logic with involution G$_\sim$, obtained by adding the corresponding axioms to SBL$_\sim$, that is axioms (Π1) and (Π2) for the product logic, and axiom (G) for Gödel logic. For these particular extensions there are stronger completeness theorems:

- $\Pi_\sim$ is complete w.r.t. the semi-standard SBL$_\sim$-algebras $([0,1], \min, \max, \cdot, \Rightarrow_\pi, n, 0, 1)$, where $\cdot$ is usual product, $\Rightarrow_\pi$ is the residuum of product (Goguen's implication) and n is an strong negation in [0, 1].
- G$_\sim$ is complete w.r.t. to the standard G$_\sim$-algebra $([0,1], \min, \max, 1 - x, 0, 1)$.

ANNEX 2: Proof of Proposition 3

Proposition 3 *With the above definitions,* $\mathbf{R_A} = \langle G_A, +, -, \times, (0,0), (1,0), \leq_G \rangle$ *is a l.o.a. ring with unit* $(1,0)$ *and with cancellation laws for* $\times$.

We only need to prove the distributivity and the associativity and cancellation laws for the product.

Proof of the distributibity law. We must prove that $(m,z) \times ((n,x) + (k,y)) = ((m,z) \times (n,x)) + ((m,z) \times (k,y))$ and the proof needs to study different cases.

A *Proof for positive pairs*: $n, k, m \geq 0$. In turn, the proof is divided in different sub-cases:

1. If $x * y = 0$ the law is given by (3) of definition 2.
2. If $x \oplus y = 1$, and $(x \odot z) * (y \odot z) = 0$, then remembering condition (4) we have
   $(m,z) \times ((n,x) + (k,y)) = (m,z) \times (n + k + 1, x * y) =$
   $= (m(n + k + 1), ((x * y) \odot z) \oplus z) + m(1, x *$

$y) + (n+k)(0,z)$.
On the other side,$((m,z) \times (n,x)) + ((m,z) \times (k,y)) =$
$(mn, z \odot x) + m(0,x) + n(0,z) + (mk, z \odot y) + m(0,y) + k(0,z) =$
$((m(n+k), (z \odot x) \oplus (z \odot y)) + m(1, x * y) + (n+k)(0,z) =$
$((m(n+k+1), (z \odot x) \oplus (z \odot y)) + m(0, x * y) + (n+k)(0,z)$
which taking into account the conditon (4) of our suposition proves the equality.

3. If $(z \odot x) \oplus (z \odot y) = 1$, then remembering condition (4) we have
$(m,z) \times ((n,x) + (k,y)) = (m,z) \times (n+k+1, x * y) =$
$(m(n+k+1), ((x * y) \odot z)) + m(1, x * y) + (n+k+1)(0,z) =$
$(m(n+k+1)+1, ((x * y) \odot z) * z)$.
On he other hand,$((m,z) \times (n,x)) + ((m,z) \times (k,y)) =$
$(mn, z \odot x) + m(0,x) + n(0,z) + (mk, z \odot y) + m(0,y) + k(0,z) =$
$((m(n+k)+1, (z \odot x) * (z \odot y)) + m(1, x * y) + (n+k)(0,z) =$
$((m(n+k+1)+1, (z \odot x) * (z \odot y)) + m(0, x * y) + (n+k)(0,z)$
which proves the equality using condition (4) of definition 2.

B *Proof for negative pairs*:

1. First we prove the sign rule, that is, $(-(m,z)) \times (n,x) = -[(m,z) \times (n,x)]$. The proof is as follows: $(-(m,z)) \times (n,x) = (-(m+1), \neg_L z) \times (n,x) = (-(m+1)n, \neg_L z \odot x)$ which is equal to $-(mn, z \odot x)$ taking into account (iii) of Lemma 1. As a consequence it also holds that $(-(m,z)) \times (-(n,x)) = (m,z) \times (n,x)$.
2. We will prove directly the following case:
$(m,z) \times ((-(n,x)) + (k,y)) =$
$((m,z) \times (-(n,x))) + ((m,z) \times (k,y))$. The proof is as follows (suppose $k > n$):
   - Suppose $\neg x * y = 0$.
   $(m,z) \times ((-(n,x)) + (k,y)) = (m,z) \times (k-(n+1), \neg x \oplus y)$
   $= (m((k-(n+1)), z \odot (\neg x \oplus y)) + m(0, \neg x \oplus y) + (k-(n+1))(0,z)$. (1)
   On the other hand,
   $((m,z) \times (-(n,x))) + ((m,z) \times (k,y)) = (-m(n+1), z \odot \neg x) + m(0, \neg x) + (n+1)(-1, \neg z) + (mk, z \odot y) + m(0,y) + k(0,z) =$
   taking into account that $(-1, \neg z) + (0,z) = (0,0)$
   $= (m(k-(n+1)), z \odot \neg x) \oplus (z \odot y)) + m(0, \neg x * y) + (k-(n+1))(0,z)$ (2)
   Of course, by condition (4), (1) = (2).
   - Suppose $\neg x \oplus y = 1$ and $(z \odot \neg x) \oplus (z \odot y) \leq 1$.
   Then $(m,z) \times ((-(n,x)) + (k,y)) = (m,z) \times (k-n, \neg x * y) =$
   $= (m(k-n), z \odot (\neg x * y)) + m(0, \neg x * y) + (k-n)(0,z)$. (1)
   On the other hand,
   $((m,z) \times (-(n,x))) + ((m,z) \times (k,y)) = (-m(n+1), z \odot \neg x) + m(0, \neg x) + (n+1)(-1, \neg z) + (mk, z \odot y) + m(0,y) + k(0,z) =$
   $= (m(k-n) - m, (z \odot \neg x) \oplus (z \odot y)) + m(1, \neg x * y) + (n+1)(-1, \neg z) + k(0,z) =$
   taking into account that $(-1, \neg z) + (0,z) = (0,0)$
   $= (m(k-n), (z \odot \neg x) \oplus (z \odot y)) + m(0, \neg x * y) + (k-(n+1))(0,z)$. (2)
   Taking $(0,z)$ from the last term of (1) and adding this to the first term of (1), condition (4) imply (1) = (2).
   - Suppose $\neg x \oplus y = (z \odot \neg x) \oplus (z \odot y) = 1$.
   Then, $(m,z) \times ((-(n,x)) + (k,y)) = (m,z) \times (k-n, \neg x * y) =$
   $= (m(k-n), z \odot (\neg x * y)) + m(0, \neg x * y) + (k-n)(0,z)$. (1)
   On the other hand,
   $((m,z) \times (-(n,x))) + ((m,z) \times (k,y)) = (-m(n+1), z \odot \neg x) + m(0, \neg x) + (n+1)(-1, \neg z) + (mk, z \odot y) + m(0,y) + k(0,z) =$
   $= (m(k-(n+1))+1, (z \odot \neg x) * (z \odot y)) + m(1, \neg x * y) + (n+1)(-1, \neg z) + k(0,z) =$
   taking into account that $(-1, \neg z) + (0,z) = (0,0)$
   $= (m(k-n)+1, (z \odot \neg x) * (z \odot y)) + m(0, \neg x * y) + (k-(n+1))(0,z)$ (2)
   Taking $(0,z)$ from the last term of (1) and adding this to the first term of (1), condition (4) imply (1) = (2).

Thus the proof is completed.

C All other cases can be proved by A and B.2 using the sign rule B.1.

Proof of the associativity law for $\times$. From distributivity, associativity of $\times$ can be easily checked.

Proof of the cancellation law for $\times$. The cancellation law is a consequence of the preservation of strict inequality by products which, in turn, is a direct and obvious consequence of the definition of $\times$ in G_A.

CHARACTERIZATION OF REVISION OPERATOR DEFINED BY DISTANCES AND ULTRAMETRICS

Pere Garcia Calvés **Eduard Giménez Funes**
Artificial Intelligence Research Institute. IIIA
Spanish Council for Scientific Research, CSIC
08193 Bellaterra, Barcelona, Spain
http://www.iiia.csic.es

Abstract

Revision Theory is about changing our beliefs or knowledge when being exposed to new evidence. Its applicability to Machine Learning makes Revision Theory an interesting field. In this paper we characterize the revision operators defined by distances in a slightly different way to Schlechta's et al. [5]. We also provide the characterization of a new class of revision operators, those defined by an ultrametric, and finally we present a new revision operator based on measures.

1 INTRUDUCTION

Revision Theory studies how our beliefs or knowledge changes when we face up to new information or new evidence. If this information is consistent with our prior knowledge, revision amounts to adding this new information. But when we are faced up to inconsistent information with our knowledge, revision is a tricky task. This problem has attracted the attention of many different research areas as philosophy [4], logic [5] or artificial intelligence [6, 2]. Alchourrón, Gädenfors and Makinson [1, 3] studied this operator from the logic point of view. They gave some reasonable, basic properties that any revision operator should satisfy and gave a full characterization of the operator satisfying those properties. Nowadays these properties, listed below are called AGM axioms. For the sake of simplicity we will work trough the whole paper with the models or worlds representing theories. Thus K and e will be subsets of models although we may refer to them as theories. Besides, as the revision operator is not associative $A * B * C$ will stand for $(A * B) * C$. **AGM Axioms:**

- $(K * 1)$ $K * e$ is a subset of models.
- $(K * 2)$ $K * e \subseteq e$
- $(K * 3)$ $K \cap e \subseteq K * e$
- $(K * 4)$ if $K \not\subseteq \neg e$ then $K * e \subseteq K \cap e$
- $(K * 5)$ if $e = \emptyset$ then $K * e = K$
- $(K * 6)$ if $K = K'$ and $e = e'$ then $K * e = K' * e'$

Following Alchorrón's et al. work, people have tried to characterize operators satisfying even more properties. One natural direction is to define operators that capture some notion of closeness between concepts or worlds. We have followed this direction proposing three different kind of operators. In section 2 we present the revision operator defined by a distance and we give a full characterizating in a slightly different to Schlechta's et al. [5]. In section 3 we present revision operators defined by ultrametrics and we provide a full characterization. The natural interest of these operators relies on the fact, that an ultrametric defines a subset partition of the space with a natural tree structure. Finally we introduce a new revision operator, defined by measures. We introduce some intuitive motivations and provide some basic properties for these revision operators.

2 REVISION OPERATOR DEFINED BY DISTANCE

Definition 2.1 *A distance is a map* $d : W \times W \to \mathbb{R}$ *such that*

- $d(w, w') = 0$ *iff* $w = w'$
- $d(w, w') = d(w', w)$
- $d(w, w'') \leq d(w, w') + d(w', w'')$

Definition 2.2 *Let* K *and* e *be sets of models. A re-*

vision operator $*$ is defined by a distance d iff :

$$K * e = \begin{cases} \{w \in e | \exists w' \in K \\ \forall (w_1, w_2) \in K \times e \\ d(w, w') \leq d(w_1, w_2)\} & if\ K, e \neq \emptyset \\ e & if\ K = \emptyset \\ K & if\ e = \emptyset \end{cases} \quad (1)$$

Our aim is to give a full characterization of the above-mentioned operator in a slightly different way to Schlechta's et al. [5]. As in [5] we consider a propositional language with a finite number of propositional variables, that is the space of models will be finite. Our axioms concerned with sets of models and do not deal with individual points. Axiom (Sym) expresses that a distance is symmetric and Axiom (Min) expresses the fact that $K * e$ is the set of all models that minimize the distance between K and e.

2.1 BASIC PROPERTIES AND DEFINITIONS

In what follows we provide the axioms that characterize the above-defined operators along with their motivation, some definitions and properties that will be used.

- (P0) If $K = \emptyset$ then $K * e = e$

This is just a technical axiom that deals with the case of a prior inconsistent knowledge.

- (Sym) $e * K * e = K * e$

As already stated this axiom is states that a distance is symmetric.

- (Min) Suppose $B \subseteq e$, $B \cap (K * e) \neq \emptyset$, $C \subseteq K$ and $C \cap (B * K) \neq \emptyset$ then $C * B \subseteq C * e$ and $B * C \subseteq e * C$

The intuitive idea underpinning this axiom is the following one: when revising K by e we are taking not only some of the models that minimize the distance between K and e, but all of them.

Definition 2.3 *Given two sets A, B we say that A, B are (K, e)-lover sets if there exist theories (sets of models) K and e fulfilling:*

$$A \subseteq B * K$$

$$B \subseteq A * e$$

Observe that $A \subseteq K$ and $B \subseteq e$. And (B, A) are (e, K)-lover sets.

Notice that in the case of a revision defined by a distance any A, B (K, e)-lover sets are (A, B)-lover. This leads us to give the following definition[1]:

Definition 2.4 *A, B are lover sets iff there exists K, e s.t. A, B are (K, e)-lover sets.*

The following result proves their existence

Lemma 2.1 *Given any revision operator satisfying $(K * 1) - (K * 6)$, $(P0)$ and $(Pe\ 1)$, and given any pair of sets K, e, there exits $B \subseteq K$ and $A \subseteq e$ such that (A, B) are (K, e)-lover, and therefore lover sets.*

Proof: Let $B = e * K$ and $A = B * e (= e * K * e)$. So let us revise A by K. Applying $(Pe\ 1)$ we get $A * K = e * K * e * K = K * e * K = e * K = B$. Notice that we have proved that $(e * K), (K * e)$ are (K, e)-lover sets. □

We can now start talking about the relation of either being closer to ... than ... or A and B are closer to each other than C and D are.

Definition 2.5 *Given two pairs of lover sets (A, B) and (A', B'). We write $(A, B) \leq (A', B')$ if there exist theories K, e s.t. $A, A' \subseteq e$, $B, B' \subseteq K$ and A, B are (K, e)-lovers.*

Observation 2.1 *Let (A, B) and (A', B') be lover sets and all the conditions of lemma 2.1 hold, then*

$$(A, B) \leq (A', B')\ iff\ (B, A) \leq (B', A')$$

At this point we can talk about the last and most important axiom. Intuitively, states that $\leq$ can be linearly ordered. Therefore from a finite series of numbers a_i proving $a_i < a_{i+1}$ $\forall i = 1, \ldots, n_0$ then $a_0 < a_{n_0}$. And it can not be in any other way.

- (L) If $(A_1, B_1) \leq (A_2, B_2) \leq \ldots \leq (A_n, B_n) \leq (A_1, B_1)$, then for any pair of theories K, e s.t. $A_i, A_j \subseteq K$ and $B_i, B_j \subseteq e$. (A_i, B_i) are lover sets iff (A_j, B_j) are.

Observation 2.2 *Condition (L) is well defined under occurrences of permuted pairs of sets,*

$$(A_1, B_1) \leq (A_2, B_2) \leq \ldots \leq (A_n, B_n) \leq (B_1, A_1)$$

then for any pair of theories K, e s.t. $A_i, A_j \subseteq K$ and $B_i, B_j \subseteq e$ (A_i, A_i) are (K, e)-lover sets iff (A_j, B_j) are.

[1] Besides when both A and B contain one single model is related to the definition of critical pairs given in [5].

Proof: Suppose $(A_1, B_1) \leq (A_2, B_2) \leq \ldots \leq (A_n, B_n) \leq (B_1, A_1)$. Following the observation above mentioned we know $(A_n, B_n) \leq (A_1, B_1)$ iff $(B_n, A_n) \leq (B_1, A_1)$. Therefore we can build the following chain

$$(B_n, A_n) \leq (A_1, B_1) \leq (A_2, B_2) \leq \ldots (A_n, B_n) \leq (B_1, A_1)$$

After a finite number of steps we obtain:

$$(B_1, A_1) \leq (B_2, A_2) \leq \ldots \leq (B_n, A_n) \leq \ldots$$
$$\ldots \leq (A_1, B_1) \leq (A_2, B_2) \leq \ldots$$
$$\ldots \leq (A_n, B_n) \leq (B_1, A_1)$$

Finally we can apply condition (L). □

2.2 THE CHARACTERIZATION RESULT

Proposition 2.1 *Any revision operator defined by a distance satisfies (K*1)-(K*6), $(P0), (Pe1) - (Pe3), (L)$*

Proposition 2.2 *Any revision operator satisfying (K*1)-(K*6), $(P0), (Pe1) - (Pe3), (L)$ is defined by a distance.*

The key point is to prove that we can build a consistent total ordering defined over $W \times W$ capturing the distance sense. The following observation will give as a hint.

Observation 2.3 *Let w_0, w_1, w_2, w_3 any four points of W and d a distance over W then*

- *If $d(w_0, w_1) < d(w_0, w_2)$ then $w_0 * \{w_1, w_2\} = w_1$*
- *If $d(w_0, w_1) = d(w_0, w_2)$ then $w_0 * \{w_1, w_2\} = \{w_1, w_2\}$*
- *If $d(w_0, w_1) < d(w_2, w_3)$, $d(w_0, w_1) < d(w_0, w_3)$ and $d(w_0, w_1) < d(w_2, w_3)$ then $\{w_0, w_2\} * \{w_1, w_3\} = w_1$ and $\{w_1, w_3\} * \{w_0, w_2\} = w_0$*

This observation helps us to decide given a revision operator defined by a distance whether $d(w_0, w_1) < d(w_2, w_3)$ or not, by comparing the results of the different revisions proposed in the observation above. Unfortunately, this problem is not always decidable. Figure 1 shows an example. No matter whether $\delta > 0 > \epsilon > 0.1$ or $\epsilon > 0 > \delta > 0.1$, both distances define the same revision operator. The example indicates that: *'If you reach the point where neither $d(w_0, w_1) < d(w_2, w_3)$ nor $d(w_0, w_1) < d(w_2, w_3)$ can be proved then, don't worry choose the one you like the most and keep on going'.*

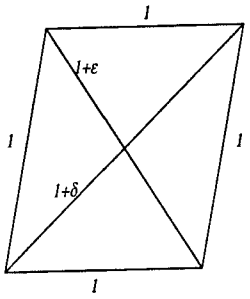


Figure 1: Example of an undecidable distance

Before starting the proof we list some technical lemmas that will ease the proof.

Lemma 2.2 *Given $(K*1)-(K*6)$, (Sym) and (Min) the following property holds.*

$(Pe\ 1)$ *$B \cap (K * e) \neq \emptyset$ and $C \cap (B * K) \neq \emptyset$ then $B \cap (C * e) \neq \emptyset$. If $B \subseteq e$ and $C \subseteq K$.*

Proof: $C * B \subseteq B$, by (Min) $C * B \subseteq C * e$ thus $B \cap C * e \neq \emptyset$ □

Lemma 2.3 *Given $(K*1)-(K*6)$, (Sym) and (Min) the following property holds.*

$(Pe\ 2)$ *$B \subseteq K * e$ then $B * K \subseteq e * K (= K * e * K)$*

Proof: Take $C = K$ then $B * K = e * K$. □

Proof: Consider the following equivalence relation $(w_0, w_1) \sim (w_2, w_3)$ iff there exists a sequence of pairs of lover sets $(w, w')_i$ $i = 0 \ldots n$ s.t. (w_0, w_1), $(w_2, w_3) \in (w, w')_i$ and $(w, w')_0 \leq (w, w')_1 \leq \ldots \leq (w, w')_n \leq (w, w')_0$. Thanks to condition (L) we can assure that the equivalence relation is well defined so we define the space:

$$X = \frac{(W \times W)}{\sim} \tag{2}$$

Let $n_i, n_j \in X$ $i \neq j$, we denote $n_i \rightarrow n_j$ iff there exists an element of each class $(w, w')_i \in n_i$ and $(w, w')_j \in n_j$ s.t. $(w, w')_i \leq (w, w')_j$ and $(w, w')_j \not\leq (w, w')_i$. $(\cup n_i, \rightarrow)$ has a natural DAG-structure (Directed Acyclic Graph), possibly not connected. Thanks to X's DAG-structure there is a mapping $\sigma : X \rightarrow I$ where I is a finite subset of $\mathbb{N}$ s.t. if $n_i \rightarrow n_j$ then $\sigma(n_i) < \sigma(n_j)$ and, from observation 2.2, if $(w, w') \in n_i$ and $(w', w) \in n_j$ then $\sigma(n_i) = \sigma(n_j)$. We define the following distance, whenever $w \neq w'$

$$d(w,w') = \frac{1}{2} + \frac{\sigma([(w,w')])}{2 * \max(I)} \quad (3)$$

Let us prove that given any pair of sets K, e then $K * e = K *_d e$. Where $*_d$ denotes the revision operator defined by d.

$\subseteq$) Consider $a \in K$, $b \in e$ s.t. $d(a,b) = \min_{x \in K,\ y \in e} \{dist(x,y)\}$ thus $b \in K *_d e$ and $a \in e *_d K$, they are (K,e)-lover sets with respect to $*_d$. Now consider $w \in K * e$ by lemma 2.3 $w * K \subseteq K * e * K = e * K$ let $w' \in w * K$ by lemma [2] 2.2 $w \in w' * e$ thus (w,w') are (K,e)-lover sets. Thus we have proved $(w,w') \leq (a,b)$ hence $w \in K *_d e$ since σ is and order preserving map hence $\sigma([(w,w')]) \leq \sigma([(a,b')])$ but $d(a,b) = \min_{x \in K,\ y \in e} \{d(x,y)\}$ so $\sigma([(w,w')]) = \sigma([(a,b')])$

$\supseteq$) Suppose $a \in K *_d e$ and let $k \in K$ s.t. $d(k,a) = \min_{x \in K, y \in e} d(x,y)$, that is $\sigma(k,a) \leq \sigma(x,y)\ \forall x,y$ in K,e. Remember that σ : *nodes* $\rightarrow \mathbb{N}$ it is an injective mapping and each node n_i represents a class of equivalence. So take $a' \in K * e$ ($\subseteq K *_d e$) there exists $k' \in a' * K$ ($\subseteq a' *_d K$ s. t. $d(k',a') = d(k,a)$ by the injectivity of σ [(k',a')]=[(k,a)] so there exists a sequence of of lover sets (A_i, B_i) $i = 1 \ldots n_0$ s.t. $(k',a'), (k,a) \in (A_i, B_i)$ and $(A_1, B_1) \leq \ldots \leq (A_n, B_n) \leq (A_1, B_1)$. Thus we can apply condition (L) and we conclude that since (a', k') is a (K,e)-lover set (k,a) it is so.

□

3 REVISION OPERATOR DEFINED BY ULTRAMETRICS

Definition 3.1 *An ultrametric is a map* $U : W \times W \rightarrow \mathbb{R}^+$.

- $U(w,w') = 0$ *iff* $w = w'$
- $U(w,w') = U(w',w)$
- $U(w,w'') \leq \max(U(w,w'), U(w',w''))$

Definition 3.2 *A revision operator is defined by an ultrametric iff* $\exists U$ *s.t.*

$$K * e = \begin{cases} \{w \in e | \exists w' \in K & \\ \forall (w_1, w_2) \in K \times e & \\ U(w,w') \leq U(w_1,w_2)\} & if\ K, e \neq \emptyset \\ e & if\ K = \emptyset \\ K & if\ e = \emptyset \end{cases} \quad (4)$$

The following definitions and properties will lead us to an easy characterization of such revision operators.

[2] $B = w$, $C = w'$

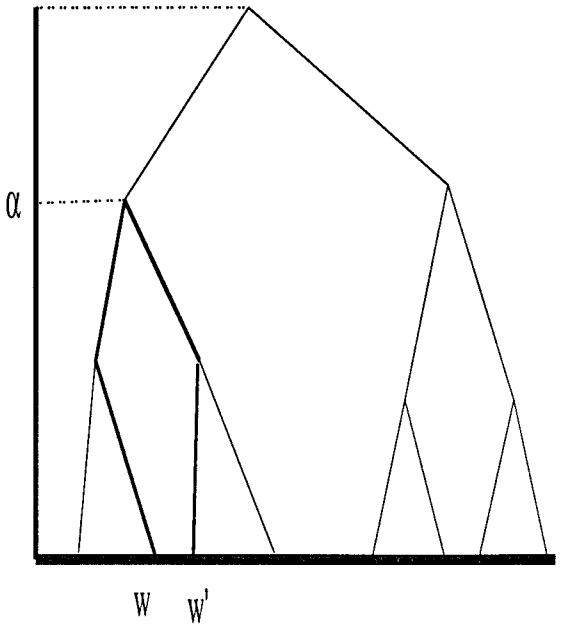


Figure 2: Tree structure of a partition induced by an Ultrametric

3.1 BASIC PROPERTIES AND DEFINITIONS

Definition 3.3 *Given* $\alpha \in Im(U) \subset \mathbb{R}$ *we define* $S_\alpha = \{C_j \in 2^W | \forall w \in W$ *and* $\forall w' \in C_j\ U(w,w') \leq \alpha$ *then* $w \in W\}$

Observation 3.1 *If* $C_i, C_j \in S_\alpha$ *and* $C_i \cap C_j \neq \emptyset$ *then* $C_i = C_j$

Proof: Since $C_i \cap C_j \neq \emptyset$ then $\exists w_0 \in C_i \cap C_j$. Let us suppose that $w_j \in C_j$ and $w_i \in C_i$. Suppose $U(w_i, w_j) \leq \max(U(w_i, w_0), (w_j, w_0)) \leq \alpha$ so this implies that $w_j \in C_i$ and $w_i \in C_j$ and hence $C_i = C_j$ □

Lemma 3.1 *Given an ultrametric defined over a finite set, it defines a unique oriented tree structure* T *and a map* $V : Nodes(T) \rightarrow I$ *s.t.* $V(v) > V(son(v))$

Proof: For each α let us consider S_α, each component $C_{j,\alpha} \in S_\alpha$ defines a node $n_{j,\alpha}$ of the tree and we assign $V(n_{j,\alpha}) = \alpha$. Given two nodes $n_{i,\alpha}, n_{l,\beta}$ they will be connected by one directed edge iff $C_{i,\alpha} \subseteq C_{l,\beta}$ and it does not exist γ s.t. $\alpha < \gamma < \beta$. The construction is well defined by the observation above. □

Figure 2 graphically describes the lemma above. All elements of W are represented as terminal nodes of T. $U(w,w')$ is least height that a path must reach in order to get from w to w'.

3.2 THE CHARACTERIZATION RESULT

At this time we should start facing the ultrametrics characterization. Revision operators defined by ultrametrics have not been characterized yet. We provide their characterization in an easy and succinct way. The next observation will give us the hint.

Observation 3.2 *Given any three points a, b, c one of them have the following property, say a:*

$$U(c,b) \leq U(a,b) \leq U(a,c) \tag{5}$$

But since $U(a,c) \leq \max(U(c,b), U(a,b))$ we deduce that

$$U(a,b) = U(a,c) \tag{6}$$

That is all possible triangles are isosceles and the base is shorter or equal than the other two sides.

So we add the following axiom.

- (U) Given (A_1, A_2), (A_2, A_3) and (A_3, A_1) lover sets if $A_i * \{A_j, A_l\} = A_j$ then $A_j * \{A_i, A_l\} = A_i$ and $A_l * \{A_i, A_j\} = \{A_i, A_j\}$

Proposition 3.1 *A revision operator, defined over a finite space, is defined by an ultrametrics iff satisfies axioms $(K*1)-(K*6), (P0), (Sym), (Min), (L), (U)$.*

Proof: Just like in the above section we can construct a map

$$\begin{array}{rccc} U: & W \times W & \to & \mathbb{N} \\ & (w, w') & \mapsto & \sigma([[w, w']]) \end{array}$$

We only have to check that given any three models w_0, w_1, w_2 $U(w_i, w_j) \leq \max(U(w_i, w_l), U(w_l, w_j))$. But this follows from condition (U) since condition (U) says that all triangles are isoseles with base shorter or equal than the other two sides. □

4 REVISION OPERATOR DEFINED BY MEASURES

4.1 MAIN CONCEPT AND BASIC DEFINITIONS

Two concerns can be made about the above defined revisions. Firstly, in a infinite space a revision defined by a distance might not be well defined since $\{w \in e | \exists w' \in K, \forall (w_1, w_2) \in K \times e \; d(w, w') \leq d(w_1, w_2)\}$ might not exists. A more pragmatic problem is the following one, the terms will be later specified but let's talk in a intuitive way for a few lines. Which is our degree of knowledge when we know K and which is our degree of knowledge when we revise by e letting to $K * e$? If K and e are consistent our degree of knowledge should increase. But if this is not the case i.e. $K \cap e = \emptyset$ as K and e are further and further apart, let's say as mutually exclusive concepts our degree of knowledge when we revise K by e should be lower and lower. With the above defined revision operators this situation is not captured by far.

The following definitions will lead us to the formalization of the above mentioned concept.

Definition 4.1 *Given a set of models B we define the knowledge degree of B with respect to a σ-additive measure $\mu : 2^B \to \mathbb{R}$ as:*

$$KD(B) = \frac{1}{\mu(B)} \tag{7}$$

That is the more I know about the environment the more complete is my theory thus there exist fewer models that satisfy it.

Example 4.1 *Let L be the propositional language defined over the propositional variables p and q.*

$$\mathcal{M}od = \{(p,q), (p,\neg q), (\neg p, q), (\neg p, \neg q)\}$$

If we assign to each model the weight of 1. So $KD(\mathcal{M}od(p)) = \frac{1}{2}$ since $\mu(\mathcal{M}od(p)) = \mu((p,q)) + \mu((p,\neg q)) = 1 + 1 = 2$

Definition 4.2 *Given a distance d over a set X and given two sets A, B we define*

$$\begin{array}{c} Dist(A,B) \\ = \\ \frac{1}{2}\Big(\max\limits_{x \in A} \min\limits_{y \in B} \{d(x,y)\} + \max\limits_{y \in B} \min\limits_{x \in A} \{d(x,y)\} \Big) \end{array} \tag{8}$$

Observation 4.1 *The following properties hold.*

1. $Dist(A,A) = 0$
2. $Dist(A,B) = Dist(B,A)$
3. $Dist(A,C) \leq Dist(A,B) + Dist(B,C)$
4. *If A and B are closed sets $Dist(A,B) = 0 \Rightarrow A = B$*

Observation 4.2 *This distance is called the Hausdorff distance and it has the following nice property*

$$\begin{array}{c} Dist(A, B \cup C) \\ \leq \\ \max(Dist(A,B), Dist(A,C)) \end{array} \tag{9}$$

We can now define the following revision operator.

Definition 4.3 *Let K be a subset of models, let*

$$e_K = \left\{ S \subseteq e \mid \int_S \frac{1}{f(d(K,x))} d\mu(x) \geq \mu(K) \right\} \quad (10)$$

where μ is a given measure and f is a strictly increasing function s.t. $f(0) = 0$ and given two sets $d(A,B) = Inf_{x \in A, y \in B} d(x,y)$, then

$$K *_D e = \begin{cases} e \text{ if } e_k = \emptyset \\ \bigcup C_\alpha \; C_\alpha \in e_K \\ \; s.t. \forall D \; D \in e_K \\ \; Dist(K, C_\alpha) \leq Dist(K, D) \end{cases} \quad (11)$$

Through the following lines we would like to motivate the definition above. $\frac{1}{d()}$ it's meant for two reasons. Firstly, if $K \cap e \neq \emptyset$ We expect $K * e = K \cap e$ but under a suitable measure $\int_{K \cap e} \frac{1}{f(d(K,x))} d\mu(x)$ will be unbounded leading us to the desired property, secondly when $d(K,e) < d(K',e)$ that is K' and e are 'more inconsistent' than K and e. We would also desire $K *_D e \subseteq K' *_D e$. Besides if $d(K,e) = d(K',e)$ and $K \subseteq K'$ we want $K *_D e \subseteq K' *_D e$ that justifies the lower bound $KD(K)$[1] in the above expression. Figure 3 pictures all this ideas.

4.2 BASIC PROPERTIES

Subsequently we provide some general basic properties and some interesting results focused on two cases of special interest. The first one is when our universe is numerable, and the second one is when our universe is a subset of $\mathbb{R}^n$ this case is specially important since it let us define revision operators over real-valued parameters spaces.

Proposition 4.1 *The above defined revision operator satisfies (K*1)-(K*6) further more satisfies $(K*7)$ $(A * e) \cap b \subseteq A * (e \cap b)$ but it does not satisfies $(K * 8)$ $A * (e \cap b) \subseteq (A * e) \cap b$*[3]

The following two propositions compare the lately defined revision operator $*_D$ with the first one $*_d$.

Proposition 4.2 *$K *_d a \subseteq K *_D a$. Therefore whenever $K *_d a$ is well defined then $K *_D a$ also is well defined.*

[3]The mentioned above axioms are taken from Alchourron's et al work.

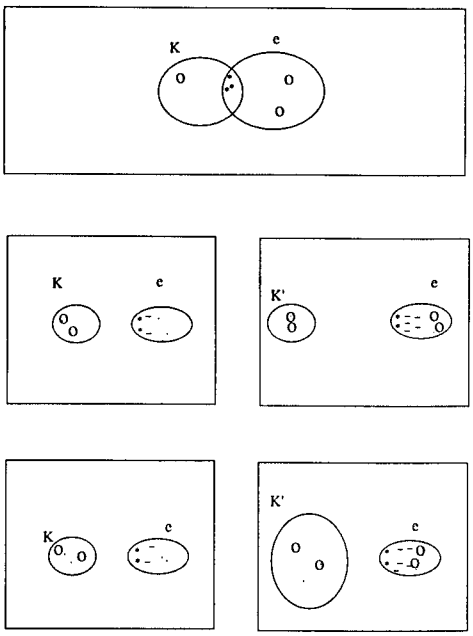


Figure 3: The symbols 0, *, - denote models. The models denoted by - belong to $K *_D e$ and the models denoted by * belong to $K *_D e \cap K * e$

In what follows we are going to be interested in the following question: *"What conditions are sufficient in order to ensure that this revision operator is well defined"*. The following result gives us a flavor of the situation.

Proposition 4.3 *Let X be a numerable space with a distance d there exists $\mu : 2^X \to \mathbb{R}$ such that $\forall x$, $x \in X$ $\mu(x) \neq 0$ and if $d(K,e) \neq 0$ then $K *_D e$ is well defined.*

Proof: Since X is numerable there exists a surjective map $n : \mathbb{N} \to X$. Let w_i a succession of real numbers such that $\sum w_i < \infty$, then given any subset S of e

$$\int_S \frac{1}{(d(K,x))} d\mu(x) = \sum_S \delta(n(i)) \frac{1}{(d(K, n(i)))} w_i < \infty \quad (12)$$

since $\frac{1}{(d(K,n(i)))}$ is bounded and $\delta(n(i)) = 1$ if $n(i) \in S$ and 0 otherwise. □

Proposition 4.4 *Given a numerable space X isometricaly embedable in $\mathbb{Q}^n$ there exists $\mu : 2^X \to \mathbb{R}$ and a function f such that if K is a finite union of $n-cubs$[4] then the revision operator is well defined.*

[4]An $n-cub$ is the Cartesian product of n intervals intersected with $\mathbb{Q}$

Proof: For the sake of simplicity we will assume that $K = [x_0, x_1]$ and $X \subseteq \mathbb{Q}$ but the proof can be easily generalized to any $\mathbb{Q}^n$.

There are basically tree possibilities

- $K \cap e \neq \emptyset$
- $\forall x,\ x \in e,\ d(K,x) > \epsilon > 0$
- $K \cap e = \emptyset$ but $\forall \epsilon\ \exists x,\ x \in e$ s.t. $d(K,x) < \epsilon$.

The last case is the interesting one since $*_d$ would not be defined.

Remember that if $\forall S,\ S \subseteq e$ and $\int_S \frac{1}{f(d(K,x))} d\mu(x) < \infty$ then $*_D$ is well defined.

Let $f(x) = \sqrt{x}$, the key idea is that $\int_{(0,\epsilon)} \frac{1}{\sqrt{x}} dx < \infty$ so we would like to find a measure over $\mathbb{Q}$ as similar to dx over $\mathbb{R}$ as possible.

An other important thing is that the superior approximation of $\int_{(0,\epsilon)} \frac{1}{\sqrt{x}} dx$ with the partition $(\frac{1}{n+1}, \frac{1}{n})$ is convergent. So we will build a measure s.t.

$$\sum_{x \in (\frac{1}{n+1}, \frac{1}{n})} \mu(x) < (\frac{1}{n} - \frac{1}{n+1}) \tag{13}$$

We will assign a weight distribution over $[0,1] \cap \mathbb{Q}$ and then it will be extended in a natural way over $\mathbb{Q}$.

W_I will denote the weight that must still be assign over the elements of $I \cap \mathbb{Q}$. w_p will denote the weight assigned to p

(1) Take $W_{(0,1)} = 1$ then $W_{(0,1/2)} + w_{1/2} + W_{(1/2,1)} \leq 1$ must hold.

(2) Let $w_{1/2}$ be any positive value < 1 and $W_{(0,1/2)} = W_{(1/2,1)} = \frac{1}{2}(1 - w_{1/2})$

(3) Now $W_{(0,1/3)} + w_{1/3} + W_{(1/3,2/3)} + w_{1/2} + W_{(1/2,2/3)} + w_{2/3} + W_{(2/3,1)} \leq 1$ must hold.

(4) Let $w_{1/3} = w_{2/3}$ be any value $< w_{1/2}$ and

$$W_{(0,1/3)} = W_{(2/3,1)} = \frac{1/3}{1/2}(W_{(0,1/2)} - w_{1/3})$$

$$W_{(1/3,1/2)} = \frac{1/2 - 1/3}{1/2}(W_{(0,1/2)} - w_{1/3})$$

At the $n-th$ step we introduce the points of the form $\frac{i}{n}$ we assign a weight much smaller than $W_{(p/q,p'/q')}$ where $\frac{p}{q}, \frac{p'}{q'}$ are the closest points to $\frac{i}{n}$ that do already have an assign weight, we distribute the remaining weight over $W_{(p/q,i/n)}$ and $W_{(i/n,p'/q')}$ proportionally to the distance of $\frac{i}{n}$ to $\frac{p}{q}$ and $\frac{p'}{q'}$. Following this procedure each rational gets a weight $\neq 0$. In order to prove that $*_D$ is well defined when K is any interval, we only have to check that

$$\sum_{x \in (x_0, x_0+\epsilon) \cap \mathbb{Q}} \frac{1}{\sqrt{x - x_0}} \mu(x) < \infty \tag{14}$$

But this is so, since for any given x_0 there exists a point of the form $\frac{p}{q}$ s.t. if $\frac{p'}{q'} \in (x_o, \frac{p}{q})$ then $q' > q$ then any interval of the form $I_i = (x_0 + \frac{1}{i+1}, x_0 + \frac{1}{i})$ s.t. $I_i \subseteq (x_0, \frac{p}{q})$ satisfies the following

$$\sum_{x \in I_i} \mu(x) \leq 2long(I_i) \tag{15}$$

so

$$\sum_{x \in (x_0, x_0+\epsilon) \cap \mathbb{Q}} \frac{1}{\sqrt{x - x_0}} \mu(x) \leq \tag{16}$$

$$\sum_{i \geq n_0}^{\infty} \frac{1}{\sqrt{(x_0 + \frac{1}{i}) - x_0}} \sum_{x \in I_i} \mu(x) \leq \tag{17}$$

$$2 \sum_{i \geq n_0}^{\infty} \frac{1}{\sqrt{(x_0 + \frac{1}{i}) - x_0}} long(I_i) < \infty \tag{18}$$

□

Proposition 4.5 *Let $X \subseteq \mathbb{R}^n$ and let $d\mu$ be the usual euclidean metric there exists f such that if $H^{n-1}(\partial K)$[5] is finite or $d(K,e) \neq 0$ then the revision is well defined.*

Proposition 4.6 *Under the same hypothesis as before $\mu(K * e)$ is monotonous increasing with this respect to $d(K,e)$.*

Corollary 4.1 *There exists a measure μ s.t. $\overline{K} *_D \overline{e}$[6] is well defined as sets embedded in $\mathbb{R}$ then $K *_D e$ is well defined.*

[5] Where H^{n-1} is the (n-1)-th Hausdorff measure

[6] Where $\overline{K}$ is the adherence of K

5 OPEN QUESTIONS AND FUTURE WORK

This results have given us a basic idea of what the general properties of the revision operator $*_D$ are. Two interesting questions still remain open.

Question 5.1 *Given a denumerable space X is there any measure μ s.t. the revision operator $*_D$ is always well defined?*

Question 5.2 *Given a distance d defined over a numerable propositional language $\mathcal{L}$ is there a measure μ s.t. if $K = Mod(\phi_1)$ and $A = Mod(\phi_2)$ there exists ϕ_3 s.t. $K * A = Mod(\phi_3)$ where ϕ_1, ϕ_2, ϕ_3 are well formed formulae?*

We have introduced some interesting revision techniques highly related to first order revision. Two major improvements remain to be taken. Firstly to provide a full characterization of the revision operator defined by measures and secondly the implementation of the above-mentioned techniques.

Acknowledgements

This research is partially supported by Spanish CICYT project TIC96-1138-C04-01 (SMASH). Eduard Giménez' research is partially supported by CIRIT-Generalitat de Catalunya, grant 1998FI 00058.

References

[1] D. Makinson C. Alchourron, P. Gardenfors. On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50:510–530, 1985.

[2] R. Mooney P. Baffes. Symbolic revision of theories with m-of-n rules. In *IJCAI-93*, pages 1135–1140, 1993.

[3] D. Makinson P. Gardenfors. On the logic of theory change: Contraction functions and their logic. *Theoria*, 48:14–37, 1982.

[4] W. Rabinowicz S. Lindstrom. The ramsey test revisited. *Theoria*, 58, 1992.

[5] K. Schlechta, D. Lehmann, and M. Magidor. Distance semantics for belief revision. *TARK'96*, pages 137–145, 96.

[6] S. Wrobel. *Advances in industive logic programming*, chapter First order theory refinement, pages 14–33. 1996.

Funciones de agregación localmente internas[1]

G. Mayor
Dpt. Matemàtiques
i Informàtica
Universitat de les
Illes Balears
07071-Palma
dmigmf0@ps.uib.es

J. Martín
Dpt. Matemàtiques
i Informàtica
Universitat de les
Illes Balears
07071-Palma

Resumen

En este trabajo se introduce el concepto de función de agregación localmente interna con dominio $\mathbb{R}^n$ y valores en $\mathbb{R}$, y se presentan una serie de resultados que conducen a la caracterización de tales funciones.

Palabras Clave. Agregación, Fórmulas bien formadas, Funciones Booleanas, Medias ponderadas, Operadores OWA.

1 Introducción

El estudio del proceso de agregación de datos (nítidos o borrosos) ha suscitado desde hace años el interés de un buen grupo de investigadores y ha dado lugar como consecuencia a abundante literatura sobre el tema ([1], [2], [3], [7]). La introducción por R. Yager del concepto de operador OWA ([6]) fue sin duda un hito importante en el estudio de funciones $f : \mathbb{R}^n \to \mathbb{R}$ que permiten modelizar la operación de agregación de una n-pla ordenada de números reales en un número real. Otro tipo de funciones ampliamente usadas en este campo son las medias ponderadas ([1]); unas y otras funciones tienen en común el hecho de ser combinaciones lineales convexas de otras funciones más simples:

$$s_i(a_1, \ldots, a_n) = \max_{1 \le j_1 < \ldots < j_i \le n} \left(\min(a_{j_1}, \ldots, a_{j_i}) \right),$$

para el caso de los operadores OWA, y

$$p_i(a_1, \ldots, a_n) = a_i,$$

para el caso de las medias. Nuestro trabajo tiene por objeto el estudio de las funciones que hemos llamado de agregación localmente internas, entre las que se encuentran como ejemplos destacados las citadas s_i y p_i.

2 Funciones de agregación localmente internas

Definición 1 Una función $f : \mathbb{R}^n \to \mathbb{R}$ diremos que es una función de agregación localmente interna si verifica:

1. Es continua
2. Es monótona creciente: si $a_i \le b_i$ para todo $i : 1, \ldots, n$, entonces $f(a_1, \ldots, a_n) \le f(b_1, \ldots, b_n)$
3. Es idempotente: $f(a, \ldots, a) = a$
4. Es localmente interna: $f(a_1, \ldots, a_n) \in \{a_1, \ldots, a_n\}$ para todo $(a_1, \ldots, a_n) \in \mathbb{R}^n$

Ejemplo 1 Para cada $i : 1, \ldots, n$ la función "proyección i-ésima" $p_i : \mathbb{R}^n \to \mathbb{R}$ definida por $p_i(a_1, \ldots, a_n) = a_i$ es una función que verifica trivialmente las condiciones 1 a 4.

Ejemplo 2 Para cada $i : 1, \ldots, n$ la función "valor i-ésimo más grande" $s_i : \mathbb{R}^n \to \mathbb{R}$ definida por

$$s_i(a_1, \ldots, a_n) = \max_{1 \le j_1 < \ldots < j_i \le n} \left(\min(a_{j_1}, \ldots, a_{j_i}) \right),$$

verifica también las cuatro condiciones. Observar que $s_1 = \max$ y $s_n = \min$. Observar también que estas funciones s_i son simétricas a diferencia de las funciones p_i que obviamente no lo son.

Ejemplo 3 La función $f : \mathbb{R}^n \to \mathbb{R}$ definida por $f(a_1, \ldots, a_n) = \max(a_1, \ldots, a_{n-1})$ es una función de agregación localmente interna que no es del tipo considerado en los ejemplos anteriores.

[1]Parcialmente financiado por la CICYT, proyecto TIC96-1393-C06-06.

Observación 1 La condición 4 de la definición 1 implica evidentemente la condición 3.

Observación 2 Las condiciones 2 y 3 (y más directamente la 4) implican

$$\min(a_1,\ldots,a_n) \leq f(a_1,\ldots,a_n) \leq \max(a_1,\ldots,a_n)$$

para todo $(a_1,\ldots,a_n) \in I\!R^n$. Obviamente esta última condición implica a su vez la condición de idempotencia.

Observación 3 Para $n = 1$, la única función que verifica las condiciones de la definición 1 es evidentemente la función identidad: $f(a) = a$ para todo $a \in I\!R$.

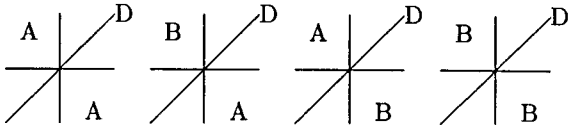
Proposición 1 En la definición 1 puede sustituirse la condición 2 por la condición 2′: si $a_i < b_i$ para todo $i : 1,\ldots,n$, entonces $f(a_1,\ldots,a_n) < f(b_1,\ldots,b_n)$

Demostración. Puede comprobarse fácilmente que las condiciones 1 y 2′ implican la condición 2. Veamos aquí que las condiciones 2 y 4 implican 2′. En efecto, supongamos $a_i < b_i$. Se trata de demostrar que $f(a) < f(b)$, en donde $f(a)$ significa $f(a_1,\ldots,a_n)$ y $f(b)$ significa $f(b_1,\ldots,b_n)$. Pongamos (condición 4) $f(a) = a_i$ y $f(b) = b_j$. Si $i = j$ ya está demostrado puesto que $a_i < b_i$. Supongamos ahora $i \neq j$; como podemos escribir (condición 2) $f(a) \leq f(b)$, se trata de ver que la hipótesis $f(a) = f(b)$, es decir $a_i = b_j$, nos conduce a una contradicción. Efectivamente, se cumpliría $f(c) = a_i = b_j$, para todo punto c del segmento ab, pero esto es absurdo ya que evidentemente existe un punto c de ab con todas sus coordenadas distintas de $a_i = b_j$. ∎

A partir de ahora indicaremos mediante $\mathcal{F}_n$ al conjunto de funciones de agregación localmente internas sobre $I\!R^n$. Veamos a continuación cómo son estas funciones en el caso $n = 2$.

Proposición 2 Las únicas funciones de agregación localmente internas sobre $I\!R^2$ son p_1, p_2, s_1 y s_2.

Demostración. En primer lugar, las funciones p_1, p_2, s_1 y s_2 pertenecen al conjunto $\mathcal{F}_2$ (ejemplos 1 y 2). Sea ahora $f : I\!R^2 \to I\!R$ continua, monótona creciente y que verifica $f(a,b) \in \{a,b\}$para todo punto (a,b) de $I\!R^2$, y sean $A = \{(a,b) \in I\!R^2;\ f(a,b) = a\}$, y $B = \{(a,b) \in I\!R^2;\ f(a,b) = b\}$. Evidentemente, $A \cup B = I\!R^2$, $A \cap B = \{(a,b) \in I\!R^2;\ a = b\} = D$. Por otra parte sean $R_1 = \{(a,b) \in I\!R^2;\ a > b\}$ y $R_2 = \{(a,b) \in I\!R^2;\ a < b\}$ las regiones limitadas por la recta $a = b$. Por la continuidad de f, A y B son conjuntos cerrados de $I\!R^2$ y también podemos decir por las propiedades de f que los conjuntos $A \cap R_i$ y $B \cap R_i$, con $i : 1,2$, son abiertos, cumpliéndose $(A \cap R_i) \cap (B \cap R_i) = \emptyset$ y $(A \cap R_i) \cup (B \cap R_i) = R_i$para $i : 1,2$. De todo ello podemos deducir que existen estas cuatro posibilidades: $A = I\!R^2$ y $B = D$, $A = R_1 \cup D$ y $B = R_2 \cup D$, $A = R_2 \cup D$ y $B = R_1 \cup D$ y $A = D$ y $B = I\!R^2$:



que corresponden respectivamente a las funciones p_1, s_1, s_2 y p_2. ∎

Observar que de las cuatro funciones anteriores sólo son simétricas s_1 y s_2, así que podemos dar la siguiente caracterización (¡una más!) (para $n = 2$) de las funciones $\max = s_1$ y $\min = s_2$:

Corolario 1 Una función $f : I\!R^2 \to I\!R$ es continua, monótona creciente, con $f(a,b) \in \{a,b\}$ para todo punto (a,b) de $I\!R^2$ y simétrica si, y sólo si, $f = \min$ ó $f = \max$.

Al plantearnos ahora el problema para $n \geq 3$, podemos considerar, de forma análoga a lo hecho para $n = 2$, los conjuntos $A_i = \{(a_1,\ldots,a_n) \in I\!R^n;\ f(a_1,\ldots,a_n) = a_i\}$. De esta forma $A_1 \cup \ldots \cup A_n = I\!R^n$ y $A_i \cap A_j \subset \{(a_1,\ldots,a_n) \in I\!R^n;\ a_i = a_j\}$. Por otra parte, si R_j son las regiones en que queda dividido $I\!R^n$ por los hiperplanos $a_i = a_j$, $i,j : 1,\ldots,n$; más concretamente $R_j = \{(a_1,\ldots,a_n) \in I\!R^n;\ a_{i_1} > \ldots > a_{i_n}\}$ siendo $i_1,\ldots,i_n$ una permutación de $1,\ldots,n$ y $j : 1,\ldots,n!$, entonces por continuidad, en cada región R_j no puede haber más que puntos de A_i para algún $i : 1,\ldots,n$; es decir ha de ser $R_j \subset A_i$. Pero a diferencia del caso $n = 2$ no todas las asignaciones $R_j \to A_i$ dan funciones de $\mathcal{F}_n$, ya que muchas de ellas no serán, por ejemplo, continuas sobre los hiperplanos. A pesar de la situación, y en el caso $n = 3$, pueden obtenerse por este procedimiento todas las funciones de $\mathcal{F}_3$ (hay 18), pero evidentemente para valores mayores de n este método resulta como mínimo tedioso.

3 Caracterización de las funciones de agregación localmente internas

Definición 2 Consideremos el alfabeto formado por los símbolos $x_1,\ldots,x_n$, $\wedge$, $\vee$, (,). A partir de él construimos el conjunto de las fórmulas bien formadas (fbf) de la manera siguiente:

1. $x_1,\ldots,x_n$ son fbf
2. Si α y β son fbf, también lo son $(\alpha \wedge \beta)$ y $(\alpha \vee \beta)$
3. No hay otras fbf aparte de las que se pueden construir a partir de 1 y 2

Ejemplo 4 En el caso n=3, el alfabeto es: x_1, x_2, x_3, $\wedge$, $\vee$, (,). Son fbf sobre este alfabeto: x_2, $x_1 \wedge x_2$, $(x_1 \wedge x_2) \vee x_3$, $(x_1 \wedge x_2) \vee (x_3 \wedge x_2)$, etc. En cambio no son fbf: $\wedge \vee x_1$, $x_1 \wedge x_2 \wedge x_3$, etc.

Para simplificar la escritura de las fbf se omiten los posibles paréntesis extremos de una fórmula: así escribimos $(x_1 \wedge x_2) \vee x_3$ en lugar de $\big((x_1 \wedge x_2) \vee x_3\big)$.

Dada una fbf α, llamamos $f_\alpha : \mathbb{R}^n \to \mathbb{R}$ a la función que aplica cada punto $(a_1, \ldots, a_n) \in \mathbb{R}^n$ en el número real que resulta al sustituir los símbolos $x_1, \ldots, x_n$ que aparecen en la fórmula por $a_1, \ldots, a_n$ respectivamente, y a continuación operar la expresión interpretando los símbolos $\wedge$ y $\vee$ como mínimo y máximo (de números reales) respectivamente.

Proposición 3 Si α es una fbf, entonces la función asociada f_α es una función de agregación localmente interna.

Demostración. En el caso $\alpha = x_i$, f_α no es más que la proyección i-ésima, y por tanto el resultado es verdadero para estas fórmulas. Para completar la demostración basta ver que si α y β son fbf con f_α y f_β funciones de agregación localmente internas entonces $f_{\alpha\wedge\beta}$ y $f_{\alpha\vee\beta}$ también son funciones de agregación localmente internas; pero esto es evidente ya que se verifica:

$$f_{\alpha\wedge\beta}(a_1, \ldots, a_n) = \min\big(f_\alpha(a_1, \ldots, a_n), f_\beta(a_1, \ldots, a_n)\big)$$

$$f_{\alpha\vee\beta}(a_1, \ldots, a_n) = \max\big(f_\alpha(a_1, \ldots, a_n), f_\beta(a_1, \ldots, a_n)\big)$$

y por tanto f_α y f_β verifican las condiciones de la definición 1. ∎

Tal como se ha señalado anteriormente, el objetivo principal de este trabajo es demostrar que cualquier función de agregación localmente interna proviene de una fbf. Más explícitamente, si g es una función de agregación localmente interna, existe una fbf α tal que $g = f_\alpha$.

Veamos ahora como tratar el problema en el caso de funciones booleanas crecientes e idempotentes.

Sea $B = \{0, 1\}$. De las 2^{2^n} funciones booleanas $f : B^n \to B$ nos interesan aquellas que son crecientes e idempotentes. Indicamos mediante $\mathcal{F}_n^*$ el conjunto de estas funciones. En este sentido se obtiene el siguiente resultado:

Proposición 4 Una función booleana $g : B^n \to B$ es creciente e idempotente si, y sólo si, existe una fbf α tal que $g = f_\alpha$.

Demostración. Si α es una fbf entonces la función f_α asociada a la fórmula es obviamente creciente e idempotente. Para demostrar el recíproco, sea $f : B^n \to B$ creciente e idempotente; consideremos el conjunto $U(f) = \{(u_1, \ldots, u_n) \in B^n;\ f(u_1, \ldots, u_n) = 1\}$. Sabemos ([5]) que por ser f booleana podemos escribir para todo $(a_1, \ldots, a_n) \in B^n$:

$$f(a_1, \ldots, a_n) = \max_{u \in U(f)} \big(\min(a_1', \ldots, a_n')\big),$$

siendo $a_i' = a_i$ si $u_i = 1$ y $a_i' = a_i^c$ si $u_i = 0$, en donde $0^c = 1$ y $1^c = 0$. Sea ahora $MU(f)$ el conjunto de puntos minimales (respecto del orden producto de B^n) del conjunto $U(f)$. Podemos entonces escribir para todo $(a_1, \ldots, a_n) \in B^n$:

$$f(a_1, \ldots, a_n) = \max_{b \in MU(f)} \Big(\max_{u \geq b} \big(\min(a_1', \ldots, a_n')\big)\Big),$$

siendo $a_i' = a_i$ si $u_i = 1$ y $a_i' = a_i^c$ si $u_i = 0$. Pero para cada $b = (b_1, \ldots, b_n) \in MU(f)$, si $b_{i_1} = \ldots = b_{i_r} = 1$ y $b_{i_{r+1}} = \ldots = b_{i_n} = 0$, entonces los puntos $u \geq b$ tienen coordenadas: $u_{i_1} = \ldots = u_{i_r} = 1$ y $u_{i_{r+1}}, \ldots, u_{i_n}$ cualesquiera. Así, para cada $b \in MU(f)$ podemos escribir:

$$\max_{u \geq b} \big(\min(a_1', \ldots, a_n')\big) = \min\Big(\min(a_{i_1}, \ldots, a_{i_r}), \max\big(\min(a_{i_{r+1}}', \ldots, a_{i_n}')\big)\Big) = \min(a_{i_1}, \ldots, a_{i_r})$$

ya que en $\min(a_{i_{r+1}}', \ldots, a_{i_n}')$ aparece el caso $\min(1, \ldots, 1) = 1$. Así nos queda finalmente:

$$f(a_1, \ldots, a_n) = \max_{b \in MU(f)} \big(\min(a_{i_1}, \ldots, a_{i_r})\big),$$

siendo $b_{i_1} = \ldots = b_{i_r} = 1$ y por tanto f deriva de una fórmula bien formada. ∎

Ejemplo 5 Sea $f : \{0,1\}^3 \to \{0,1\}$ definida por $f(0,0,0) = f(0,1,0) = f(0,0,1) = 0$ y $f(1,0,0) = f(1,1,0) = f(1,0,1) = f(0,1,1) = f(1,1,1) = 1$. Evidentemente f es creciente e idempotente. En este ejemplo $U(f) = \{100, 110, 101, 011, 111\}$ y $MU(f) = \{100, 011\}$, así que $f(a_1, a_2, a_3) = \max(\min(a_1), \min(a_2, a_3)) = \max(a_1, \min(a_2, a_3))$ para todo $(a_1, a_2, a_3) \in \{0,1\}^3$, y por tanto f deriva de la fbf $\alpha = x_1 \vee (x_2 \wedge x_3)$.

Proposición 5 La correspondencia $W : f \to f^*$ (restricción de f a $\{0,1\}^n$) es una biyección entre el conjunto $\mathcal{F}_n$ de las funciones de agregación localmente internas y el conjunto $\mathcal{F}_n^*$ de las funciones booleanas crecientes e idempotentes.

Demostración. En primer lugar, W está bien definida ya que si f verifica las cuatro condiciones de la definición 1, entonces su restricción, f^*, a $\{0,1\}^n$ toma valores en $\{0,1\}$ y además es creciente e idempotente (por serlo f).

La exhaustividad de W se demuestra fácilmente: Sea g booleana cerciente e idempotente. Sabemos por la proposición 4 que g deriva de una fbf α. Sea entonces f_α la función de $\mathcal{F}_n$ asociada a α (proposición 3); evidentemente la restricción de f_α a $\{0,1\}^n$ es g.

Veamos ahora que f_α es la única función de $\mathcal{F}_n$ cuya restricción es la función booleana dada g. En efecto, sea $f \in \mathcal{F}_n$ tal que su restricción a $\{0,1\}^n$ es g, sea $\overline{R}_j = \{(a_1,\ldots,a_n) \in I\!R^n;\ a_{i_1} \geq \ldots \geq a_{i_n}\}$, siendo $i_1,\ldots,i_n$ una permutación de $1,\ldots,n$ y veamos que f queda determinada en cada región $\overline{R}_j$ $(j : 1,\ldots,n!)$. Para no cargar la notación, supongamos que la región es $\overline{R}_1 = \{(a_1,\ldots,a_n) \in I\!R^n;\ a_1 \geq \ldots \geq a_n\}$. Observemos que en esta región están los siguientes $n+1$ puntos de $\{0,1\}^n$: $(0,0,\ldots,0)$, $(1,0,\ldots,0)$, $(1,1,\ldots,0)$,...,$(1,1,\ldots,1)$, y por la monotonía e idempotencia de g podemos escribir: $0 = g(0,0,\ldots,0) \leq g(1,0,\ldots,0) \leq g(1,1,\ldots,0) \leq \ldots \leq g(1,1,\ldots,1) = 1$, y por tanto también ($f$ y g coinciden sobre $\{0,1\}^n$) $0 = f(0,0,\ldots,0) \leq f(1,0,\ldots,0) \leq f(1,1,\ldots,0) \leq \ldots \leq f(1,1,\ldots,1) = 1$; como consecuencia existirá s: $1 \leq s \leq n$ tal que los s primeros puntos tendrán imagen (por f) 0 y los $n+s-1$ restantes tendrán imagen 1; es decir:

$$f(0,0,\ldots,0) = f(1,0,\ldots,0) = \ldots =$$
$$= f(\overbrace{1,\ldots,1}^{s-1};0,\ldots,0) = 0,\ y$$

$$f(\overbrace{1,\ldots,1}^{s-1};1,0,\ldots,0) = f(\overbrace{1,\ldots,1}^{s-1};1,1,\ldots,0) =$$
$$= \ldots = f(\overbrace{1,\ldots,1}^{s-1};1,1,\ldots,1) = 1$$

Por otra parte sabemos que f en la región considerada tiene que ser de la forma $f(a_1,\ldots,a_n) = a_p$ para un cierto p fijo, pero por lo anterior tiene que ser $p = s$ y así f queda determinada en $\overline{R}_1$. La misma argumentación podría hacerse para cualquier región $\overline{R}_j$, $j : 1,\ldots,n!$, y por tanto f queda completamente determinada en $I\!R^n$ por g. ∎

Podemos dar ahora el resultado crucial de este trabajo, que nos caracteriza las funciones de agregación localmente internas:

Proposición 6 Una función $f : I\!R^n \to I\!R$ es una función de agregación localmente interna si, y sólo si, deriva de una fórmula bien formada.

Demostración. En la proposición 3 hemos demostrado que si α es una fbf entonces la función asociada f_α es una función de agregación localmente interna. El recíproco se deduce de lo demostrado en la proposición 5: Si f es una función de agregación localmente interna, entonces f deriva de una fbf, que es la que corresponde (proposición 3) a la restricción de f a $\{0,1\}^n$. ∎

Demos finalmente una consecuencia de la proposición anterior:

Proposición 7 Las únicas funciones $f : I\!R^n \to I\!R$ de agregación localmente internas que son simétricas (conmutativas) son las de la forma

$$s_i(a_1,\ldots,a_n) = \max_{1\leq j_1<\ldots<j_i\leq n} (\min(a_{j_1},\ldots,a_{j_i})),\ (*)$$

con $i : 1,\ldots,n$.

Demostración. Por la proposición 5 podemos decir que f es simétrica si, y sólo si, f^α es simétrica. Por otra parte se sabe ([4]) que las únicas funciones booleanas que son crecientes, idempotentes y simétricas son las de la forma $(*)$, así que por lo demostrado en las proposiciones anteriores f tiene que ser también de esta forma. ∎

4 Sobre el cardinal del conjunto $\mathcal{F}_n$

El número de funciones de agregación localmente internas, $|\mathcal{F}_n|$, es, según la proposición 5, el mismo que el número, $|\mathcal{F}_n^*|$, de funciones booleanas crecientes e idempotentes. El problema de calcular el cardinal de $\mathcal{F}_n^*$ fue planteado en 1900 por Dedekind, y hasta 1988 sólo se habían hallado resultados parciales (por ejemplo, Korshunov en 1981 describe el comportamiento de $|\mathcal{F}_n^*|$ para $n \to \infty$). Es en 1988 cuando se publica el artículo de A. Kisielewicz "A solution of Dedekind's problem on he number of isotone Boolean functions" en el que al fin se resuelve el problema, basándose en una representación numérica de B^n. Es, por otra parte, interesante observar que el problema de Dedekind puede formularse también así:

Hallar el número de familias, $M_1,\ldots,M_p$, de subconjuntos de $\{1,\ldots,n\}$ tales que $M_i \neq \emptyset$ y $M_i \not\subset M_j$ para todo $i \neq j$. O también:

Hallar el cardinal del retículo distributivo libre con n generadores.

Referencias

[1] J. Aczél, T.L. Saaty. "Procedures for synthesizing ratio fudgements". *Journal of Mathematical Psychology* **27** (1983), 93-102.

[2] G. Mayor, E. Trillas. "On the representation of some Aggregation Functions. *Proceedings of the XVI IEEE-ISMVL* (1986), 110-114.

[3] G. Mayor, T. Calvo. "On Extended Aggregation Functions". *Proceedings of the IFSA'97* (1986), 281-295.

[4] R.A. Melter, S. Rudeanu. "A mesure for boolean algebras of central tendency. *Analele Stiintifice ale Universitatii Al.I.Cuza din Iasi*, Tomul XXVII, s.I. a, f.2 (1981).

[5] S. Rudeanu. "Boolean functions and equations". North-Holland Publishing Co. New York (1974).

[6] R.R. Yager. "On Ordered Weighted Averaging operators in multicriteria decision making". *IEEE Transactions on Systems, Man and Cybernetics*, **18** (1988), 183-190.

[7] H.J. Zimmermann, P. Zysno. "Latent Connectives in Human Decision Making". *Fuzzy Sets and Systems*, **4** (1980), 37-51.

SUBESPACIOS VECTORIALES T-DIFUSOS ASOCIADOS A APLICACIONES LINEALES ENTRE ESPACIOS VECTORIALES

Pedro Burillo
Dpto. Automática y Computación.
Universidad Pública de Navarra
Campus de Arrosadía
31006 Pamplona. España
e-mail: pburillo@upna.es

Luis Garmendia
Dpto. Estadística y Econometría
Universidad Carlos III de Madrid.
España
e-mail: luisgarm@est-econ.uc3m.es

Adela Salvador
Dpto. de Matemática Aplicada
Universidad Politécnica de
Madrid. E.T.S.I. Caminos
Ciudad Universitaria s.n.
28040 Madrid. España.
e-mail: ma09@dumbo.caminos.upm.es

RESUMEN

Este artículo es continuación de otros trabajos de Burillo [5], [6], y [7]. Analiza el comportamiento de la imagen por un isomorfismo, un epimorfismo, un monomorfismo o un homomorfismo de un subespacio vectorial T-difuso. En particular se estudia la imagen del subespacio vectorial difuso y del mínimo subespacio vectorial T-difuso con valores de pertenencia prefijados para una base.

Keywords: Algebra; triangular norm; vector space; fuzzy vector subspace; T-fuzzy vector subspace; homomorphism.

Palabras Clave: Álgebra; t-norma; espacio vectorial; subespacio vectorial difuso; subespacio vectorial T-difuso; homomorfismo.

1. INTRODUCCIÓN

Este artículo es continuación de otros trabajos de Burillo [5], [6], y [7], donde son estudiadas las propiedades algebraicas de los subespacios vectoriales T-difusos. Los conceptos de espacio vectorial y de homomorfismo son muy importantes en las aplicaciones ordinarias, por lo que el concepto de subespacio vectorial T-difuso inducido por un homomorfismo puede también ser aplicado. Estos conceptos se utilizan, por ejemplo, en los métodos usados para generar códigos, en las estructuras semánticas del lenguaje natural, en inferencia probabilística, en sistemas de control difuso, e incluso en el reconocimiento del camino en la conducción de vehículos. Rosenfeld [12] introdujo el concepto de grupo difuso, que fue redefinido por Anthony y Sherwood [3] utilizando la noción de norma triangular. El concepto de espacio vectorial difuso fue introducido por Katsaras y Liu [9], y estudiado por Nanda [11], Abu Osman [1], Adbukhalikov [2], y otros. Finalmente, el concepto de subespacio vectorial T-difuso fue definido por Das [8], y estudiado por Abu Osman [1] y Yu [13].

En [7] se analizó como, fijada una norma triangular T, un subespacio vectorial T-difuso contiene al mínimo subespacio vectorial T-difuso que asigna unos valores de pertenencia prefijados a los vectores de una base; y está contenido en el subespacio vectorial difuso que asigna esos mismos valores de pertenencia. En este artículo se estudia el comportamiento de los subespacios vectoriales T-difusos cuando se tiene definido un homomorfismo entre espacios vectoriales, analizándose en particular los subespacios transformados de esos dos tipos de subespacios vectoriales, estudiándose el caso en que el homomorfismo sea inyectivo, sea un epimorfismo o sea un isomorfismo.

2. PRELIMINARES

En esta sección se revisan algunas definiciones y algunos resultados que posteriormente serán usados.

Sean E y F dos espacios vectoriales ordinarios de dimensión finita sobre un cuerpo K, T una norma triangular, $\{e_1, e_2, ..., e_n\}$ una base de E, $\alpha_1, \alpha_2, ..., \alpha_n \in [0, 1]$, con $1 \geq \alpha_1 \geq \alpha_2 \geq ... \geq \alpha_n \geq 0$ y $L\{a, ..., b\}$ el subespacio vectorial ordinario engendrado por los vectores $\{a, ..., b\}$.

Definición 2.1: [5]

Un subconjunto difuso V de E se llama *subespacio vectorial T-difuso* de E si, y sólo si verifica las siguientes propiedades:
i) $V(x+y) \geq T\{V(x), V(y)\}$, para todo $x, y \in E$
ii) $V(kx) \geq V(x)$, para todo $k \in K$ y para todo $x \in E$
iii) $V(0) = 1$

Se denomina *subespacio vectorial difuso* al subespacio vectorial mín-difuso.

Se denomina V_α al subconjunto de nivel α del subconjunto difuso V.

Definición 2.2: [5]

Sea f: E $\rightarrow$ F un homomorfismo entre espacios vectoriales, sea A un subconjunto difuso de E y sea B un subconjunto difuso de F.

Se define f(A) y $f^{-1}(B)$ por:

$$f(A)(y) = \begin{cases} \text{Sup}\{A(x): f(x) = y\}, & \text{si } y \in f(E) \\ 0, & \text{si } y \notin f(E); \end{cases}$$

y $f^{-1}(B)(x) = B(f(x))$, para todo $x \in E$.

Proposición 2.1: [5]

Si V es un subespacio vectorial T-difuso de E y T es una norma triangular semicontinua inferiormente, entonces f(V) es un subespacio vectorial T-difuso de F.

Si W es un subespacio vectorial T-difuso de F entonces:

$f^{-1}(W)$ es un subespacio vectorial T-difuso de E.

Proposición 2.2: [7]

Si V es un subespacio vectorial T-difuso tal que asigna a los vectores de una base los valores de pertenencia: $V(e_k) = \alpha_k$, para todo k, $1 \leq k \leq n$. Entonces :

1. $\{0\} \subset V_1$ y $L\{e_k\} \subset V_{\alpha k}$
2. $L\{e_k, e_j, ..., e_s\} \subset V_{T\{\alpha k, \alpha j, ..., \alpha s\}}$
3. $E = V_{T\{\alpha 1, \alpha 2, ..., \alpha n\}}$

Consecuencia 1: Si T es igusl a Mín entonces se verifica el teorema de representación de Lowen.

Consecuencia 2: Si T es una norma triangular continua, y si W_T es el mínimo subespacio vectorial T-difuso tal que $W_T(e_k) = \alpha_k$, para todo k, $1 \leq k \leq n$, entonces W_T tiene a lo sumo 2^n subconjuntos de nivel que son unión de subespacios vectoriales ordinarios de la forma: $L\{e_k, ..., e_s\} \cup ... \cup L\{e_j, ..., e_i\}$

3. SUBESPACIO VECTORIAL T-DIFUSO ASOCIADO A UN ISOMORFISMO

Proposición 3.1:

Sea T una norma triangular semicontinua inferiormente. Si f es un isomorfismo entre los espacios vectoriales E y F, y V es un subespacio vectorial T-difuso de E entonces, f(V) es un subespacio vectorial T-difuso isomorfo a V y $f(V)(f(e_k)) = \alpha_k$, $f(V_\alpha) = (f(V))_\alpha$ para cualquier valor α.

Demostración:

Sea V un subespacio vectorial T-difuso de E, inducido por la base $\{e_1, e_2, ..., e_n\}$ con $V(e_k) = \alpha_k$; al ser f una biyección entonces para todo vector x de E, existe un único vector *y* de F, tal que y = f(x), luego $f(V)(y) = \text{Sup}\{V(x): f(x) = y\} = V(x)$. Por tanto la base $\{e_1, e_2, ..., e_n\}$ se transforma en una base de F, $\{f(e_1), f(e_2), ..., f(e_n)\}$ siendo f(V) el subespacio vectorial T-difuso de F tal $f(V)(f(e_k)) = V(e_k) = \alpha_k$.

Cada subconjunto de nivel de V, V_α se transforma en un subconjunto de nivel isomorfo en f(V), $(f(V))_\alpha$, verificándose que, $f(V_\alpha) = (f(V))_\alpha$ para todo valor α. ■

Corolario 3.2:

Sea T una norma triangular semicontinua inferiormente. Si f: E $\rightarrow$ F es un isomorfismo entre espacios vectoriales y V es un subespacio vectorial difuso de E, tal que $V(e_k) = \alpha_k$, para todo k, $1 \leq k \leq n$, entonces f(V) es un subespacio vectorial difuso de F con a lo sumo n+1 subconjuntos de nivel $\{0\} \subset f(V)_1 \subset f(V)_{\alpha 1} \subset .... \subset f(V)_{\alpha k} \subset ... \subset f(V)_{\alpha n} = F$ que son subespacios vectoriales ordinarios de la forma $(f(V))_{\alpha k} = L\{f(e_1), f(e_2), ..., f(e_k)\}$.

Demostración:

En el caso particular de que V sea un subespacio vectorial mín-difuso sabemos que por el teorema de representación de Lowen [10], sus subconjuntos de nivel son $\{0\} \subset V_1 \subset V_{\alpha 1} \subset .... \subset V_{\alpha k} \subset ... \subset V_{\alpha n} = E$ luego entonces f(V) es un subespacio vectorial mín-difuso de F de subconjuntos de nivel $\{0\} \subset f(V)_1 \subset f(V)_{\alpha 1} \subset .... \subset f(V)_{\alpha k} \subset ... \subset f(V)_{\alpha n} = F$ siendo $f(V)_{\alpha k}$ isomorfo a $V_{\alpha k}$, y como $V_{\alpha k}$ es el subespacio vectorial ordinario engendrado por $\{e_1, ..., e_k\}$, entonces $(f(V))_{\alpha k}$ es el subespacio vectorial ordinario engendrado por $\{f(e_1), ..., f(e_k)\}$. ■

Corolario 3.3:

Sea T una norma triangular semicontinua inferiormente. Si f es un isomorfismo de E en F y W_T es el mínimo subespacio vectorial T-difuso de E inducido por la base $\{e_1, ..., e_n\}$ con $W_T(e_k) = \alpha_k$, para todo k, $1 \leq k \leq n$, entonces $f(W_T)$ es el mínimo subespacio vectorial T-difuso de F inducido por la base $\{f(e_1), ..., f(e_n)\}$ con valores de pertenencia $f(W_T)(f(e_k)) = \alpha_k$ y tiene a lo sumo 2^n subconjuntos de nivel que son unión de subespacios vectoriales ordinarios de la forma $L\{f(e_k), ..., f(e_s)\} \cup ... \cup L\{f(e_j), ..., f(e_i)\}$.

Demostración:

En el caso particular en que W_T sea el mínimo subespacio vectorial T-difuso de E inducido por la base $\{e_1, ..., e_n\}$ entonces $W_T(e_k) = \alpha_k$, para todo k, $1 \leq k \leq n$, y

sus $N \leq 2^n$ subconjuntos de nivel son $(W_T)_1$, $(W_T)_{\alpha k}$, $(W_T)_{T\{\alpha k,\alpha j,...,\alpha s\}}$ siendo $(W_T)_\alpha$ unión de subespacios vectoriales ordinarios $L\{e_k, ..., e_s\} \cup ... \cup L\{e_j, ..., e_i\}$, entonces como consecuencia de la proposición 3.1 $f(W_T)$ es el mínimo subespacio T-difuso de F inducido por la base $\{f(e_1), ..., f(e_n)\}$ y tiene $N \leq 2^n$ subconjuntos de nivel de la forma $f(W_T)_1$, $f(W_T)_{\alpha k}$, ...,$f(W_T)_{T\{\alpha k, \alpha j, ..., \alpha s\}}$ y por ser $f((W_T)_\alpha)$ igual a $(f(W_T))_\alpha$ entonces $(f(W_T))_\alpha$ es unión de subespacios vectoriales ordinarios engendrados por los vectores $\{f(e_j)\}$, es decir, es de la forma $L\{f(e_k), ..., f(e_s)\} \cup ... \cup L\{f(e_j), ..., f(e_i)\}$. ■

4. SUBESPACIO VECTORIAL T-DIFUSO ASOCIADO A UNA APLICACIÓN LINEAL INYECTIVA

Sea E un espacio vectorial de dimensión n y sea F un espacio vectorial de dimensión m, $n<m$. Sea $\{e_1, e_2, ..., e_n\}$ una base de E, y $\alpha_1, \alpha_2, ..., \alpha_n \in [0, 1]$, tales que $1 \geq \alpha_1 \geq \alpha_2 \geq ... \geq \alpha_n \geq 0$. Sea $f: E \rightarrow F$ una aplicación lineal inyectiva. Sea T una norma triangular semicontinua inferiormente.

Proposición 4.1:

Si V es un subespacio vectorial T-difuso de E tal que $V(e_k)=\alpha_k$, entonces f(V) es el subespacio vectorial T-difuso de F tal que sus subconjuntos de nivel $(f(V))_\alpha$, si $\alpha \neq 0$, son isomorfos a los subconjunto de nivel de V, $f(V_\alpha) = (f(V))_\alpha$, y si $\alpha=0$ entonces $(f(V))_0 = F$. Si $T\{\alpha_1, \alpha_2, ..., \alpha_n\}$ es distinto de cero, entonces Imf coincide con $f(V)_{T\{\alpha 1,\alpha 2,...,\alpha n\}}$.

Demostración:

Si f es una aplicación inyectiva entonces, si existe un x tal que $y = f(x)$, x es único, por tanto:

$$f(V)(y)=\begin{cases} \text{Sup}\{V(x): f(x)=y\}=V(x), & \text{si } y \in f(E) \\ 0, & \text{si } y \notin f(E); \end{cases}$$

Como el conjunto $\{f(e_1), ..., f(e_n)\}$ es linealmente independiente y es una base de Imf, pero no es necesariamente una base de F, completamos este conjunto con m-n vectores de F que formen una base, $\{f(e_1), ..., f(e_n), u'_{n+1},..., u'_m\}$ y todo vector *y* de F puede escribirse

$y = \Sigma y_k.f(e_k) + \Sigma y_s.u'_s$

Si $y \in$ Imf, $y = f(x)$ entonces $(f(V))(y) = V(x)$, y en particular, los transformados de la base son: $(f(V))(f(e_j))=V(e_j)$. Existe una biyección entre E e Imf, por tanto si $\alpha \neq 0$ cada subconjunto de nivel $(f(V))_\alpha$ es isomorfo al subconjunto de nivel V_α. Si $y \in f(E)$, entonces $f(V)(y) = f(V)(y_1 f(e_1) + ... + y_n f(e_n)) \geq T\{f(V)(f(e_1)), ..., f(V)(f(e_n))\} = T\{V(e_1), ... V(e_n)\} = T\{\alpha_1, \alpha_2, ..., \alpha_n\}$, luego $y \in f(V)_{T\{\alpha 1,\alpha 2,...,\alpha n\}}$, por tanto Imf $\subset f(V)_{T\{\alpha 1,\alpha 2,...,\alpha n\}}$. Si $y_k, y_j, ..., y_s$ son distintos de cero entonces $f(V)(y) \geq T\{\alpha_k, \alpha_j, ..., \alpha_s\}$. Si $T\{\alpha_1, \alpha_2, ..., \alpha_n\}$ es distinto de cero e $y \in f(V)_{T\{\alpha 1,\alpha 2,...,\alpha n\}}$ entonces se verifica que $f(V)(y) \geq T\{\alpha_1, \alpha_2, ..., \alpha_n\} \neq 0$, luego y pertenece a Imf, e Imf = $f(V)_{T\{\alpha 1,\alpha 2,...,\alpha n\}}$.

Pero si $y \notin$ Imf entonces $f(V)(y) = 0$. En consecuencia f(V) tiene a lo sumo tantos subconjuntos de nivel como V más uno, $f(V)_0$: $f(V)_1 \subset f(V)_{\alpha 1} \subset .... \subset f(V)_{T\{\alpha k,\alpha j, ...,\alpha s\}} \subset ... \subset f(V)_{T\{\alpha 1,\alpha 2, ...,\alpha n\}} \subset f(V)_0 = F$. ■

Corolario 4.2:

Si f es una aplicación inyectiva y V es un subespacio vectorial difuso de E, tal que $V(e_k)=\alpha_k$, entonces f(V) es el subespacio vectorial difuso de F que tiene a lo sumo $n + 2 \leq m + 1$ subconjuntos de nivel siendo, si $\alpha_k \neq 0$, $f(V)_{\alpha k} = L\{f(e_1), f(e_2), ..., f(e_k)\}$ y $f(V)_0 = F$.

Demostración:

Como consecuencia inmediata de la proposición anterior, si V es un subespacio vectorial Mín-difuso, entonces f(V) es también un subespacio vectorial min-difuso y puede tener un subconjunto de nivel más que V, y lo tiene si α_n es distinto de cero, siendo entonces $f(V)_{\alpha n} = Imf$ y $f(V)_0 = F$. Como el subconjunto de nivel $(f(V))_\alpha$ es isomorfo al subconjunto de nivel V_α si $\alpha \neq 0$, tenemos que $\{0\} \subset f(V)_1 \subset f(V)_{\alpha 1} \subset .... \subset f(V)_{\alpha k} \subset ... \subset f(V)_{\alpha n} \subset f(V)_0 = F$ con $f(V)_{\alpha k} = L\{f(e_1), f(e_2), ..., f(e_k)\}$. ■

Corolario 4.3:

Si f es una aplicación lineal inyectiva de E en F, y W_T es el mínimo subespacio vectorial T-difuso inducido por la base $\{e_1, e_2, ..., e_n\}$, con valores de pertenencia $\alpha_1 \geq \alpha_2 \geq ... \geq \alpha_n$ entonces $f(W_T)$ es el mínimo subespacio vectorial T-difuso de F inducido por la base $\{f(e_1), ..., f(e_n), u'_{n+1},..., u'_m\}$ con valores de pertenencia $(f(W_T))f(e_k)=\alpha_k$ y $(f(W_T))u'_r=0$, y a lo sumo $2^n+1 \leq 2^m$ subconjuntos de nivel donde $f(W_T)_\alpha$, si $\alpha \neq 0$, es unión de subespacios vectoriales ordinarios de la forma $L\{f(e_k), ..., f(e_s)\} \cup ... \cup L\{f(e_j), ..., f(e_i)\}$ y $f(W_T)_0 = F$

Demostración:

$f(W_T)$ es el subespacio vectorial T-difuso de F inducido por la base $\{f(e_1), ..., f(e_n), u'_{n+1},..., u'_m\}$ ampliado al valor de pertenencia cero si y no pertenece a Imf. Por tanto $f(W_T)$ tiene a lo sumo $2^n + 1$ subconjuntos de nivel,

siendo $f(W_T)_\alpha$, si $\alpha \neq 0$ isomorfo a $(W_T)_\alpha$. Al ser $(W_T)_\alpha$ unión de subespacios vectoriales de la forma $L\{e_j, ..., e_s\}$entonces $f(W_T)_\alpha$ es unión de subespacios vectoriales de la forma $L\{f(e_j), ..., f(e_s)\}$.

Si $T\{\alpha_1, \alpha_2, ..., \alpha_n\}$ es distinto de cero entonces $f(W_T)_{T\{\alpha 1, \alpha 2, ..., \alpha n\}} = Imf$ y $f(W_T)_0 = F$, siendo $f(V)(y) = 0$ cuando *y* pertenece a F-Imf. Pero si $T\{\alpha_1, \alpha_2, ..., \alpha_n\}$ es igual a cero entonces $f(W_T)_{T\{\alpha 1,\alpha 2,...,\alpha n\}} = f(W_T)_0 = F$. ■

5. SUBESPACIO VECTORIAL T-DIFUSO ASOCIADO A UN EPIMORFISMO

Proposición 5.1:

Sea T una norma triangular semicontinua inferiormente. Si f es una aplicación lineal suprayectiva de E sobre F, siendo E y F espacios vectoriales de dimensiones n y m respectivamente, r la dimensión del núcleo y V el subespacio vectorial difuso de E inducido por la base $\{e_1, e_2, ..., e_n\}$, tal que $V(e_j) = \alpha_j$, entonces f(V) es el subespacio vectorial difuso de F inducido por una base de $Imf = F$, $\{f(e_{j1}), f(e_{j2}), ..., f(e_{js})\}$, siendo $s = n-r$; $f(V)(f(e_{jk})) = V(e_{jk}) = \alpha_{jk}$, y tiene f(V) a lo sumo s + 1 subconjuntos de nivel: $f(V)_1 \subset f(V)_{\alpha j1} \subset ... \subset f(V)_{\alpha js}$ donde $f(V)_1$ contiene a $f(Kerf) = \{0\}$; $f(V)_{\alpha j1} = L\{f(e_{j1})\}$; $f(V)_{\alpha jk} = L\{f(e_{j1}), ..., f(e_{jk})\}$; $f(V)_{\alpha js} = L\{f(e_{j1}), ..., f(e_{js})\} = Imf = F$.

Demostración:

Sabemos que E/Kerf es isomorfo a Imf = F. Buscamos una base de Kerf $\{u_1, u_2, ..., u_r\}$ y la ampliamos de forma ordenada con los vectores de la base que genera al subespacio Mín-difuso V, es decir si $1 \geq \alpha_1 \geq ... \geq \alpha_n$ entonces probamos si $\{u_1, ..., u_r, e_1\}$ es linealmente independiente. Si lo es, incluimos e_1 en la nueva base, y si no lo es probamos con e_2, y así sucesivamente. Construimos de esta forma una nueva base de E $\{u_1, ..., u_r, e_{j1}, ..., e_{js}\}$ donde $V(e_{jk}) = \alpha_{jk}$ y $\alpha_{j1} \geq ... \geq \alpha_{jk} \geq ... \geq \alpha_{js}$.

Si e_1 pertenece a la nueva base $f(V)(f(e_1)) = Sup_{f(x)=f(e1)}\{V(x)\}$ y como x es distinto de cero, el máximo valor que puede tomar V(x) es α_1 pues $\{0\} = V_1 \subset V_{\alpha 1} \subset ... \subset V_{\alpha n}$ luego $f(V)(f(e_1)) = \alpha_1$. Si e_1 no pertenece a la nueva base, entonces $\{u_1, ..., u_r, e_1\}$ es linealmente dependiente, por lo que $e_1 = \Sigma l_j u_j$, luego e_1 pertenece a Kerf, y por tanto $f(V)(f(e_1)) = 1$.

Sea e_{j1} el primer vector que añadimos a la base, entonces $f(V)(f(e_{j1})) = Sup_{f(x)=f(ej1)}\{V(x)\}$, luego x debe ser igual a $e_{j1} + \Sigma m_k u_k = e_{j1} + v$, donde v pertenece a Kerf ya que $f(x) = f(e_{j1}+v) = f(e_{j1}) + f(v) = f(e_{j1})$. Luego x pertenece a $L\{u_1, ..., u_r, e_{j1}\}$, y por tanto para que x no pertenezca a Kerf el mayor valor de pertenencia debe ser α_{j1}. Vamos a comprobarlo:

Caso a: Si $V(u_k) \leq V(e_{j1})$ en este caso $Sup_{f(x)=f(ej1)}\{V(x)\} = Sup_{f(x)=f(ej1)}\{Mín\{V(u_1), ..., V(u_r), V(e_{j1})\} = V(e_{j1}) = \alpha_{j1}$.

Caso b: Si $V(v) \geq V(e_{j1})$. Podemos suponer que $V(v) \geq V(x)$ pues en particular $V(0) = 1 \geq V(x)$. Como $x = e_{j1} + v$, entonces $V(x) \geq Mín\{V(e_{j1}), V(v)\} = \alpha_{j1}$ y despejando $e_{j1} = x - v$, luego $V(e_{j1}) = \alpha_{j1} \geq Mín\{V(x), V(v)\} = V(x)$, y por tanto $V(x) = \alpha_{j1} = V(e_{j1})$.

Análogamente se demuestra que $f(V)(f(e_{jk})) = \alpha_{jk}$ si e_{jk} pertenece a la base construida.

La aplicación suprayectiva *g* que transforma E en E/Kerf nos transforma la base ampliada $\{u_1, ..., u_r, e_{j1}, ..., e_{js}\}$ en la base de E/Kerf $\{e_{j1}+Kerf, ..., e_{js}+Kerf\}$ siendo $g(V)(e_{jk} + Kerf) = \alpha_{jk}$. Entre E/Kerf y Imf = F existe un isomorfismo luego una base de Imf = F es $\{f(e_{j1}), ..., f(e_{js})\}$, por tanto f(V) es el subespacio Mín-difuso de Imf = F definido por la base $\{f(e_{j1}), ..., f(e_{js})\}$ con valores de pertenencia $f(V)(f(e_{jk})) = V(e_{jk}) = \alpha_{jk}$, y f(V) tiene a lo sumo s+1 subconjuntos de nivel $f(V)_1 \subset f(V)_{\alpha j1} \subset ... \subset f(V)_{\alpha js} = Imf = F$ donde $f(V)_1$ contiene a $f(Kerf) = \{0\}$, y si α_{j1} es distinto de 1 entonces $f(V)_1 = f(Kerf) = \{0\}$; $f(V)_{\alpha j1} = L\{f(e_{j1})\}$; $f(V)_{\alpha jk} = L\{f(e_{j1}), ..., f(e_{jk})\}$; $f(V)_{\alpha js} = L\{f(e_{j1}), ..., f(e_{js})\} = Imf = F$.

Veamos ahora cuanto vale $f(V)(f(e_j))$ si e_j no pertenece a la base ampliada. Si e_j pertenece a Kerf entonces $f(V)(f(e_j)) = 1$. Si e_j no pertenece a Kerf entonces si α_j es el valor de pertenencia comprendido entre $\alpha_{jk+1} < \alpha_j \leq \alpha_{jk}$ al ser $e_j = \Sigma l_i u_i + m_1 e_{j1} + ... + m_s e_{js}$, $V(e_j) = \alpha_j \geq Mín\{V(u_i), V(e_{js})\} > \alpha_{jk+1}$, los coeficientes de e_{jk+1} y siguientes deben valer 0, ya que hemos supuesto que $\alpha_{jk+1} < \alpha_j$, por lo que e_j no puede ser combinación lineal de $e_{jk+1}, ..., e_{js}$ y por tanto $e_j = \Sigma l_i u_i + m_1 e_{j1} + ... + m_k e_{jk}$ y como $g(e_j)$ pertenece a $L\{e_{j1}+Kerf, ..., e_{jk}+Kerf\}$ entonces $f(e_j)$ pertenece a $L\{e_{j1}, ..., e_{jk}\} = f(V)_{\alpha jk}$. ■

Corolario 5.2:

Sea T una norma triangular semicontinua inferiormente. Si f es un epimorfismo entre E y F, T es una norma triangular semicontinua inferiormente, y V es un subespacio vectorial T-difuso tal que $V(e_j) = \alpha_j$, entonces f(V) es un subespacio vectorial T-difuso tal que:

i) $\{0\} = f(Kerf) \subset f(V)_1$

ii) Si $\{e_{j1}, ..., e_{js}\}$ son los vectores de $\{e_1, ..., e_n\}$ con los que hay que ampliar una base de Kerf para obtener una base de E, entonces $f(V)(f(e_{jk})) = V(e_{jk}) = \alpha_{jk}$

iii) $f(V)_{T\{\alpha_{j1},...,\alpha_{js}\}} = Imf = F$

Demostración:

i) Si x pertenece a Kerf entonces $f(V)(f(x)) = 1$ ya que entonces $f(x) = 0$ y $f(V)(f(x)) = Sup_{f(x')=0}\{V(x')\} = Sup_{f(x')=0}\{V(0), V(x')\} = Sup\{1, ...\} = 1$. Por tanto $f(Kerf) \subset f(V)_1$

ii) Buscamos una base de Kerf, $\{u_1, ..., u_r\}$ y la ampliamos como en la proposición anterior, de forma ordenada con los vectores de $\{e_1, ..., e_n\}$, construyendo una nueva base de E $\{u_1, ..., u_r, e_{j1}, ..., e_{js}\}$ donde $V(e_{jk}) = \alpha_{jk}$ y $\alpha_{j1} \geq ... \geq \alpha_{jk} \geq ... \geq \alpha_{js}$.

$f(V)(f(e_{jk})) \geq Sup_{f(x)=f(ejk)}\{V(x)\}$, pero para que $f(x) = f(e_{jk})$ debe ser $x = e_{jk} + v$, donde $v \in Kerf$; luego entonces $V(x) \geq T\{V(e_{jk}), V(v)\}$ y por tanto $f(V)(f(e_{jk})) \geq Sup_{f(x)=f(ejk)}\{T\{V(e_{jk}), V(v)\}, V(e_{jk})\} \geq \alpha_{jk}$. Por otro lado sabemos que Mín $\geq$ T, para toda norma triangular T, luego si W es el subespacio vectorial difuso que verifica que $W(e_j) = \alpha_j$ entonces $W \geq V$. En la proposición anterior hemos probado que $f(W)(f(e_{jk})) = \alpha_{jk}$, luego se debe verificar que $\alpha_{jk} \geq f(V)(f(e_{jk})) = Sup_{f(x)=f(ejk)}\{V(x)\} \geq \alpha_{jk}$ y por tanto $f(V)(f(e_{jk})) = \alpha_{jk}$.

iii) Si $\{u_1, ..., u_r, e_{j1}, ..., e_{js}\}$ es una base de E, entonces $\{e_{j1}+Kerf, ..., e_{js}+Kerf\}$ es una base de E/Kerf, y como E/Kerf es isomorfo a Imf entonces $\{f(e_{j1}), ..., f(e_{js})\}$ es una base de Imf. Sabemos que f(V) es un subespacio vectorial T-difuso de Imf y por tanto para todo *y* de Imf, $f(V)(y) = f(V)(\Sigma l_k f(e_{jk})) \geq T\{\alpha_{j1}, ..., \alpha_{js}\}$, luego *y* pertenece a $f(V)_{T\{\alpha_{j1},..., \alpha_{js}\}}$, y $f(V)_{T\{\alpha_{j1},..., \alpha_{js}\}} = Imf = F$ si f es sobreyectiva. ■

Proposición 5.3:

Sea T una norma triangular semicontinua inferiormente. Si f es un epimorfismo entre E y F, T es una norma triangular semicontinua inferiormente, y V es el mínimo subespacio vectorial inducido por la base $\{e_1, ..., e_n\}$ siendo $V(e_j) = \alpha_j$, para todo j, $1 \leq j \leq n$, entonces f(V) es el mínimo subespacio vectorial T-difuso de F inducido por la base $\{f(e_{j1}), ..., f(e_{js})\}$ y por tanto tiene a lo sumo 2^s subconjuntos de nivel de la forma: $f(V)_{T\{\alpha jk,..., \alpha jp\}}$ que son unión de subespacios vectoriales ordinarios de la forma $L\{f(e_{jk}), ..., f(e_{jp})\}$.

Demostración:

Es consecuencia inmediata de la proposición anterior y de la construcción de los mínimos subespacios vectoriales T-difusos asociados a una base. ■

6. SUBESPACIO VECTORIAL T-DIFUSO ASOCIADO A UN HOMOMORFISMO

Proposición 6.1:

Sea T una norma triangular semicontinua inferiormente. Si f es una aplicación lineal de E en F, T una norma triangular semicontinua inferiormente, y V es un subespacio vectorial T-difuso de E tal que $V(e_j) = \alpha_j$, para todo j, $1 \leq j \leq n$, entonces f determina en F un subespacio vectorial T-difuso f(V) tal que

i) $f(Kerf) = \{0\} \subset f(V)_1$ y si α_{j1} es distinto de 1 entonces $f(V)_1 = f(Kerf)$

ii) Si $\{e_{j1}, ..., e_{js}\}$ son los vectores de $\{e_1, ..., e_n\}$ con los que se amplia una base de Kerf para obtener una base de E, entonces $f(V)(f(e_{jk})) = V(e_{jk}) = \alpha_{jk}$

iii) $Imf \subset f(V)_{T\{\alpha j1,..., \alpha js\}}$

iv) Si $T\{\alpha_{j1}, ..., \alpha_{js}\}$ es distinto de 0 entonces: $Imf = f(V)_{T\{\alpha j1,..., \alpha js\}}$; $f(V)(y)=0$ si y sólo si y pertenece a F - Imf; $f(V)_0 = F$

Demostración:

Sea $\{e_1, ..., e_n\}$ la base de E tal que $V(e_j) = \alpha_j$, para todo j, $1 \leq j \leq n$, y sea $\{u_1, ..., u_r\}$ una base de Kerf. Sea s = n-r = dimE – dimKerf. La ampliamos de forma ordenada con vectores de la base $\{e_1, ..., e_n\}$ hasta obtener otra base de E: $\{u_1, ..., u_r, e_{j1}, ..., e_{js}\}$ tal que $1 \geq \alpha_{j1} \geq ... \geq \alpha_{js} \geq 0$. Entonces $\{e_{j1}+Kerf, ..., e_{js}+Kerf\}$ es una base de E/Kerf y al ser E/Kerf isomorfo a Imf, $\{f(e_{j1}), ..., f(e_{js})\}$ es una base de Imf. Ampliamos esta base hasta obtener una base de F $\{f(e_{j1}), ..., f(e_{js}), u'_{s+1}, ..., u'_m\}$.

Sabemos que $f(Kerf) \subset f(V)_1$ y $f(V)(f(e_{jk})) = V(e_{jk}) = \alpha_{jk}$. Si α_{j1} es distinto de 1 entonces $f(V)_1 = f(Kerf)$.

Si *y* pertenece a Imf entonces $y = \Sigma m_k.f(e_{jk})$, $f(V)(y) = f(V)(\Sigma m_k.f(e_{jk})) \geq T\{\alpha_{j1}, ..., \alpha_{js}\}$. Si $T\{\alpha_{j1}, ..., \alpha_{js}\}$ es distinto de 0, y si *y* pertenece a $f(V)_{T\{\alpha j1,..., \alpha js\}}$ entonces $f(V)(y) \geq T\{\alpha_{j1}, ..., \alpha_{js}\} > 0$, luego $Imf = f(V)_{T\{\alpha j1,..., \alpha js\}}$. Si *y* no pertenece a Imf, $f(V)(y)=0$ luego si $T\{\alpha_{j1}, ..., \alpha_{js}\}$ es distinto de 0 entonces $f(V)(y)=0$, si y sólo si *y* pertenece a F - Imf. Por último es trivial que $f(V)_0 = F$. ■

Corolario 6.2:

Sea T una norma triangular semicontinua inferiormente. Si f es una aplicación lineal de E en F y V es un subespacio vectorial difuso de E tal que $V(e_j) = \alpha_j$, para todo j, $1 \leq j \leq n$, entonces f determina en F un subespacio vectorial difuso f(V) inducido por una base $\{f(e_{j1}), ..., f(e_{js}), u'_{s+1}, ..., u'_m\}$ tal que $f(V)(f(e_{jk})) = \alpha_{jk}$, $f(V)(u'_j) = 0$ con a lo sumo s + 2 subconjuntos de nivel (siendo r =

dim Kerf, n = dim E y s = n - r = dim Imf = dimE – dimKerf) tales que:
i) $f(V)_1 \subset f(V)_{\alpha j1} \subset ... \subset f(V)_{\alpha jk} \subset ... \subset f(V)_{\alpha js} \subset f(V)_0 = F$
ii) f(Kerf) = $\subset f(V)_1$, y si α_{j1} es distinto de 1 entonces $f(V)_1$ = f(Kerf)
iii) $f(V)_{\alpha j1} = L\{f(e_{j1})\}$; ...; $f(V)_{\alpha jk} = L\{f(e_{j1}), ..., f(e_{jk})\}$; $f(V)_{\alpha js} = L\{f(e_{j1}), ..., f(e_{js})\}$ que contiene a Imf
iv) Si α_{js} es distinto de 0 entonces Imf = $f(V)_{\alpha js}$; $f(V)_0$ = F

Demostración:

Al ser $\{f(e_{j1}), ..., f(e_{js})\}$ una base de Imf y f(V) un subespacio vectorial Mín-difuso sabemos que f(V) tiene sobre Imf a lo sumo s + 1 subconjuntos de nivel $f(V)_1 \subset f(V)_{\alpha j1} \subset ... \subset f(V)_{\alpha jk} \subset ... \subset f(V)_{\alpha js}$ tales que f(Kerf) $\subset f(V)_1$; $f(V)_{\alpha jk} = L\{f(e_{j1}), ..., f(e_{jk})\}$; e Imf $\subset f(V)_{\alpha js}$

Si *y* no pertenece a Imf, f(V)(y)=0, luego a lo sumo en F habrá un subconjunto de nivel más, $f(V)_0$ cuando α_{js} sea distinto de cero. Si α_{js} es distinto de 0 entonces $T\{\alpha_{j1},..., \alpha_{js}\} = Mín\{\alpha_{j1},..., \alpha_{js}\} = \alpha_{js}$ distinto de 0, y Imf = $f(V)_{\alpha js}$. ■

Corolario 6.3:

Sea T una norma triangular semicontinua inferiormente. Si f es una aplicación lineal de E en F, T una norma triangular semicontinua inferiormente y W_T es el mínimo subespacio vectorial T-difuso de E inducido por la base $\{e_1, ..., e_n\}$ tal que $W_T(e_j)=\alpha_j$, para todo j, $1 \leq j \leq n$, entonces f determina en F el mínimo subespacio vectorial T-difuso $f(W_T)$ inducido por una base $\{f(e_{j1}), ..., f(e_{js}), u'_{s+1}, ..., u'_m\}$ tal que $f(W_T)(f(e_{jk})) = \alpha_{jk}$ y que $f(W_T)(u'_j) = 0$ que tiene a lo sumo 2^s+2 subconjuntos de nivel diferentes, tales que: $f(W_T)_1 \subset f(W_T)_{\alpha j1} \subset ... \subset f(W_T)_\gamma \subset ... f(W_T)_{T\{\alpha j1,...,\alpha js\}} \subset f(W_T)_0 = F$, con f(Kerf) $\subset f(W_T)_1$, y siendo $\gamma = \alpha_{jk}$, o bien $\gamma = T\{\alpha_{jk},..., \alpha_{jp}\}$ y $f(W_T)_\gamma$ es unión de subespacios vectoriales ordinarios de la forma $L\{f(e_{jk}), ..., f(e_{jp})\}$; Imf $\subset f(W_T)_{T\{\alpha j1,...,\alpha js\}}$ y si $T\{\alpha_{j1}, ..., \alpha_{js}\}$ es distinto de cero, entonces Imf = $f(W_T)_{T\{\alpha j1,..., \alpha js\}}$.

Demostración:

Es consecuencia inmediata de la estructura de los mínimos subespacios vectoriales T-difusos asociados a una base y de la proposición 6.1. ■

CONCLUSIÓN:

La estructura de los subespacios vectoriales T-difusos se conserva mediante las aplicaciones lineales cuando previamente se han fijado los valores de pertenencia de los vectores de una base.

BIBLIOGRAFÍA

[1] M. T. Abu Osman: On t-fuzzy subfield and subespacio vectorial T-difusos. *Fuzzy Sets and Systems, 33* (1989) 111-117.

[2] K. S. Adbukhalikov y al.: On fuzzy bases de vector spaces. *Fuzzy Sets and Systems, 63* (1994) 201-206.

[3] J. M. Anthony y H. Sherwood: Fuzzy groups redefined. *J. Math. Appl.* 69. (1979). 124-130.

[4] R. Biswas: Fuzzy fields and fuzzy linear spaces redefined. *Fuzzy Sets and Systems, 33* (2). (1989) 257-259.

[5] P. Burillo, y A. Salvador: Espacios Vectoriales T-Difusos. Aplicaciones Lineales y operaciones. Actas XIII Jornadas Hispano-Lusas. (1988).

[6] P. Burillo, R. Bravo y A. Salvador: Relaciones de predominancia en normas y conormas triangulares generalizadas. Actas XIV Jornadas Hispano-Lusas. (1989).

[7] P. Burillo, L. Garmendia y A. Salvador: Representation theorem from T-fuzzy vector subspaces. *International Journal of Approximate Reasoning*. To appear.

[8] P. Das: Fuzzy vector spaces under triangular norms. *Fuzzy Sets and Systems, 25* (1988) 73-85.

[9] A. K. Katsaras y D. B. Liu: Fuzzy vector spaces and fuzzy topological vector spaces. *J. Math. Anal. Appl., 58* (1977) 135-146.

[10] R. Lowen: Convex fuzzy sets. *Fuzzy Sets and Systems, 3* (1980) 291-310.

[11] S. Nanda: Fuzzy fields and fuzzy linear spaces. *Fuzzy Sets and Systems, 19* (1986) 89-94.

[12] A. Rosenfeld: Fuzzy groups. *J. Math. Anal. Appl.* 35. (1971). 512-517.

[13] Y. Yu: Finitely generated t-fuzzy linear spaces. *Fuzzy Sets and Systems, 30* (1989) 69-81.

TRATAMIENTO DE INFORMACIÓN DIFUSA EN UN MODELO DE MEZCLA DE COMPONENTES GAUSSIANAS*

M.C. Garrido†, J.M. Cadenas
Dpto. de Informática, Inteligencia Artificial y Electrónica
Universidad de Murcia
mgarrido@dif.um.es, jcadenas@dif.um.es

A. Ruiz
Dpto. de Informática: Lenguajes y Sistemas Informáticos
Universidad de Murcia
aruiz@dif.um.es

Resumen

En este trabajo se presenta una forma de introducir información difusa en el aprendizaje e inferencia de un modelo probabilístico constituido por una mezcla de componentes gaussianas generalizadas. De esta forma dicho modelo, que ya admite datos con incertidumbre arbitraria, podrá trabajar también con datos imprecisos. La introducción de la información difusa se realiza desde el marco de la Teoría de Evidencias de Dempster-Shafer que permite englobar la Teoría de la Probabilidad y la Teoría de la Posibilidad.

Palabras clave: Tratamiento de información imperfecta, Teoría de Evidencias de Dempster-Shafer, Aprendizaje, Inferencia, Mezcla de componentes gaussianas.

1 INTRODUCCIÓN

La imprecisión y la incertidumbre pueden ser considerados como dos aspectos complementarios de información imperfecta [2].

Desde el punto de vista práctico, un item de información puede ser representado por una cuádrupla (atributo,objeto,valor,confianza).

El atributo es una función que asigna un valor (o conjunto de valores) al objeto. El valor es un subconjunto del dominio de referencia asociado al atributo. La confianza es una indicación de la veracidad del item de información.

Veamos en este contexto la diferencia entre imprecisión e incertidumbre: imprecisión está relacionado con el valor del item, mientras incertidumbre está relacionado con la confianza del item.

El modelo clásico para tratar con la incertidumbre ha sido el modelo probabilístico, mientras que el modelo clásico para tratar la imprecisión ha sido el modelo de análisis de intervalos.

Cuando expresamos imprecisión mediante análisis de intervalos, no se permite incertidumbre: si no se conoce el valor exacto de un parámetro, se conocen sus límites exactos.

Las medidas de posibilidad y conjuntos difusos generalizan el análisis de intervalos para que se permitan gradaciones.

Sería interesante trabajar con una teoría más general, que englobe el modelo probabilístico y el modelo posibilístico. Para ello vamos a trabajar con la Teoría de Evidencias de Dempster-Shafer que se encuentra, en la jerarquía de medidas difusas, por encima de ambas teorías [8].

A continuación veremos de forma breve los aspectos más relevantes de las distintas teorías. En la sección 3 veremos cómo representar la información imperfecta mediante funciones de masas. A continuación, en la sección 4 se presentará brevemente el modelo de Mezcla de Gaussianas Generalizadas Factorizadas (MFGG). En la sección 5 presentaremos el tratamiento de la información difusa en el modelo MFGG y por último terminaremos, en la sección 6, con un ejemplo ilustrativo.

2 INTRODUCCIÓN A LAS DISTINTAS TEORÍAS

2.1 TEORÍA DE EVIDENCIAS DE DEMPSTER-SHAFER

Como hemos dicho anteriormente la Teoría de Evidencias de Dempster-Shafer engloba la Teoría de la Probabilidad y Posibilidad. Veremos las nociones básicas de esta teoría [3], [4], [5], [6], [9].

* Trabajo soportado por el proyecto CICYT TIC97-1343-C02-02

† Autor para correspondencia

Sea x una variable cuyo dominio es Ω_x y $\mathcal{P}(\Omega_x)$ el conjunto de partes de Ω_x. La Teoría de Evidencias de Dempster-Shafer es caracterizada completamente por una función:

$$m : \mathcal{P}(\Omega_x) \rightarrow [0,1]$$

cumpliendo que $m(\emptyset) = 0$ y $\sum_{A \subseteq \Omega_x} m(A) = 1$

Esta función es llamada función de masas y $m(A)$ expresa el grado de evidencia sobre A. Cuando $m(A) > 0$, el conjunto A se llama elemento focal.

Dadas dos funciones de masa definidas sobre $\mathcal{P}(\Omega_x)$, m_1 y m_2, la función de masa m_3 combinación de las anteriores $m_3 = m_1 \oplus m_2$, se obtiene mediante la "Regla de combinación de evidencias de Dempster-Shafer":

$$m_3(A) = \sum_{B \cap C = A} m_1(B) m_2(C), \quad B, C \subseteq \Omega_x$$

Esta teoría permite expresar incertidumbre imprecisa sobre valores imprecisos.

2.2 TEORÍA DE LA PROBABILIDAD

Si los elementos focales son singletons, la función de masa es una medida de probabilidad [5], [2].

Una medida de probabilidad está totalmente determinada por su función de distribución:

$$p : \Omega_x \rightarrow [0,1]$$

con $m(x_i) = p(x_i)$; $\forall x_i \in \Omega_x$ y $\sum_{x_i \in \Omega_x} p(x_i) = 1$.

Esta teoría sólo permite expresar incertidumbre precisa sobre valores precisos.

2.3 TEORÍA DE LA POSIBILIDAD

Si los elementos focales están anidados (ordenados por inclusión), tenemos de nuevo un caso especial de evidencias que son las evidencias consonantes [5], [2]. Estas evidencias dan lugar a dos medidas: Posibilidad y Necesidad.

Una medida de Posibilidad está determinada por su función de distribución de posibilidad

$$r : \Omega_x \rightarrow [0,1]$$

donde $Pos(A) = max_{x_i \in A} r(x_i) \quad \forall A \subseteq \Omega_x$ y $max_{x_i \in \Omega_x} r(x_i) = 1$.

Si asumimos que $\Omega_x = \{x_1, x_2, ..., x_n\}$, $A_1 \subset A_2 \subset ... \subset A_n$ donde $A_i = \{x_1, x_2, ..., x_i\}, \forall i$ y si $m(A) > 0$ implica que $A \in \{A_1, A_2, ..., A_n\}$, es decir, la función de masas m se define sobre elementos focales ordenados por inclusión, entonces:

$$m(A_i) = r(x_i) - r(x_{i+1}) \qquad (1)$$

asumiendo que $r(x_{n+1}) = 0$. Por lo tanto, una función de distribución de posibilidad define una función de masa [5].

Esta teoría nos va a permitir expresar con qué grado pertenece un determinado parámetro a un intervalo de valores, es decir, evita la rigidez del análisis de intervalos.

3 INFORMACIÓN IMPERFECTA COMO FUNCIONES DE MASA

3.1 CONJUNTO DIFUSO COMO FUNCIÓN DE MASA

Un conjunto difuso A con función de pertenecia μ_A puede ser considerado como una función de distribución de posibilidad $r_A(x_i)$ ([1]), donde

$$r_A(x_i) = \mu_A(x_i), \forall x_i \in \Omega_x$$

La interpretación de un conjunto difuso como distribución de posibilidad es la siguiente:

Sea A un conjunto no difuso de Ω_x y v una variable sobre Ω_x. Decir que v toma sus valores en A indica que cualquier elemento en A puede ser un valor de v y que cualquier elemento que no esté en A no puede ser un valor de v. La sentencia "v toma sus valores en A" puede inducir una distribución de posibilidad r sobre Ω_x asociando a cada elemento $x_i \in \Omega_x$ la posibilidad de que x_i sea el valor de v:

$$r(v = x_i) = r(x_i) = \begin{cases} 1 & si\ x_i \in A \\ 0 & en\ otro\ caso \end{cases}$$

Supongamos ahora que A es un conjunto difuso que actúa como una restricción difusa sobre el posible valor de v. Entonces una extensión de la interpretación anterior es que A induce una distribución de posibilidad que es igual a la función de pertenencia μ_A:

$$r(v = x_i) = r(x_i) = \mu_A(x_i)$$

En esta interpretación los elementos focales se corresponden con los distintos α-cortes $A_\alpha = \{x_i \in A : \mu_A(x_i) \geq \alpha\}$ del conjunto difuso A, ya que $A_\alpha \subseteq A_\beta$ si $\alpha > \beta$.

Por otro lado, si conocemos la distribución de posibilidad de una variable, podemos calcular la función de masas asociada como se indicó en (1).

Por lo tanto, dado un conjunto difuso, se puede obtener la función de masas asociada.

3.1.1 Ejemplo

Veremos un ejemplo de cómo un conjunto difuso puede ser transformado en una función de masas.

Sea el conjunto difuso "joven" que tiene asociado la siguiente función de pertenencia:

$$\mu_{joven}(x) = \begin{cases} 1 & 0 \leq x \leq 25 \\ -0.1x + 3.5 & 25 \leq x \leq 35 \end{cases}$$

Si tomamos los α-cortes $\{1, 0.9, 0.8, 0.7, ..., 0.1\}$ formamos 10 subconjuntos del conjunto difuso "joven" y calculamos las masas de cada subconjunto según la expresión (1) como queda reflejado en la tabla 1.

Tabla 1: Conjunto difuso como función de masas.

SUNCONJUNTOS	MASAS
$joven_1 = [0, 25]$	$m([0, 25]) = \mu(25) - \mu(26) = 1 - 0.9 = 0.1$
$joven_{0.9} = [0, 26]$	$m([0, 26]) = 0.9 - 0.8 = 0.1$
$joven_{0.8} = [0, 27]$	$m([0, 27]) = 0.1$
$joven_{0.7} = [0, 28]$	$m([0, 28]) = 0.1$
$joven_{0.6} = [0, 29]$	$m([0, 29]) = 0.1$
$joven_{0.5} = [0, 30]$	$m([0, 30]) = 0.1$
$joven_{0.4} = [0, 31]$	$m([0, 31]) = 0.1$
$joven_{0.3} = [0, 32]$	$m([0, 32]) = 0.1$
$joven_{0.2} = [0, 33]$	$m([0, 33]) = 0.1$
$joven_{0.1} = [0, 34]$	$m([0, 34]) = 0.1$

3.2 NORMAL COMO FUNCIÓN DE MASAS

Cuando queremos expresar información incierta en un marco probabilístico podemos usar una función de densidad de probabilidad. Obtenemos la función de masas correspondiente a una función de densidad Normal de media μ y desviación σ, $N(x, \mu, \sigma)$, discretizando el dominio de x, Ω_x, en intervalos.

Para ello vamos a dividir el dominio $[\mu - 5\sigma, \mu + 5\sigma]$ en intervalos de tamaño *salto*. Es decir, obtendremos el siguiente conjunto de intervalos:

$$[\mu - 5\sigma, (\mu - 5\sigma) + salto],$$
$$[(\mu - 5\sigma) + salto, (\mu - 5\sigma) + 2salto],$$
$$[(\mu - 5\sigma) + 2salto, (\mu - 5\sigma) + 3salto],$$
$$...,$$
$$[(\mu + 5\sigma) - salto, \mu + 5\sigma]$$

La masa asignada a cada intervalo será la probabilidad de dicho intervalo calculada como la integral definida $\int_a^b N(x, \mu, \sigma)$, donde a y b son los distintos extremos de los intervalos.

Veremos a continuación una breve descripción del modelo probabilístico MFGG, para ver despues cómo tratar la información difusa en dicho modelo.

4 MODELO MFGG

El modelo MFGG permite realizar inferencia y aprendizaje probabilístico a partir de información con incertidumbre arbitraria.

MFGG obtiene una expresión de la densidad conjunta que modela al sistema mediante un algoritmo que se denomina Algoritmo EM extendido (fase de aprendizaje), y un conjunto de datos de entrada que contiene descripciones de objetos representativos del sistema a modelar mediante atributos que podrán ser tanto discretos o nominales como continuos y que además pueden contener incertidumbre probabilística arbitraria, [7].

Una vez obtenida la densidad conjunta del modelo, MFGG permite inferir el valor de determinados atributos desconocidos total o parcialmente de un objeto, a partir de información de entrada acerca de dicho objeto (fase de inferencia del modelo). Este proceso de inferencia se realiza mediante las densidades condicionales que se obtienen a partir de la densidad conjunta de manera inmediata.

Veremos a continuación de forma breve las principales expresiones del modelo MFGG.

Supongamos que un objeto z del sistema está descrito por n atributos $z = (z^1, z^2, ..., z^n)$ que pueden ser tanto discretos o nominales como continuos.

La descripción de los objetos en la fase de inferencia y de aprendizaje, tiene la siguiente estructura:

El k-ésimo dato, $S^{(k)}$ estará descrito por:

$$S^{(k)} = \sum_{r=1}^{R} \pi_r^{(k)} \prod_{j=1}^{n} s_r^{j(k)} \qquad (2)$$

donde:

- n es el número de atributos con los que se describe un objeto.
- $\pi_r^{(k)}$ expresa el grado de certeza asignado a cada componente de la observación k.
- $s_r^{j(k)}$ expresa el valor (incierto o no) asignado al atributo z^j en la r-ésima componente de la observación k. Por lo tanto, este valor tomará distintas formas dependiendo del tipo de incertidumbre que refleje.

Es decir, $S^{(k)}$ tiene la siguiente forma

$$S^{(k)} = \{\pi_1^{(k)}(s_1^{1(k)} s_1^{2(k)} ... s_1^{n(k)}) + ... +$$
$$+ \pi_R^{(k)}(s_R^{1(k)} s_R^{2(k)} ... s_R^{n(k)})\}$$

Un ejemplo de la estructura de una observación de entrada tanto para la inferencia como para el aprendizaje del modelo puede ser:

$$0.3[B \cdot N(z^2, 150, 1)] + 0.7[(0.4 \cdot B + 0.6 \cdot A) \cdot 180]$$

donde aparecen valores de atributos exactos, por ejemplo $z^1 = B$, y valores de atributos inciertos, por ejemplo $N(z^2, 150, 1)$.

En el marco MFGG, las dependencias de los atributos en el dominio son modeladas mediante la densidad conjunta en la forma de una mezcla finita de componentes factorizados $C_1, C_2, ..., C_l$:

$$p(z) = p(z^1, z^2, ..., z^n) = \sum_{i=1}^{l} P(C_i) \prod_{j=1}^{n} p(z^j / C_i)$$

donde:

- $P(C_i)$ es la probabilidad del componente C_i.
- Para z^j continuo $p(z^j/C_i) = N(z^j, \mu_i, \sigma_i)$.
- Para z^j discreto o nominal $p(z^j/C_i) = \sum_\omega t^j_{i\omega} N(z^j, \omega)$ donde $t^j_{i\omega} = P(z^j = \omega/C_i)$ y $N(z^j, \omega)$ es una delta de Dirac.

Podemos decir que cada componente C_i agrupa objetos con independencia de atributos.

Con el propósito de inferir y aprender a partir de las observaciones de objetos con incertidumbre arbitraria (2), MFGG obtiene la densidad conjunta modelo-observación con la siguiente forma:

$$p(z, S^{(k)}) = \sum_{i=1}^{l} \sum_{r=1}^{R} P(C_i) \pi_r^{(k)} \prod_{j=1}^{n} \beta_{ir}^{j(k)} \Psi_{ir}^{j(k)}(z^j) \quad (3)$$

donde:

- $P(C_i)$ es la probabilidad del componente C_i.
- $\pi_r^{(k)}$ es el grado de certeza de la r-ésima componente en la observación k.
- $\beta_{ir}^{j(k)} = P(s_r^{j(k)}/C_i)$ es la verosimilitud en el componente C_i del valor de z^j en la r-ésima componente de la observación k.
- $\Psi_{ir}^{j(k)}(z^j) = p(z^j/s_r^{j(k)}, C_i)$ es el efecto en el componente C_i de la información acerca de z^j en la r-ésima componente de la observación k.

4.1 INFERENCIA

De la expresión anterior (3), la estructura del inferidor, es decir, de la densidad condicional es:

$$p(z^j/S^{(k)}) = \sum_{ir} \alpha_{ir}^{(k)} E\left\{\Psi_{ir}^{j(k)}(z^j)\right\}$$

donde:

- $\alpha_{ir}^{(k)} = P(C_i, s_r^{(k)}) = \frac{P(C_i)\pi_r^{(k)}\beta_{ir}^{(k)}}{\sum_{kl} P(C_k)\pi_l^{(k)}\beta_{kl}^{(k)}}$
- $\beta_{ir}^{(k)} = \prod_j \beta_{ir}^{j(k)}$
- $E\left\{\Psi_{ir}^{j(k)}(z^j)\right\} = E\left\{p(z^j/s_r^{j(k)}, C_i)\right\}$

El valor concreto de las expresiones necesarias para inferir en el modelo MFGG, β_{ir}^j y $E\left\{\Psi_{ir}^j(z^j)\right\}$, para las observaciones con incertidumbre arbitraria no es el objeto de este trabajo (que se desarrolla en [7]) y sólo serán obtenidas en la siguiente sección para la información difusa.

4.2 APRENDIZAJE

El aprendizaje del modelo MFGG se realiza con una extensión del algoritmo EM clásico que es resumido a continuación.

En este algoritmo hay dos pasos principales:

Paso E (Expectation step):

Se calculan las verosimilitudes del conjunto de las M observaciones de entrada, $k = 1, ..., M$:

$$\beta_{ir}^{(k)} = \prod_j \beta_{ir}^{j(k)};\ \beta^{(k)} = \sum_i \sum_r P(C_i)\pi_r^{(k)}\beta_{ir}^{(k)}$$

A continuación se calcula la probabilidad de que el r-ésimo componente del k-ésimo ejemplo haya sido generado por el i-ésimo componente:

$$q_{ir}^{(k)} = P(C_i, s_r^{(k)}/S^{(k)}) = \alpha_{ir}^{(k)} = \frac{P(C_i)\pi_r^{(k)}\beta_{ir}^{(k)}}{\beta^{(k)}}$$

Paso M (Maximation step):

Modifica los parámetros de cada componente C_i usando todos los ejemplos ponderados por sus probabilidades:

$$P(C_i) = \frac{1}{M} \sum_k \sum_r q_{ir}^{(k)}$$

Para atributos continuos:

$$\mu_i^j = \frac{1}{P(C_i)M} \sum_k \sum_r \left[q_{ir}\nu_{ir}^j\right]^{(k)};\ (\sigma_i^j)^2 =$$

$$= \frac{1}{P(C_i)M} \sum_k \sum_r \left[q_{ir}\left[(\nu_{ir}^j)^2 + (\lambda_{ir}^j)^2\right]\right]^{(k)} - (\mu_i^j)^2$$

Las expresiones concretas para ν_{ir}^j y λ_{ir}^j para cada tipo de información con incertidumbre se pueden consultar en [7]. En este trabajo sólo las obtendremos para la información difusa.

Para atributos discretos:

$$t_{i\omega}^j = \frac{1}{P(C_i)M} \sum_k \sum_r \left[q_{ir} t_{r\omega}^j t_{i\omega}^j / \beta_{ir}^j\right]^{(k)}$$

donde $t_{r\omega}^j = P(s_r^j = \omega)$ en la observación k.

5 CONJUNTOS DIFUSOS EN EL MODELO MFGG

5.1 INFERENCIA

Tan solo veremos qué forma concreta toman las expresiones $\beta_{ir}^{j(k)}$ y $E\{\Psi_{ir}^{j(k)}(z^j)\}$ cuando $s_r^{j(k)}$ en la k-ésima observación está dada por un conjunto difuso

(usaremos las variables mayúsculas $Z^1, ..., Z^n$ para indicar argumentos de funciones que son subconjuntos de $\Omega_{z^1}, ..., \Omega_{z^n}$).

$$\beta_{ir}^{j(k)} = V_{m_{ir}^{j(k)}(Z^j)} = \sum_{A \cap B \neq \emptyset} m_i^j(A) m_r^{j(k)}(B) \qquad (4)$$

donde $A, B \subseteq \Omega_{z^j}$, $m_i^j(Z^j)$ es la función de masas correspondiente a $p(z^j/C_i) = N(z^j, \mu_i^j, \sigma_i^j)$ y $m_r^{j(k)}(Z^j)$ es la función de masas correspondiente a la función de pertenencia del conjunto difuso '*etiqueta*', $\mu_{etiqueta}(z^j)$, cuando $s_r^{j(k)} = etiqueta$ en la observación de entrada. Es decir, el valor $V_{m_{ir}^{j(k)}(Z^j)}$ mide el grado de ausencia de conflicto entre las funciones m_i^j y $m_r^{j(k)}$ y puede ser considerado como la verosimilitud en el componente C_i del valor $s_r^{j(k)} = etiqueta$.

Por otro lado

$$E\{\Psi_{ir}^{j(k)}(z_j)\} = \bar{m}_{ir}^{j(k)} =$$

$$= \frac{1}{V_{m_{ir}^{j(k)}(Z^j)}} \sum_{h=0}^{p-1} \left(\frac{e_h + e_{h+1}}{2} \right) \cdot m_{ir}^{j(k)}([e_h, e_{h+1}]) \quad (5)$$

donde $e_0, e_1, ..., e_{p-1}, e_p$ son los extremos de los intervalos que constituyen la función $m_{ir}^{j(k)}$ y donde el valor de p dependerá de la función de masas concreta con la que estemos trabajando.

$\bar{m}_{ir}^{j(k)}$ puede ser considerado como el valor medio en la función de masas combinada $m_{ir}^{j(k)}(Z^j)$.

5.2 APRENDIZAJE

Veremos cómo incluir información dada como conjuntos difusos en la obtención del modelo MFGG mediante el algoritmo EM extendido:

Paso E: Expectation step.

Calcula las verosimilitudes de los datos $s_r^{j(k)}$ como hemos visto en (4).

Los restantes cálculos de este paso del algoritmo se realizan de la misma forma que vimos anteriormente en la sección 4.2.

Paso M: Maximation step.

Puesto que la información difusa la consideramos sobre atributos continuos, sólo tenemos que incorporarla en las siguientes expresiones (el resto de parámetros se obtienen como antes):

Tomamos $m_r^{j(k)}(Z^j)$ como la función de masas obtenida a partir de la función de pertenencia del conjunto difuso que representa el valor de z^j en el ejemplo k y combinamos esta información con $m_i^j(Z^j)$ que es la función de masas correspondiente a la discretización de la función $p(z^j/C_i)$ del modelo actual. A partir de la función de masas resultante, $m_{ir}^{j(k)}(Z^j)$, obtenemos el valor de $\nu_{ir}^{j(k)}$ y $\lambda_{ir}^{j(k)}$ como:

- $\nu_{ir}^{j(k)}$ es el valor medio de $m_{ir}^{j(k)}(Z^j)$, $\bar{m}_{ir}^{j(k)}$, calculado como en (5).
- $(\lambda_{ir}^{j(k)})^2$ es una medida de dispersión que se obtiene como:

$$d^2_{m_{ir}^{j(k)}(Z^j)} = \frac{1}{V_{m_{ir}^{j(k)}(Z^j)}} \cdot$$

$$\cdot \sum_{h=0}^{p-1} \left(\left(\frac{e_h + e_{h+1}}{2} \right) - \bar{m}_{ir}^{j(k)} \right)^2 \cdot m_{ir}^{j(k)}([e_h, e_{h+1}])$$

donde $e_0, e_1, ..., e_{p-1}, e_p$ son los extremos de los intervalos que constituyen la función $m_{ir}^{j(k)}(Z^j)$.

6 EJEMPLO

Supongamos que partimos de una mezcla bivariable $p(z) = p(x, y)$ como la siguiente:

$$p(x, y) = 0.5N(x, 22, 1)N(y, 80, 2) +$$
$$+0.5N(x, 40, 2)N(y, 60, 2)$$

Expresamos las normales $N(x, 22, 1)$ y $N(x, 40, 2)$ mediante las funciones de masa $m_1^x(X)$ y $m_2^x(X)$ respectivas de la tabla 2 .

Las funciones de masas resultantes de la combinación de las dos funciones anteriores $m_1^x(X)$ y $m_2^x(X)$ y el valor de entrada $x = joven$ transformado a función de masas $(m_r^x(X))$ como se vió en la sección 3.1.1, se muestran como $m_{1r}^x(X)$ y $m_{2r}^x(X)$ respectivamente en la tabla 3.

Como se puede observar, la verosimilitud, $V_{m_{ir}^x(X)}$, del concepto 'joven' es mayor en el componente C_1 que en el componente C_2.

Cuando queremos inferir el valor de x, es decir, $p(x/x\ es\ joven)$, tomamos el valor medio de la función de masa resultante en cada componente. En este ejemplo, el valor medio en $m_{1r}^x(X)$ es $\bar{m}_{1r}^x = 22$ y para $m_{2r}^x(X)$ es $\bar{m}_{2r}^x = 33.13$.

Por lo tanto, el resultado vendrá dado por:

$$p(x/x\ es\ joven) = \alpha_{1r} \cdot \bar{m}_{1r}^x + \alpha_{2r} \cdot \bar{m}_{2r}^x$$

donde

$$\alpha_{1r} = \frac{0.5 \cdot 1 \cdot 0.996}{0.498 + 0.00008105} = 0.99984$$

$$\alpha_{2r} = \frac{0.5 \cdot 1 \cdot 0.0001621}{0.498 + 0.00008105} = 0.00016$$

Entonces

$p(x/x\ es\ joven) = 0.99984 \cdot 22 + 0.00016 \cdot 33.13 = 22.002$

Tabla 2: Normales expresadas con masas.

$m_1^*(X)$	$m_2^*(X)$
(17,17.5) = 3.11102e-06	(30,30.5) = 7.30432e-07
(17.5,18) = 2.82736e-05	(30.5,31) = 2.38059e-06
(18,18.5) = 0.000200958	(31,31.5) = 7.29085e-06
(18.5,19) = 0.00111727	(31.5,32) = 2.09827e-05
(19,19.5) = 0.00485977	(32,32.5) = 5.6746e-05
(19.5,20) = 0.0165405	(32.5,33) = 0.000144212
(20,20.5) = 0.0440571	(33,33.5) = 0.000344396
(20.5,21) = 0.0918481	(33.5,34) = 0.000772873
(21,21.5) = 0.149882	(34,34.5) = 0.00162987
(21.5,22) = 0.191462	(34.5,35) = 0.0032299
(22,22.5) = 0.191462	(35,35.5) = 0.00601481
(22.5,23) = 0.149882	(35.5,36) = 0.0105257
(23,23.5) = 0.0918481	(36,36.5) = 0.017309
(23.5,24) = 0.0440571	(36.5,37) = 0.026748
(24,24.5) = 0.0165405	(37,37.5) = 0.0388426
(24.5,25) = 0.00485977	(37.5,38) = 0.0530055
(25,25.5) = 0.00111727	(38,38.5) = 0.0679721
(25.5,26) = 0.000200958	(38.5,39) = 0.0819102
(26,26.5) = 2.82736e-05	(39,39.5) = 0.0927561
(26.5,27) = 3.11102e-06	(39.5,40) = 0.0987063
(27,27.5) = 2.67666e-07	(40,40.5) = 0.0987063
	(40.5,41) = 0.0927561
	(41,41.5) = 0.0819102
	(41.5,42) = 0.0679721
	(42,42.5) = 0.0530055
	(42.5,43) = 0.0388426
	(43,43.5) = 0.026748
	(43.5,44) = 0.017309
	(44,44.5) = 0.0105257
	(44.5,45) = 0.00601481
	(45,45.5) = 0.0032299
	(45.5,46) = 0.00162987
	(46,46.5) = 0.000772873
	(46,46.5) = 0.000772873
	(46,46.5) = 0.000772873
	(46,46.5) = 0.000772873
	(47,47.5) = 0.000144212
	(47.5,48) = 5.6746e-05
	(48,48.5) = 2.09827e-05
	(48.5,49) = 7.29085e-06
	(49,49.5) = 2.38059e-06
	(49.5,50) = 7.30432e-07
	(50,50.5) = 2.106e-07

Tabla 3: Masas resultantes de la combinación.

$m_{1r}^*(X)$	$m_{2r}^*(X)$
(17,17.5) = 1· 3.11102e-06	(30,30.5) = 0.4· 7.30432e-07
(17.5,18) = 1· 2.82736e-05	(30.5,31) = 0.4· 2.38059e-06
(18,18.5) = 1· 0.000200958	(31,31.5) = 0.3· 7.29085e-06
(18.5,19) = 1· 0.00111727	(31.5,32) = 0.3 · 2.09827e-05
(19,19.5) = 1· 0.00485977	(32,32.5) = 0.2 · 5.6746e-05
(19.5,20) = 1· 0.0165405	(32.5,33) = 0.2 · 0.000144212
(20,20.5) = 1· 0.0440571	(33,33.5) = 0.1· 0.000344396
(20.5,21) = 1· 0.0918481	(33.5,34) = 0.1· 0.000772873
(21,21.5) = 1· 0.149882	(34,34.5) = 0
(21.5,22) = 1· 0.191462	(34.5,35) = 0
(22,22.5) = 1· 0.191462	(35,35.5) = 0
(22.5,23) = 1· 0.149882	(35.5,36) = 0
(23,23.5) = 1· 0.0918481	(36,36.5) = 0
(23.5,24) = 1· 0.0440571	(36.5,37) = 0
(24,24.5) = 1· 0.0165405	(37,37.5) = 0
(24.5,25) = 1· 0.00485977	(37.5,38) = 0
(25,25.5) = 0.9· 0.00111727	(38,38.5) = 0
(25.5,26) = 0.9· 0.000200958	(38.5,39) = 0
(26,26.5) = 0.8· 2.82736e-05	(39,39.5) = 0
(26.5,27) = 0.8· 3.11102e-06	(39.5,40) = 0
(27,27.5) = 0.7· 2.67666e-07	...
$V_{m_{1r}^*(X)} = 0.996$	$V_{m_{2r}^*(X)} = 0.0001621$

7 CONCLUSIONES

Hemos introducido el tratamiento de un nuevo tipo de información en el modelo probabilítico MFGG, el cual ya permitía el uso de incertidumbre arbitraria en el marco de la Teoría de la Probabilidad.

Este tratamiento de información difusa en el modelo probabilístico, situados en el marco de trabajo de la Teoría de Evidencias de Dempster-Shafer, permite añadir al modelo el tratamiento de evidencias o funciones de masas genéricas sobre atributos discretos o nominales y continuos, que serán objeto de futuros estudios.

Por lo tanto, disponemos de un modelo de aprendizaje e inferencia que nos va a permitir trabajar con información incierta e imprecisa sobre atributos o variables tanto discretas como continuas.

Referencias

[1] **D. Dubois and H. Prade**, Fuzzy Sets and Systems: Theory and Applications, *Academic Press, 1980.*

[2] **D. Dubois and H. Prade**, Possibility Theory: An approach to Computerized Processing of Uncertainty, *Plenum Press, New York, 1988*

[3] **J.W. Guan, D.A. Bell**, Evidence Theory and its Applications, *Studies in Computer Science and Artificial Intelligence, Volume 1, North-Holland, 1991.*

[4] **J.W. Guan, D.A. Bell**, Evidence Theory and its Applications, *Studies in Computer Science and Artificial Intelligence, Volume 2, North-Holland, 1992.*

[5] **G.J. Klir**, Probabilistic versus possibilistic conceptualization of uncertainty, *Analysis and Management of Uncertainty: Theory and Applications, Elsevier Science Publishers B.V., pp.13-25, 1992.*

[6] **I. Kramosil**, Believeability and plausibility functions over infinite sets, *Int. Journal of General Systems, vol.23, pp.173-198, 1995.*

[7] **A. Ruiz, P.E. López de Teruel, M.C. Garrido**, Probabilistic Inference from Arbitrary Uncertainty using Mixtures of Factorized Generalized Gaussians, *Technical Report del Departamento de Informática y Sistemas. Universidad de Murcia. TR DIS 2-98. Sometido a revisión en JAIR. 1998.*

[8] **J.F. Verdegay**, Tratamiento de la incertidumbre mediante medidas difusas. *Algunos aspectos del tratamiento de la incertidumbre en la Inteligencia Artificial. Universidad de Granada. 1991.*

[9] **L.P. Wesley**, Evidential knoweledge-based computer vision. *Optical Engineering, vol.25, n.3, 1986.*

GENERACIÓN DE MORFOLOGÍAS MATEMÁTICAS BORROSAS*

P. Burillo
Dpto de Automática y Computación. Universidad Pública de Navarra Campus de Arrosadía 31006. Pamplona. Navarra.
E-mail:pburillo @upna.es

N. Frago
Dpto de Automática y Computación. Universidad Pública de Navarra Campus de Arrosadía 31006. Pamplona. Navarra.
noe.frago @upna.es,

R. Fuentes
Dpto de Automática y Computación. Universidad Pública de Navarra Campus de Arrosadía 31006. Pamplona. Navarra.
rfuentes @upna.es

Resumen

En este trabajo, partiendo de implicaciones borrosas, se propone una forma general para obtener operadores: erosión y dilatación borrosos para el tratamiento de imágenes. Propuesta que recoge como casos particulares, las dadas por otros autores en la teoría de Morfología Matemática Borrosa. Hacemos un estudio de las propiedades básicas en el procesamiento de imágenes y se analiza los efectos prácticos utilizando una imagen con distintos niveles de gris y 18 implicaciones borrosas usuales en la literatura.

Palabras Clave: Morfología Matemática Borrosa, Grado de Inclusión, Funciones Implicación, Erosión, dilatación.

1 INTRODUCCIÓN

En la Morfología Matemática [17], el modelo algebraico que representa el estudio de las transformaciones de imágenes binarias A, B, (subconjuntos ordinarios del referencial $U = R^2$ o $U = Z^2$), se basa en la consideración de operadores Φ isótonos sobre el conjunto ordenado $(\wp(U), \subseteq)$, es decir operadores tales que: $A \subseteq B \Rightarrow \Phi(A) \subseteq \Phi(B)$. En particular tienen especial interés las transformaciones básicas: erosiones $\xi(\_,B)$ y dilataciones $\mathcal{D}(\_,B)$ asociadas a imágenes fijadas B denominadas elementos estructurantes. Su expresión algebraica viene dada por:

Imagen erosionada de A por el elemento estructurante B:

$$\xi(A,B)= \cap\{A_b/b\in \overset{\vee}{B}\},$$

donde $A_b=\{a+b/a\in A\}$ y $\overset{\vee}{B} = \{-b/b\in B\}$.
Imagen dilatada de A por el elemento estructurante B:

$$\mathcal{D}(A,B)= (\xi(A^c, \overset{\vee}{B}))^c.$$

Estos operadores son base de otros utilizados en la teoría, como los llamados *apertura*: $@(A, B) = \mathcal{D}(\xi(A, B), B)$, y *cierre*: $\zeta(A, B) = \xi(\mathcal{D}(A, \overset{\vee}{B}), \overset{\vee}{B})$.

* Este trabajo ha sido parcialmente financiado por el Gobierno de Navarra.

Todos estos operadores tienen su correspondiente versión en la Morfología Matemática sobre el espacio de funciones "grey-level"[17] que modelizan imágenes con distintos niveles de gris.

En la literatura, aparece una versión borrosa de estos operadores en [3-6] (Divyendu-Dougherty).Estos autores definen las operaciones básicas de la *Morfología Matemática Borrosa* utilizando el operador R, grado de inclusión [4-5] del subconjunto borroso $A\in F(U)$ en el subconjunto borroso $B\in F(U)$:

$$R(A, B) = \inf_{x\in U}\{\min(1, \lambda(A(x))+\lambda(1-B(x)))\}$$

.donde λ: [0, 1] $\longrightarrow$ [0, 1] es una aplicación que verifica las propiedades:

1.- λ decreciente
2.- $\lambda(1) = 0$ y $\lambda(0) = 1$
3 - Si p, q$\in$[0, 1] y $\lambda(p)=\lambda(q) \geq 0.5$, entonces p=q
4.- La ecuación $\lambda(p)=0$ tiene solución única
5.- $\forall p\in[0, 1]$ se verifica $\lambda(p)+\lambda(1-p)\geq 1$

El grado de inclusión de A en B está dado por un número perteneciente a [0, 1] cuyo valor queda determinado por R(A, B). A partir de este operador grado de inclusión, definen la erosión de un subconjunto borroso A por otro B llamado elemento estructurante, como el subconjunto borroso $\xi(A, B)$ cuya función característica está dada por:
$\xi(A, B)(z) = R(B_z, A) = \inf_{x\in U}\{\min(1, \lambda(B_z(x))+\lambda(1-A(x)))\}$

A partir de aquí Divyendu y Dougherty definen la dilatación de forma análoga al caso binario

$$\mathcal{D}(A, B)(z)=1-R((\overset{\vee}{B})_z, 1-A).$$

y deducen el resto de las operaciones básicas de esta morfología como,

apertura: $@(A, B) = \mathcal{D}(\xi(A, B), B)$,

cierre: $\zeta(A, B) = \xi(\mathcal{D}(A, \overset{\vee}{B}), \overset{\vee}{B})$, etc.

Posteriormente, Block y Maître [2], proponen definiciones de erosiones y dilataciones borrosas a partir de conormas **s** y t-normas **t** según las siguientes definiciones:

$\xi(A, B)(z) = \inf_{x \in U} \{s(A(x), B(x-z)\}$,

$\mathcal{O}(A, B)(z) = \sup_{x \in U} \{t(A(x), n(B(x-z))\})$,

siendo n una negación fuerte en [0,1].

Presentamos aqui una propuesta más general para definir operadores erosión sobre subconjuntos borrosos. Propuesta que incluirá como casos particulares los citados en los párrafos anteriores.

Nuestras definiciones se basan en el operador *grado de inclusión* del subconjunto borroso A en el B, que es un elemento R(A, B) $\in$[0,1].propuesto por Bandler y Kohout [1] y dado por ,

$$R(A, B) = \inf_{x \in U} \{I(A(x), B(x))\},$$

siendo I una *implicación borrosa* [16] en [0,1].

En este trabajo utilizamos el termino implicación borrosa en el sentido más amplio, como aplicaciones I:[0,1]x[0,1]$\longrightarrow$[0,1], que extienden alguna de las implicaciones de la lógica booleana [21]. En particular las implicaciones aquí analizadas o cumple alguna de las definiciones dadas en [22] o bien extienden alguna implicación booleana y verifican alguna de las propiedades exigidas por Dubois-Prade [9] para los operadores implicación.

Se estudian alguna de las propiedades de los operadores definidos y se inicia el análisis de los efectos prácticos de erosiones obtenidas con las 18 implicaciones borrosas más utilizadas y que aparecen recogidas en un trabajo de Ruan y Kerre [16]. En este análisis el subconjunto borroso a tratar A (fig. 1 en el anexo A) es una imagen con distintos tonos de gris y se trata con tres elementos estructurantes B que son subconjuntos borrosos con valor constante. Los resultados aparecen el anexo A, al final del trabajo.

2 PRELIMINARES

Señalamos a continuación algunos operaciones y resultados básicos usuales en Morfología Matemática Borrosa.

Se llama *translación* de un subconjunto borroso A de F(**U**), por un elemento $v \in \mathbf{U}$, al elemento A_v de F(**U**), tal que $A_v(x) = A(x-v) \quad \forall x \in \mathbf{U}$.

La *reflexión* de un subconjunto borroso A de F(**U**), es el elemento $\check{A}$ de F(**U**), tal que $\check{A}(x) = A(-x) \quad \forall x \in \mathbf{U}$.

Propiedades: Dado un subconjunto borroso A de F(**U**) y $x \in \mathbf{U}$, entonces: Si A^C es el borroso $A^C = 1-A(x)$

1) $\check{A}_x = \check{A}_{(-x)}$, **2)** $(A_x)^C = (A^C)_x$, **3)** $(\check{A})^C = (A^C)\check{}$

Demostración: **1)** $\check{A}_x(y) = A_x(-y) = A(-y-x) = A(-(x+y)) = (\check{A})(x+y) = (\check{A})_{(-x)}(y)$

2) $(A_x)^C(y) = 1-A_x(y) = 1-A(y-x) = A^C(y-x) = (A^C)_x(y)$

3) $(\check{A})^C(y) = 1-(\check{A})(y) = 1-A(-y) = A^C(-y) = (A^C)\check{}(y)$. □.

Teorema 1: Dados A, B$\in[0, 1]^U$ se verifica:

1) $R(A, B) = R(\check{A}, \check{B})$ **2)** $R(A, B) = R(A_z, B_z) \quad \forall z \in \mathbf{U}$

Demostración:

$R(\check{A}, \check{B}) = \inf_{x \in U} \{I((\check{A})(x), (\check{B})(x))\} =$

$\inf_{x \in U} \{I(A(-x), B(-x)\} = \inf_{x \in U} \{I(A(x), B(x)\}$

$R(A_z, B_z) = \inf_{x \in U} \{I(A_v(x), B_v(x)\} =$

$\inf_{x \in U} \{I(A(x-v), B(x-v)\} = \inf_{x \in U} \{I(A(x), B(x)\} = R(A, B)$

Quedando demostrado el teorema. □.

3 RESULTADOS GENERALES

Dada una implicación borrosa I, si R representa el grado de inclusión correspondiente, proponemos las siguientes

Definiciones 1: La *erosión / dilatación borrosas* de una imagen A por el elemento estructurante B, ambos subconjuntos borrosos pertenecientes a F(**U**), es otro subconjunto borroso denotado por $\xi(A, B)$ / $\mathcal{O}(A, B)(z)$, y definido por

$$\xi(A, B)(z) = R(B_z, A) \quad \forall z \in \mathbf{U}.$$

$$\mathcal{O}(A, B)(z) = 1-R((\check{B})_z, A^C) \quad \forall z \in \mathbf{U}.$$

Las dilataciones y erosiones anteriores son operaciones duales como se desprende de la propia definición:

$\mathcal{O}(A, B)(z) = 1-\xi(A, (\check{B}))(z) \quad \forall z \in \mathbf{U}$

Estos operadores permiten la construcción de otros interesantes desde el punto de vista de la Morfología:

Definición 2: Los operadores *apertura* y *cierre borrosos* se definen, como en la morfología ordinaria, a partir de la erosión y la dilatación

$$@(A,B)(z) = \mathcal{O}(\xi(A, B), B)(z) \quad \forall z \in \mathbf{U}.$$

$$\zeta(A,B)(z) = \xi(\mathcal{O}(A, \check{B}), \check{B})(z) \quad \forall z \in \mathbf{U}.$$

Analizaremos alguna de sus propiedades .

Teorema 2: Sean A y B dos subconjuntos borrosos de F(**U**) y un elemento v perteneciente a **U**. Entonces se verifica:

1) $\xi(Av, B) = \xi(A, B)$, **2)** $\xi(A, Bv) = \xi(A, B)(-v)$

3) $\mathcal{O}(A_v, B) = \mathcal{O}(A, B)_v$ **4)** $\mathcal{O}(A, B_v) = \mathcal{O}(A, B)$

Demostración: **1)** $\xi(A_v, B)(z) = R(B_z, A_v) = R(B_{z-v}, A) = \xi(A, B)(z-v) = \xi(A, B)_v(z) \quad \forall z \in \mathbf{U}$

2) $\xi(A, B_v)(z) = R((B_z)_v, A) = R(B_{z+v}, A) =$
$\xi(A, B)(z+v) = \xi(A, B)_{(-v)}(z) \quad \forall z \in \mathbf{U}$

3) $\mathcal{Q}(A_v, B)(z) = 1-\xi((A_v)^c, \check{B})(z) =$

$1-\xi((A^C)_v, \check{B})(z) = 1-\xi(A^C, B)_v(z) = \mathcal{Q}(A, B)_v(z)$
$\forall z \in \mathbf{U}$.

4) $\mathcal{Q}(A, B_v)(z) = 1-\xi(A^C, -(B_v))(z) =$
$1-\xi(A^C, (-B)_{(-v)})(z) = 1-\xi(A^C, -B)_v(z) =$

$1-\xi(A^C, \check{B})(z-v) = \mathcal{Q}(A, B)_v(z) \quad \forall z \in \mathbf{U}$. □.

Este teorema prueba que las dilataciones y erosiones definidas a partir de los grados de inclusión definidos por Bandler y Kohout [2], verifican propiedades de invarianza para traslaciones a semejanza con lo que ocurre en la Morfología Matemática ordinaria (Primer principio establecido por Serra [17] para las transformaciones morfológicas).

Veamos a continuación que esta Morfología Borrosa que proponemos difiere en algunos aspectos del tratamiento usual de imágenes con niveles de gris en la Morfología Matemática en el espacio de funciones "grey-level".

Sea **U** cualquiera de los siguientes conjuntos **R**, $\mathbf{R}^2$, **Z** ó $\mathbf{Z}^2$. Sean f, g : **U** $\longrightarrow$ [0, 1] funciones "grey-level",(que evidentemente son subconjuntos borrosos de **U**). Utilizando la implicación de Lukasiewicz , la erosión borrosa ξ_F de f utilizando g como elemento estructurante es, según la definición 1:

$$\xi_F(f, g)(z) = \inf_{x \in \mathbf{U}} \{\min(1, f(x+z)+1-g(x))\} \quad z \in \mathbf{U}$$

que difiere de la definición usual de erosión "grey-level":

$$\xi_G(f, g)(z) = \inf_{x \in \mathbf{U}} \{f(x+z)-g(x)\} \quad z \in \mathbf{U}$$

Por otra parte, podemos observar que :

1) Los valores de $\xi_F(f, g)(z)$ pertenecen a [0, 1], mientras que los valores de $\xi_G(f, g)(z)$ pertenecen a [-1, 1].

2) En la erosión borrosa no es posible la invarianza de la traslación tal como se da en la grey-level:

$$\xi_G(f_h \Diamond \alpha, g)(z) = \xi_G(f \Diamond \alpha, g)_h(z)+\alpha, \quad \alpha \in \mathbf{R} \text{ y}$$

$$(f \Diamond \alpha)(x)=\min(1, \max(0, f(x)+\alpha)), \; f_h(x)=f(x-h)).$$

El teorema 2 nos da un tipo de invarianza de la traslación para la erosión y dilatación borrosa, pero sin posibilidad de establecer una fórmula similar a la clásica traslación de las funciones "grey-level".

El segundo principio establecido por Serra [17] para las transformaciones morfológicas ordinarias se refiere a la invarianza de la erosión y dilatación para homotecias (cambios de escala), o formalmente $\psi(\alpha A) = \alpha\psi(A)$ con α número real mayor o igual que cero. Vamos a ver en los lemas siguientes que, aun limitándonos a valores $\alpha \in [0, 1]$, la erosión y la dilatación no verifican la invarianza por homotecias.

Lema 1: Sea $\alpha \in [0, 1]$. Consideremos el producto $\alpha \inf_{x \in \mathbf{U}} \{I(B_z(x), A(x))\} = \inf_{x \in \mathbf{U}} \{I(B_z(x), \alpha A(x))\}$.

Entonces, $\alpha I(B_z(x), A(x)) = I(B_z(x), \alpha A(x))$.

Demostración: Supongamos que existe un $x \in \mathbf{U}$ tal que $\alpha I(B_z(x), A(x)) \neq I(B_z(x), \alpha A(x))$.

- $\alpha \inf_{x \in \mathbf{U}} \{I(B_z(x), A(x))\} \leq I(B_z(x), \alpha A(x)) \Rightarrow$

$\alpha I(B_z(y), A(y)) \leq I(B_z(x), \alpha A(x)) \quad \forall y \in \mathbf{U} \Rightarrow$
$\alpha I(B_z(x), A(x)) \leq I(B_z(x), \alpha A(x))$.

- $\inf_{x \in \mathbf{U}} \{I(B_z(x), \alpha A(x))\} \leq \alpha I(B_z(x), A(x)) \Rightarrow$

$I(B_z(y), \alpha A(y)) \leq \alpha I(B_z(x), A(x)) \quad \forall y \in \mathbf{U} \Rightarrow$
$I(B_z(x), \alpha A(x)) \leq \alpha I(B_z(x), A(x))$.

No existe un $x \in \mathbf{U}$ tal que
$\alpha I(B_z(x), A(x)) \neq I(B_z(x), \alpha A(x))$. □.

Lema 2: Ninguna de las 18 implicaciones borrosas más habituales en la bibliografía (Kerre [14,16]) verifica que $\alpha I(a, b)=I(a, \alpha b) \quad \forall a, b \in [0, 1]$, para cualquier $\alpha \in [0, 1]$.

Demostración: Es una comprobación. □.

Los lemas anteriores nos demuestran que para las erosiones definidas con las 18 implicaciones borrosas no se verifica la invarianza para homotecias establecida por Serra [17] como uno un principio de las transformaciones morfológicas, resultado que recogemos en el siguiente.

Teorema 3: Las erosiones definidas con las 18 implicaciones borrosas que aparecen habitualmente en la bibliografía (Kerre [14,16]) no son invariantes para homotecias.

Demostración: Es inmediato se tenemos en cuenta los Lemas 1 y 2. □.

Finalmente, veremos en este apartado el equivalente en nuestro contexto del tercer principio establecido por Serra [17] para las transformaciones morfológicas.

Teorema 4: Sea dos imágenes borrosas A y $B \in F(\mathbf{U})$, y un conjunto acotado $K' \subseteq [0,1]$, en el cual queremos conocer $\xi(A, B)$ y $\mathcal{Q}(A,B)$ ($\xi(A, B) \cap K'$ y $\mathcal{Q}(A,B) \cap K'$). Podemos encontrar un conjunto acotado $K \subseteq [0, 1]$ tal que $\xi(A \cap K, B) = \xi(A, B) \cap K'$ y $\mathcal{Q}(A \cap K, B) = \mathcal{Q}(A,B) \cap K'$.

Demostración: Veamos que esto es cierto para las erosiones: sea $z \in \mathbf{U}$ tal que $\xi(A, B)(z) \in K'$, $\xi(A, B)(z) = \inf_{x \in \mathbf{U}} \{I(Bz(x), A(x))\}$. Al estar la sucesión $\{I(Bz(x), A(x)) \mid x \in \mathbf{U}\}$ acotada inferiormente, existe una subsucesión convergente a dicha cota. Si consideramos la sucesión $\{x_i\}_{i \in N} \subseteq \mathbf{U}$ tal que $I(Bz(x_i), A(x_i))$ pertenece a la subsucesión convergente a $\xi(A, B)(z)$ y tomamos como K el conjunto de valores $\{A(x_i)\}_{i \in N}$, resulta que $\xi(A \cap K, B)(z) = \xi(A, B) \cap K' = \xi(A, B)(z)$.

De forma similar se puede demostrar para las dilataciones. □.

El cuarto principio es el de la semicontinuidad de las transformaciones morfológicas, que como se puede comprobar Frago [11], depende de la semicontinuidad del operador grado de inclusión utilizado

4.- PROPIEDADES QUE DEPENDEN DEL OPERADOR IMPLICACIÓN

Se pueden demostrar los siguientes resultados.

Teorema 5: Sean I y J operadores implicación borrosa tales que I≤J, entonces se verifica: $\xi_I(A,B)(x)\geq\xi_J(A,B)(x)$, $\mathcal{D}_I(A,B)(x)\leq\mathcal{D}_J(A,B)(x)$ $\forall x\in U$

$@_I(A,B)(x)\leq@_J(A,B)(x)$, $\zeta_I(A,B)(x)\geq\zeta_J(A,B)(x)$ $\forall x\in U$

Teorema 6: Sean A, A',B y B'∈F(U), e I un operador implicación borrosa verificando las condiciones

i) a≤a' ⇒ I(a,b)≥I(a',b), a, a' y b∈[0,1]

ii) b'≥b ⇒ I(a,b)≥I(a,b') b, b' y a∈[0,1]

Entonces se verifica:

1) B⊆B' ⇒ R(A, B) ≤ R(A, B')

2) A⊆A' ⇒ R(A', B)≤R(A, B)

3) R(A∪A', B)= min(R(A,B), R(A', B))

4) R(A, B∩B')= min(R(A, B), R(A, B'))

Corolario 1: Sean A ⊆ A' e I un operador implicación borrosa verificando la condición a≤a' ⇒ I(a,b) ≥ I(a',b), a, a' y b∈[0,1]. Entonces se verifica:

$\xi(A,B)(x)\leq\xi(A',B)(x)$ $\forall x\in U$,

$\mathcal{D}(A,B)(x)\leq\mathcal{D}(A',B)(x)$ $\forall x\in U$

$@(A,B)(x)\leq@(A',B)(x)$ $\forall x\in U$,

$\zeta(A,B)(x)\geq\zeta(A',B)(x)$ $\forall x\in U$

Corolario 2: Sean B ⊆ B' e I un operador implicación borrosa verificando la condición

- b'≥b ⇒ I(a,b)≥I(a,b') b, b' y a∈[0,1]

Entonces se verifica:

$\xi(A,B')(x)\leq\xi(A,B)(x)$ $\forall x\in U$,

$\mathcal{D}(A,B)(x)\leq\mathcal{D}(A,B')(x)$ $\forall x\in U$

Corolario 3: Sean A,A',B y B'∈F(U), e I un operador implicación borrosa verificando las condiciones

a≤a' ⇒ I(a,b)≥I(a',b), a, a' y b∈[0,1].

b'≥b ⇒ I(a,b)≥I(a,b') b, b' y a□**[0,1]**

Entonces se verifica:

$\xi(A, B\cup B') = \xi(A, B)\cap\xi(A\cap B')$,

$\xi(A\cap A', B) = \xi(A, B)\cap\xi(A'\cap B)$,

$\mathcal{D}(A, B\cup B') = \mathcal{D}(A, B)\cup\mathcal{D}(A\cap B')$

Teorema 7: Si el operador Implicación borrosa verifica I(a, b) = I(n(b), n(a)), entonces $R(A, B) = R(B^C, A^C)$.

Corolario 3: Si la dilatación se ha definido utilizando un operador implicación borrosa que verifica I(a, b) = I(n(b), n(a)), siendo n una negación fuerte. Entonces la dilatación es conmutativa, $\mathcal{D}(A, B) = \mathcal{D}(B, A)$.

5.- ANÁLISIS DE LAS DISTINTAS DEFINICIONES DE EROSIÓN: EFECTOS PRACTICOS EN EL PROCESAMIENTO DE IMÁGENES

Se presenta a continuación un estudio de los efectos del elemento estructurante en las modificaciones de los niveles de gris de la imagen de salida ($\xi(A, B)$) al procesar una imagen digitalizada A con el operador erosión.

Consideramos como elemento estructurante, una imagen constante (representada aquí por una matriz nxn, con todos sus elementos con valor α), con el nivel de gris de cada pixel determinado por el valor α∈[0, 1].

La formula de la erosión utilizada es: $\xi(A, B)(x) = \inf_{y\in D_B}\{I(B(x), A(x+y))\}$, $x\in D_A$, siendo I un operador Implicación, D_B dominio de B. Para este caso particular en el que $B(y) = \alpha$ $\forall y\in D_B$, la formula de la erosión resulta $\xi(A, B)(x) = \inf_{y\in D_B}\{I(\alpha, A(x+y))\}$,

Se puede observar (Anexo A) que para las erosiones definidas con las implicaciones 1, 2, 3, 5, 7, 8, 9, 10, 11, 13, 14, 17 y 18, que verifican la propiedad I(1, b) = b $\forall b\in[0,1]$, tienen todas ellas la misma imagen de salida cuando el elemento estructurante es una imagen blanca (todo unos). La imagen de salida es una contracción de la de entrada con algunos cambios en los niveles de gris determinados por la expresión

$$\xi(A, B)(x) = \inf_{y\in D_B}\{A(x+y)\} \quad \forall x\in D_A.$$

La propiedad I(0, b) = 1 $\forall b\in[0, 1]$, que cumplen todas las implicaciones 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 18, determina que las erosiones respectivas den una mancha blanca de salida para cualquier imagen de entrada cuando el elementos estructurante es una imagen negra (todo ceros), ya que $\xi(A, B)(x) = \inf_{y\in D_B}\{I(0, A(x+y))\}=1$..

Realizamos a continuación un análisis que nos permita una mejor comprensión de los efectos del operador erosión. Efectos que puede verse parcialmente reflejados en el Anexo A.

1. Implicación de Zadeh.

- Si 1> α >0.5, $\xi(A, B)(x) \leq \alpha$, por tanto todos los tienen un nivel de gris inferior a α, toda la imagen se oscurece y desaparecen las zonas blancas.
- Si α≤0.5, $\xi(A, B)(x) = 1-\alpha$, todos los pixels tienen el mismo nivel de gris 1-α, la imagen es una mancha gris.

2. Implicación de Lukasiewicz.

➢ Si $1>\alpha$, $\xi(A, B)(x)=$

$\inf_{y\in D_B}\{\min(1.\ 1-\alpha+A(x+y)\} > \inf_{y\in D_B}\{A(x+y)\}$,

toda la imagen se hace más clara dependiendo del valor de α.

3. Implicación de Mandani.

➢ Si $1> \alpha \geq 0$. $\xi(A, B)(x) =$ $\inf_{y\in D_B}\{$Si $A(x+y)\leq \alpha$ entonces $A(x+y)$ en otro caso $\alpha\}$,

el nivel de gris de cada pixel siempre está por debajo de α, la imagen se hace más oscura.

➢ Si $\alpha=0$, $\xi(A,B)(x)=0$,

todos los pixel de la imagen tienen nivel de gris 0, la imagen es una mancha negra.

4. Implicación Estándar estricto.

➢ Si $\alpha=1$,

$\xi(A, B)(x) = \inf_{y\in D_B}\{$Si $1\leq A(x+y)$ entonces 1 en otro caso $0\}$,

mantiene los blancos y convierte todos los grises en negro.

➢ Si $\alpha<1$,

$\xi(A, B)(x) = \inf_{y\in D_B}\{$Si $\alpha\leq A(x+y)$ entonces 1 en otro caso $0\}$,

mantiene las zonas blancas, convierte los grises con valor menor que α en negro y el resto los pasa a blancos.

5. Implicación de Godel.

➢ Si $\alpha<1$,

$$\xi(A, B)(x) = \begin{cases} 1 & si \quad \alpha \leq A(y+x) \quad \forall y \in D_B \\ \inf_{y\in D_B}\{A(y+x)\} & en \quad otro \quad caso \end{cases},$$

las zonas con grises más claros que el elemento estructurante pasan a blancos, en los demás casos toma el mínimo de los elementos de la imagen implicados.

6. Implicación Estándar Strict-Star.

➢ Si $\alpha=1$,

$$\xi(A, B)(x) = \begin{cases} 1 & si \quad A(y+x)=1 \quad \forall y \in D_B \\ 0 & en \quad otro \quad caso \end{cases},$$

deja solamente las zonas blancas.

➢ Si $\alpha<1$,

$$\xi(A, B)(x) = \begin{cases} \inf_{y\in D_B}\{A(y+x)\} & si \quad \alpha \leq A(y+x) \quad \forall y \in D_B \\ 0 & en \quad otro \quad caso \end{cases}$$

deja las zonas con grises más claros que el elemento estructurante toma el mínimo de los elementos de la imagen implicados, el resto los pasa a negros.

7. Implicación Estándar Star-Strict.

$$\text{Si } \alpha<1,\ \xi(A, B)(x) = \begin{cases} 0 & si \quad \alpha \leq A(y+x) \quad \forall y \in D_B \\ \inf_{y\in D_B}\{A(y+x)\} & en \quad otro \quad caso \end{cases},$$

deja las zonas blancas o grises más claros que el elemento estructurante y pasan a negro el resto toma el mínimo los puntos da la imagen implicados.

8. Implicación Estándar Star-Star.

➢ Si $\alpha<1$,

$$\xi(A, B)(y) = \begin{cases} \inf_{y\in D_B}\{A(x+y)\} & si \quad \alpha > A(y+x) \quad \forall y \in D_B \\ \in \{A(y+x), 1-A(y+x) \mid y \in D_B\} & en \quad otro \quad caso \end{cases},$$

en los puntos grises más obscuros que el elemento estructurante toma el mínimo de los pixel de la imagen implicados.

9. Implicación Estándar Strict-Strict.

➢ Si $\alpha=1$,

$$\xi(A, B)(x) = \begin{cases} 1 & si \quad A(y+x)=1 \quad \forall y \in D_B \\ 0 & en \quad otro \quad caso \end{cases},$$

solo deja las zonas blancas.

➢ Si $\alpha<1$, $\xi(A, B)(x) = 0$.

10. Implicación de Kleene-Dienes.

➢ Si $\alpha<1$,

$$\xi(A, B)(x) = \begin{cases} \inf_{y\in D_B}\{A(y+x)\} & si \quad 1-\alpha \leq A(y+x) \quad \forall y \in D_B \\ 1-\alpha & en \ otro \ caso \end{cases}.$$

11. Implicación de Gaines.

➢ Si $\alpha<1$,

$$\xi(A, B)(x) = \begin{cases} 1 & si \quad \alpha \leq A(y+x) \quad \forall y \in D_B \\ \inf_{y\in D_B}\left\{\dfrac{A(y+x)}{\alpha}\right\} & en \quad otro \quad caso \end{cases},$$

las zonas con niveles de gris mayor que α las convierte en zonas blancas, en las otras el nivel de gris queda determinado por los cocientes $\dfrac{A(y+x)}{\alpha}$.

12. Implicación de Gaines modificado.

➢ Si $\alpha=1$,

$$\xi(A, B)(x) = \begin{cases} 1 & si \quad A(y+x)=1 \quad \forall y \in D_B \\ 0 & en \quad otro \quad caso \end{cases}$$

deja solo las zonas blancas.

➢ Si $0<\alpha<1$,

$$\xi(A, B)(x) = \inf_{y\in D_B}\left\{min(1, \frac{A(y+x)}{\alpha}, \frac{1-\alpha}{1-A(y+x)})\right\}.$$

➢ Si $\alpha=0$, $\xi(A, B)(x) = 1$.

13. Implicación de Kleene-Dienes-Lukasiewicz.

➢ Si $0<\alpha<1$,

$\xi(A, B)(x) = \inf_{y\in D_B}\{1-\alpha+A(y+x)\alpha\}$,.

➢ Si $\alpha=0$, $\xi(A, B)(x) = 1$.

14. Implicación de Willmott.

- Si $0.5<\alpha<1$, $\xi(A, B)(x) =$

$$\begin{cases} \alpha & si \quad A(y+x)=1 \quad \forall y \in D_B \\ \inf\limits_{i \in D_B} \{min(\alpha, A(y+x)\} & si \quad A(y+x) \geq 0.5 \\ \inf\limits_{i \in D_B} \{max(1-\alpha, A(y+x)\} & si \quad A(y+x) < 0.5 \end{cases}$$

- Si $0.5>\alpha$, $\xi(A, B)(x) =$

$$\begin{cases} 1-\alpha & si \quad A(y+x)=1 \quad \forall y \in D_B \\ \inf\limits_{i \in D_B} \{min(1-\alpha, A(y+x)\} & si \quad A(y+x) \geq 0.5 \\ \inf\limits_{i \in D_B} \{min(1-\alpha, 1-A(y+x)\} & si \quad A(y+x) < 0.5 \end{cases}$$

15. Implicación Estándar Sharp.

- Si $\alpha=1$,

$$\xi(A, B)(x) = \begin{cases} 1 & si \quad A(y+x)=1 \quad \forall y \in D_B \\ 0 & en \quad otro \quad caso \end{cases},$$

deja solo las zonas blancas.

- Si $\alpha<1$, $\xi(A, B)(x) = 1$.

16. Implicación de Wul.

- Si $\alpha=1$,

$$\xi(A, B)(x) = \begin{cases} 1 & si \quad A(y+x)=1 \quad \forall y \in D_B \\ 0 & en \quad otro \quad caso \end{cases},$$

deja solo las zonas blancas.

- Si $\alpha<1$,

$$\xi(A, B)(x) = \begin{cases} 1 & si \quad \alpha \leq A(y+x)=1 \quad \forall y \in D_B \\ \inf\limits_{y \in D_B} \{A(y+x)\} & si \quad A(y+x)<1-\alpha, \alpha \quad \forall y \in D_B. \\ 1-\alpha & en \quad otro\ caso \end{cases}$$

17. Implicación de Wu2.

- Si $\alpha<1$,

$$\xi(A, B)(x) = \begin{cases} 0 & si \ \exists y \in D_B \ni \alpha < A(y+x) \\ \inf\limits_{y \in D_B} \{A(y+x) & si \ \alpha \geq A(y+x) \ \forall y \in D_B \end{cases}$$

- Si $\alpha=0$, $\xi(A, B)(x)=0$

18. Implicación de Yager.

- Si $0<\alpha<1$,

$$\xi(A, B)(x) = \inf_{y \in D_B} \{A(y+x)^{\alpha}\} > \inf_{y \in D_B} \{A(y+x)\}.$$

5.- CONCLUSIONES

La dilatación y erosión para subconjuntos borrosos, definidos aquí a partir de los grados de inclusión, recuperan propiedades de estas mismas operaciones de la Morfología Matemática ordinaria. En particular, verifican el primer principio, (invarianza para traslaciones) y el segundo, (suficiencia del conocimiento local de un objeto para obtener un conocimiento local de la imagen transformada), establecidos por Serra [17] para las transformaciones morfológicas. El cumplimiento del segundo y cuarto principio establecidos por Serra [17] depende de que la implicación borrosa utilizada sea semicontinua e invariante para homotecias.

En el caso de utilizar imágenes y elementos estructurantes binarios, las erosiones y dilataciones borrosas definidas aquí producen en general los mismos efectos que los operadores análogos en la Morfología Matemática ordinaria.

Agradecimientos: Los autores agradecen a los referees anónimos las sugerencias aportadas a este trabajo.

REFERENCIAS

[1] W. Bandler and L. Kohout, *Fuzzy Power sets and Fuzzy Implication Operator.* Fuzzy Sets and Systems, 4(1980) 13-30.

[2] I. Bloch and H. Maître, Constructing a Fuzzy Mathematical Morphology: Alternative ways, Telecom Paris 92 C 002, September 1992.

[3] S. Divyendu and E. R. Dougerty, Characterization of Fuzzy Minkowski Algebra, SPIE Vol. 1769 Image Algebra and Morphological Image Processing III (1992) 59-69.

[4] S. Divyendu and E. R. Dougherty, Fuzzification of Set inclusión, SPIE Vol. 1708 Applications of Artificial Intelligence X: Machine Vision and Robotics (1992) 440-449.

[5] S. Divyendu and E. R. Dougherty, Fuzzification of set inclusión: Theory and applications, Fuzzy Sets and Systems 55 (1993) 15-42.

[6] S. Divyendu and E. R. Dougherty, Fuzzy Mathematical Morphology, Journal of Visual Communication and Image Representation, Vol. 3, No. 3, September 1992, 286-302.

[7] D. Dubois and H. Prade H., Fuzzy Logics and the generalized modus ponens revisited. Cybernetics and Systems: An International Journal, 15 (1984) 293-331.

[8] D. Dubois and H. Prade, Fuzzy Sets and Systems: Theory and Applications, Mathematics in Science and Engineering, (Academic Press, New York 1980).

[9] D. Dubois and H. Prade, Fuzzy sets in approximate reasoning, Part 1: Inference with possibility distributions. Fuzzy Sets and Systems 40 (1991) 143-202.

[10] D. Dubois, L. Jérôme and H. Prade, Fuzzy Sets in approximate reasoning, Part 2: Logical approaches. Fuzzy Sets and Systems 40 (1991) 203-244.

[11] N. Frago, Morfología Matemática Borrosa basada en operadores generalizados de Lukasiewicz: Procesamiento de imagenes. Ph.D Thesis, Universidad Pública de Navarra, Spain 1996.

[12] R. G. Giardina, E. R. Dougherty, Morphological Methods in Image and Signal Processing, (Prentice-Hall, New Yersy 1988).

[13] A. Haas, G. Materon and J. Serra. Morphological convolution for image processing, APIE Vol 1199, Visual

Communications and Image Procssing IV (1989) 1177-1182.

[14] E. Kerre, A comparative study of the behavior of some popular fuzzy implication operators on the generalized modus ponens. Fuzzy Logic for The Management of Uncertainty (Zadeh L. A., Kacprzyk J. eds, Wiley J., New York, 1992), 281-295.

[15] G. Matheron, Random Sets and Integral Geometry, (John While & Sons, New York 1975).

[16] D. Ruan and E. E. Kerre, Fuzzy implication operators and generalized fuzzy method of cases, Fuzzy Sets and Systems 54 (1993) 23-27.

[17] J. Serra, Image Analysis and Mathematical Morphology, Vol. 1 y 2, Academic Press, London 1982.

[18] P. Smets and P. Magrez, Implications in Fuzzy Logic. International Journal of Approximate easoning, 1 (1987) 327-347.

ANEXO A

Presentamos la Tabla 1, donde se recogen los efectos producidos sobre una imagen A con distintos niveles de gris, al erosionarla con los diferentes operadores erosión asociados a las 18 implicaciones anteriormente mencionadas. Hemos utilizado tres elementos estructurantes 3x3 definidos con valor constante, 1, 0.7, 0.4. La imagen de entrada A utilizada es la de la Figura 1.

[19] E. Trillas and L. Valverde, On inference in fuzzy logic, Proc. 2nd IFSA Congress, Tokyo, Japan, (1987) 294-297.

[20] E. Trillas and L. Valverde, On mode and implication in approximate reasoning, in: Aproximate Reasoning in Expert System (M. M Gupta et al., Eds.), North-Holland, Amsterdam, (1985) 105-112.

[21] E. Trillas y S. Cubillo, Modus Ponens on Boolean Algebras Reviseted, Mathware & Soft Computing, 3(1996), pp. 47

[22] R, Trillas, C. Campos, S. Cubillos, Nota sobre el concepto de implicación, Estylf'98.

[23] J. Xu and C. R. Giardina, The algebraic structure of the generalizatión of binari Morphology using Fuzzy set concept. Inst.Graphic Commun Waltham, MA, USA, 1988, 292-297

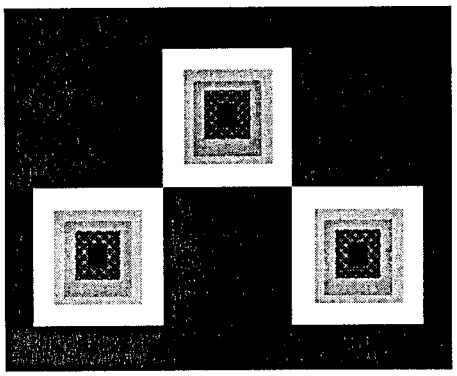
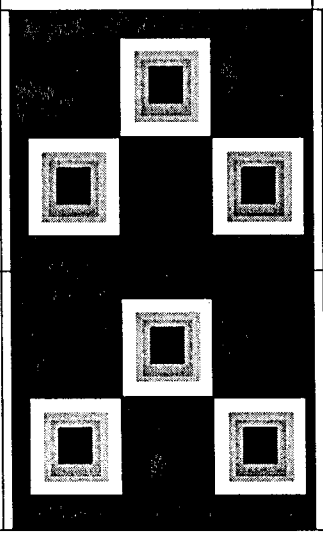
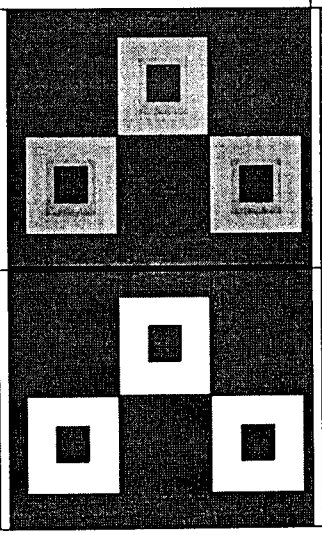
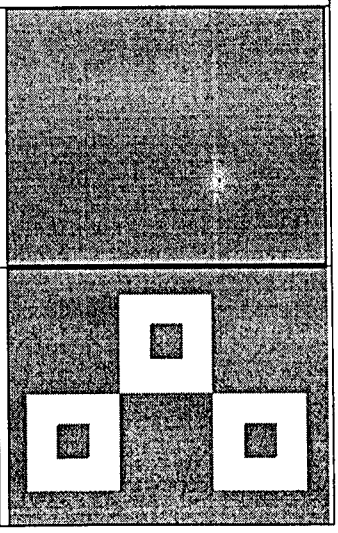
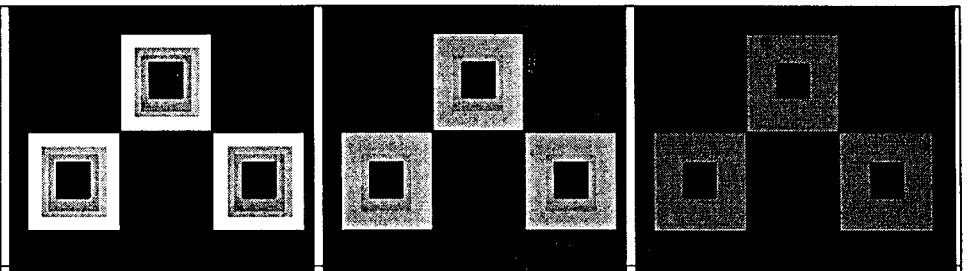
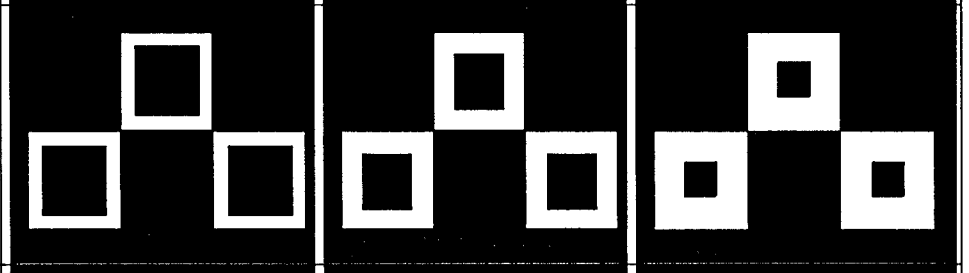
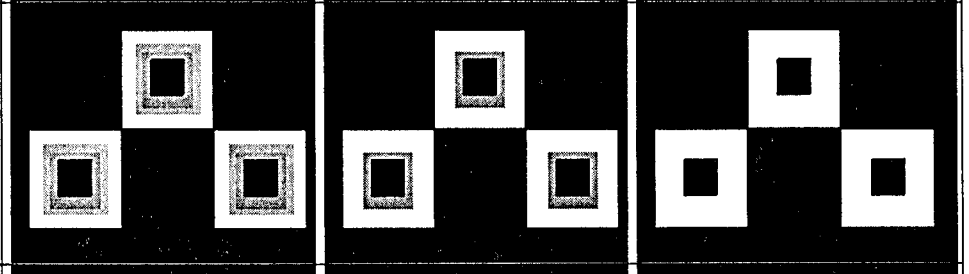
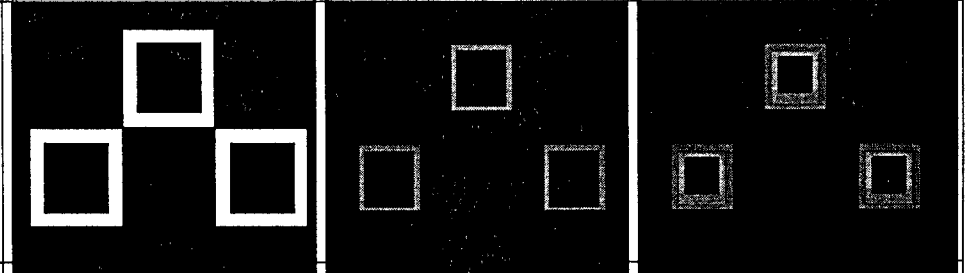
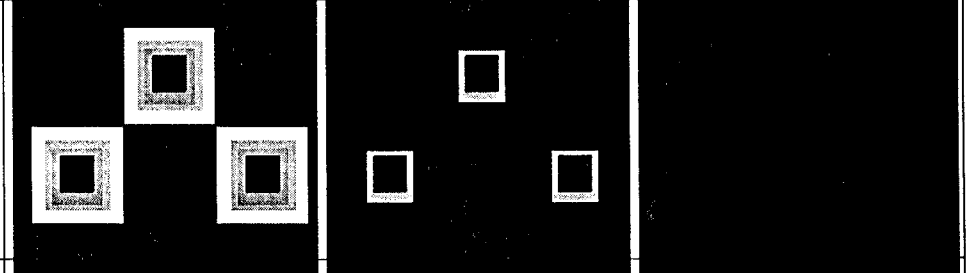
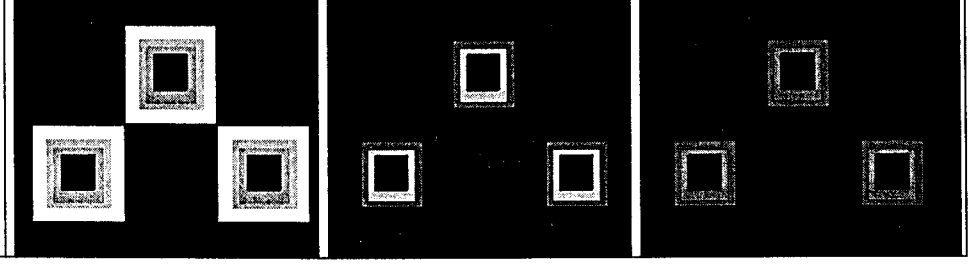
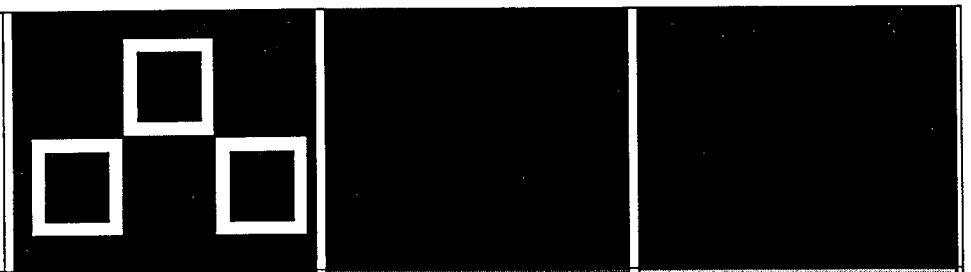
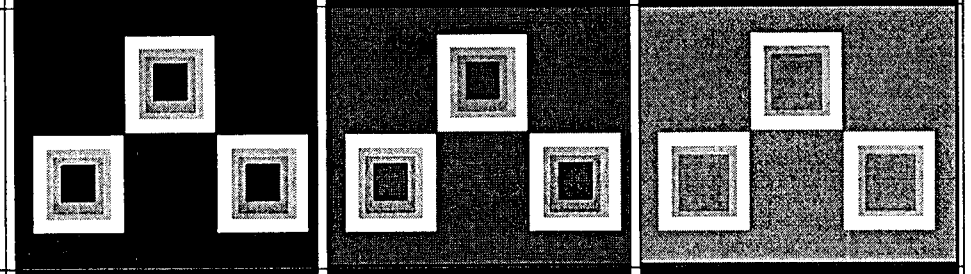
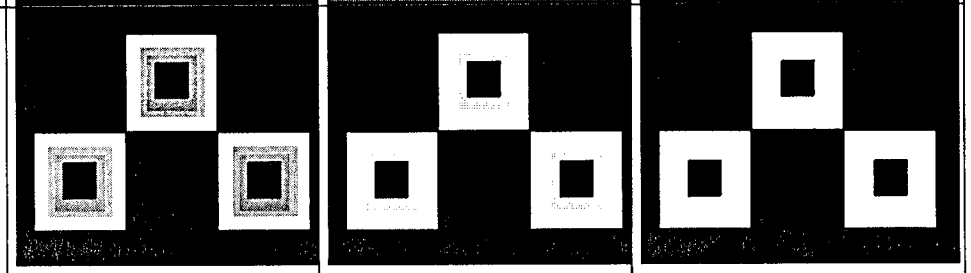
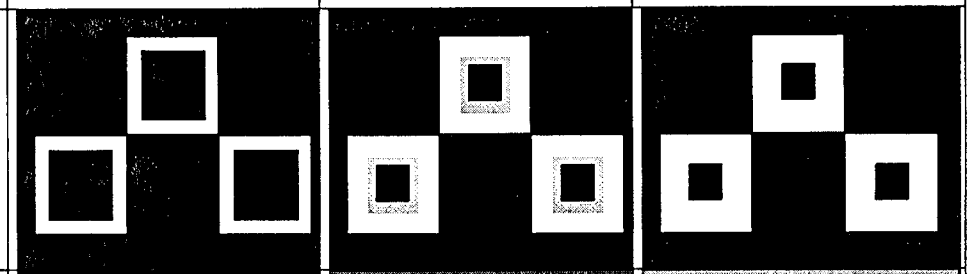
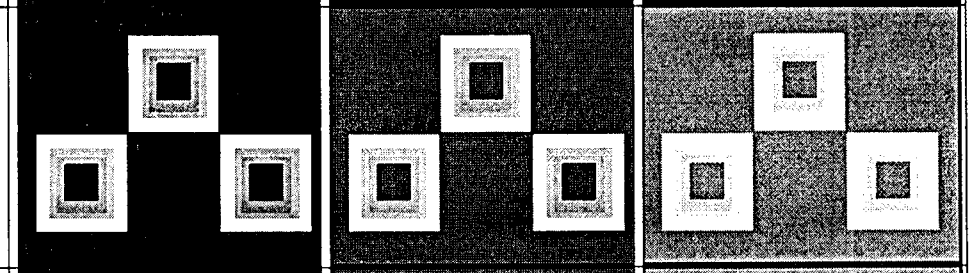
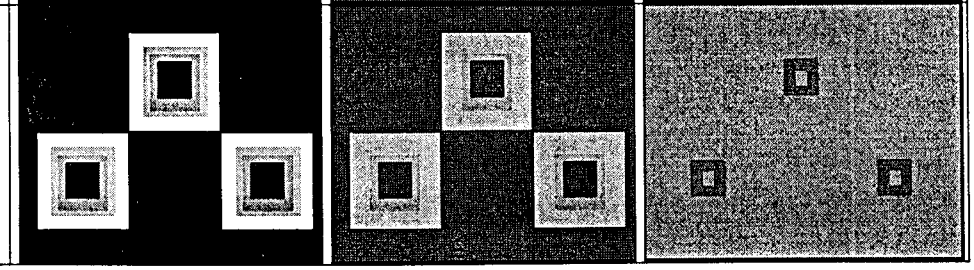


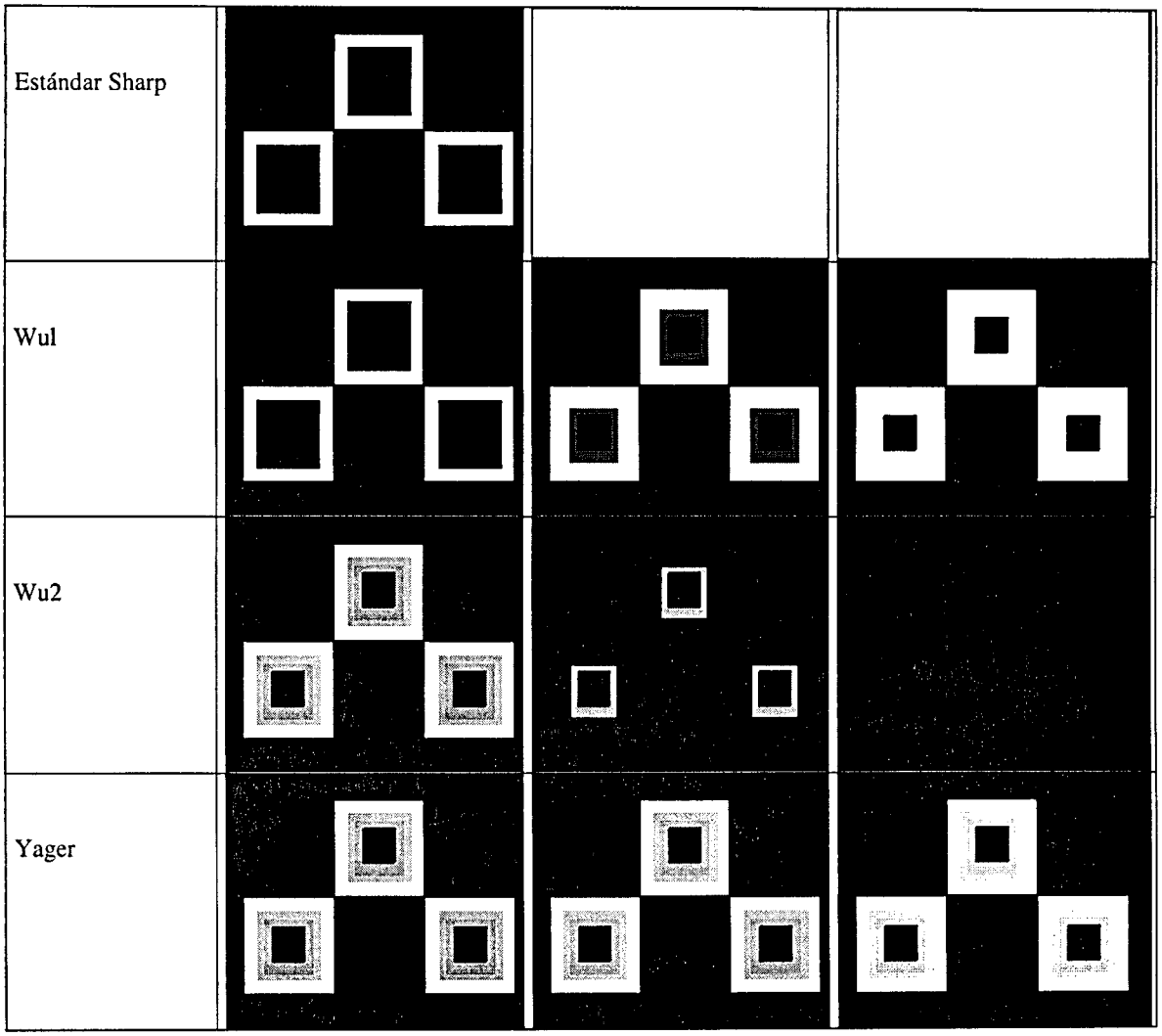
Figura 1: imagen A

Tabla 1

E.Estructurante / Implicación	B(i,j)=1	B(i,j)=0,7	B(i,j)=0,4
Zadeh			
Lukasiewicz			

Mandani	
Estándar "Strict".	
Godel	
Estándar "Strict Star".	
Estándar "Star Strict".	
Estándar "Star Star".	

Estándar "Strict Strict".	
Keene-Dienes	
Gaines	
Gaines modificado	
Lkeene Dienes Lukasiewicz	
Willmot	


Estándar Sharp
Wul
Wu2
Yager

Adquisición Automática de Conocimiento basada en Constructores Personales y en aprendizaje automático

J.L. Castro, J.J. Castro-Schez*, J.M. Zurita

Dpto. Ciencias de la Computación e I.A.
E.T.S.I. Informática. Universidad de Granada
Avenida de Andalucía, 38, 18071 Granada (Spain)
castro@decsai.ugr.es zurita@decsai.ugr.es

* Dpto. Sistemas
Caja Rural de Granada
Circunvalación, 2, 18006 Granada
administ.granada@cajarural.com

Resumen

La adquisición del conocimiento (AC) necesario para la construcción de un Sistema Experto, es una de las actividades clave en la Ingeniería del Conocimiento. La calidad del conocimiento adquirido determinará en un alto porcentaje la utilidad y aceptación del sistema desarrollado. La aparición de problemas como el excesivo tiempo dedicado a la fase de AC, la dificultad que entraña la comunicación del conocimiento experto, la dificultad para acceder al experto, la pobre selección y la mala aplicación de las distintas técnicas de AC, las traducciones erróneas del conocimiento adquirido a una representación funcional u operacional,... provoca que se comience a investigar y desarrollar metodologías y herramientas que automaticen de alguna manera parte o todo el proceso de AC. De esta manera podemos encontrar herramientas desarrolladas para tal propósito que utilizan por lo general lógica binaria para representar el conocimiento del dominio. Sin embargo esta restricción es insuficiente para modelar sistemas complejos (experto) ya que por lo general las cosas no suelen ser absolutamente ciertas o falsas. En este trabajo pretendemos desarrollar una metodología que hace uso de ideas del campo de la Psicología (La *Teoría de los Constructores Personales de Kelly*) y de la Inteligencia Artificial (Aprendizaje Automático) y cuyo mecanismo de representación son los conjuntos difusos.

Palabras Clave: Sistemas Expertos, Ingeniería del conocimiento, Adquisición del conocimiento.

1. INTRODUCCIÓN

La mayoría de herramientas desarrolladas para automatizar el proceso de AC (ETS[1], AQUINAS[2], MOLE[7], etc.) utilizan como método de representación la lógica binaria. Esto provoca que muchas veces las expresiones aprendidas sean poco naturales y que no se represente completamente el conocimiento del experto. Existe en todo experto y en general en todo humano una imprecisión e incertidumbre que hace que estos asignen valoraciones que se encuentran entre la verdad y la falsedad absoluta. De esta manera se presenta el concepto de *Conjunto Difuso* y *Lógica Difusa* en los sistemas expertos, como un método natural de representar la manera de razonar del experto.

Otra dificultad a la que hacen frente tanto las herramientas automáticas de AC como los ingenieros de conocimiento es la accesibilidad del experto. Hay que tener en cuenta que los expertos son bienes escasos de las empresas u organizaciones a los cuales pertenecen y estas tienden a proteger su tiempo de una manera especial. Sólo permiten pequeñas interrupciones y a veces el ingeniero de conocimiento debe emplear las más diversas tretas para acceder a éste. Sin embargo la empresa siempre está dispuesta a proporcionar datos o registros de actuaciones históricas del experto. Una vez que el ingeniero de conocimiento posee algún conocimiento sobre el dominio, gracias al análisis de los registros proporcionados, ya está en disposición de aprovechar mejor los contactos con el experto y hacer preguntas más interesantes en entrevistas mejor planificadas. Como ya se ha mencionado anteriormente las herramientas automáticas de AC también comparten este problema, aunque de una manera diferente. Las herramientas de AC automáticas reducen de alguna manera el problema de accesibilidad del experto ya que éste podrá trabajar con la herramienta cuando desee. Pero hay que señalar que el experto no estará dispuesto a entablar "conversación" con una herramienta que parte siempre de lo que él le dice, es decir la herramienta debe de "cautivarlo" y muchas veces es necesario que la herramienta aprenda cosas a través del estudio previo de ciertos datos y haga cuestiones al experto basándose en este conocimiento adquirido.

La investigación conduce al empleo de la lógica difusa en el desarrollo de sistemas expertos, sin embargo las reglas difusas y las funciones de pertenencia son difíciles de definir. Se hace necesario el desarrollo de herramientas que adquieran este conocimiento, o por lo menos faciliten al experto la tarea de comunicarlo. También se comienzan a desarrollar herramientas que toman provecho del conocimiento presente en un conjunto de ejemplos, ya que en la mayoría de ocasiones un ingeniero de conocimiento se va a encontrar a lo largo del proceso de construcción de un SE con un conjunto de ejemplos.

Nosotros pretendemos desarrollar una metodología que ayude al experto a expresar su conocimiento de forma natural, de manera que el experto vaya creando y depurando su base de conocimiento. Como mencionamos anteriormente el problema de la accesibilidad del experto es un problema añadido al de la AC. Es por esto por lo que la metodología propuesta prescinde del experto siempre que sea posible.

En este trabajo, queremos mostrar una metodología para la AC automática que integra ideas provenientes de dos de los campos de investigación abiertos actualmente en el área de la AC: la psicología y la inteligencia artificial, *Castro*[5]. Así como presentar la lógica difusa como un buen método para representar el conocimiento adquirido que facilitará la comunicación entre el experto y la herramienta de AC automática.

2. DECRIPCIÓN DEL PROBLEMA Y SOLUCIÓN PROPUESTA

Partimos de la suposición de que queremos extraer un conocimiento que resuelve un problema de clasificación o decisión. Es decir se pretende decidir a que clase o tipo pertenece un objeto o que decisión tomar a partir de cierta información de entrada que por lo general podrá ser propiedades del objeto o variables que de alguna manera influyen sobre la decisión a tomar.

De partida solamente sabemos lo anteriormente expuesto, es decir que tratamos con un problema en el cual el experto lo que hace es clasificar o decidir (tomar una decisión) en función de una información de entrada, y para obtener más información contamos únicamente con la presencia del experto (de cara a la automatización).

En este contexto proponemos un enfoque basado en la *Teoría de los Constructores Personales* de Kelly[10] para obtener un conocimiento inicial. La esencia de esta teoría subyace en que cada persona observa y comprende el entorno en el que se mueve según su punto de vista, usando sus propias reglas y tomando conclusiones conforme a su propio criterio. Estas reglas y criterios son denominados *constructores personales*. Si nosotros queremos conocer la manera en que razona un experto será muy útil conocer cuáles son sus constructores personales. La *técnica de la rejilla de repertorio*[11] (repertory grid) es una manera muy eficiente de obtenerlos. Esta técnica ha sido adoptada por los ingenieros de conocimiento para obtener el conocimiento de los expertos del dominio.

Se propone está técnica como un método útil para identificar las variables de entrada y los diferentes resultados que utiliza y produce respectivamente el experto en su razonamiento. Realizando un análisis completo podemos obtener las decisiones o diferentes tipos y las variables o propiedades que ayudan al experto a decidir, así como un pequeño conjunto de reglas que solucionan el problema en un primer acercamiento (reglas evidentes para el experto).

Una vez hemos obtenido las variables o propiedades que utiliza el experto para decidir ($X_1,..,X_n$), podemos solicitar a la organización a la cual el experto pertenece un conjunto de ejemplos χ, cuyos elementos tendrán la siguiente forma *($x_1,..,x_n$:decisión de su experto)* con x_i siendo un valor perteneciente al dominio de la variable X_i y "decisión de su experto" un elemento detectado como una posible decisión del experto.

Llegados a este punto proponemos el uso del algoritmo desarrollado en *Castro*[6] para obtener el conocimiento presente en los ejemplos generando un conjunto de reglas claras.

Una vez tenemos este conjunto de reglas (reglas iniciales y reglas claras), podemos utilizarlas para desarrollar un conjunto de estrategias de entrevista, como hace Kahn[9], que conduzcan a una base de reglas más poderosa. Estas estrategias proporcionan un mecanismo para entrevistar a los expertos del domino y no sólo liberan al ingeniero de conocimiento de la tarea de entrevista sino que además permiten una rápida medida de la fortaleza de la actual base de conocimiento y la selección de las cuestiones que solventan las debilidades encontradas. Una técnica de entrevista que proponemos está basado en el algoritmo presentado por *Castro*[3] para validar SE. Lo que hace este algoritmo es extraer entradas donde el sistema aún no ha producido respuestas o donde se han producido inconsistencias. Nosotros lo que solicitamos una vez encontradas estas entradas es la resolución del conflicto o la

incorporación de nuevas reglas que den respuesta al caso para el cual no se ha especificado aún nada. La idea de ir construyendo una base de conocimiento en función de mejoras sobre una base de conocimiento anterior no es nueva y ha sido propuesta por otros autores como *Tecuci*[13].

La metodología propuesta sigue la línea presentada por *Castro*[4] para el aprendizaje automático de un SBC con la colaboración de un experto, a la cual incorporamos un método para extraer las variables relevantes del domino, las cuales serán necesarias en el proceso de aprendizaje automático. No aceptamos cualquier conjunto de ejemplos sino solamente aquel que tiene la *estructura útil*, entendiendo por estructura útil aquella que contiene las variables que realmente nos interesan, y estas serán proporcionadas por la técnica de la rejilla de repertorio como ya hemos visto.

3. LOS CONSTRUCTORES PERSONALES Y LA REJILLA DE REPERTORIO

Como ya hemos comentado en la sección anterior una persona (experto) observa y comprende su entorno (dominio) según su punto de vista, usando sus propias reglas y tomando conclusiones conforme a su propio criterio. Todo esto vendrá determinado por sus *constructores personales*, que son dimensiones de conocimiento. Cada constructor, por definición es una distinción bipolar simple. Por ejemplo si un profesor quiere conocer como son sus alumnos, este podría utilizar el constructor diligente/indolente. Un constructor no tendrá sentido completo sin considerar el significado de los dos polos del constructor.

Un constructor personal se puede definir como un atributo o propiedad cuyos diferentes valores puede distinguir un objeto o grupo de objetos de cualquier otro. Cada experto posee su propio conjunto de constructores que le permiten comprender un dominio. Por esto al afrontar un problema de AC, necesitaremos conocer cuáles son y cómo tiene organizado el experto los constructores (relación entre ellos). El trabajo de extraer los constructores y analizar las relaciones entre ellos va a ser difícil, Kelly propone el repertory grid o rejilla de repertorio como una metodología para realizarlo.

La técnica de la rejilla de repertorio es esencialmente una prueba de elección compleja en la cual una lista de elementos es juzgada sucesivamente sobre la base de un conjunto de constructores bipolares. La forma más básica de una rejilla de repertorio es una matriz rectangular en la que los estados o eventos alternativos (elementos) constituyen las columnas y las maneras (constructores) en las cuales nosotros medimos la similaridad o diferencia entre elementos forman las filas. Cada intersección columna-fila en la matriz contiene la aplicación de uno u otro polo (1 ó 0) a un elemento. En la Figura 1 se muestra un ejemplo de Rejilla de Repertorio.

	Juan	Miguel	Eloy	
Diligente	1	0	0	Indolente
Hábil	1	1	0	Torpe
Conciso	0	0	1	Prolijo
Enérgico	1	1	1	Débil

Figura 1. Una rejilla de repertorio bipolar.

Esta fue la forma original acuñada por la rejilla de repertorio, pero esta presenta algunos problemas como pueden ser:

- ¿Qué ocurre si un atributo tiene sentido solamente para un subconjunto del conjunto total de elementos? Por ejemplo, "Buena relación con los padres" no está definido para aquellos alumnos huérfanos.
- ¿Qué ocurre si un atributo no es bipolar? Por ejemplo, "Aficiones de los alumnos" no se puede contestar con 1 ó 0, ya que este constructor podrá tomar valores como Deportes, Música, Dibujo, Teatro,...
- ¿Qué ocurre si el experto no es capaz de valuar a un objeto con valores extremos? Por ejemplo, puede que el experto no sea capaz de decir que la alumna Sofía sea Torpe pero tampoco puede decir que sea Hábil.

Esto conduce a otro tipo de rejillas de repertorio a las cuales se permiten entradas a NUL, es decir atributos no aplicables, a la introducción de atributos multivaluados, y a permitir la introducción de valuaciones o graduaciones. El experto gradúa la aplicación de un atributo a un elemento, asigna un valor de 1 a *n*, donde 1 significa la aplicación de un polo del constructor y *n* la aplicación del polo opuesto. Los valores entre 1 y *n* designan estados intermedios. Por lo general el valor *n* será un valor manejable 5, 7.

Este tipo de rejillas presenta otro tipo de problema que ya ha sido mencionado en la sección primera. Los expertos suelen emplear términos imprecisos para definir determinados atributos. De esta manera se puede construir una nueva rejilla de repertorio con valores difusos, como

ejemplo véase *Hwang*[8]. En la Figura 2, se muestra un ejemplo.

	E_1	E_2	E_3	E_4	E_5	
Variable 1	3/S	0/N	-3/S	1/S	3/S	BAJO-MEDIO-ALTO
Variable 2	2/S	1/S	3/S	2/S	0/S	PEQUEÑO-MEDIO-GRANDE
Variable 3	2/S	2/S	3/S	1/N	2/S	LENTO-NORMAL-RAPIDO
Variable 4	-1/N	2/S	-2/S	-3/S	3/S	BAJO-MEDIO-ALTO

Figura 2. Rejilla de repertorio con valores difusos (Adaptada de *Hawng*)

Donde por ejemplo, 3 en la variable 1 significa MUY ALTO, 2 ALTO, 1 MÁS O MENOS ALTO, 0 MEDIO, -1 MÁS O MENOS BAJO, -2 BAJO y -3 MUY BAJO. El grado de certeza para cada valoración es descrito por medio de "S" y "N", las cuales representan MUY SEGURO y NO MUY SEGURO, respectivamente.

Para nosotros este tipo de rejillas de repertorio tiene un problema y este es el siguiente, la mayoría de las veces un experto no será capaz de dar un valor categórico de una variable para un elemento. Es decir, se producirán frecuentemente situaciones en las que el experto asignará a una variable varios valores (por dualidad o por incertidumbre), por ejemplo en el caso anterior (Figura 3) puede ocurrir que la variable 3 para el elemento E_1 sea RAPIDO o MUY RAPIDO.

Nosotros apostamos por rejillas de repertorio en las que convivan todos estos tipos de valores, a saber bipolares, multivaluados, graduaciones y valores difusos. Además pretendemos que un experto pueda dar más de un valor para un constructor del mismo elemento. Es decir, preguntado un experto por el valor que toma un determinado constructor, por ejemplo Diligente/Indolente, en un elemento, por ejemplo el alumno Miguel, este puede responder un doble valor 4,5. En la Figura 3 se puede ver un ejemplo de la rejilla de repertorio propuesta.

Para desarrollar esta rejilla, la herramienta deberá extraer los elementos que utiliza el experto. Esto se podrá realizar de varias formas: la primera mediante la especificación directa por parte del experto, esta se utilizará cuando el experto tenga suficientemente claro cuáles son los elementos con los que él trabaja y la segunda empleando una batería de preguntas cuyo objetivo final será obtener dicho conjunto.

Una vez tenemos los elementos, pasamos a extraer los constructores (las características que discriminan o marcan la similaridad entre los elementos). Para hacer esto proponemos el *método de las ternas* que consiste en presentar al experto los elementos de tres en tres y preguntarle cómo dos de ellos se diferencian del tercero. El experto dará un constructor, y el programa interrogará sobre el tipo de atributo con el que se está trabajando (bipolar, multivaluado, graduado, difuso), se definirá el dominio de esta variable. Si es difuso se construirán los conjuntos difusos y sus funciones de pertenencia, así como el número de etiquetas lingüísticas empleadas. Si es multivaluada se pedirán los distintos valores que puede tomar la variable y si es graduada se pedirá la graduación empleada. Todo esto también se hará mediante preguntas-respuestas. Una vez tenemos el constructor y su tipo el experto valorará cada elemento según este constructor extraído.

El método de las ternas nos parece apropiado para extraer los constructores ya que un experto por lo general no es capaz de comunicar de forma exhaustiva los constructores que emplea. Ya que la extracción en ternas se basa en distinciones entre elementos, el experto va a generar un conjunto mínimo de discriminantes. Cuando el experto tiene que comunicar un conjunto de discriminantes suele generar un conjunto más amplio, la mayoría de los cuales tendrán poco uso y lo único que hacen es introducir ruido en el proceso de AC.

Para obtener más información sobre el método de las ternas y cómo se lleva a cabo el proceso de construcción de la rejilla de repertorio, recomendamos la lectura de *Ruqian*[12].

Una vez tenemos construida la rejilla de repertorio, los resultados podrán ser analizados con el objetivo de descubrir similaridades y distancias entre objetos (elementos y constructores) de cara a la depuración del conocimiento extraído y al descubrimiento de relaciones entre los constructores y elementos.

Como resultado final en esta fase obtendremos un conjunto de reglas que contienen el conocimiento evidente del experto. Es decir aquellas reglas que el experto comunicaría al ingeniero de conocimiento en las primeras fases del proceso de AC, así como las variables importantes (aquellas en las que el experto basa su razonamiento).

4. UNA TÉCNICA DE APRENDIZAJE AUTOMATICO PARA OBTENER EL CONOCIMIENTO TÁCITO

Una vez conocemos las variables importantes, es decir aquellas en las que basa su razonamiento el experto, pedimos a la organización a la cual este pertenece un

	Alicia	Gloria	Miguel	Jóse	
Agilidad mental	BAJA,MEDIA	BAJA	MEDIA,ALTA	ALTA	
Estudiante BUP	1	0	1	1	Estudiante FP
Diligente	1	4	4,5	2,3	Indolente
Aficiones	Culturales	Deportivas	Deportivas	NUL	

Figura 3. Rejilla de repertorio propuesta.

conjunto de ejemplos que reflejen los valores que tenían esas variables y la respuesta que dio el experto.

Con este conjunto de ejemplos nosotros vamos a obtener el conocimiento que el experto no es capaz de proporcionar de forma directa, aplicando un algoritmo de aprendizaje automático encaminado a obtener las reglas más generales o claras (*Castro*[6]), que podremos contrastar con las obtenidas en el paso anterior. De esta manera vamos a depurar las reglas evidentes que representan las respuestas obvias del experto.

El algoritmo de aprendizaje propuesto lo que hace es construir un conjunto inicial de reglas difusas con el conjunto de ejemplos suministrados y a partir de estás elaborar un conjunto de reglas claras las cuales son generalizaciones de las reglas iniciales.

Para lograr esto se convierte cada ejemplo proporcionado por la organización en una regla difusa. El valor de cada variable de entrada será representado por medio de una etiqueta lingüística, y la salida del ejemplo será codificada (a cada valor diferente de salida le corresponderá un código distinto).

Ejemplo 4.1. Sea *((a,b,c,d),z)* un ejemplo de

entrenamiento, donde *a,b,c* y *d* son valores concretos del dominio de las variables de entrada X_0, X_1, X_2 y X_3 y *z* es el valor de la variable de salida. Este ejemplo será convertido en una regla difusa: $((L_{1i},L_{2j},L_{3k},L_{4l}),y_t)$, la cual es una manera abreviada de representar la regla:

IF X_0 is L_{1i} and X_1 is L_{2j} and X_2 is L_{3k} and X_3 is L_{4l} THEN Y is y_t

donde L_{1i}, L_{2j}, L_{3k} y L_{4l} son funciones de pertenencia y y_t el código que se asocia a la salida *z*.

La definición de las funciones de pertenencia dependerá del tipo de variable con el que estemos trabajando, es decir difusa, graduación, multivaluada, bipolar,... De manera que cuando se construyen las reglas iniciales a partir del conjunto de ejemplos de entrenamiento, lo que se hace es asignar al valor de cada variable de entrada del ejemplo la etiqueta que presenta un grado de pertenencia máximo.

En este punto será construido el conjunto de las reglas difusas que identifican o modelan el sistema. El proceso que conduce a la construcción de este conjunto de reglas definitivas es el siguiente:

Tomamos cada regla inicial y comprobamos si esta ya está *subsume* en alguna de las reglas del conjunto de reglas definitivas. Recordamos que una regla R_i $((E_{i1},E_{i2},...,E_{in}),y_j)$ subsume en otra regla R_k $((E_{k1},E_{k2},...,E_{kn}),y_x)$ si $E_{i1}\subseteq E_{k1}$, $E_{i2}\subseteq E_{k2}$,..., $E_{in}\subseteq E_{kn}$ e $y_j=y_x$.

Las etiquetas lingüísticas como tales sólo tendrán sentido en las variables de tipo difuso, en el resto de variables aunque se representen como etiquetas lingüísticas sólo interesarán las funciones de pertenencia asociadas a dichas etiquetas.

Ejemplo 4.2. La regla $((SN,L_{12},L_{21},ALTO),y_1)$ subsume en la regla: IF X3 is {ALTO,MUY ALTO} THEN Y is y_1. Cuando una variable no aparece en la regla, significa que puede tomar cualquier valor del conjunto de etiquetas en los cuales se ha dividido el dominio de esa variable. De esta manera si el conjunto de etiquetas en las cuales se dividen las variables X_0, X_1 y X_2 es:

$$X_0 \longrightarrow \{HN,MN,SN,Z,SP,MP,HP\}=\ell_1$$
$$X_1 \longrightarrow \{L_{11},L_{12}\}=\ell_2$$
$$X_2 \longrightarrow \{L_{21},L_{22},L_{23},L_{24},L_{25}\}=\ell_3$$

La regla $((SN,L_{12},L_{21},ALTO),y_1)$ subsume en la regla $((\ell_1,\ell_2,\ell_3,\{ALTO,MUY\ ALTO\}),y_1)$ ya que $SN\subseteq\ell_1$, $L_{12}\subseteq\ell_2$, $L_{21}\subseteq\ell_3$, $ALTO\subseteq\{ALTO,MUY\ ALTO\}$ e $y_1=y_1$.

Si la regla subsume en alguna regla perteneciente al conjunto de reglas iniciales entonces esta se ignora ya que la regla ya está generalizada y se toma una nueva regla del

conjunto de reglas iniciales. Sin embargo, si la regla no subsume en ninguna regla definitiva, el algoritmo trabaja con esa regla inicial aplicando un proceso de generalización o amplificación. Este proceso consiste en lo siguiente:

Comienza con la regla $((E_1,E_2,...,E_n),y_j)$ que queremos generalizar o convertir en una regla general o maximal y vamos realizando generalizaciones en cada variable, es decir ampliamos cada E_i (conjunto de etiquetas lingüísticcas) para i=1,..,n con las etiquetas que todavía no han sido consideradas para dicha variable.

Ejemplo 4.3. Si el conjunto de etiquetas es $\mathcal{L}$={Bajo,Medio,Alto} y nosotros tenemos la siguiente regla ((Bajo,Medio,Alto),y_3). Las ampliaciones se realizarán de la siguiente manera:

((Bajo,Medio,Alto),3) → (({Bajo,Medio},Medio,Alto),3) → (({Bajo,Medio,Alto},Medio,Alto),3) → (({Bajo,Medio,Alto},{Medio,Bajo},Alto),3) → (({Bajo,Medio,Alto},{Medio,Bajo,Alto},Alto),3) → (({Bajo,Medio,Alto},{Medio,Bajo,Alto},{Alto,Bajo}),3)→(({Bajo,Medio,Alto},{Medio,Bajo,Alto},{Alto,Bajo,Medio}),3).

Cuando se realiza una ampliación en una variable se comprueba si dicha ampliación es *posible*.

Recordamos que una ampliación de R_i a $R_{i'}$, con $R_{i'}$ siendo la regla IF X is $E_{i'}$ THEN y_p, es posible si no existe ninguna regla R_j, siendo R_j la regla IF X is E_j THEN y_q, perteneciente al conjunto de reglas iniciales que verifique que $E_j \subseteq E_{i'}$ y no subsume $(R_j,R_{i'})$. Es decir no existe ninguna regla R_j que verifique que $E_j \subseteq E_{i'}$ y $y_p \neq y_q$.

Ejemplo 4.4. Si el conjunto de etiquetas es $\mathcal{L}$={Bajo,Medio,Alto} y tenemos la regla ((Bajo,Bajo,Medio),y_2) en el conjunto de reglas iniciales. Una ampliación de (({Bajo,Medio,Alto},Medio,Medio),y_3) a (({Bajo,Medio,Alto},{Medio,Bajo},Medio),y_3) no es posible ya que Bajo$\subseteq${Bajo,Medio,Alto} Bajo$\subseteq${Medio,Bajo}, Medio$\subseteq$Medio y $y_3 \neq y_2$.

Cuando no se puede realizar una ampliación de la regla R_i a $R_{i'}$, continuamos trabajando con Ri y descartamos esa ampliación como una generalización no posible. Si la ampliación es posible continuamos generalizando la nueva regla obtenida $R_{i'}$.

Al final obtendremos un conjunto de reglas que representan de algún modo el conocimiento tácito del experto. Dentro de estas reglas tendremos también las reglas obvias o evidentes que el experto es capaz de utilizar en cualquier momento.

Para la obtención de un menor número de reglas nosotros proponemos dos mejoras, las cuales son el orden previo de las variables del conjunto de manera que las que más información proporcionen se coloquen en el último lugar en el ejemplo, y la eliminación de ejemplos ruidosos. Ya que lo que nosotros pretendemos no es identificar completamente el sistema (en este caso el experto) sino que nuestro objetivo será obtener un conjunto de reglas que contengan las reglas claras empleadas por el experto y aquellas que reflejan el conocimiento heurístico u oculto del experto.

5. UNA TÉCNICA DE VALIDACIÓN PARA LA AC

Para la obtención de la Base de Reglas "definitivas", es decir aquellas que simulan la manera de razonar del experto, se propone un método de validación que lo que "evoluciona" una base de reglas inicial.

El punto de partida como hemos mencionado anteriormente será la base de reglas evidentes, obtenidas al aplicar la técnica de la rejilla de repertorio, y las reglas claras que expresan el conocimiento tácito y evidente presente en el conjunto de ejemplos proporcionados, obtenidas por medio del algoritmo de aprendizaje automático. Puesto que el conjunto de ejemplos difícilmente será completo, es decir no se van a contemplar todas las situaciones posibles y puesto que en algunas ocasiones el experto trabaja en función de "ideas felices o inspiraciones" es decir que en situaciones similares el experto ofrezca soluciones no similares, se propone un método de validación (*Castro*[3]) como un método útil para detectar los casos en los que es necesario aportar nuevas soluciones y aquellos en los que hay contradicciones o diferentes soluciones.

Este refinamiento de la base de reglas está basado en la cooperación entre el experto y el sistema de AC. En función del error detectado se lanzará la batería de preguntas adecuada, y cuya misión será resolver el conflicto detectado.

El método de validación se lanzará cada vez que se aporte una solución y se deja de aplicar en el momento en el que no se encuentre ningún conflicto basándose en la información disponible.

No entramos en detalle sobre el método de validación propuesto, mecanismos de entrevista, ni en las posibles soluciones que puede aportar el experto en función del

conflicto detectado. Este método utilizará el aprendizaje basado en explicaciones cómo mecanismo para la resolución de aquellos casos para los que no se han aportado soluciones y sugerirá la incorporación de variables, redireccionamientos a zonas de conflictos donde se aplicarán metareglas, la corrección de reglas erróneas,... para la solución de conflictos.

6. CONCLUSIONES

Cuando el ingeniero se dispone a adquirir el conocimiento que hace a una persona experta en un determinado dominio, la mayoría de las veces se va a encontrar con expertos que no son capaces de expresarlo, o que no disponen de tiempo para atenderles. Esta última postura normalmente será apoyada por la organización a la cual pertenece, debido a que el tiempo de un experto está muy bien valorado y altamente cotizado. Sin embargo la organización estará muy interesada en la construcción de un SE que trabaje de una manera similar a como lo hace su experto, ya que la construcción de este SE permitirá reducir el trabajo que asfixia a su experto (recordemos que este es un bien escaso y caro) e incrementar la productividad de la organización. De esta manera, en las primeras fases del proceso de AC la organización tenderá a proporcionar registros o ejemplos de actuaciones pasadas de su experto en situaciones problemáticas del dominio en el cual se quiere construir el SE. Por otro lado, cuando el experto no es capaz de expresar el conocimiento que lo convierte en tal, debido a factores como pueden ser que este viene dado en función de habilidades, ideas felices, inspiración,... El experto y organización tienden a proporcionar ejemplos de situaciones posibles y cómo el experto las solucionaría.

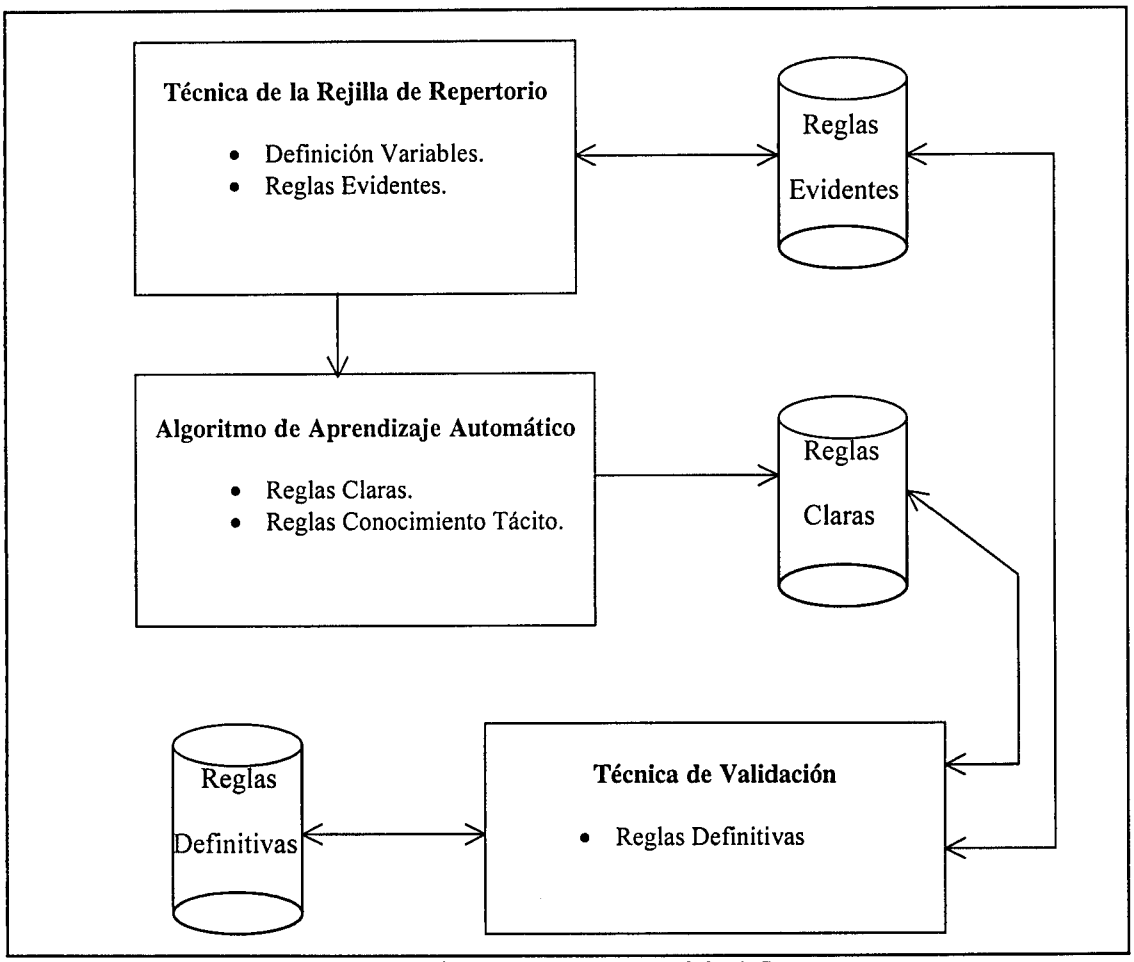


Figura 6. Esquema Metodología Propuesta.

Para solventar estos problemas proponemos una metodología para la AC automática que hace uso de:

- La *lógica difusa* como mecanismo de representación, por ser este un método natural de representar la manera de razonar del experto
- La *técnica de la rejilla de repertorio* para adquirir la información necesaria sobre las variables en las que basa el experto su razonamiento así como para adquirir las reglas evidentes del experto.
- Un algoritmo de aprendizaje automático para adquirir el conocimiento que el experto no es capaz de comunicar en su totalidad. El aprendizaje automático se muestra como una herramienta útil en el proceso de análisis de datos para sacar conclusiones claras a partir de un conjunto de ejemplos particulares.
- Una técnica de validación para extraer el conocimiento que permite resolver los casos conflictivos y aquellos casos que necesitan ser examinados.

El esquema de la metodología propuesta es el que se muestra en la Figura 6.

La utilidad de la metodología propuesta es evidente en casos de problemas simples, en los cuales se tiene una serie de entradas y en función de estas se toma una decisión (salida del sistema). En problemas más complejos la metodología propuesta sigue siendo útil. Siguiendo la idea que se propugna desde la investigación dentro del campo de la Ingeniería del Software, según la cual divide un problema en subproblemas más simples de manera que el problema de AC se ve ahora como la identificación de los modelos que tratan cada uno de estos subproblemas. Obviamente siempre se van a encontrar problemas del tipo del que se ha tratado.

Referencias

[1.] Boose, J. "A Knowledge Acquisition program for expert systems based on personal construct psychology", *Int. J. Man-Mach. Stu.*, 23, 495-525 (1985).

[2.] Boose, J. and J.M. Bradshaw, "Expertise transfer and complex problems: using AQUINAS as a knowledge acquisition workbench for knowledge-based systems". *Int. J. Man-Mach. Stu.*, 26, 3-28 (1987).

[3.] Castro, J., E. Trillas and J. Zurita. "Expert Systems Validation". *Proceedings of the sixth IFSA Congress*, Sao Paulo, 41-44 (1995).

[4.] Castro, J., J. Castro-Schez and J. Zurita, "Una Metodología para el desarrollo de SBC", *Congreso Español sobre Tecnologías y Lógica Fuzzy*, Tarragona, - (1997).

[5.] Castro, J., J. Castro-Schez and J. Zurita, "Un primer acercamiento a la adquisición automática de conocimiento", *Informática Automática*, 30, 4:100-116 (1997).

[6.] Castro, J., J. Castro-Schez and J. Zurita, "Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems", *Fuzzy Sets and Systems*, (Por aparecer 1998).

[7.] Esheman, L. D. Ehret, J. McDermott, and M. Tan, "MOLE: a tenacious knowledge acquisition tool", *Int. J. Man-Mach. Stud.*, 26, 41-54 (1987).

[8.] Hwang, G. "Knowledge Acquisition for Fuzzy Expert Systems", Int. J. Intelligent Systems, 10, 541-560 (1995).

[9.] Kahn, G., S. Nowlan and J. McDermott. "Strategies for Knowledge Acquisition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5,511-522 (1985).

[10.] Kelly, G. The Psychology of Personal Constructs. New York: Norton (1995).

[11.] _____, "Repertory Grid technique", in Concise Encyclopedia of Psychology, R. Cornisi, Ed. New York: Wiley-Interscience, 1987.

[12.] Ruqian, L. New approaches to Knowledge Acquisition. World Scientific, Singapore, (1994).

[13.] Tecuci, G. "Automating Knowledge Acquisition as Extending, Updating and Improving a Knowledge Base", *IEEE Transactions on Systems, Man, and Cybernetics*, 22, 6:1444-1460 (1992).

UN ESTUDIO SOBRE LA INCORPORACIÓN DE MODIFICADORES SEMÁTICOS EN UN ALGORITMO DE APRENDIZAJE DE REGLAS DIFUSAS*

Antonio GONZÁLEZ
Dpto. Ciencias de la Computación e I.A.
E.T.S.I. Informática
Universidad de Granada
18071-Granada (España)
A.Gonzalez@decsai.ugr.es

Raúl PÉREZ
Dpto. Ciencias de la Computación e I.A.
E.T.S.I. Informática
Universidad de Granada
18071-Granada (España)
fgr@decsai.ugr.es

Resumen

Un problema muy importante asociado al uso de algoritmos de aprendizaje de modelos difusos es el poder establecer una asignación correcta de las etiquetas difusas de las variables implicadas en el problema. En este trabajo partiendo de una asignación inicial, proponemos la inclusión dentro del algoritmo de aprendizaje de reglas difusas SLAVE, de modificadores semánticos que permitan alterar ligeramente la semántica de las etiquetas inicialmente propuestas. Esta alteración se hace con la idea de obtener descripciones más próximas a la información que refleja el conjunto de ejemplos. En concreto se propone la utilización de un tipo genérico de modificador semántico parámetrizable que altera la distribución de posibilidad de las etiquetas manteniendo fijo su soporte.

Palabras Clave: Aprendizaje Automático, Reglas Difusas, Algoritmos Genéticos, Modificadores Semánticos.

1 INTRODUCCIÓN

Uno de los principales problemas con los que nos podemos encontrar cuando tratamos de emplear un algoritmo de aprendizaje basado en modelos difusos consiste en establecer una discretizacion correcta de las variables que se mueven sobre dominios continuos. Algunos algoritmos de este tipo [4, 7, 11, 13, 14] requieren como entradas al mismo la discretización mediante conjuntos difusos de los dominios de las variables continuas. Para este tipo de algoritmos, es necesario usar la información dada por un experto sobre el problema que se desea resolver en la construcción de la discretización de los dominios.

Una inadecuada discretización de las variables continuas conlleva en general una limitación para el algoritmo de aprendizaje, ya que existe un fuerte dependencia entre la discretización que se realiza sobre los dominios de las variables involucradas y la capacidad de clasificación del algoritmo empleado. Por tanto, si se desea incrementar la capacidad de aprendizaje, es necesario que el algoritmo sea capaz de establecer una discretización apropiada o bien sea capaz de modificar la discretización que inicialmente se le propone.

Dentro de los algoritmos de aprendizaje que construyen o modifican la discretización de las etiquetas y que se pueden encontrar en la literatura, debemos distinguir fundamentalmente dos formas diferenciadas de actuación para obtener dichas discretizaciones:

- Aquellos que construyen las discretizaciones durante el proceso de aprendizaje. Dentro de estos algoritmos, podemos distinguir los algoritmos de aprendizaje basados en árboles de decisión, como C4.5 [15] o basados en reglas como CN2 [2]. Normalmente, es una discretización intervalar obtenida utilizando métodos estadísticos.
- Aquellos que parten de una discretización inicial del rango de las variables continuas a partir de la cual obtienen un conocimiento inicial, y una vez obtenido este conocimiento, en un proceso posterior y separado del algoritmo de aprendizaje, se modifica la discretización inicial para refinar este conocimiento. Estos algoritmos de ajuste se han usado ampliamente en el campo de las reglas difusas aplicadas al problema del control. Un ejemplo de estos sistemas es [3].

Otra posibilidad consiste en la definición de un modelo que parta de una discretización inicial de las variables, posiblemente propuesta por un experto, y durante el proceso de aprendizaje sea capaz de modificar

Este trabajo ha sido financiado parcialmente por la CICYT, dentro del Proyecto TIC95-0453

la semántica de dicha discretización inicial.

En este trabajo, utilizando esta idea, proponemos una metodología que permite la inclusión de la capacidad de alterar el significado de las etiquetas linguísticas dentro del propio proceso de aprendizaje del sistema SLAVE. Dicha capacidad implica el enriquecimiento del lenguaje de descripción de reglas del algoritmo. Para ello, modificamos el modelo de regla de forma que incluimos **modificadores semánticos** [17]. Los modificadores semánticos permiten pequeñas alteraciones de la semántica de las etiquetas definidas inicialmente, es decir, permiten la definición de nuevas etiquetas durante el proceso de aprendizaje, modificando ligeramente el significado de las etiquetas originales.

En la siguiente sección realizamos una descripción general del algoritmo de aprendizaje SLAVE. En la sección 3 proponemos la inclusión de los modificadores semáticos dentro del algoritmo genético de SLAVE, presentando en la secciones 4 y 5 estudios experimentales y conclusiones sobre este nuevo sistema.

2 UNA BREVE DESCRIPCIÓN DE SLAVE

SLAVE (Structural Learning Algorithm in Vague Environment) [7, 11] es un algoritmo de aprendizaje inductivo de reglas difusas. El esquema básico del algoritmo SLAVE es el siguiente:

1. Se usa un algoritmo genético para obtener UNA REGLA.
2. La regla se añade al conjunto final de reglas.
3. Se penaliza la regla para que no vuelva a ser seleccionada.
4. Si el conjunto de reglas es suficiente para representar los ejemplos del conjunto de entrenamiento, el sistema termina devolviendo dicho conjunto de reglas como solución del problema. En otro caso, se vuelve al paso 1.

La tarea principal de SLAVE consiste en encontrar la mejor regla que describe las relaciones existentes entre las variables y las clases (paso 1). Esta tarea puede formalizarse como un problema de búsqueda. De entre los distintos algoritmos de búsqueda que son aplicables, nosotros hemos usado algoritmos genéticos ya que estos proporcionan una poderosa técnica de búsqueda en espacios grandes y complejos. El problema de encontrar la mejor regla se plantea de la siguiente manera: fijada una clase, encontrar el mejor antecedente para esta clase. Una vez seleccionada la regla, una forma muy simple para penalizarla (paso 3) es eliminar los ejemplos del conjunto de entrenamiento que son recubiertos por la misma en un grado elevado.

Para la búsqueda del mejor antecedente para una clase concreta, SLAVE utiliza un algoritmo genético que trabaja a dos niveles [9, 10]. El primero de estos niveles busca las variables relevantes para la descripción de la clase, mientras el segundo nivel busca la mejor asignación de valores a estas variables relevantes.

El primero de los niveles, denominado **nivel de variable** codifica la relevancia o irrelevancia de una variable para una regla particular. En este nivel se usa una codificación real con n+1 componentes (siendo n el número de variables implicadas en el problema). El elemento i-ésimo contiene un valor entre 0 y 1 que representa el grado de relevancia de la variable i-ésima con respecto a la clase. El valor n+1 es un valor entre 0 y 1 que representa un umbral de activación, de manera que la variable i-ésima se considerará relevante para el antecedente de la regla si su valoración es mayor o igual que este umbral de activación. En otro caso la variable se considerará como irrelevante y será eliminada de la descripción de la regla. En la generación de la población inicial, se calcula el grado de relevancia de cada variable con respecto a la clase usando una medida de información [10]. El proceso evolutivo cambiara estos valores iniciales para obtener una mejor estimación de la relevancia de cada una de las variables. Los operadores genéticos definidos para alterar estos valores son el operador de cruce BLX_α y el operador de mutación no uniforme.

El segundo de los niveles denominado **nivel de valor** representa las asignaciones de valores a las variables. En este nivel se ha utilizado codificación binaria que expresa la presencia o ausencia en la asignación de una variable de cada uno de los valores de su dominio. Este nivel se hereda de la representación genética utilizada en las versiones anteriores de SLAVE y utiliza un conjunto más extenso de operadores genéticos que puede ser consultado en [8].

La seleccion de la mejor regla es el objetivo del algoritmo genético. La medida fundamental utilizada para valorar una regla se basa en las nociones de consistencia y completitud. En [11, 12] se proponen definiciones que suavizan y adaptan las definiciones clásicas de estos conceptos para trabajar con reglas difusas ($\Lambda(R)$) y $\Gamma_{k_1 k_2}(R)$ grado de completitud y grado consistencia débil respectivamente de la regla R. Además, en la evaluación de las reglas, tambien se tienen en cuenta dos funciones que miden la simplicidad y la comprensibilidad de las mismas ($svar(R)$ y $sval(R)$) respectivamente. Estas funciones se han propuesto en [1].

La función de evaluación del algoritmo genético combina las medidas anteriores utilizando una evaluación

funcional lexicográfica con los siguientes criterios:

Orden	Criterio
1	$\max\{\Lambda(R) \times \Gamma_{k_1 k_2}(R)\}$
2	$\max\{svar(R)\}$
3	$\max\{sval(R)\}$

3 INCLUSIÓN DE MODIFICADORES SEMÁTICOS EN SLAVE

Cuando un experto en un campo determinado realiza la discretización de una variable que se mueve sobre un referencial continuo, puede conocer de forma aproximada el significado que para él tiene una determinada etiqueta lingüística, pero difícilmente la conocerá de forma exacta.

Es fácil comprobar como diferentes personas, con un mismo nivel de conocimiento sobre el tema, pueden tener una percepción diferente para una misma etiqueta lingüística. Si bien las diferentes percepciones sobre el significado de una etiqueta lingüística no ocasiona en general problemas de comprensión en procesos de comunicación entre seres humanos, en procesos de extracción automática de conocimiento, la representación semántica del concepto lingüístico tiene un papel fundamental, ya que para ese concepto se establece una valoración numérica sobre su idoneidad. En este caso, ya no todas las representaciones posibles de un concepto lingüístico, aunque todas representen una noción similar del concepto, son igualmente apropiadas para formar parte del conocimiento que se desea extraer. En este sentido, los modelos difusos que no pueden modificar la semántica de las etiquetas iniciales, encuentran una fuerte restricción para mejorar su capacidad para representar el comportamiento de un determinado sistema.

Por esta razón, parece conveniente dotar a los algoritmos de aprendizaje basados en modelos difusos de la capacidad de alterar la semántica de las etiquetas que componen los dominios de las variables, con el fin de determinar el significado más correcto para la descripción del sistema.

Los encargados de modificar el significado de un determinado concepto son los modificadores semánticos. Los modificadores semánticos son elementos que aparecen habitualmente en los procesos de comunicación estructurada de alto nivel que permiten ampliar o restringir el significado de una etiqueta lingüística. Estos elementos juegan un papel importante en la comunicación humana, y en [17] se destaca su importancia y su utilidad en la representación del conocimiento en los procesos de razonamiento aproximado.

En modelos de representación del conocimiento basados en reglas difusas, los modificadores semánticos, en muchos casos, aparecen como funciones que permiten alterar la definición de las funciones de pertenencia de las etiquetas difusas. Ya que las etiquetas difusas establecen una distribución de posibilidad de pertenencia al mismo sobre el referencial y los modificadores alteran dicha distribución.

Por tanto, consideraremos un modificador semántico como una función β definida sobre el conjunto de etiquetas y con valores en el conjunto de valores difusos $\mathcal{R}$, es decir,

$$\begin{array}{rccc} \beta\colon & D & \longrightarrow & \mathcal{R} \\ & \mathrm{L} & \longrightarrow & \beta(L) \end{array}$$

en donde $\mu_{\beta(L)}(x) = \beta(\mu_L(x)) \in [0,1]$.

En este trabajo, proponemos un modelo de regla que incorpora en la definición de un termino lingüístico:

- un conjunto inicial de etiquetas lingüísticas T, también denominados *términos atómicos* que establecen una primera discretización del referencial de las variables, y
- un conjunto de modificadores semánticos $S = \{\beta_1, \beta_2, \ldots, \beta_m\}$ que permiten alterar la distribución de posibilidad asociada a los términos atómicos,

de manera que el modelo de regla asociado es

$$\text{IF } X \text{ IS } \phi(A) \text{ THEN } Y \text{ IS } B$$

donde $\phi(A)$ representa la composición de funciones

$$\beta_1 \circ \beta_2 \circ \ldots \circ \beta_t$$

que son aplicables a la etiqueta A de la variable X.

Por tanto, siguiendo esta idea es posible incluir un número arbitrario de modificadores semánticos en el modelo de aprendizaje. Sin embargo, en este trabajo vamos a explorar la inclusión de un único tipo de modificador semántico.

Las siguientes subsecciones describen el modificador semántico incluido y el nuevo modelo de regla, asi como las modificaciones que es necesario hacer sobre el algoritmo genético para que pueda trabajar con el nuevo modelo de regla.

3.1 MODIFICADOR DE CAMBIO DE FORMA

El tipo de operador que estamos interesados en incluir en el algoritmo de aprendizaje es aquel que modifica la distribución de posibilidad que refleja la función

de pertenencia, y que caracterizamos con la siguiente definición.

Definición 3.1 *Dadas dos etiquetas L_1 y L_2, diremos que L_2 es una etiqueta resultante de una alteración de cambio de forma de la etiqueta L_1, que notaremos por*

$$L_2 = ACF(L_1)$$

si el soporte de L_2 coincide con el soporte de L_1, es decir,

$$Sop(L_1) = Sop(L_2).$$

De entre el conjunto de familias de funciones que satisfacen la anterior definición, estamos interesados en aquellas funciones que alteren la distribución de la etiqueta manteniendo la ordenación relativa en la valoración de posibilidad de los elementos pertenecientes al soporte de la misma. Por ejemplo, si en la etiqueta original L, dados cualesquiera dos elementos del soporte de la misma a y b, se verifica que

$$\mu_L(a) \geq \mu_L(b)$$

entonces un operador β aplicado a dicha etiqueta debe satisfacer que

$$\beta(\mu_L(a)) \geq \beta(\mu_L(b))$$

Estos operadores son interesantes desde nuestro punto de vista, porque preservan la ordenación de los elementos del soporte de la etiqueta original con respecto a su posibilidad relativa de pertenencia a la misma, y en consecuencia, la etiqueta resultante de la aplicación del operador presenta tan solo una semántica ligeramente diferente a la propuesta inicialmente por el experto.

El operador $\beta(L) = L^a$ es un operador que representa a una familia de funciones que verifica la definición del operador ACF y además preserva la propiedad anterior. Un caso particular de este operador es aquel que transforma la etiqueta original mediante $\mu_L(x)^2$ que fué inicialmente propuesto en [17] para representar al modificador semántico **MUY**.

Como se muestra en las figuras 1, 2 y 3 este operador tiene dos interpretaciones diferentes en función del valor del parámetro a. Para valores de $a \in]1, \infty[$, se satisface que $\mu_L(x) \geq \mu_L(x)^a$ para todos los elementos que pertenecen al soporte de la etiqueta. En consecuencia, tras aplicar el operador, aquellos elementos con valores de posibilidad cercanos a uno en la etiqueta original mantienen o decrementan poco su posibilidad de pertenencia a la etiqueta. Por el contrario, los valores de posibilidad cercanos a cero, sufren un decremento más pronunciado en la etiqueta resultante. Por consiguiente, este operador puede entenderse como

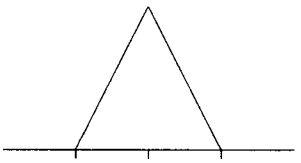


Figura 1: Etiqueta original

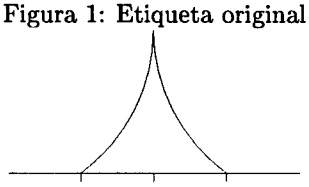


Figura 2: Transformación con $a = 2$

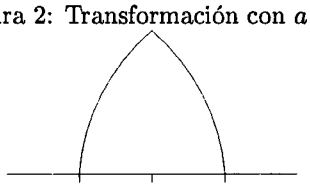


Figura 3: Transformación con $a = \frac{1}{2}$

una modificación en la imprecisión de la etiqueta, ya que los primeros (los que poseen valores cercanos a uno) tras la aplicación del operador, aumentan la posibilidad de pertenecer a la nueva etiqueta en relación a los que poseen valores cercanos a cero en la etiqueta original.

Sin embargo, para valores de $a \in]0, 1[$, se produce el efecto contrario, es decir, la etiqueta resultante aumenta su grado de imprecisión, ya que todos los valores del soporte de la etiqueta mantienen o aumentan su posibilidad de pertenencia a la etiqueta, y este cambio es más evidente sobre aquellos elementos que poseían valores de posibilidad cercanos a cero en la etiqueta original.

3.2 INCLUSIÓN DE MODIFICADORES SEMÁNTICOS EN EL ALGORITMO GENÉTICO

La introducción del operador semántico anterior implica una modificación en la representación del conocimiento extraído, y por consiguiente, una nueva propuesta de modelo de regla. Así, proponemos el siguiente modelo de regla:

IF X_1 *is* $A_1(a_1)$ *and* ... *and* X_p *is* $A_p(a_p)$ THEN Y *is* B

donde cada variable X_i tiene un conjunto referencial U_i y toma valores en un dominio finito D_i, para $i \in \{1, \ldots, p\}$. El conjunto referencial para Y es V y su dominio es F. El valor de la variable Y es B, donde $B \in F$ y el valor de la variable X_i es $A_i(a_i)$, donde $A_i \in P(D_i)$ y $P(D_i)$ denota el conjunto de las partes

D_i, y a_i es el parámetro del modificadores ACF. Así,

$$A_i(a_i) \equiv A_i^{a_i}.$$

Readaptar SLAVE implica una redefinición del módulo de extracción de reglas del sistema que se implementó mediante un algoritmo genético. En esta redefinición, necesitamos incorporar los nuevos elementos que se han de buscar, así, además de las variables relevantes para la descripción del concepto y de la asignación más apropiada de valores, también tendremos que incluir en la búsqueda los valores más apropiados para los parámetros del modificador semántico. Por tanto, es necesario incluir en la codificación de los cromosomas la información relativa a los operadores semánticos.

Como se describió anteriormente, SLAVE usa un algoritmo genético con dos niveles, donde en el nivel de variable se busca el conjunto de variables que son relevantes para describir el concepto, mientras que en el nivel de valor se busca la mejor asignación de valores para las variables relevantes. La mejor solución se busca mediante la evolución conjunta de los dos niveles.

Como los modificadores semánticos afectan directamente al nivel de valor, proponemos la modificación de este nivel a través de la descomposición del mismo en dos subniveles: **subnivel de activación** que corresponde al nivel original de valor que mantiene su codificación binaria y sus operadores genéticos y el **subnivel ACF** para representar los parámetros del modificador semántico.

Con esta representación en dos subniveles proponemos nuevamente un modelo basado en la evolución conjunta para encontrar la mejor solución. El tamaño de cada subnivel es el mismo y coincide con el número total de etiquetas implicadas en el problema, ya que el modificador se aplica para todas y cada una de las etiquetas que constituyen los dominios de las variables. Por tanto, los dos subniveles son de longitud fija.

El subnivel de activación representa la información relativa a la presencia o ausencia de cada valor en la descripción de la regla. En este sentido, mantiene el significado del nivel anterior de valor, y al igual que éste, conserva la codificación binaria, de manera que el valor uno indica presencia, mientras que el valor cero indica la no consideración de dicha etiqueta.

El subnivel ACF está formado por el trozo de cromosoma que hace referencia a los parámetros del modificador semántico ACF. El rango de valores que puede tomar este parámetro se mueve en el intervalo [0,∞). Un valor en [0,1) establece un crecimiento de la imprecisión con respecto a la etiqueta original, un valor en (1, ∞), una reducción de la imprecisión, mientras que el valor uno preserva la distribución de la etiqueta original.

En cuanto a la generación de la población inicial del proceso evolutivo, en el nivel de variable y en el subnivel de activación, este proceso se realiza de forma semejante al modelo anterior, ya que estos dos niveles se heredan directamente de su representación. Con respecto al subnivel ACF, tomamos el criterio de no alterar inicialmente la definición de las etiquetas de los dominios. Por esta razón, todos los parámetros a_{ij} del modificador semántico ACF se inicializan a 1.

Este criterio de inicialización de los parámetros de los modificadores semánticos es justificable, ya que deseamos que las etiquetas utilizadas para representar el conocimiento extraido sean lo más parecidas posibles a las que se proponen inicialmente.

Al igual que sucede con la generación de la población inicial, los operadores genéticos aplicados al nivel de variable y al subnivel de activación coinciden con los ya propuestos anteriormente. Sin embargo, es necesario establecer los operadores genéticos para el subnivel ACF. En este subnivel se definen únicamente los dos operadores usuales: el operador de mutación y el operador de cruce.

El criterio utilizado para la construcción del operador de mutación se basa en provocar pequeñas modificaciones que consisten en incrementos o decrementos de los parámetros de los operadores semánticos, de manera que el algoritmo genético determine el sentido más apropiado en el que los parámetros deben evolucionar. El uso estricto del criterio anterior provoca que los parámetros de los modificadores semánticos tiendan a tomar valores que modifican las etiquetas originales. Por esta razón, en el operador de mutación aplicado a estos subniveles aumentamos la probabilidad de que cada parámetro tome el valor que no modifica la etiqueta original.

El operador de cruce definido para este subnivel es el cruce por intercambio sobre dos puntos. A pesar de que este subnivel mantienen codificación real, no aplicamos cruces del tipo BLX_α como se hace en el nivel de variable, porque no deseamos que se produzcan fuertes modificaciones en los párametros de los operadores semánticos, ya que esto aumentaría el tiempo de convergencia del proceso evolutivo, aún a pesar de que posiblemente las descripciones que se obtengan sean mejores.

La característica fundamental del proceso de coevolución que se propone consiste en que no se establece una evaluación individualizada por niveles, sino que existe una única función de evaluación que determina la bondad para cada individuo dependiendo de la información global (es decir, de la agrupación de todos los niveles). Debido a esto, el proceso de

selección no se realiza de forma individualizada, sino que en función de la valoración de cada individuo se establece un orden lineal sobre todos los elementos de la población. De esta forma, la nueva población se construye seleccionando individuos completos de la última población. Por consiguiente, mientras la aplicación de los operadores genéticos es particular para cada nivel, las operaciones de selección y evaluación son aplicadas a individuos completos.

La función de evaluación usada en SLAVE se definió en base a una evaluación funcional lexicográfica (ELF) sobre varios criterios. Evaluación que sigue siendo válida para esta nueva redefinición del módulo de selección de reglas, sin más que incorporar en el cálculo, tanto del grado de satisfacción de la condición de consistencia débil, como del grado de completitud, las modificaciones provocadas por los modificadores en las etiquetas originales. Además, deseamos incorporar un nuevo criterio a la EFL de forma que premie en situaciones de empate de los criterios anteriores la menor modificación posible sobre las etiquetas iniciales.

Se puede establecer un criterio apropiado atendiendo a una medida de distancia, tomando como referencia el valor del operador semántico que no modifica la etiqueta original.

Definición 3.2 *Sean* $(d_1, \ldots d_n)$ *los parámetros del operador ACF que afectan a las etiquetas iniciales que aparecen para cada variable en la descripción de un individuo I. Definimos la diferencia entre las etiquetas resultantes con respecto a las inicialmente propuestas, que notaremos por* $Difer_{ACF}(I)$*, como la distancia entre los parámetros* $(d_1, \ldots d_n)$ *con respecto a* $(1, \ldots, 1)$*, es decir,*

$$Difer_{ACF}(I) = \sqrt[2]{\sum_{i=1}^{n}(d_i - 1)^2}.$$

Por consiguiente, la nueva función *Fitness* de evaluación de individuos queda definida como una EFL con cuatro criterios ordenados de la siguiente forma: Dada una regla R, definimos la valoración de la regla R como una evaluación funcional lexicográfica con los siguientes criterios:

Orden	Criterio
1	$\max\{\Lambda(R) \times \Gamma_{k_1 k_2}(R)\}$
2	$\max\{S(R)\}$
3	$\max\{CP(R)\}$
4	$\min\{Difer_{ACF}(R)\}$

4 ESTUDIOS EXPERIMENTALES

Una vez planteado el modelo de aprendizaje con la inclusión del modificador semántico y la nueva reformulación de la función de evaluación, deseamos comprobar experimentalmente el comportamiento de este algoritmo de aprendizaje.

En esta estudio experimental utilizaremos el conocido problema del péndulo. Así dado un conjunto de ejemplos que describe el comportamiento del sistema, deseamos aprender el conjunto de reglas que permiten controlarlo. Si asumimos que el valor de $|\theta| << 1$ (radianes) la ecuación diferencial no lineal que describe el comportamiento del péndulo es

$$m\frac{l^2}{3}\frac{d^2\theta}{dt^2} = \frac{l}{2}(-f + m\ g\ sin\theta - k\frac{d\theta}{dt})$$

donde $k\frac{d\theta}{dt}$ es una aproximación de la fuerza de fricción.

Por tanto, el sistema se puede describir a través de dos variables de estado θ (ángulo) y ω (velocidad angular) y la variable de control f (fuerza). Para la simulación real se ha considerado un péndulo de 5 kg de peso y 5 m de largo, al que se han aplicado fuerzas sobre el centro de gravedad cada 10 ms. Con estos parámetros, los referenciales de las variables son los siguientes:

$$\theta \in [-0.277, 0.277],$$

$$\omega \in [-0.458, 0.458],$$

$$f \in [-1593, 1593].$$

De forma experimental, hemos obtenido dos conjuntos de ejemplos bajo las anteriores condiciones, el primero de ellos tiene 213 ejemplos y se ha usado para aprender (conjunto de entrenamiento) y el segundo contiene 125 ejemplos y se ha utilizado para comprobar el comportamiento de las reglas aprendidas (conjunto de test). Estos conjuntos de ejemplos se han obtenido de las ecuaciones anteriores teniendo en cuenta dos condiciones iniciales diferentes:

a) θ=-0.277 y ω=0,

b) θ=0.277 y ω=0.

El primer paso para aprender consiste en discretizar el rango de las variables mediante etiquetas difusas. La figura 4 muestras las etiquetas construidas para cada variable, mientras que la figura 5 muestra la función que describe el comportamiento del péndulo.

Esta base de datos se ha aplicado a los siguientes tres algoritmos, SLAVE[1] que representa al algoritmo SLAVE propuesto en la sección 2 de este trabajo, SLAVE[2] que es el algoritmo SLAVE con la inclusión de modificadores semánticos y WM algoritmo de aprendizaje basado en modelos difusos propuesto en [16].

Los resultados obtenidos se muestran en la Tabla 1. Esta Tabla muestra tres parámetros: el error

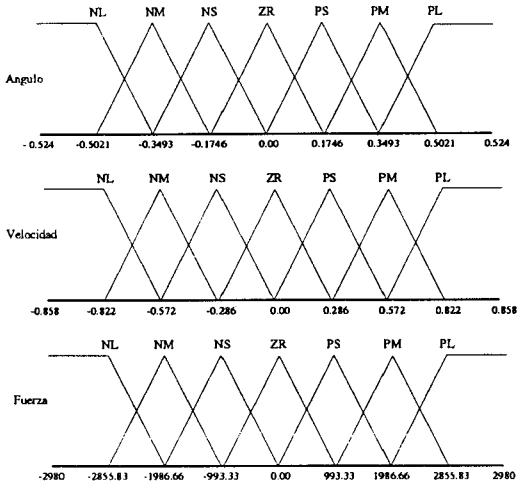


Figura 4: El problema del péndulo

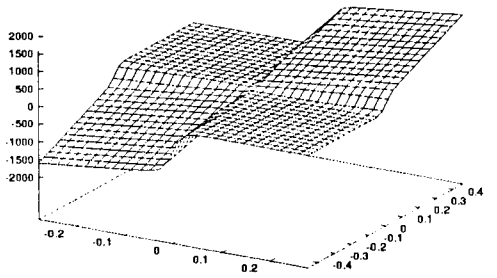


Figura 5: Superficie original del problema del péndulo

cuadrático medio sobre el conjunto de entrenamiento y el conjunto de prueba, así como el número de reglas de la base de datos obtenida por cada algoritmo de aprendizaje.

Los resultados muestran que la incorporación del modificador semántico provoca una mejora sustancial en la capacidad de aproximación de SLAVE[2] sobre el algoritmo original SLAVE[1]. Podemos observar que el número de reglas para ambos sistemas es el mismo. Esto significa que las reglas obtenidas contienen en sus descripciones las mismas etiquetas originales, pero la alteración de la semántica de dichas etiquetas provocadas por el modificador semántico permite a SLAVE[2] una mejor representación del comportamiento que refleja el conjunto de ejemplos.

SLAVE[2] también mejora sensiblemente el resultado obtenido por el algoritmo WM, necesitando un número muy inferior de reglas para representar el comportamiento del sistema.

Tabla 1: Resultados obtenidos por los distintos sistemas sobre la base de datos PENDULO

	Entrenamiento	Prueba	N.Reglas
SLAVE[1]	5.0690	6.3751	5
SLAVE[2]	0.1236	0.1524	5
WM	1.2044	7.6490	13

5 CONCLUSIONES

En este trabajo hemos presentado una modificación del algoritmo de aprendizaje SLAVE que incluye en su modelo de regla modificadores semánticos. Estos modificadores semánticos permiten alterar la semántica de las etiquetas modificando ligeramente las etiquetas propuestas inicialmente para la resolución del problema. Los resultados experimentales muestran que la incorporación de modificadores mejora los resultados obtenidos por el sistema.

Referencias

[1] L. Castillo, A. González R. Pérez, *Including a simplicity criterion in the selection of the best rule in a genetic fuzzy learning algorithm*, Sometido a Fuzzy Set and System.

[2] P. Clark , T. Niblett, *Learning If Then Rules in Noisy Domains.* TIRM 86-019, The Turing Institute, Glasgow (1986).

[3] Cordón O., Herrera F. *A Three-Stage Evolutionary Process for Learning Descriptive and Approximate Fuzzy-Logic-Controller Knowledge Bases From Examples.* International Journal of Approximate Reasoning, 17, pp. 369-407, (1997).

[4] Cordón O., Herrera F. *A Two-Stage Evolutionary Process for Designing TSK Fuzzy Rule-Based Systems.* Submitted to IEEE Transactions on systems, man and cybernetics.

[5] A. González, *A learning methodology in uncertain and imprecise environments*, International Journal of Intelligent Systems, Vol 19, (1995), pp. 357-371.

[6] A. González, F. Herrera, *Multi-stage Genetic Fuzzy Systems Based on the Iterative Rule Learning Approach*, Mathware and Soft Computing, Vol 4, n. 3, (1997), pp. 233-249.

[7] A. González, R. Pérez, J.L. Verdegay, *Learning the Structure of a Fuzzy Rule: an Genetic Approach.* Proceedings del First European Congress on Fuzzy and Intelligent Technologies, Vol 2, (1993), pp. 814-819, también publicado en Fuzzy Systems and Artificial Intelligence, Vol 3, 1, (1994), pp. 57-70.

[8] A. González, R. Pérez, *A Learning System of Fuzzy Control Rules.* In: F. Herrera, J.L. Verdegay (Eds.), Genetic Algorithms and Soft Computing, Physica-Verlag, (1996), pp. 202-225.

[9] A. González, R. Pérez, *A two level genetic fuzzy learning algorithm for solving complex problems.* Proceedings de IFSA'97, Vol 2, (1997), pp. 192-197.

[10] A. González, R. Pérez, *Using information measures for determining the relevance of the predictive variables in learnings problemas*, proceedings del FUZZ-IEEE'97, Vol 3, (1997), pp. 1423-1428.

[11] A. González, R. Pérez, *Completeness and Consistency conditions for learning fuzzy rules.* Fuzzy Sets and Systems 96 (1998) pp. 37-51.

[12] A. González R. Pérez, *SLAVE: A genetic learning system based on an iterative approach*, Sometido al IEEE Transaction on Fuzzy System.

[13] Magdalena L. *Adapting the Gain of an FLC with Genetic Algorithms.* International Journal of Approximate Reasoning, 17, pp. 327-349, 1997.

[14] Magdalena L., Monasterio F. *A Fuzzy Logic Controller with Learning Through the Evolution of its Knowledge Base.* International Journal of Approximate Reasoning, 16, pp. 335-358, 1997.

[15] J.R. Quinlan, *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, San Francisco (1993).

[16] L. Wang, M. Mendel, *Generating fuzzy rules by learning from examples.* IEEE Transactions on systems, man and cybernetics, vol.22 n.6, pp.1414-1427 (1992)

[17] L.A. Zadeh, *The concept of a linguistic variable and its applications to approximate reasoning.* Part I Information Sciences vol.8, pp. 199-249, Part II Information Sciences vol. 8, pp. 301-357, Part III Information Sciences vol. 9, pp. 43-80. (1975).

MÉTODOS DE RAZONAMIENTO APROXIMADO BASADOS EN EL CONCEPTO DE MAYORÍA DIFUSA PARA SISTEMAS DE CLASIFICACIÓN *

Oscar Cordón
Dpto. de Ciencias de la Computación e I.A.
E.T.S. de Ingeniería Informática
Universidad de Granada
18071 - Granada
e-mail: ocordon@decsai.ugr.es

Maria José del Jesus
Dpto. de Informática
Escuela Politécnica Superior
Universidad de Jaén
23071 - Jaén
e-mail: mjjesus@ujaen.es

Francisco Herrera
Dpto. de Ciencias de la Computación e I.A.
E.T.S. de Ingeniería Informática
Universidad de Granada
18071 - Granada
e-mail: herrera@decsai.ugr.es

Resumen

El método clásico de razonamiento en Sistemas de Clasificación Basados en Reglas Difusas clasifica un patrón con la clase indicada en el consecuente de la regla con la que tiene mayor grado de asociación. Algunos métodos de razonamiento alternativos consideran la información proporcionada por todas las reglas compatibles con el ejemplo. En este trabajo se proponen dos métodos de razonamiento aproximado que seleccionan la información a considerar en el proceso de inferencia, en base al concepto de mayoría difusa y al uso de operadores de agregación de la familia OWA.

Palabras Clave: Clasificación, Reglas difusas, Métodos de Razonamiento.

1 INTRODUCCIÓN

Los Sistemas Basados en Reglas Difusas se han aplicado con éxito en el diseño de Sistemas de Clasificación. Combinan precisión en la predicción con un alto nivel de interpretabilidad, lo que los hace muy adecuados para el diseño de sistemas de clasificación en problemas reales.

En un Sistema de Clasificación Basado en Reglas Difusas (SCBRD) se distinguen dos componentes principales: 1) La Base de Conocimiento (BC) formada por una Base de Reglas (BR) y una Base de Datos (BD) para un problema de clasificación específico, y 2) el Método de Razonamiento Difuso (MRD).

En la bibliografía especializada, se han propuesto distintos métodos de aprendizaje supervisado para la generación de BCs, algunos de los cuales se pueden encontrar en [1, 4, 8, 12, 13, 14, 17].

Respecto al MRD, el más utilizado clasifica un ejemplo con la clase determinada por la regla con mayor grado de asociación con el mismo ([1, 5, 13, 14, 16, 17]). Este MRD no considera la información proporcionada por otras reglas difusas con etiquetas lingüísticas distintas que representan el valor del ejemplo para la variable correspondiente, aunque en menor grado. En [3, 5, 6, 15] se mostró el poder de generalización de un SCBRD constituido por un MRD que utiliza la información de todas las reglas compatibles con el ejemplo, agregándola a través de las operaciones de suma normalizada y media. En [7] se propuso un modelo general de razonamiento que incluye como instancias particulares, el MRD clásico, y los basados en los operadores suma normalizada, media aritmética, media quasiaritmética, Badd, y Sowa or like entre otros.

La característica común a todos estos MRDs alternativos al clásico, es la participación en la clasificación de un nuevo ejemplo de todas las reglas compatibles con el mismo. En BRs con cardinalidad alta, esto puede llevar a errores en la clasificación por la fuerte influencia que ejercen muchas reglas con un grado de asociación pequeño con el ejemplo, frente a otras -en ocasiones menos numerosas- que representan mejor el conocimiento extraido sobre la zona del espacio en la que se incluye el ejemplo.

En este trabajo, desarrollamos nuevos MRDs que seleccionan la información a considerar en el proceso de inferencia. Esta selección es modelada a través de algunos operadores de la familia OWA, bajo el concepto de mayoría difusa. La inclusión de un MRD de este tipo en un SCBRD, aumenta la capacidad de predicción del sistema y añade descripción lingüística al proceso de inferencia y por tanto al SCBRD resultante.

La contribución se organiza como sigue: En la Sección 2, se describen brevemente los aspectos básicos de un SCBRD. En la Sección 3, se presentan las propuestas de MRDs basados en el concepto de mayoría difusa. Posteriormente, en la Sección 4, se analiza el poder de generalización aportado por los MRDs propuestos. En la Sección 5, se exponen las conclusiones.

*Trabajo realizado en el proyecto CICYT-TIC96-0778

2 SISTEMAS DE CLASIFICACIÓN BASADOS EN REGLAS DIFUSAS

Como se ha mencionado, un SCBRD está formado por dos componentes: La BC y el MRD. El diseño de un SCBRD implica la determinación de ambos, y si se realiza a través de un proceso de aprendizaje supervisado, parte de un conjunto de ejemplos correctamente clasificados (ejemplos de entrenamiento) y tiene como objetivo final el diseño de un Sistema de Clasificación que asigne etiquetas de clase a nuevos ejemplos con el mínimo error. Finalmente, se calcula el rendimiento del sistema sobre datos de prueba para obtener una estimación del error real del SCBRD. Este proceso se describe en la figura 1 y la estructura de sus componentes, se explica brevemente en las siguientes subsecciones.

2.1 Base de Conocimiento

La BC está formada por la BR y la BD. Para el diseño de la BR, en la literatura especializada se han utilizado distintos tipos de reglas: Reglas difusas con una clase en el consecuente ([1, 12]), con una clase y un grado de certeza asociado a la clasificación en esa clase en el consecuente ([14]) y aquellas que describen en el consecuente el grado de certeza asociado a la clasificación en cada una de las clases posibles ([17]). En [8] se desarrolló un método genético de aprendizaje utilizando BRs de cada uno de los tipos mencionados, y se mostró que las reglas difusas con una clase y el grado de certeza asociado en el consecuente, representan mejor el conocimiento sobre la zona del espacio correspondiente. En este trabajo, consideraremos SCBRDs formados por una BR de este tipo:

$$R_k: \quad \text{Si } x_1 \text{ es } A_1^k \text{ y } \ldots \text{ y } x_N \text{ es } A_N^k \text{ entonces } Y \text{ es } C_j \text{ con } r^k \tag{1}$$

donde:

- $x_1, \ldots, x_N$ son las características seleccionadas para el problema de clasificación,
- $A_1^k, \ldots, A_N^k$ son etiquetas lingüísticas usadas para discretizar el dominio contínuo de las variables,
- Y es la clase $C_j \in \{C_1, \ldots, C_M\}$ a la que pertenece el ejemplo, y
- r^k es el grado de certeza de la clasificación en la clase C_j para un ejemplo perteneciente al subespacio difuso delimitado por el antecedente de la regla R_k.

La BD contiene la definición de los conjuntos difusos asociados a los términos lingüísticos usados en la BR. Es común a toda la BR para mantener el carácter lingüístico del SCBRD.

2.2 Método de Razonamiento Difuso

Un MRD es un procedimiento de inferencia que deriva conclusiones entre un conjunto de reglas difusas y un ejemplo. En [7] presentamos un modelo general de razonamiento que, particularizado para una BR formada por reglas con una clase y su grado de certeza en el consecuente, se describe a continuación.

En la clasificación de un ejemplo $E^t = (e_1^t, \ldots, e_N^t)$ la BR $R = \{R_1, \ldots, R_L\}$ se divide en M subconjuntos de acuerdo a la clase indicada por su consecuente,

$$R = R_{C_1} \cup R_{C_2} \cup \ldots \cup R_{C_M} \tag{2}$$

y se sigue el siguiente esquema:

1. **Grado de compatibilidad.** Calcular el *grado de activación del antecedente de todas las reglas de la BR con el ejemplo*, usando una t-norma ([2, 9]).

$$\begin{array}{l} R^k(E^t) = T(\mu_{A_1^k}(e_1^t), \ldots, \mu_{A_N^k}(e_N^t)) \\ k = 1, \ldots, L \end{array} \tag{3}$$

2. **Grado de asociación.** Calcular el *grado de asociación del ejemplo E^t con las M clases de acuerdo a cada regla de la BR.*

$$\begin{array}{l} b_i^k = h(R^k(E^t), r^k),\ k = 1, \ldots, |R_{C_i}| \\ i = 1, \ldots, M \end{array} \tag{4}$$

3. **Función de ponderación.** *Ponderar* los valores obtenidos mediante una función g. Una posibilidad es incrementar los valores más altos y penalizar los menores.

$$\begin{array}{l} B_i^k = g(b_i^k), k = 1, \ldots, |R_{C_i}| \\ i = 1, \ldots, M \end{array} \tag{5}$$

4. **Grado de clasificación del ejemplo en cada una de las clases.** Para calcularlo se usa una función de agregación ([2, 9]) que combina -para cada clase- los grados de asociación positivos calculados en el paso anterior, y obtiene el grado de clasificación del ejemplo en esa clase.

$$\begin{array}{l} Y_i = f(B_i^k,\ k = 1, \ldots, |R_{C_i}| \text{ y } B_i^k > 0), \\ i = 1, \ldots, M \end{array} \tag{6}$$

siendo f un operador de agregación con comportamiento entre el máximo y el mínimo. Seleccionando para la función f el operador máximo, tenemos el MRD clásico.

5. **Clasificación.** Se aplica una función de decisión F al grado de clasificación del ejemplo en todas las clases. Esta función devolverá la etiqueta de clase que corresponde al máximo valor.

$$C_l = F(Y_1, \ldots, Y_M) \text{ tal que } Y_l = \max_{j=1,\ldots,M} Y_j \tag{7}$$

Algunas alternativas para los diferentes operadores implicados en el modelo general se describen en la tabla 7.

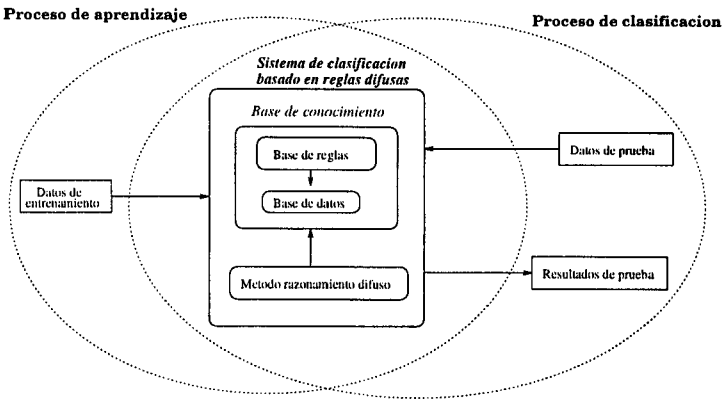


Figure 1: Diseño de un SCBRD (aprendizaje/clasificación).

3 MÉTODOS DE INFERENCIA BASADOS EN EL CONCEPTO DE MAYORÍA DIFUSA

En el modelo general de inferencia todas las funciones propuestas como operador de agregación (Tabla 7) consideran la información aportada por todas las reglas compatibles con el ejemplo. Este comportamiento puede provocar que, en bases de reglas grandes, muchas reglas con un grado pequeño de asociación con el ejemplo tengan más influencia que otras con mayor grado, y orienten el sistema hacia una clasificación errónea.

La incorporación, de un MRD que considere en cada nueva predicción sólo la información proporcionada por el subconjunto de reglas más adecuado para el ejemplo, puede mejorar el rendimiento del sistema. Esta selección puede realizarse con operadores de la familia OWA, bajo el concepto de mayoría difusa.

Los operadores OWA (Ordered Weighted Averaging) [11, 18, 19, 20], constituyen una clase de operadores de agregación que cubren completamente el intervalo entre los operadores máximo y mínimo. Su característica más importante es la utilización de pesos para los valores a agregar no asociados a valores particulares sino a su posición ordenada. Esta propiedad, nos permite asociar, en el proceso de inferencia, pesos a los grados de asociación del ejemplo con las distintas reglas de forma ordenada, y por tanto, potenciar las reglas más predictivas para ese ejemplo.

Un *operador OWA* es una función de agregación definida como

$$f_{\vec{\tau}}(a_1, \ldots, a_s) = \sum_{h=1}^{s} \omega^h \cdot b_h \qquad (8)$$

donde $(b_1, \ldots, b_s)$ es una permutación del vector $(a_1, \ldots, a_s)$ en la que los elementos están ordenados de forma decreciente, y $W = (\omega^1, \ldots, \omega^s)$ es un vector de pesos asociados que verifica que

$$\begin{array}{ll} 1) & \omega^z \in [0,1], \quad \forall z = 1, \ldots, s \\ 2) & \sum_{z=1}^{s} \omega^z = 1 \end{array} \qquad (9)$$

Se puede encontrar más información sobre las propiedades de esta familia de operadores en [11, 18, 19, 20].

Existen al menos dos formas de obtener los pesos ω^z. La primera implica la utilización de un mecanismo de aprendizaje que use un conjunto de ejemplos, sus argumentos y los valores agregados, e intente ajustar los pesos a esa colección de datos a través de algún tipo de modelo de regresión. Otra alternativa es dar significado a los pesos. Este trabajo se centra en este último enfoque, y en él se calculan pesos para la agregación realizada mediante el operador OWA usando *cuantificadores lingüísticos* que representen el concepto de *mayoría difusa*.

La mayoría difusa es un concepto relajado de mayoría representado habitualmente mediante un cuantificador difuso ([21]) que constituye un intento tanto de establecer un puente entre los sistemas formales y el discurso humano, como de proporcionar una herramienta de representación del conocimiento más flexible, y cuya semántica se puede representar con un conjunto difuso. Trabajaremos con cuantificadores proporcionales no decrecientes Q, definidos de forma que para cualquier $r \in [0,1]$, $Q(r)$ indica el grado con el cual la proporción r es compatible con el significado que representa el cuantificador. En el caso de los cuantificadores proporcionales no decrecientes, la función de pertenencia puede representarse así:

$$Q(r) = \begin{cases} 0, & \text{si } r < a \\ \frac{r-a}{b-a}, & \text{si } a \leq r \leq b \\ 1, & \text{si } r > b \end{cases} \qquad (10)$$

con $a, b, r \in [0,1]$,

En la figura 2, se muestran algunos ejemplos de cuantificadores proporcionales, donde los parámetros a y b son (0.3, 0.8), (0, 0.5), y (0.5, 1).

Una forma de calcular los pesos del OWA usando cuantificadores difusos es la siguiente ([19, 20]): Para un cuantificador proporcional no decreciente Q, el

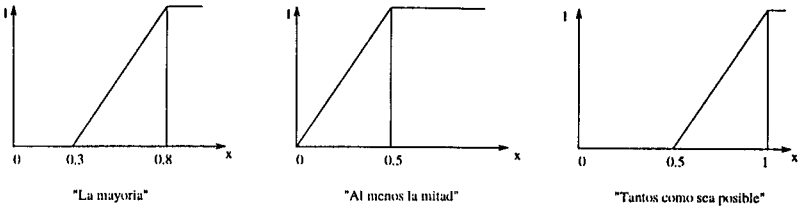


Figure 2: Algunos ejemplos de cuantificadores difusos proporcionales.

cálculo de los pesos vendrá dado por

$$\omega_i = Q\left(\frac{i}{s}\right) - Q\left(\frac{i-1}{s}\right), \quad i = 1, \ldots, s \qquad (11)$$

siendo s el número de valores a agregar.

El operador OWA junto con esta forma de cálculo de los pesos asociados, nos permite determinar la proporción de reglas más compatibles con el ejemplo que queremos que intervengan en el proceso de inferencia. Para ello, no tendremos más que especificar los parámetros a y b que definen el cuantificador difuso y calcular el vector de pesos para el proceso de agregación bajo el cual subyace ese cuantificador. De esta forma estamos definiendo un método de inferencia en el que no intervienen todas las reglas, sino "la mayoría", y este concepto de mayoría difusa está representado por un cuantificador difuso.

Basándonos en el concepto de agregación que proporciona la familia de operadores OWA, podemos definir otro método de inferencia basado en el operador *Quasiowa* definido en [11]:

$$f_8(a_1, \ldots, a_s) = h^{-1}\left[\sum_{k=1}^{s} \omega^k \cdot h(b_k)\right] \qquad (12)$$

siendo h una función contínua estrictamente monótona.

En nuestros experimentos hemos utilizado la función $h(x) = x^p$ con $p \in R$ que forma parte de la definición del operador media generalizada y proporciona un efecto compensativo en los valores a agregar [9, 10].

4 EXPERIMENTACIÓN

La experimentación se ha realizado sobre dos bases de ejemplos conocidas: IRIS y PIMA, de la siguiente forma:

- Se ha considerado una partición difusa constituida por tres etiquetas con funciones de pertenencia triangulares en el caso de PIMA, y cinco en el caso de IRIS.
- Para el cálculo de una estimación del error real del SCBRD, se han usado cinco particiones aleatorias de cada base de ejemplos en conjuntos de entrenamiento y prueba (70% y 30% respectivamente).
- Los resultados que aparecen en las tablas son medias del porcentaje de clasificación correcta en cada caso.
- Para mostrar la independencia de los resultados del método de generación de reglas empleado, se han utilizado dos técnicas de aprendizaje: la extensión del algoritmo de Wang y Mendel (W-M) a reglas de clasificación ([5]), y las dos primeras etapas del método genético de aprendizaje de SCBRD desarrollado en [8]. En las tablas, los SCBRDs resultantes, se notarán como SCBRD-T1 y SCBRD-T2, respectivamente.
- Los distintos MRDs están referenciados por la función de agregación en la que se basan, y el MRD clásico, se nota como f_0.

4.1 Iris

Los resultados obtenidos para esta base de ejemplos, con una BR generada con la extensión del algoritmo W-M, y los MRDs difusos basados en los operadores Owa y Quasiowa, se describen en la tabla 1. En la tabla 2, se muestran los porcentajes de clasificación correcta alcanzados con los MRDs que consideran todas las reglas.

I	a	b	p	Entr.	Pru.	Entr.	Pru.
				g_1		g_2	
f_0	0.0	0.0		97.31	94.32		
f_7	0.0	0.3		96.39	95.23	96.57	95.23
f_7	0.0	0.5		96.39	94.81	96.39	94.81
f_7	0.0	0.8		96.76	92.98	96.41	93.89
f_7	0.3	0.8		93.88	91.05	94.06	91.48
f_7	0.5	1.0		87.88	89.88	89.87	89.88
f_8	0.0	0.3	20	97.11	94.81	97.11	94.81
f_8	0.0	0.5	20	97.31	94.81	97.31	94.81
f_8	0.0	0.8	20	97.31	94.32	97.31	94.32
f_8	0.3	0.8	20	95.70	93.44	95.70	93.93
f_8	0.5	1.0	20	92.45	91.97	92.62	91.97
f_8	0.0	0.3	50	97.11	94.81	97.11	94.81
f_8	0.0	0.5	50	97.31	94.81	97.31	94.81
f_8	0.0	0.8	50	97.31	94.32	97.31	94.32
f_8	0.3	0.8	50	95.70	94.36	95.88	94.36
f_8	0.5	1.0	50	92.62	91.97	92.62	91.44

Tabla 1: IRIS. SCBRD-T1 (1)

Inferencia	Entr.	Pru.
f_3 $(p = 10) + g_1$	96.77	94.32
f_5 $(\alpha = 0.5) + g_1$	96.77	94.32
f_6 $(p = 5) + g_1$	97.12	94.32

Tabla 2: IRIS. SCBRD-T1 (2)

Se puede observar que los MRDs basados en el concepto de mayoría difusa, incrementan el poder de generalización del SCBRD.

Este incremento, se muestra también en un SCBRDs obtenido con el método genético de aprendizaje de SCBRD propuesto en [8] (Tabla 3). Además, la inclusión de MRDs alternativos al clásico en el proceso de aprendizaje inductivo permite la obtención de un SCBRD con mejor comportamiento.

Inferencia	Entr.	Pru.
f_0	98.57	94.72
f_7 $(a = 0\, b = 0.8) + g_1$	100	95.73
f_8 $(a = 0\, b = 0.8\, p = 20) + g_2$	99.65	96.28

Tabla 3: IRIS. SCBRD-T2

4.2 Pima

Los resultados obtenidos con los MRDs basados en los operadores Owa y Quasiowa, y con los MRDs que consideran todas las reglas, con una BR generada con el método de W-M, se muestran en las tablas 4 y 5 respectivamente.

I	a	b	p	Entr.	Pru.	Entr.	Pru.
				g_1		g_2	
f_0	0.0	0.0		85.11	73.23		
f_7	0.0	0.3		83.97	73.12	84.38	72.71
f_7	0.0	0.5		83.37	73.12	84.21	72.71
f_7	0.0	0.8		83.37	71.77	85.12	72.09
f_7	0.3	0.8		77.93	70.23	78.70	70.85
f_7	0.5	1.0		74.06	68.38	74.24	68.70
f_8	0.0	0.3	20	84.63	73.53	84.63	73.53
f_8	0.0	0.5	20	84.59	73.95	84.63	73.95
f_8	0.0	0.8	20	85.78	73.44	85.81	73.13
f_8	0.3	0.8	20	79.95	70.75	80.02	71.05
f_8	0.5	1.0	20	75.46	69.00	75.70	68.79
f_8	0.0	0.3	50	84.59	73.43	84.59	73.54
f_8	0.0	0.5	50	84.56	73.64	84.56	73.75
f_8	0.0	0.8	50	85.74	73.23	85.74	73.34
f_8	0.3	0.8	50	80.02	71.05	80.06	70.95
f_8	0.5	1.0	50	75.77	73.12	84.38	72.71

Tabla 4: PIMA. SCBRD-T1 (1)

Inferencia	Entr.	Pru.
f_3 $(p = 50) + g_2$	85.78	73.44
f_6 $(p = 5) + g_2$	85.89	73.53
f_6 $(p = 10) + g_1$	85.85	73.53

Tabla 5: PIMA. SCBRD-T1 (2)

Para esta Base de ejemplos, el método de razonamiento que presenta mejor comportamiento es el basado en el operador Quasiowa. Esto es debido a que este operador, además de la selección impuesta por el concepto de mayoría difusa, realiza por su definición, una compensación de los grados de asociación, comportamiento que tiene buenos resultados en bases de reglas de cardinalidad alta, como la generada para este ejemplo por el método W-M.

También para este ejemplo, con el segundo método de generación se observa que se incrementa el poder de generalización del SCBRD con la utilización de MRDs que seleccionan la información a agregar. Los resultados se muestran en la tabla 6.

Inferencia	Entr.	Pru.
f_0	78.67	73.66
f_7 $(a = 0\, b = 0.8) + g_1$	82.71	75.70
f_8 $(a = 0\, b = 0.8\, p = 20) + g_2$	81.56	74.26

Tabla 6: PIMA. SCBRD-T2

5 Conclusiones

En este trabajo hemos presentado una nueva propuesta de métodos de razonamiento aproximado para Sistemas de Clasificación Basados en Reglas Difusas. Los MRDs propuestos seleccionan la información a considerar en el proceso de inferencia a través de operadores de la familia OWA y bajo el concepto de mayoría difusa. Hemos estudiado los resultados obtenidos por estos MRDs con dos bases de ejemplos y dos métodos de aprendizaje inductivo para la generación de reglas, comparándolos con los obtenidos por otros MRDs.

Se ha mostrado que los MRDs propuestos, al seleccionar una mayoría difusa de los mejores grados de asociación entre las reglas y el ejemplo, incrementan el poder de generalización y de descripción del sistema. Además se ha apuntado la mejora obtenida por la integración de los MRDs un proceso de aprendizaje inductivo, para la obtención de una BR cooperativa con el MRD utilizado por el sistema.

References

[1] S. Abe, R. Thawonmas, "A fuzzy classifier with ellipsoidal regions," IEEE Transactions on Fuzzy Systems **5** 358-368 (1997).

[2] C. Alsina, E. Trillas, L. Valverde, "On some logical connectives for fuzzy set theory," J. Math. Anal. Appl. **93** 149-163 (1983).

[3] A. Bárdossy, L. Duckstein, "Fuzzy rule-based modeling with applications to geophysical, biological and engineering systems, CRC Press (1995).

[4] Z. Chi, J. Wu, H. Yan, "Handwritten numeral recognition using self-organizing maps and fuzzy rules," Pattern Recognition **28** 59-66 (1995).

[5] Z. Chi, H. Yan, T. Pham, "Fuzzy algorithms with applications to image processing and pattern recognition, World Scientific, (1996).

[6] O. Cordón, M.J. del Jesus, F. Herrera, E, López, "Selecting fuzzy rule based classification systems with specific reasoning methods using genetic algorithms," Proc. Seventh International Fuzzy Systems Association World Congress, Prague 424-429. 1997.

[7] O. Cordón, M.J. del Jesus, F. Herrera, "A proposal on reasoning methods in fuzzy rule-based classification systems," Technical Report DECSAI-970126, Dept. Computer Science and Artificial Intelligence, University of Granada, Granada (Spain, November 1997).

[8] 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 (1998) Por aparecer.

[9] D. Dubois, H. Prade, "A review of fuzzy set aggregation connectives," Information Sciences **36** (1985) 85-121.

[10] H. Dyckhoff, W. Pedrycz "Generalized means as models compensative conectives," Fuzzy Sets and Systems **14** 143-154 (1984).

[11] J. Fodor, J-L. Marichal, M. Roubens, "Characterization of the Ordered Weighted Averaging Operators," IEEE Transactions on Fuzzy Systems **3:2** 236-239 (1995).

[12] L. Fu, "Rule generation from neural networks," IEEE Transactions on Systems, Man and Cybernetics **24** 1114-1124 (1994).

[13] A. González, R. Pérez, "Completeness and consistency conditions for learning fuzzy rules," Fuzzy Sets and Systems (1998) Por aparecer.

[14] H. Ishibuchi, K. Nozaki, H. Tanaka, "Distributed representation of fuzzy rules and its application to pattern classification," Fuzzy Sets and Systems **52** 21-32 (1992).

[15] H. Ishibuchi, T. Morisawa, T. Nakashima, "Voting schemes for fuzzy-rule-based classification systems," Proc. Fifth IEEE Int. Conference on Fuzzy Systems 614-620. 1996.

[16] L. I. Kuncheva, "On the equivalence between fuzzy and statistical classifiers," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **15** 245-253 (1996).

[17] D.P. Mandal, C.A. Murthy, S.K. Pal, "Formulation of a multivalued recognition system," IEEE Trans. on Systems, Man, and Cybernetics **22** 607-620 (1992).

[18] A.L. Ralescu, D.A. Ralescu, "Extension of fuzzy aggregation," Fuzzy Sets and Systems **77** 125-150 (1997).

[19] R.R. Yager, "On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking," IEEE Trans. on Systems, Man and Cibernetics **18** 183-190 (1988).

[20] R.R. Yager, "Families of OWA operators," Fuzzy Sets and Systems **59** 125-148 (1993).

[21] L.A. Zadeh, "A computational approach to fuzzy quantifiers in natural languajes," Comps. and Maths. with Appls. **9** 149-184 (1983).

1. *Grado de compatibilidad*
$R^k(E^t) = \min_{i=1,\ldots,N} \mu^k_{A_i}(e^t_i)$

2. *Grado de Asociación*
$h(R^k(E^t), r^k_j) = R^k(E^t) \cdot r^k_j$

3. *Función de ponderación. Propuestas:*
$g_1(x) = x, \quad \forall x \in [0,1]$

$$g_2(x) = \begin{cases} x^2, & \text{si } x < 0.5 \\ \sqrt{x}, & \text{si } x \geq 0.5 \end{cases}$$

4. *Función de agregación. Propuestas:*

Clásico

$f_0(a_1, \ldots, a_s) = \max_{i=1,\ldots,s} a_i$

Suma Normalizada

$$f_1(a_1, \ldots, a_s) = \frac{\sum_{i=1}^{s} a_i}{f_{1_{max}}}$$

$$f_{1_{max}} = \max_{j=1,\ldots,M} \sum_{i=1}^{s} a_i$$

Media Aritmética

$$f_2(a_1, \ldots, a_s) = \frac{\sum_{i=1}^{s} a_i}{s}$$

Media Quasiaritmética

$$f_3(a_1, \ldots, a_s) = H^{-1}\left[\frac{1}{s}\sum_{i=1}^{s} H(a_i)\right]$$

$H(x) = x^p, \quad p \in \mathrm{R}$

Sowa and-like

$$f_4(a_1, \ldots, a_s) = \alpha \cdot a_{min} + (1-\alpha)\frac{1}{s}\sum_{i=1}^{s} a_i$$

$\alpha \in [0,1], \quad a_{min} = min\{a_1, \ldots, a_s\}$

Sowa or-like

$$f_5(a_1, \ldots, a_s) = \alpha \cdot a_{max} + (1-\alpha)\frac{1}{s}\sum_{i=1}^{s} a_i$$

$\alpha \in [0,1], \quad a_{max} = max\{a_1, \ldots, a_s\}$

Badd

$$f_6(a_1, \ldots, a_s) = \frac{\sum_{i=1}^{s} a_i^{\alpha+1}}{\sum_{i=1}^{s} a_i^{\alpha}}, \quad \alpha \in \mathrm{R}$$

siendo $a_1, \ldots, a_s$ los valores a agregar para un ejemplo E^t respecto a la clase C_j

Tabla 7: *Propuestas para los operadores del Modelo General de Razonamiento*

CASE-BASED DECISION: A CHARACTERIZATION OF PREFERENCES IN A QUALITATIVE SETTING

Lluís Godo

IIIA-CSIC,
Campus UAB s/n
08193 Bellaterra, Spain.
godo@iiia.csic.es

Adriana Zapico

IIIA-CSIC,
Campus UAB s/n
08193 Bellaterra, Spain.
zapico@iiia.csic.es

Abstract

Gilboa and Schmeidler have proposed a so-called case-based decision theory, where decisions are chosen according to their performance in previously experienced decision problems. Recently, it has been suggested to adapt Dubois and Prade's possibilistic decision model to the framework of case-based decision problems. However, some technical problems that may appear have already been pointed out. In this paper we further focus on this issue. In particular, when solving some of those problems we are led to extend the original Dubois and Prade's decision model to cope with belief states represented by non-normalized possibility distributions, accounting for belief states that can be inconsistent to some extent. We provide axiomatic characterizations of preference relations among decisions induced by qualitative utility functions which are formally similar to the original ones up to modifying factors coping with the possible lack of full consistency in the belief representations.

Keywords: Case-based Decision, Qualitative Utilities.

1 INTRODUCTION

Gilboa and Schmeidler [8] claim that Decision Making under uncertainty is, at least, partly case-based. They suggest that people choose acts based on their performance in the past and they propose the so-called Case-based Decision Theory (CBDT), as an alternative to the classical Expected Utility Theory (EUT). This theory assumes a set **M** of decision problem instances storing the performance of decisions taken in different past situations as triples (*situation, decision, consequence*), and *some measure* S *of similarity* between situations. The Decision Maker, in face of a new situation s_0, is proposed to choose a decision d which maximises a counterpart of the expected utility, namely the expression

$$U_{s_0,M}(d) = \Sigma_{(s,d,x)\in M}\, S(s_0,s)\cdot u(x)$$

where S is a non-negative function which estimates the similarity of situations and u provides a numerical utility for each consequence x. Gilboa and Schmeidler axiomatically characterise the preference relation induced by this U-maximisation.

Dubois and Prade [7] propose another approach to Case-Based Decision. Instead of averaging the utility of consequences obtained in similar situations, weighted by a similarity value, they propose to look for decisions that always gave good results in experienced similar situations. With this objective they estimate the utility of a decision d in terms of the degree of inclusion of the fuzzy set of situations which are similar to the new situation and where the decision d was experienced into the fuzzy set of situations where decision d led to good results.

Dubois and Prade [6] have also suggested a qualitative counterpart of Expected Utility Theory, in the spirit of the Von Neumann and Morgenstern's axiomatic foundations for EUT [9]. They assume that uncertainty is of possibilistic nature, i.e. belief states are represented by normalised possibility distributions π, and make only use of commensurate ordinal preference and uncertainty scales, equipped with the maximum, minimum and an order reversing operations. This (Possibilistic) Qualitative Decision Theory (QDT) appears as the natural decision theory related to Possibility Theory. In a recent paper [5] it is proposed an improved axiomatic setting for this Possibilistic Qualitative Decision Theory [6]. Two

qualitative criteria are axiomatized in a finite setting that generalise the well-known maximin and maximax criteria.

In [5] a link is established between Dubois and Prade's Case-based and Qualitative Decision models by estimating how much plausible x is a consequence of a decision d in the current situation s_0 in terms of what extent s_0 is similar to situations in which x was experienced after taking the decision d. So again, a decision or action d can be identified with a possibility distribution on consequences. This link between similarity on situations and possibility distributions on consequences allow us to apply the previously mentioned qualitative criteria to case-based decision problems.

However unsatisfactory results may appear if the distributions associated with the decisions are non normalised. Indeed, when the distributions are normalised the optimistic criterion scores a decision higher than the pessimistic one, as it is expected, but when the distributions are non normalised the pessimistic utility may result higher than the optimistic utility. In possibility theory, non-normalised distributions account for partially inconsistent representations of belief states. In order to solve this problem and to include these kind of distributions in the model, in this paper we extend the model and provide their corresponding characterisations.

Before analysing the problematic of non normalised distributions, in Section 2 we summarise already known results about the possibilistic framework for decision making. In Section 3 we propose an extension of the framework and we provide an axiomatic setting characterising generalised utility functions for non normalised distributions. Finally, in Section 4 we relate them to case-based decision and we end up with some conclusions.

2 BACKGROUND

2.1 POSSIBILISTIC QUALITATIVE DECISION THEORY

In Dubois and Prade's Qualitative Decision Theory, the working hypothesis is that uncertainty is modelled by possibility distributions. Also, it is assumed the existence of a utility function $u: X \to U$, being U a linearly ordered finite scale modelling Decision Maker's preferences on consequences, i.e. the greater is u(x), the more preferred is x. In a similar way, an increasing in $\pi(x)$ means the plausibility of x increases.

In such a framework we may identify each decision with a normalised possibility distribution on X. Therefore, choosing the "best" decision is equivalent to choose its associated possibility distribution maximising a utility function $\mathcal{U}$ on Pi(X), being Pi(X) the set of normalised possibility distributions on X, that is,

$$d \sqsubseteq d' \quad \text{iff} \quad \pi_d \sqsubseteq \pi_{d'} \quad \text{iff} \quad \mathcal{U}(\pi_d) \leq \mathcal{U}(\pi_{d'}).$$

In what follows, **S** will denote a finite set of situations and **X** will denote a finite set of consequences of acts. A decision or act **d**, is represented by a function $\mathbf{d}: S \to X$ which provides the consequence of the decision in each possible situation. **V** will denote a finite linear scale of uncertainty, with inf(V) = 0, sup(V) = 1, and **Pi(X)** the set of normalised possibility distributions also called possibilistic lotteries on X over V, i.e. $\mathbf{Pi(X)} = \{\pi: X \to V \mid \exists\, x \in X : \pi(x) = 1\}$. For the sake of simplicity, we shall use x for denoting both an element belonging to X and the possibility distribution on X such that $\pi(x) = 1$ and $\pi(z) = 0$ for $z \neq x$. In general, we shall also denote by A both a subset $A \subseteq X$ and the possibility distribution on X such that $\pi(x) = 1$ if $x \in A$ and $\pi(x) = 0$ otherwise. With this convention, we can consider X as included in Pi(X). The so-called **Possibilistic mixture** is an operation defined on Pi(X) that combines two possibility distributions π_1 and π_2 into a new one, denoted $(\lambda/\pi_1, \mu/\pi_2)$, with $\lambda, \mu \in V$ and $\max(\lambda, \mu) = 1$, and defined as

$$(\lambda/\pi_1, \mu/\pi_2)(x) = \max(\min(\lambda, \pi_1(x)), \min(\mu, \pi_2(x))).$$

In particular the possibilistic lottery $(\lambda/x, \mu/y)$ is defined as the possibility distribution on X such that

$$(\lambda/x, \mu/y)(z) = \begin{cases} \lambda & \text{if } z = x \\ \mu & \text{if } z = y \\ 0 & \text{otherwise} \end{cases}$$

Finally, **U** will denote a finite linearly ordered scale of preference, with sup(U) = 1 and inf(U) = 0, $\mathbf{n_U}: U \to U$ will denote the order reversing involution in U.

As a consequence of the Possibilistic mixture definition we have the following property (reduction of lotteries):

$$(\lambda/x, \mu/(\alpha/x, \beta/y)) = (\max(\lambda, \min(\mu, \alpha))/x, \min(\mu, \beta)/y).$$

In this framework a conservative qualitative criterion is to look for distributions such that all its plausible consequences are preferred, while a more optimistic looks for distributions such that at least one of its plausible consequences is preferred. In order to define these criteria, Dubois and Prade [6] assume the existence of an order preserving mapping h between the plausibility scale V and the preference scale U such that h(1) = 1 and h(0) = 0. Then the following pessimistic and optimistic qualitative utilities are defined respectively:

$$QU^-(\pi) = \min_{x \in X} \max(n(\pi(x)), u(x))$$
$$QU^+(\pi) = \max_{x \in X} \min(h(\pi(x)), u(x))$$

where $n: V \to U$ is the order reversing function $n = n_U \circ h$. The optimistic criterion has been first proposed by Yager [15] and the pessimistic criterion by Whalen [14]. One can easily notice that $QU^-(\pi)$ and $QU^+(\pi)$ are nothing but the *necessity* and *possibility* measures of the fuzzy set u, with U-valued membership function, w.r.t. the distribution π^* (with $\pi^*(x) = h(\pi(x))$), namely $QU^-(\pi) = Nec_{\pi^*}u$ and $QU^+(\pi) = Pos_{\pi^*}u$.

Some Remarks:

- QU^- and QU^+ coincide with the well know maximin and maximax criteria when π is an all or nothing distribution.
- QU^- can be seen as a degree of inclusion of the fuzzy set π of plausible consequences into the fuzzy set u of preferred consequences, while QU^+ can be seen as a degree of intersection of both sets. Indeed, under usual conditions, $QU^-(\pi) = 1$ iff $u(x) = 1$ for all x that are somewhat plausible ($\pi(x) > 0$), while $QU^+(\pi) = 1$ as soon as there exists a fully plausible x ($\pi(x) = 1$) such is that $u(x) = 1$.
- QU^- and QU^+ preserve the possibilistic mixture, in the sense that the following expressions hold:

$$QU^-(\lambda/\pi_1, \mu/\pi_2) = \\ = \min\{\max(n(\lambda), QU^-(\pi_1)), \max(n(\mu), QU^-(\pi_2))\}.$$

$$QU^+(\lambda/\pi_1, \mu/\pi_2) = \\ = \max\{\min(h(\lambda), QU^+(\pi_1)), \min(h(\mu), QU^+(\pi_2))\}.$$

A first set of axioms was proposed by Dubois and Prade in [6] for a preference relation $\sqsubseteq$ on Pi(X) to be represented by the (conservative) qualitative utility defined above. In [5] an improvement of those axioms is proposed, together with a new proof of the representation theorem. In the following $\pi \sim \pi'$ means $\pi' \sqsubseteq \pi$ and $\pi \sqsubseteq \pi'$.

- **A1**: $\sqsubseteq$ is a total pre-order (i.e. $\sqsubseteq$ is reflexive, transitive and complete).
- **A2** (uncertainty aversion): if $\pi \leq \pi' \Rightarrow \pi' \sqsubseteq \pi$.
- **A3** (independence):
  $\pi 1 \sim \pi 2 \Rightarrow (\lambda/\pi 1, \mu/\pi) \sim (\lambda/\pi 2, \mu/\pi)$.
- **A4** (continuity):
  For all $\pi \in$ Pi(X) there exists $\lambda \in V$ such that $\pi \sim (1/\overline{x}, \lambda/\underline{x})$, where $\overline{x}$ and $\underline{x}$ are a maximal and a minimal elements on $(X, \sqsubseteq)$ respectively.

It has been shown that the set of axioms AX = {A1, A2, A3, A4} faithfully represent the preference orderings induced by the above pessimistic qualitative utilities.

Representation Theorem 1. *(Pessimistic Utility)* A preference relation $\sqsubseteq$ on Pi(X) satisfies axioms A1, A2, A3 and A4 if, and only if, there exist

(i) a linearly ordered preference scale U with inf(U) = 0 and sup(U) = 1,
(ii) a preference function $u: X \to U$ such that $u^{-1}(1) \neq \emptyset \neq u^{-1}(0)$, and
(iii) an onto order reversing function $n: V \to U$ such that $n(0) = 1$ and $n(1) = 0$,

in such a way that it holds:

$$\pi' \sqsubseteq \pi \text{ iff } \pi' \preccurlyeq_u \pi,$$

where $\preccurlyeq_u$ is the ordering on Pi(X) induced by the qualitative utility QU^-, i.e. $\pi \preccurlyeq_u \pi'$ iff $QU^-(\pi) \leq QU^-(\pi')$, with $QU^-(\pi) = \min_{x \in X} \max(n(\pi(x)), u(x))$.

With respect to the optimistic preference criterion, the uncertainty aversion axiom A2 is changed by a uncertainty-prone postulate

- **A2$^+$**: if $\pi \leq \pi'$ then $\pi \sqsubseteq \pi'$

and axiom A4 is adequately modified into

- **A4$^+$**: for all $\pi \in$ Pi(X), there exists $\lambda \in V$ such that $\pi \sim (\lambda/\overline{x}, 1/\underline{x})$, where $\overline{x}$ and $\underline{x}$ are a maximal and a minimal elements on $(X, \sqsubseteq)$.

It is also shown in [5] that $AX^+ = \{A1, A2^+, A3, A4^+\}$ faithfully represent the preference orderings induced by the above optimistic qualitative utilities, providing its respective representation theorem.

2.2 A GENERALISATION FOR MAX-T POSSIBILISTIC MIXTURES

The possibilistic mixture operation considered in the previous section to combine possibilistic lotteries was a max-min combination: $(\alpha/\pi_1, \beta/\pi_2) = \max(\min(\alpha, \pi_1), \min(\beta, \pi_2))$. Possibilistic mixtures, definable as consensus functions on $\perp$*–decomposable measures*, $\perp$ being a t-conorm operation[1], have been studied in [9]. It is shown there that for possibility measures, i.e. max-decomposable measures, an admissible class of mixture operations is obtained by defining

$$M_T(\pi, \pi'; \alpha, \beta) = \max(\alpha T \pi, \beta T \pi')$$

where T is a t-norm[2] on V and $\max(\alpha, \beta) = 1$. The following generalised qualitative utilities are proposed in [5], as extensions of the qualitative utilities QU^- and

[1] A measure $g: 2^X \to V$ is $\perp$-decomposable if $g(A \cup B) = g(A) \perp g(B)$ when $A \cap B = \emptyset$, where $\perp$ is a non-decreasing, associative and commutative binary operation on V, with $v \perp 0 = v$ and $v \perp 1 = 1$ for all $v \in V$.

[2] A t-norm operation T on V is a non-decreasing, associative and commutative binary operation on V, verifying $vT0 = 0$ and $vT1 = v$ for all $v \in V$.

QU^+ when $M_\top$ is considered instead of the usual possibilistic mixture,

$$GQU^-(\pi) = \min_{x_i \in X} n(\pi(x_i) \top \lambda_i)$$
$$GQU^+(\pi) = \max_{x_i \in X} h(\pi(x_i) \top \delta_i)$$

where $n(\lambda_i) = u(x_i)$ and $h(\delta_i) = u(x_i)$. As usual, in these expressions n and h are an order reversing mapping and an order preserving mapping from V to U, but further verifying the following *coherence* conditions w.r.t. $\top$

$$n(\lambda) = n(\mu) \Rightarrow n(\alpha \top \lambda) = n(\alpha \top \mu)$$
$$h(\lambda) = h(\mu) \Rightarrow h(\alpha \top \lambda) = h(\alpha \top \mu)$$

for all $\alpha, \lambda, \mu \in V$. These condition guarantees the correctness of the above definitions. Now, the reduction of lotteries follows the next rule:

$M_\top(M_\top(\pi_1, \pi_2; \lambda_1, \lambda_2), M_\top(\pi_1, \pi_2; \mu_1, \mu_2), \alpha, \beta) = M_\top(\pi_1, \pi_2; \max(\alpha \top \lambda_1, \beta \top \mu_1), \max(\alpha \top \lambda_2, \beta \top \mu_2))$.

In order to characterise GQU^- it is proposed in [5] the set of axioms $AX_\top = \{A1, A2, A3_\top, A4_\top\}$, where

- **$A3_\top$** (independence):
  $\pi_1 \sim \pi_2 \Rightarrow M_\top(\pi_1, \pi; \alpha, \beta) \sim M_\top(\pi_2, \pi; \alpha, \beta)$
- **$A4_\top$:** For all $\pi \in Pi(X)$ there exists $\lambda \in V$ such that $\pi \sim M_\top(\overline{x}, \underline{x}; 1, \lambda)$, where $\overline{x}$ and $\underline{x}$ are a maximal and a minimal element of X with respect to $\sqsubseteq$ respectively.

together with the following representation theorem.

Representation Theorem 2. A preference relation $\sqsubseteq$ on Pi(X), equipped with the mixture operation $M_\top$, satisfies the axiom set $AX_\top$ if and only if there exist

(i) a 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 reversing function n: $V \to U$ such that $n(0) = 1$, $n(1) = 0$, satisfying coherence w.r.t. $\top$, i.e. $n(\lambda) = n(\mu) \Rightarrow n(\alpha \top \lambda) = n(\alpha \top \mu)$, for all $\alpha, \lambda, \mu \in V$,

in such a way that it holds:

$$\pi' \sqsubseteq \pi \text{ iff } \pi' \preccurlyeq_u \pi,$$

where $\preccurlyeq_u$ is the ordering on Pi(X) induced by the qualitative utility GQU^-.

For the modelization of the optimistic behaviour, in [5] they consider the axiom set $AX^+_\top = \{A1, A2^+, A3_\top, A4^+_\top\}$, where

- **$A4^+_\top$:** For all $\pi \in Pi(X)$ there exists $\lambda \in V$ such that $\pi \sim M_\top(\overline{x}, \underline{x}, \lambda, 1)$ where $\overline{x}$ and $\underline{x}$ are a maximal and a minimal element on X with respect to $\sqsubseteq$ respectively,

and provide the following representation theorem

Representation Theorem 3. A preference relation $\sqsubseteq$ on Pi(X), equipped with the mixture operation $M_\top$, satisfies the axiom set $\{A1, A2^+, A3_\top, A4^+_\top\}$ if, and only if, there exist

(i) a 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 order onto preserving function h: $V \to U$ such that $h(0) = 0$, $h(1) = 1$, satisfying coherence w.r.t. $\top$, i.e. $h(\lambda) = h(\mu) \Rightarrow h(\alpha \top \lambda) = h(\alpha \top \mu)$ for all $\alpha, \lambda, \mu \in V$,

in such a way that it holds:

$$\pi' \sqsubseteq \pi \text{ iff } \pi' \lhd_u \pi,$$

where $\lhd_u$ is the ordering on Pi(X) induced by the qualitative utility GQU^+.

2.3 POSSIBILISTIC CASE-BASED DECISION

As already mentioned, Dubois and Prade [7] propose an approach to case-based decision, based on possibility and necessity measures. They assume a given memory of cases M, a "similarity"[3] function $\mathcal{S}: S \times S \to [0,1]$ that measures the degree of similarity between two situations, and a utility function u: $X \to [0, 1]$ representing preferences on consequences. Given a new situation s_0 and a decision d, let $\mathcal{S}^d: \{ s \mid (s, d, x) \in M\} \to [0,1]$ be the fuzzy set of situations which are similar to s_0 and where decision d was experienced, with membership function $\mathcal{S}^d(s) = \mathcal{S}(s, s_0)$ and let $G^d: \{s \mid (s, d, x) \in M\} \to [0,1]$ be the fuzzy set of situations where decision d led to good results, with membership function $G^d(s) = u(x)$. They propose the following utility function

$$U^-_{s_0,M}(d) = \min_{(s,d,x) \in M} \mathcal{S}(s, s_0) \to u(x),$$

where $\to$ is chosen as $(x \to y) = N(x) \perp y$ with $\perp$ a conorm and N an involutive negation, that measures the degree of inclusion of $\mathcal{S}^d$ into G^d. When only ordinal interpretations are meaningful, $\perp$ = maximum, so

$$U^-_{s_0,M}(d) = \min_{(s,d,x) \in M} \max(N(\mathcal{S}(s, s_0)), u(x)).$$

The corresponding dual "optimistic" behaviour consists in selecting decisions which led to good results in at least one situation similar to s_0, using the dual criteria

$$U^+_{s_0,M}(d) = \max_{(s,d,x) \in M} \min(\mathcal{S}(s_0, s)), u(x)).$$

Applying the following Case-Based Reasoning Principle [4], *"The more similar is s_0 to s, the more possible x is a*

---

[3] Actually we are speaking about fuzzy proximity relation on S, i.e. $\mathcal{S}$ is symmetric and reflexive.

consequence of d at s_0" to each case (s, d, x)∈ M, in [5] they define the possibility distribution π_{d,s_0}: X → V associated to d and s_0 (depending also of $\mathcal{S}$ and M) as

$$\pi_{d,s_0}(x) = \max\{\mathcal{S}(s_0, s) \mid (s, d, x) \in M\},$$

where, by convention, we take max ∅ = 0. $\pi_{d,s_0}(x)$ represents the plausibility of x of being the consequence of s_0 by d. So, a decision or act d, at the new situation s_0, can be identified with its possibility distribution π_{d,s_0}. If there exists an element s in S such that (s, d, x) ∈ M and $\mathcal{S}(s_0, s) = 1$, π_{d,s_0} will be normalised. Taking U = V = [0,1], it can be shown that

$$U^-_{s_0}(d) = QU^-(\pi_{d,s_0}) \text{ and } U^+_{s_0}(d) = QU^+(\pi_{d,s_0}).$$

If π_{d,s_0} is normalised it is always the case that $U^+_{s_0}(d) \geq U^-_{s_0}(d)$, as expected. However, if π_{d,s_0} is not normalised we may obtain unsatisfactory results like the fact that the pessimistic utility $U^-_{s_0}(d)$ may be higher than the optimistic utility $U^+_{s_0}(d)$. Indeed, when $\max_{(s,d,x)\in M} S(s, s_0) < 1$, i.e. when decision d has been never experienced on a situation completely similar to s_0, for example if for all x, $\{s \mid (s, d\ x) \in M,\ S(s, s_0) > 0\} = \emptyset$, we have $U^-_{s_0}(d) = 1$ which is counterintuitive. In order to solve this problem, it is proposed in [4], taking U = V = [0, 1], to modify the previous definitions as

$$U^-_{s_0}(d) = \min\{\mathcal{H}(\pi_{d,s_0})), QU^-(\mathcal{N}(\pi_{d,s_0}))\}$$
$$U^+_{s_0}(d) = \max\{n_U(\mathcal{H}(\pi_{d,s_0})), QU^+(\mathcal{N}(\pi_{d,s_0}))\}$$

with $\mathcal{H}$ being the height of the distribution, i.e. $\mathcal{H}(\underline{\pi}) = \max_{x\in X} \underline{\pi}(x)$, and $\mathcal{N}(\underline{\pi})$ the normalisation of $\underline{\pi}$, i.e. the distribution $\mathcal{N}(\underline{\pi})(x) = 1$ if $\underline{\pi}(x) = \mathcal{H}(\underline{\pi})$ and $\mathcal{N}(\underline{\pi})(x) = \underline{\pi}(x)$ otherwise. In the case we are requiring V and U to be finite, linear and commensurate scales, it is proposed in [5]

$$U^-_{s_0}(d) = \min\{h(\mathcal{H}(\pi_{d,s_0})), QU^-(\mathcal{N}(\pi_{d,s_0}))\}$$
$$U^+_{s_0}(d) = \max\{n(\mathcal{H}(\pi_{d,s_0})), QU^+(\mathcal{N}(\pi_{d,s_0}))\}$$

with h and n as usual.

3 REPRESENTATION OF POSSIBILISTIC UTILITIES FOR NON NORMALISED DISTRIBUTIONS

3.1 THE PURE ORDINAL CASE

We consider now the set of possibilistic lotteries as the set $Pi^{ex}(X)$ of non necessarily normalised distributions on V. We keep the same notion of possibilistic mixture in $Pi^{ex}(X)$ as in Pi(X), i.e.

$$PME(\underline{\pi}_1, \underline{\pi}_2, \lambda, \mu) = (\lambda/\underline{\pi}_1, \mu/\underline{\pi}_2) = \\ = \max\{\min(\lambda, \underline{\pi}_1(x)), \min(\mu, \underline{\pi}_2(x))\}.$$

where $\max(\lambda, \mu) = 1$. Thus the reduction property

$$(\lambda/\underline{\pi}_1\ \mu/(\alpha/\underline{\pi}_1, \beta/\underline{\pi}_2)) = \\ = (\max(\lambda, \min(\mu, \alpha))/\underline{\pi}_1, \min(\mu, \beta)/\underline{\pi}_2)$$

still holds. Now we consider the qualitative (or ordinal) utility functions on $Pi^{ex}(X)$, corresponding to those considered at the end of section 2.3 in the case-based decision framework:

$$\underline{QU}^-(\underline{\pi}) = \min\{QU^-(\mathcal{N}(\underline{\pi}), h(\mathcal{H}(\underline{\pi}))\}$$
$$\underline{QU}^+(\underline{\pi}) = \max\{QU^+(\mathcal{N}(\underline{\pi}), n(\mathcal{H}(\underline{\pi}))\}.$$

Let $\sqsubseteq_c$ be a preference relation in $Pi^{ex}(X)$. We will denote by $\sqsubseteq$ its restriction to Pi(X), $\sim_c$ and $\sim$ the corresponding indifference relations and *inv* the reversing involution in Pi(X)/$\sim$[4]. In order to characterise the preference orderings induced by the utility $\underline{QU}^-$, we extend the axiom set AX, defined on (Pi(X), $\sqsubseteq$) in Subsection 2.1 with the following additional axiom

- **A5**: For all $\underline{\pi} \in Pi^{ex}(X)$ there exists $\lambda \in V$ such that $\underline{\pi} \sim_c (1/\mathcal{N}(\underline{\pi}), \lambda/\underline{x})$, where $\overline{x}$ and $\underline{x}$ are a maximal and a minimal elements of (X, $\sqsubseteq$) respectively, and $\lambda \in V$ is such that $\pi_\lambda \in inv[(1/\overline{x}, \mathcal{H}(\underline{\pi})/\underline{x})]$, with $\pi_\lambda = (1/\overline{x}, \lambda/\underline{x})$.

The intuitive idea behind this axiom is that, according to the above utility function $\underline{QU}^-$, a non-normalised possibilistic lottery $\underline{\pi}$ is indifferent to the corresponding normalised lottery $\mathcal{N}(\underline{\pi})$ provided that this is modified by a uniform uncertainty level corresponding to the normality degree of the lottery expressed by the its height $\mathcal{H}(\underline{\pi})$. Notice that $\underline{x}$ is indifferent to the whole set X, so $\underline{\pi}$ is indifferent to the lottery $(1/\mathcal{N}(\underline{\pi}), \mathcal{H}(\underline{\pi})/X)$.

Representation Theorem 4 *("Pessimistic" Utility).* A preference relation $\sqsubseteq_c$ on $Pi^{ex}(X)$ satisfies axiom set $\underline{AX}$ = AX ∪ {A5} if, and only if, there exist

(i) a linearly ordered preference scale U with inf(U) = 0 and sup(U) = 1,

(ii) a preference function u: X → U such that $u^{-1}(1) \neq \emptyset \neq u^{-1}(0)$, and

(iii) an onto order preserving function h: V → U such that h(0) = 0 and h(1) = 1,

in such a way that it holds:

$$\underline{\pi}' \sqsubseteq_c \underline{\pi} \text{ iff } \underline{\pi}' \preccurlyeq \underline{\pi}$$

where $\preccurlyeq$ is the ordering induced on $Pi^{ex}(X)$ by the qualitative utility $\underline{QU}^-$.

Proof.

←) Let us introduce the order-reversing mapping n:V→ U defined as $n(\lambda) = n_U(h(\lambda))$. Consider now the utility function QU^- defined in terms of n and u. Axioms AX are

---

[4] When we have both scales of uncertainty and preferences, Pi(X)/~ is "isomorph" with U, and then *inv* is like n_U.

verified because of $\underline{QU}^-$ restricted to Pi(X) is equal to QU^- and by theorem 1, we have that $\preccurlyeq_u$ satisfies AX, where $\preccurlyeq_u$ is the ordering induced by QU^- in Pi(X). Now we verify A5. Using that h is onto, we may choose λ such that $n(\lambda) = h(\mathcal{A}(\underline{\pi}))$, then as QU^- preserves possibilistic mixtures

$$QU^-(1/\mathcal{N}(\underline{\pi}), \lambda/\underline{x}) = \min\{QU^-(\mathcal{N}(\underline{\pi})), n(\lambda)\} = \\ = \min\{QU^-(\mathcal{N}(\underline{\pi})), h(\mathcal{A}(\underline{\pi})\} = \underline{QU}^-(\underline{\pi}).$$

$\rightarrow$) Since $\sqsubseteq$ satisfies axioms AX we may apply theorem 1. So, we have determined the existence of U, n and u satisfying (i) and (ii) such that QU^- represents $\sqsubseteq$, with $QU^-(\pi) = \min_{x \in X} \max(n(\pi(x)), u(x))$. Now we define $h = n_U \circ n$, h satisfies (iii). Since A5 guarantees that $\exists\, \lambda$ such that $\underline{\pi} \sim_c (1/\mathcal{N}(\underline{\pi}), \lambda/\underline{x})$, we define

$$\underline{QU}^-(\underline{\pi}) = QU^-(1/\mathcal{N}(\underline{\pi}), \lambda/\underline{x}).$$

Now we verify that $\underline{QU}^-$ represents $\sqsubseteq_c$ in the sense that $\underline{\pi}' \sqsubseteq_c \underline{\pi}$ iff $\underline{QU}^-(\underline{\pi}') \leq \underline{QU}^-(\underline{\pi})$. By A5 and A4 we have that exist λ, λ' such that $\underline{\pi} \sim_c \pi_\lambda$, $\underline{\pi}' \sim_c \pi_{\lambda'}$ and $\underline{QU}^-(\underline{\pi}) = QU^-(\pi_\lambda)$, $\underline{QU}^-(\underline{\pi}') = QU^-(\pi_{\lambda'})$. So $\underline{\pi}' \sqsubseteq_c \underline{\pi}$ iff $\pi_{\lambda'} \sqsubseteq_c \pi_\lambda$, and as QU^- represents $\sqsubseteq$ we have that $QU^-(\pi_{\lambda'}) \leq QU^-(\pi_\lambda)$. Then recalling that QU^- coincides with $\underline{QU}^-$ on Pi(X), we obtain that indeed

$$\underline{\pi}' \sqsubseteq_c \underline{\pi} \text{ iff } \underline{QU}^-(\underline{\pi}') \leq \underline{QU}^-(\underline{\pi}).$$

It remains to prove that $\underline{QU}^-(\underline{\pi}) = \min\{QU(\mathcal{N}(\underline{\pi})), h(\mathcal{A}(\underline{\pi})\}$. Since QU^- preserves mixtures, and A5 guarantees that $\exists\, \lambda$ such that $\underline{\pi} \sim_c (1/\mathcal{N}(\underline{\pi}), \lambda/\underline{x})$,

$$\underline{QU}^-(\underline{\pi}) = QU^-(1/\mathcal{N}(\underline{\pi}), \lambda/\underline{x}) = \min\{QU^-(\mathcal{N}(\underline{\pi})), n(\lambda)\},$$

and since $\pi_\lambda \in inv[(1/\overline{x}, \mathcal{A}(\underline{\pi})/\underline{x})]$ we have that $n(\lambda) = h(\mathcal{A}(\underline{\pi}))$, so we finally obtain that

$$\underline{QU}^-(\underline{\pi}) = \min\{QU^-(\mathcal{N}(\underline{\pi})), h(\mathcal{A}(\underline{\pi}))\}. \qquad \square$$

In a similar way, we may characterise the "optimistic" utility on $Pi^{ex}(X)$, using the set of axioms AX^+ plus

- $\underline{A5}^+$: For all $\underline{\pi} \in Pi^{ex}(X)$ there exists $\lambda \in V$ such that $\underline{\pi} \sim_c (1/\mathcal{N}(\underline{\pi}), \lambda/\overline{x})$, where $\overline{x}$ and $\underline{x}$ are a maximal and a minimal elements on $(X, \sqsubseteq)$ respectively and $\lambda \in V$ is such that $\pi_\lambda \in inv[(h(\underline{\pi})/\overline{x}, 1/\underline{x})]$, with $\pi_\lambda = (\lambda/\overline{x}, 1/\underline{x})$.

We then have a corresponding representation theorem for th eoptimistic behaviour, whose proof is very analogous to the pessimistic case and it will be omitted.

Representation Theorem 5 *("Optimistic" Utility).* A preference relation $(Pi^{ex}(X), \sqsubseteq_c)$ satisfies axioms A1, A2$^+$, A3, A4$^+$ and $\underline{A5}^+$, if and only if there exist

(i) a linearly ordered preference scale U, with inf(U) = 0, sup(U) = 1,

(ii) a preference function u: $X \rightarrow U$ such that $u^{-1}(1) \neq \emptyset \neq u^{-1}(0)$, and

(iii) an onto order reversing function n: $V \rightarrow U$ such that n(0) = 1, n(1) = 0,

in such a way that it holds:

$$\underline{\pi}' \sqsubseteq \underline{\pi} \text{ iff } \underline{\pi}' \lhd \underline{\pi},$$

where $\lhd$ is the ordering on $Pi^{ex}(X)$ induced by the qualitative utility $\underline{QU}^+$.

3.2 THE CASE OF MAX-T POSSIBILISTIC MIXTURES

In order to extend the above decision model coping with possibly non normalised distributions to the case of generalised max-T mixtures introduced in Section 2.2 of the form

$$\underline{M}_T(\underline{\pi}, \underline{\pi}'; \alpha, \beta) = \max(\alpha \mathrel{T} \underline{\pi}, \beta \mathrel{T} \underline{\pi}')$$

with $\max(\alpha, \beta) = 1$, now defined in $Pi^{ex}(X)$, we propose to consider the following utilities

$$\underline{GQU}^-(\underline{\pi}) = \min\{GQU^-(\mathcal{N}(\underline{\pi})), h(\mathcal{A}(\underline{\pi}))\}$$
$$\underline{GQU}^+(\underline{\pi}) = \max\{GQU^+(\mathcal{N}(\underline{\pi})), n(\mathcal{A}(\underline{\pi}))\}.$$

In a very mimetic way to previous sections, we consider the axiom set $\underline{AX}_T = \{A1, A2, A3_T, A4_T, \underline{A5}_T\}$, where the new axiom $\underline{A5}_T$ is the suitable adaptation of previous axiom A5 for the present type of mixtures.

- $\underline{A5}_T$: For all $\underline{\pi} \in Pi^{ex}(X)$ there exists $\lambda \in V$ s.t. $\underline{\pi} \sim_c M_T(\mathcal{N}(\underline{\pi}), \underline{x}, 1, \lambda)$, where $\overline{x}$ and $\underline{x}$ are a maximal and a minimal elements on $(X, \sqsubseteq)$ respectively and $\lambda \in V$ is such that $\pi_\lambda \in inv[M_T(\overline{x}, \underline{x}, 1, \mathcal{A}(\underline{\pi})]$, with $\pi_\lambda = M_T(\overline{x}, \underline{x}, 1, \lambda)$.

Representation Theorem 6. A preference relation $\sqsubseteq_c$ on $(Pi^{ex}(X), M_T)$ satisfies axioms $\underline{AX}_T$ if, and only if, there exist

(i) a linearly ordered preference scale U with inf(U) = 0 and sup(U) = 1,

(ii) a preference function u: $X \rightarrow U$ such that $u^{-1}(1) \neq \emptyset \neq u^{-1}(0)$, and

(iii) an onto order preserving function h: $V \rightarrow U$ such that h(0) = 0 and h(1) = 1, satisfying coherence w.r.t. T, i.e. $h(\lambda) = h(\mu) \Rightarrow h(\alpha T \lambda) = h(\alpha T \mu)$ for all $\alpha, \lambda, \mu \in V$,

in such a way that it holds:

$$\underline{\pi}' \sqsubseteq_c \underline{\pi} \text{ iff } \underline{\pi}' \preccurlyeq \underline{\pi},$$

where $\preccurlyeq$ is the ordering in $Pi^{ex}(X)$ induced by the qualitative utility $\underline{GQU}^-$.

Proof. The proof is very similar to the case when T is the minimum. Consider $n = n_U \circ h$. To verify axiom $\underline{A5}_T$, take in account that GQU^- preserves extended mixtures, in the sense that

$GQU^-(M_T(\pi_1, \pi_2, \lambda, \mu)) = \min\{n(\lambda T \delta_1), n(\mu T \delta_2)\}$ with $n(\delta_j) = GQU^-(\pi_j)$, and the fact that by the coherence condition, we have that

$$n(\delta_2) = 0 \text{ implies } n(\mu \top \delta_2) = n(\mu).$$

So, we only sketch the proof for the "only if" part. As $\sqsubseteq$ satisfies AX_T, we may establish the existence of n, U and u, such that $GQU^-(\pi) = \min_{x_i \in X} n(\pi(x_i) \top \lambda_i)$, where $n(\lambda_i) = u(x_i)$, represents $\sqsubseteq$. Consider $h = n_U \circ n$. Since for any $\underline{\pi} \in Pi^{ex}(X)$ $\underline{A5}_T$ guarantees that $\exists\ \lambda$ such that $\underline{\pi} \sim_c M_T(\mathcal{N}(\underline{\pi}), \underline{x}, 1, \lambda)$, we define

$$\underline{GQU}^-(\underline{\pi}) = GQU^-(M_T(\mathcal{N}(\underline{\pi}), \underline{x}, 1, \lambda)).$$

As before, we may verify that $\underline{GQU}^-$ represents $\sqsubseteq_c$ using the fact that GQU^- represents $\sqsubseteq$ and axiom $\underline{A5}_T$ and $A4_T$. It remains to prove that

$$\underline{GQU}^-(\underline{\pi}) = \min\{GQU^-(\mathcal{N}(\underline{\pi})), h(\mathcal{A}(\underline{\pi}))\}.$$

By $\underline{A5}_T$, there exists λ s.t. $\underline{\pi} \sim_c M_T(\mathcal{N}(\underline{\pi}), \underline{x}, 1, \lambda)$, so as $\underline{GQU}^-(\underline{\pi}) = GQU(M_T(\mathcal{N}(\underline{\pi}), \underline{x}, 1, \lambda))$ and

$$GQU^-(M_T(\pi_1, \pi_2, \lambda, \mu)) = \min\{n(\lambda T \delta_1), n(\mu \top \delta_2)\}$$

with $n(\delta_j) = GQU^-(\pi_j)$, we have that

$$GQU^-(M_T(\mathcal{N}(\underline{\pi}), \underline{x}, 1, \lambda)) = \min\{n(1 T \delta_1), n(\lambda \top \delta_2)\}$$

with $n(\delta_1) = GQU^-(\mathcal{N}(\underline{\pi}))$ and $n(\delta_2) = GQU^-(\underline{x}) = 0 = n(1)$. Therefore using the coherence condition of n w.r.t. T and the fact that $n(\delta_2) = n(1)$, we obtain that

$$\underline{GQU}^-(\underline{\pi}) = \min\{GQU^-(\mathcal{N}(\underline{\pi})), n(\lambda)\},$$

and because of $\pi_\lambda \in inv[M_T(\overline{x}, \underline{x}, 1, \mathcal{A}(\underline{\pi})]$ we have that $n(\lambda) = h(\mathcal{A}(\underline{\pi}))$, we obtain that

$$\underline{GQU}^-(\underline{\pi}) = \min\{GQU^-(\mathcal{N}(\underline{\pi})), h(\mathcal{A}(\underline{\pi}))\}. \quad \square$$

As usual, we also have a similar representation theorem for an optimistic case with the extended mixture, corresponding to the following axiom set $\underline{AX}^+_T = \{A1, A2^+, A3_T, A4^+_T, \underline{A5}^+_T\}$, where

- $\underline{A5}^+_T$: For all $\underline{\pi} \in Pi^{ex}(X)$ there exists $\lambda \in V$ such that $\underline{\pi} \sim_c M_T(\mathcal{N}(\underline{\pi}), \overline{x}, 1, \lambda)$, where $\overline{x}$ and $\underline{x}$ are a maximal and a minimal elements on $(X, \sqsubseteq)$ respectively and $\lambda \in V$ is such that $\pi_\lambda \in inv[M_T(\overline{x}, \underline{x}, \mathcal{A}(\underline{\pi}), 1]$, with $\pi_\lambda = M_T(\overline{x}, \underline{x}, \lambda, 1)$.

4 BACK TO CASE-BASED DECISION

Finally, using the link between similarity on situations and possibility distributions on consequences described in Section 2.3, we just propose here to apply the generalised qualitative utility functions $\underline{GQU}^-$ and $\underline{GQU}^+$ (for a given t-norm T defined on the uncertainty scale V, see 2.3) for case-based decision.

First, we recall our hypotheses. We assume as given a deterministic memory of cases $M \subseteq S \times D \times X$, being S a finite set of situations, X a finite set of consequences and D the set of decisions available. We have two finite qualitative scales linearly ordered V and U for similarity and preferences respectively, being commensurable between them, and a "similarity" function $\mathcal{S}: S \times S \to V$, that measures the degree of similarity between two situations, and a utility function $u: X \to U$ representing preferences on consequences. As previous, let $\mathcal{S}^d$ be the V-fuzzy set of situations which are similar to s_0 and where decision d was experienced, with membership function $\mathcal{S}^d(s) = \mathcal{S}(s, s_0)$ and let G^d be the U-fuzzy set of situations where decision d led to good results, with membership function $G^d(s) = u(x)$.

So, if we are interested in acts s.t. that in *all* the situations similar to s_0, d led in good results, we are looking for decisions that maximise this function,

$$GU^-_{s_0}(d) = \underline{GQU}^-(\pi_{d,s_0}) = \\ = \min\{h(\mathcal{A}(\pi_{d,s_0})), GQU^-(\mathcal{N}(\pi_{d,s_0}))\}$$

while if we are looking for decisions which gave a good result is a similar situation we may want to maximise

$$GU^+_{s_0}(d) = \underline{GQU}^+(\pi_{d,s_0}) = \\ = \max\{n(\mathcal{A}(\pi_{d,s_0})), GQU^+(\mathcal{N}(\pi_{d,s_0}))\}.$$

Finally, let us remark that $GQU^-(\mathcal{N}(\pi_{d,s_0}))$ can be still regarded as the degree of inclusion $[\mathcal{S}^d \subseteq G^d]$ of the fuzzy set of situations similar to s_0 into the fuzzy set of situations in which d led to good results if we define

$$[\mathcal{S}^d \subseteq G^d] = \mathrm{Min}_{s/(s,d,x) \in M} (\mathcal{S}^d(s) \to G^d(s))$$

where $a \to b$ ($a \in V$, $b \in U$) is a many-valued implication function of the type "not(a and not b)", interpreting the "and" by a t-norm T, and because of the different domains involved (V, U) has to be formally expressed as $a \to b = n(a \top \beta)$, where $n(\beta) = b$. Analogously, $GQU^+(\mathcal{N}(\pi_{d,s_0}))$ is still a degree of intersection $[\mathcal{S}^d \cap G^d]$ provided that we define

$$[\mathcal{S}^d \cap G^d] = \mathrm{Max}_{s/(s,d,x) \in M} (\mathcal{S}^d(s) \otimes G^d(s))$$

where $\otimes$ is a t-norm like function (a many-valued conjunction operation) defined as $a \otimes b = h(a)\ T_U\ b$, where T_U is a transform of the t-norm T (defined on V) into U.

5 CONCLUSIONS

In this paper we have been concerned with representational issues of preferences in the framework of a possibilistic (qualitative) decision model under uncertainty, originally introduced by Dubois and Prade. This decision model differs from other alternatives to EUT found in the

literature (see for instance [1, 2, 3, 6, 8, 10, 12, 13]) in that only ordinal scales of uncertainty and preference are required. The model is extended here to cope with decision problems where the belief state may be partially inconsistent. This amounts to let the model allow for ranking possibly non-normalised possibility distributions on the set of decision consequences. The model provides dual conservative and optimistic evaluation functions, formally similar to the original expressions which cope with ordinal uncertainty up to modifying factors which cope with the lack of normalisation of belief states. The lack of normalisation may appear quite naturally when applying the possibilistic decision model for case-based decision problems, as it has been discussed in the paper. Finally, generalisations of the model by allowing non purely ordinal expressions in the utility functions have been also considered.

Acknowledgements

Lluís Godo has been partially supported by the CICYT project SMASH (TIC96-1138-C04-01) and Adriana Zapico acknowledges partial support by the Universidad Nacional de Río Cuarto (Argentina). The authors are also grateful to anonymous referees for their comments that have helped in improving the paper.

References

[1] Bonet B. and Geffner H. (1996): Arguing for decisions: A qualitative model of decision making. *Proc. of the 12th Conf. on Uncertainty in Artificial Intelligence* (E. Horwitz, F. Jensen, eds.), Portland, OR, pp. 98-105.

[2] Boutilier C. (1994): Toward a logic for qualitative decision theory, in *Proc. 4th. Inter. Conf. on Principles of Knowledge Representation and Reasoning (KR'94)*, Bonn, pp. 75-86.

[3] Brafman R.I. and Tennenholtz M. (1997): On the axiomatization of qualitative decision criteria, in *Proc. 14th Nat. Conf. on A.I. (AAAI'97)*, pp. 76-81.

[4] Dubois D., Esteva F., Garcia P., Godo L., L. de Màntaras R. and Prade H. (1997): Fuzzy Set modelling in case-based reasoning, in *International Journal of Intelligent System* , p.p. 345 - 373.

[5] Dubois D. , Godo L., Prade H. and Zapico A. (1998): Making decision in a qualitative setting: from decision under uncertainty to case-based decision, in *Sixth International Conference on Principles of Knowledge Representation and Reasoning(KR'98).* Trento, p.p. 594 - 605.

[6] Dubois D. and Prade H. (1995): Possibility theory as a basis for qualitative decision theory, in *Proc. of the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI'95)*, Montreal, pp. 1924-1930.

[7] Dubois D. and Prade H. (1997): A fuzzy set approach to case-based decision, in *Proc. of the Second European Workshop on Fuzzy Decision Analysis and Neural networks for Management, Planning and Optimization (EFDAN97)*, Dortmund, pp. 1-9.

[8] Gilboa I. and Schmeidler D. (1995): Case-based Theory, *The Quarterly Journal of Economics*, pp. 607-639.

[9] von Neumann J. and Morgenstern O. (1944): *Theory of Games and Economic Behaviour* (Princeton Univ. Press, Princeton, NJ).

[10] Pearl J. (1993): From qualitative utility to conditional "ought to", in *Proc. of the 9th Inter. Conf. on Uncertainty in Artificial Intelligence*, (D. Heckerman, H. Mamdani, eds.), pp. 12-20.

[11] Savage L. J. (1972): The Foundations of Statistics. Dover, New York.

[12] Schmeidler D. (1986): Integral representation without aditivity, in *Proc. Amer. Math. Soc.* 9 , pp 255-261.

[13] Schmeidler D. (1989): Subjective probability and expected utility without additivity. in *Econometrica*, 57, pp. 571-587.

[14] Whalen T. (1984): Decision making under uncertainty with various assumptions about available information. IEEE *Trans. on Systems, Man and Cybernetics*, 14, pp. 888-900.

[15] Yager R.R. (1979): Possibilistic decision making. *IEEE Trans. on Systems, Man and Cybernetics*, 9, pp. 388-392.

SEMANTICA PARA SEGMENTACIONES SIN CLASES PREVIAS, USANDO REGLAS DIFUSAS

José M. Benítez, Ignacio Requena
Dpto Ciencias Computación e Inteligencia A.
Universidad de Granada
requena@decsai.ugr.es

Enrique López
Dpto Contabilidad y Direcc. Empresas.
Universidad de León
ddeelg@unileon.es

Resumen

La segmentación sin clases previas en situaciones reales, tiene el problema de la asignación de una semántica a las clases obtenidas. Una vez realizada la clasificación, difusa o no, el experto necesita manejar las clases, poniéndoles un nombre (o una semántica) para que les sean útiles en su quehacer cotidiano. En este trabajo se propone una metodología, basada en reglas difusas, para la asignación de esta semántica

Palabras Clave: Segmentación, reglas difusas, asignación de semántica a clases..

1 INTRODUCCIÓN

El problema de la segmentación es muy importante en muchas situaciones de la vida real. Muchas empresas o instituciones realizan continuamente segmentaciones de sus clientes, productos, ... que permiten a la empresa realizar acciones directas sobre los mismos, controlar procesos, etc..

Por ejemplo, en un banco o entidad financiera, las relaciones con los clientes es una de las componentes más importantes para el éxito de la empresa. Las necesidades de cada cliente han de estar claras en la mente de los expertos que desarrollan la planificación estratégica de la entidad. La mejora y diversificación de los productos ofertados es una constante diaria, debido a la creciente competencia entre diferentes entidades.

La segmentación juega un papel crucial en el conocimiento del comportamiento de los clientes, y este es básico para el desarrollo de estrategias de actuación en temas de concesión de créditos, introducción y ofertas ce nuevos productos etc..

Lo habitual es definir unas clases de comportamiento y asignar a cada cliente una de las clases, de acuerdo con la información que se tiene de los mismos. Esta segmentación suele ser excluyente, aunque últimamente se están realizando segmentaciones difusas. segmentación suele ser excluyente, aunque últimamente se están realizando segmentaciones difusas. La identificación de las clases se suele hacer por expertos de la propia empresa o contratando

una consultora. La segmentaciones excluyentes tienen el problema de ser muy rígidas en las fronteras de separación entre unas clases y otras, lo que con frecuencia distorsiona un poco la situación. Por ello se tiende cada vez más al uso de las segmentaciones no excluyentes, entre las que se abren paso de forma importante las segmentaciones difusas.

En muchas situaciones reales, la determinación previa de las clases distorsiona la realidad, por lo que sería muy interesante hacer la segmentación sin establecer previamente las clases. En este caso, y suponiendo una clasificación aceptada de los items (clientes, productos, ...), surge un problema adicional, que es el de establecer una semántica adecuada a las clases obtenidas, para que los diferentes departamentos de la empresa puedan hacer útil la segmentación realizada.

En trabajos anteriores, hemos desarrollado metodologías basadas en Redes Neuronales, para obtener un conjunto de reglas difusas que describan correctamente un sistema [1,2]. En otro trabajo [3], apuntamos la posibilidad de realizar las asignaciones de una semántica a las clases, e indicábamos la forma de hacerlo. En este trabajo se desarrolla lo indicado en [3], usando los métodos señalados en [1,2]

2 REGLAS DIFUSAS PARA UNA SEGMENTACION SIN CLASES PREVIAS

En [1] se describe un procedimiento de obtención de reglas difusas, que llamamos metodología de selección, mediante un proceso de selección de reglas, a partir del conjunto de todas las reglas posibles, de acuerdo con un conjunto de etiquetas previas para cada variable. Se entrena una RNA con los ejemplos disponibles del sistema y se comprueba para cada regla, de todas las posibles, su adaptación a la RNA entrenada, rechazando las que no se adaptan bien. Sobre las reglas no rechazadas, se aplica un algoritmo tipo greedy (voraz), que va seleccionando las reglas que cubren mejor los ejemplos del sistema, según un grado de cobertura. El proceso finaliza cuando la cobertura sobre los ejemplos es suficiente.

En [2] se describe un procedimiento constructivo, que permite obtener las variables relevantes para cada salida del sistema, y no necesita etiquetas previas para las variables, ya que se construyen directamente.

En primer lugar se obtiene un conjunto de pre-reglas para cada salida posible del sistema (cada clase en problemas de segmentación). En segundo lugar, para construir las etiquetas, se consideran los valores de las variables que intervienen en una salida y se realiza una clasificación (clustering) difusa, mediante un método de clasificación difusa (C-medias difusa, Mapas de Kohonen difusos, ...). Cada clase dará lugar a una regla efectiva.

Para construir las etiquetas de la regla, se ordenan los datos de la clase para cada variable en la pre-regla, y se le asigna a cada valor una frecuencia proporcional al grado de pertenencia a la clase. Se obtiene así, una distribución estadística de los valores de esa variable en cada clase.

Construimos la etiqueta correspondiente a cada variable, como un número difuso trapezoidal, considerando los valores extremos para el soporte, y los cuartiles 1 y 3 de la distribución para la moda.

Estos procesos, aplicados a problemas de segmentación, necesitan que las clases sean conocidas previamente, ya que se utiliza aprendizaje supervisado. Cuando las clases no se conocen previamente, antes de aplicar los métodos señalados más arriba, se realiza una segmentación sin clases previas adecuada. Nosotros hemos utilizado el método de Chiu [4]. Este método permite obtener una clasificación difusa sin fijar previamente el número de clases ni los centros.

Para ello, asigna a cada ejemplo, una medida de su potencialidad como centro de una clase, y selecciona como centros los que parece que mejor van a representar al conjunto de ejemplos. Los datos han de estar normalizados en el mismo rango (normalmente el intervalo [0,1]. El potencial de cada punto depende de la distancia a los demás

puntos, y de un parámetro inversamente proporcional al radio de la clase (cuantos más puntos haya dentro de ese entorno,

Con la segmentación obtenida, ya podemos aplicar los procesos de obtención de reglas difusas, descritos anteriormente.

.

3 ASIGNACION DE UNA SEMANTICA A LAS CLASES

En los procesos de segmentación sin clases previamente definidas, un aspecto interesante es la asignación de una semántica a las clases obtenidas, es decir, describir la clase en forma suscinta y clara, para que el usuario pueda manejar las clases, sin tener que controlar la lista completa de los items en la misma. Este aspecto, que en la literatura científica es poco relevante ya que importan sobre todo el número de clases y los centros de las mismas, es un aspecto crucial en aplicaciones y situaciones reales, ya que para los expertos que han de manejar las clases obtenidas, es fundamental ponerles un "nombre" a cada clase, que refleje de forma adecuada la pertenencia de cada item a las clases.

Proponemos la siguiente metodología para asignar una semántica a las clases obtenidas en un proceso de segmentación sin clases previas, en situaciones de la vida real, donde la opinión de los expertos que posteriormente van a manejar las clases tiene gran importancia.

Suponemos que disponemos de un conjunto de ejemplos que incluyen los valores de las variables que vamos a usar en la clasificación. No tenemos un conjunto de clases previamente definido. Separamos los ejemplos en un conjunto para el aprendizaje, y otro para la verificación (test) de los resultados obtenidos. Entonces, los pasos a seguir son:

1) Con los ejemplos de aprendizaje, realizamos una clasificación sin clases previas, siguiendo el método de Chiu [4].
2) Junto con los expertos del sistema, se revisa la segmentación obtenida, usando por un lado el conocimiento que los expertos tienen del sistema, para analizar la homogeneidad intraclases y la heterogeneidad interclases, y por otro lado el conjunto de ejemplos de test. Este análisis se basa fundamentalmente en la experiencia de los expertos, que conocen, de acuerdo con los datos de los registros, si dos registros pueden y/o deben estar en la misma clase, y si las clases obtenidas parecen adecuadas.

Para obtener los resultados adecuados al criterio de los expertos, a veces habrá que modificar levemente los parámetros del método de Chiu.

Generalmente, la aportación de los expertos suele ser la necesidad de incorporar clases no obtenidas por el proceso de clasificación sin clases previas, pero que son fundamentales en la planificación estratégica de la entidad.

3) Con la clasificación obtenida, aplicamos la metodología constructiva para obtener un conjunto de reglas que describan adecuadamente el sistema obtenido en los pasos 1) y 2).
4) Asignamos una semántica inicial a cada clase, teniendo en cuenta lo siguiente:
   - A cada variable en la(s) regla(s) que definen la clase le asignamos un intervalo de valoración, obtenido a partir del centroide de la etiqueta de la variable, de forma que su amplitud a izquierda y derecha recoja la forma de la etiqueta.
   - Cada clase puede estar definida por una o más reglas, y en cada regla puede haber varias variables. Los intervalos de las variables en la misma regla se unen con el conector "Y". Los

correspondientes a distintas reglas, se unen con el conector "O".

- La semántica tendrá la forma:

SI a11 < var11 < b11 Y a12 <var21<b21 Y .....

O

SI a21 < var21 < b21 Y a22 es valor22 Y .....

O

. . . . . .

ENTONCES el item está en la clase C1

La variable a22 refleja una variable cualitativa, no numérica.

5) De nuevo se dialoga con los expertos del sistema o entidad, con dos objetivos:
   - Estar seguros de que los expertos comprenden bien la semántica establecida
   - Buscar junto con los expertos un "lenguaje" más cercano al uso que se hará de la segmentación que describa la semántica establecida.
6) Establecer la semántica definitiva y fin del proceso

La intervención de los expertos del sistema que tratamos de clasificar es fundamental en dos fases de la metodología. En el paso 2, para estar seguros de que las clases obtenidas y la asignación de los items a las clases responde realmente al sistema. En el paso 5, para estar seguros de que la forma en que las clases están descritas son adecuadas al uso que de estas descripciones se va a hacer en la realidad.

Teniendo en cuenta que el método de obtención de reglas difusas es válido, tanto si la clasificación de partida es crisp o difusa, la asignación de la semántica a las clases obtenidas puede realizarse tanto para una segmentación crisp como para una segmentación difusa. En este caso, el grado de pertenencia de los items a la clase se calcula como agregación (usando p.e. el mínimo para los conectores Y y el máximo para los conectores O) a partir de una asignación lineal entre el centroide (valor 1) y los extremos (valor 0) del intervalo asociado a cada variable en el paso 4 de la metodología descrita.

.

4 CASO DE ESTUDIO

Hemos utilizado el procedimiento descrito en el apartado anterior, durante el desarrollo de un Proyecto de Investigación sobre Segmentación Difusa de Clientes, realizado para una entidad bancaria.

Se trataba de verificar con la tecnología señalada, si la segmentación que la Entidad venía usando era ratificada o no por nuestro procedimiento, tanto en número de clases consideradas, como en la asignación de los items a las distintas clases.

La Entidad disponía de una segmentación crisp de los clientes en 8 clases. Para la asignación de los items a las clases se utilizaban hasta 15 variables diferentes. En el preprocesamiento de los datos, necesario para otras fases del proyecto, estas 15 variables fueron resumidas en 9 variables de entrada. Dispusimos de 3000 items, de los cuales, 600 fueron usados para el aprendizaje, y el resto para la fase de verificación.

Describimos ahora la aplicación del proceso.

PASO 1.-

Tras la aplicación del método de Chiu, les presentamos a los expertos de la entidad, dos segmentaciones crisp ligeramente diferentes con 6 clases cada una. También se les presentó una segmentación con 5 clases. Estas segmentaciones diferentes se obtuvieron modificando ligeramente los parámetros internos del método de Chiu.

PASO 2.-

En el dialogo con los expertos, se desechó la segmentación de 5 clases, y se aceptó de forma básica, una de las de 6 clases. Aunque nuestro sistema no las consideró

como clases diferenciadas, los expertos añadieron 2 clases más que consideraban importantes, de acuerdo con la planificación estratégica de la Entidad.

PASO 3.-

La aplicación del método constructivo para obtener las reglas difusas nos permitió obtener un conjunto de 10 reglas, 1 por clase, excepto 2 clase con 2 reglas cada una. A título de ejemplo, la regla obtenida para una de las clases era:

SI var1 (0, 0, 0, 0) Y var3 (.45, .45, .46, .47)
Y var7 (0, 0, 0.01, 0.1)
ENTONCES C4

PASO 4.-

Se obtuvieron las descripciones de las clases en la forma ya señalada anteriormente. A titulo de ejemplo, la semántica asignada a la clase C4 fue:

La semántica inicial obtenida para esta clase fue:

SI var1 aprox. 0 Y var3 menor que 17
Y var 7 menor que 900
ENTONCES C4

PASO 5.-

En el dialogo con los expertos de la Entidad, pudimos observar, ya que era el objetivo principal del experimento, que las clases obtenidas, con sus semánticas respectivas estaban básicamente de acuerdo con las clases iniciales que la Entidad manejaba, teniendo en cuenta las dos clases añadidas por planificación estratégica de la propia entidad.

Donde sí había diferencias, importantes en algunos casos, era en la composición de las clases, es decir, en la asignación de items a algunas de las clases. Esto hacía que las semánticas asociadas a las clases, válidas básicamente, para la segmentación obtenida con nuestro procedimiento, no fuesen totalmente válidas para 3 clases de la segmentación de la Entidad.

PASO 6.-

Se establecieron las semánticas definitivas para la segmentación obtenida con este procedimiento, y se ratificó su comprensión con expertos que no habían participado en el proceso de asignación de semántica a las clase.

5 CONCLUSIÓNES

Se ha dado un procedimiento, aplicable a situaciones de la vida real, para asignar semánticas, y a partir de éstas, posiblemente nombres simples, a las clases obtenidas mediante procesos de segmentación sin clases predefinidas. Para ello, se ha utilizado una metodología basada en reglas difusas.

Se ha aplicado en un proceso real de obtención de clasificaciones difusas, como alternativa a las clasificaciones excluyentes, en clientes bancarios.

Agradecimientos

Este trabajo ha sido financiado en parte por el proyecto SEDI. Caja Asturias. Oviedo

Referencias

[1.] Benítez J.M., Blanco A., Requena I. . An empirical procedure to obtain fuzzy rules using neural networks. Procc. 6th IFSA Congress. Sao Paulo. Brasil. Vol II pp 663-669. 1995.

[2] Benítez, J.M., Blanco A., Delgado M., Requena I. .Neural methods to obtaining fuzzy rules. Mathware & Soft Computing, 3, pp 371-382. 1996

[3] Requena, I., López, E., Segmentación difusa de clientes bancarios. Proc. ESTYLF'96 .Oviedo 1996, pp 231-234

[4] Chiu, S.L. A cluster estimation method with extension to fuzzy model identification. Procc. IEEE Inter. Congress on Fuzzy Systems, pp 1240-1245. 1994

EL MÉTODO DE ACUMULACIÓN APLICADO AL RAZONAMIENTO NUMÉRICO

Jaume Casasnovas Casasnovas
Departamento de Matemáticas e Informática
Universidad de las Islas Baleares
Crra.Valldemossa Km. 7'5
07071 Palma de Mallorca, Spain
e-mail: dmijcc0@ps.uib.es

RESUMEN

En los problemas de razonamiento numérico solemos conocer solamente una proporción o porcentaje del total de elementos de un conjunto que cumple determinada propiedad, tal como pertenecer a otro conjunto. Si tenemos varios de estos datos , las reglas de razonamiento numérico tratan de determinar o de acotar el conocimiento relativo a porcentajes de total de elementos que cumplen otras propiedades no explicitadas en los datos de partida. El método que se presenta en este trabajo trata de extender a este tipo de razonamiento la operación acumulación y la inferencia de resultados que puede realizarse cuando los datos que se tienen son relativos a la verdad o la falsedad de los enunciados

Palabras Clave: razonamiento numérico, razonamiento aproximado,acumulación,agregación. Propagation rules

1. INTRODUCCIÓN

En [2] y [4] se presentó un concepto y un método para realizar una acumulación de los conocimientos , exactos o aproximados,relativos a la valoración en el intervalo [0 1] de la verdad de enunciados referidos a un universo finito. Esta acumulación es aplicable también a enunciados valorados en {0,1} mediante una operación interna del conjunto de símbolos {0,1,*,k} (Casasnovas [3], estudiada ya por Belnap[1]), que representan la afirmación , la negación, la ignorancia y la contradicción, respectivamente.En [5] se extiende el método a medidas de Posibilidad y de Necesidad

El interés de esta acumulación consiste en que , además de ser un método operativo, es capaz de mostrar al final del proceso el total de la información disponible sobre cada uno de los elementos del álgebra de enunciados, con lo cual es posible enunciar nuevas proposiciones verdaderas o calcular la valoración exacta o aproximada de estas nuevas proposiciones.

En este trabajo se realiza la aplicación del método a un ejemplo sencillo de razonamiento numérico que se encuentra , entre otras, en [7]. En este ejemplo se comprueba que con la acumulación de la información disponible se consigue un conocimiento numérico, aunque aproximado, superior al precisado para la deducción del resultado buscado y que, por tanto, es posible por una parte precisar más el resultado y por otra realizar valoraciones de otros enunciados.

La cuestión que se plantea es la siguiente:

Sean A, B y C predicados clásicos sobre un universo de discurso U, finito. Supongamos que se conocen los datos:

Por una parte : pA son B , qB son C
y, por otra: p'B son A , q'C son B

Donde p,q,p',q' son números en [0 1], por ejemplo porcentajes

La pregunta es : *¿qué puede concluirse en la forma* :
rA son C, r'C son A?

La solución que aparece en [7] es la siguiente:

TEOREMA:
Se cumple:

$$p/p' \text{ Max } (0,p'+q-1) \le r \le \min(1,1-p+q\cdot p/p') \quad (1)$$
$$q'/q \text{ Max } (0,q+p'-1) \le r' \le \min 1,\ 1-q'+p\cdot q'/q) \quad (2)$$

y la demostración es aritmética, después de expresar convenientemente el predicado "A y C"

2. EXPLICACIÓN DEL MÉTODO QUE SE PROPONE

El método que aquí se quiere aplicar es el siguiente:

La *información completa* referente a la situación que nos ocupa consistiría en conocer:

la parte del total de elementos de A que son de B y de C
la parte del total de elementos de A que son de B pero no son de C
la parte del total de elementos de A que no son de B y son de C
la parte del total de elementos de B que no son de A y son de B
etc.

Es decir, información numérica referente a cada uno de los enunciados que hemos colocado en un diagrama para mayor capacidad de síntesis

Tabla 1

$\neg A \wedge \neg B \wedge \neg C$	$\neg A \wedge B \wedge \neg C$
$\neg A \wedge \neg B \wedge C$	$\neg A \wedge B \wedge C$
$A \wedge \neg B \wedge C$	$A \wedge B \wedge C$
$A \wedge \neg B \wedge \neg C$	$A \wedge B \wedge \neg C$

Pero el problema es que esta información en general *no está disponible*, sinó que sólo poseemos los datos numéricos de algunas de las casillas de la tabla 1,o del total de un grupo de ellas.
Por ejemplo, en el caso que nos acupa , *disponemos de la siguiente información*:

Si llamamos $|A|$ al cardinal del conjunto A y analogamente para los otros conjuntos ; si representamos por el símbolo * la ausencia de información (sea o no deducible por el método que sea)

Tabla 2

*	$(1-p')\,\lvert B\rvert$
*	
*	*
*	*

Tabla 3

*	*
$(1-q')\,\lvert C\rvert$	$q\,\lvert B\rvert = q'\,\lvert C\rvert$
*	*

Tabla 4

*	*
*	*
$(1-p)\,\lvert A\rvert$	$p\,\lvert A\rvert = p'\,\lvert B\rvert$

Cualquier afirmación sobre las proporciones cuantitativas en cada casilla que no produzca contradicciones al acumularla a los datos es *una situación posible*

Lo que queremos averiguar es precisamente el máximo y el mínimo posible de la suma $x + y$ en una afirmación como la siguiente:

Tabla 5

*	*
*	*
$y\,\lvert A\rvert$	$x\,\lvert A\rvert$
*	*

De forma que la acumulación con la información disponible no produzca contradicciones

Tal como se expone en [2], *si tenemos información sobre un conjunto de n casillas y la acumulamos con información no contradictoria sobre un subconjunto de m casillas, se produce información sobre el subconjunto de n – m casillas restantes*

Así, si acumulamos toda la información disponible (expresada en las tablas 2,3 y 4) con el resultado incógnita (expresado en la tabla 5) obtendríamos:

Tabla 6

*	$(1-q)\|B\| - p\|A\| + x\|A\|$
$(1-q')\|C\| - y\|A\|$	$Q\|B\| - x\|A\|$
$y\|A\|$	$X\|A\|$
$(1-p)\|A\| - y\|A\|$	$P\|A\| - x\|A\|$

Las condiciones para que el resultado de la acumulación sean *vàlidos* , es decir *no contradictorios* es que los resultados obtenidos en cada una de la casillas no sean negativos, es decir :

$$x|A| \geq 0 \quad (3)$$
$$q|B| - x|A| \geq 0 \quad (4)$$
$$p|A| - x|A| \geq 0 \quad (5)$$
$$(1-q)|B| - p|A| + x|A| \geq 0 \quad (6)$$

que puede expresarse:

$$\max(0, p|A| - (1-q)|B|) \leq x|A| \leq \min(p|A|, q|B|) \quad (7)$$

$$0 \leq y|A| \leq \min((1-p)|A|, (1-q')|C|) \quad (8)$$

Este resultado es más fuerte que la tesis del teorema enunciado, expresado en (1) ,ya que si sumamos miembro a miembro las desigualdades (7) y (8), y teniendo en cuenta que:

$$B| = p/p'|A| \; ; \; |C| = q/q'|B| \quad (9)$$

obtenemos una cota inferior para x + y igual precisamente a max (0, p/p'(p' + q − 1)), mientras que obtenemos una cota superior menor o igual que min(p + 1 − p , 1 − p + + q · p/p')

Analogamente, si acumulamos la información en función de la parte del total de elementos de C, obtenemos:

Tabla 7

*	$(1-q)\|B\| - -p'\|B\| + +x'\|C\|$
$(1-q')\|C\| - -y'\|C\|$	$q'\|C\| - -x'\|C\|$
$Y'\|C\|$	$x'\|B\|$
$(1-p)\|A\| - -y'\|C\|$	$p'\|B\| - -x'\|C\|$

Y la condición para la ausencia de contradicción es la no negatividad en cada casilla, con lo cual obtenemos :

$$\text{Max}(0, q'/q\,(p'-1+q)|C|) \leq x'|C| \leq \leq \min(q'|C|, p'q'/q|C|) \quad (10)$$

$$0 \leq y'|C| \leq \min((1-q')|C|, p'q'/pq\,(1-p)|C|) \quad (11)$$

lo cual es más fuerte también que lo precisado para demostrar la tesis (2) del teorema propuesto

Además, hemos obtenido acotaciones para cada una de las cantidades x,y,x', e y' por separado y , sustituyendo valores obtendriamos facilmente acotaciones para cada casilla.

3. EJEMPLOS:

3.1.- Supongamos los datos:

1/5 parte de elementos de A son de B y una ½ parte de elementos de B son de C, por otra parte, sabemos que 5/12 parte de B son de A y que 6/32 parte de C son de B

Entonces :

$$\max(0,\ 1/5/5/12(5/12 - 1 + ½) \leq x + y \leq \leq \min(1, 1 - 1/5 + 1/2 \cdot 1/5/5/12) \quad (12)$$

Es decir,

$$0 \leq x + y \leq 1 \quad (13)$$

Sin embargo, aplicando (1) y (2), resulta, respectivamente:

$$0 \leq x \leq \min(1/5,\ ½ \cdot 1/5/5/12) = 1/5 \quad (14)$$

$$0 \leq y \leq \min((1 - 1/5), (1 - 6/32)\ ½ \cdot 1/5 / 6/32 \cdot 5/12) = = 4/5 \quad (15)$$

cuyas acotaciones determinan(tomando por ejemplo, $|A| = 25$) las situaciones extremas expresadas en las tablas 8 y 9:

Tabla 8

*	1
26	6
0	0
20	5

Tabla 9

*	6
6	1
20	5
0	0

Siendo posible cualquier situación intermedia, como:

Tabla 10

*	3
8	4
18	2
2	3

Pero no del tipo:

Tabla 11

*	6
6	0
17	6
2	0

A pesar de que se cumple: $0 \le x + y \le 1$

3.2.- Ejemplo:

Supongamos que 1/5 de elementos de A son de B, ½ de elementos de B son de C, que 5/28 elementos de B son de A y que 14/16 de C son de B. (y que tomamos el número de elementos de A es de 25)
Entonces:

$$0 = \text{Max}(0, 1/5/5/28 \cdot (5/28 + ½ - 1) \le x + y \le$$
$$\le \min(1, 1 - 1/5 + ½ \cdot 1/5/5/28) = 1 \quad (16)$$

Pero:

$$0 = \text{Max}(0, 1/5/5/28 \cdot (5/28 + ½ - 1) \le x \le$$
$$\le \min(1/5, 1/5 \cdot 1/2/5/28) = 1/5 \quad (17)$$

$$0 \le y \le$$
$$\le \min(4/5, (1 - 14/16) \cdot 1/5 \cdot 1/2/(5/28 \cdot 14/16)) = 2/25 \quad (18)$$

que detemina las siguientes situaciones extremas:

Tabla 12

*	9
2	14
0	0
20	5

Tabla 13

*	14
0	9
2	5
18	0

Y no es posible :

Tabla 14

*	14
–2	9
4	5
16	0

Aunque sigue cumpliendo $0 \le x + y \le 1$

3.3.- Ejemplo: (propuesto en [7])

El 90% de los estudiantes son jóvenes; 25% de los jóvenes son estudiantes
El 90% de los jóvenes son solteros; 60% de los solteros son jóvenes

En [7]se obtiene:

Al menos el 54% de los estudiantes son solteros
El porcentaje de solteros que son estudiantes varia entre el 10% y el 56'7%

Mientras que con la acumulación se obtiene, además de los anteriores, resultados como:

Al menos el 54% de los estudiantes son jóvenes y solteros
Como màximo el número de estudiantes jóvenes que son solteros es de 90%

Como máximo el número de estudiantes que son solteros pero no jóvenes es del 10%
Etc.

5. CONCLUSIONES

El método que se ha aplicado en este trabajo no es una demostración en sentido clásico de un resultado propuesto, sinó que se realiza una acumulación de la información disponible independientemente de los resultados buscados y , posteriormente se estudia la veracidad del resultado propuesto acumulàndolo a la información disponible.
Además, la acumulación de la información disponible permite estudiar la veracidad de cualquier otro resultado que se proponga.
La aplicación de un método que estaba formulado para un tipo de resultados verdadero-falso de enunciados de la propia álgebra al razonamiento numérico permite albergar la opinión de que el método es extensible a un razonamiento que utilice proposiciones borrosas

Agradecimientos

A los doctores Josep Miró y Gaspar Mayor de la Universidad de les Illes Balears por sus conversaciones y respaldo

Referencias

[1]. N.D. Belnap,Jr. *A useful four – valued logic.* In G.Epstein, editor, Modern Uses of Multiple Valued Logic, pages 8 – 49. Boston ,MA ,1977

[2].J.Casasnovas. *Contribución a una formalización de la Inferencia Directa.* Ph.D. Thesis, Universitat de les Illes Balears, Palma de Mallorca.Spain, 1989

[3]. J.Casasnovas y J.Miró. *Accumulation and Inference over Finite-Generated Algebras for Mapping Approximations.* Proceedings of IPMU'92, 149 - 157

[4]. J.Casasnovas. *A Deduction Rule for the Approximated Knowledge of a Mapping.* Proceedings of IPMU'92, 314 - 320

[5]. J.Casasnovas and J.Rifà:.Inference of Approximate Values of a Possibility Measure by Database Accumulation..Fifth Intern.Sym. on Knowledge Engineering.Proceedings.216-219,Sevilla1992

[6]. Driankov D.: *Towards a Many-Valued Logic of quantified belief: the Information Lattice.*International Journal of Intelligent Systems, 6:135 –166,1991

[7]. E.Trillas,C.Alsina,JMTerricabras: *Introducción a la Lógica Borrosa.Ariel Matemàtica.* Barcelona 1995

Índice de Autores

Acevedo, M. I. 225
Agustench, E. 85
Al-Hadithi, B. M. 303
Albertos, P. 309
Aranguren, G. 425
Arregi, G. 449

Barriga, A. 437
Barro, S. 3,63,109
Barrón, M. 425
Baturone, Y. 437
Benítez, J. M. 205,413
Bernad, P. E. 327
Boixader, D. 47,151
Bugarín, A. 109
Burillo, P. 41,361,373
Bustince, H. 85

Cadenas, J. M. 367
Calvo, T. 163
Campo, C. del 239
Canet, P. 171
Cánovas, J. 431
Carbonell, M. 163
Cariñena, M. P. 109
Carmona, P. 285
Casasnovas, J. 419
Castiñeira, E. 53
Castro, J. L. 15,77,205,383
Castro-Sánchez, J. J. 383
Comino, M. A. 91
Cordón, O. 199,399
Couso, I. 279
Cubillo, S. 53,137,239
Chavarría, A. 425

Delgado, M. 77,97,193
Deschamps, J. P. 297
Díaz, Y. 125

Elorza, J. 41
Esteva, F. 339
Félix, P. 63
Fernández, M. 297
Fernández, S. 117
Flores-Sintas, A. 431
Fortes, Y. 91
Fraga, S. 63
Frago, N. 373
Fuente, D. de la 211
Fuentes, R. 373

García, C. J. 225
García, J. 431
García, P. 347,443
Garmendia, L. 57,361
Garrido, M. C. 367
Gil, P. 157,279
Giménez, E. 347
Godo, L. 185,339,405
Gómez-Skarmeta, A. F. 97,271,321,443
González, A. 391
González, H. 225
Gutiérrez, J. 327

Herencia, J. A. 259
Hernández, E. 103
Herrera , E. 333
Herrera, F. 199,333,399

Ibañez, J. 321
Isassi, P. 131

Jacas, J. 47,151
Jesús , M. J. del 399
Jiménez, A. 303
Jiménez, C. J. 437
Jiménez, F. 321
Junco, L. 315

Kacprzyk, J. 143

Lamata, M. T. 259,265
León, T. 71
Liern, V. 71
Linares, L. J. 97
López de Arroyabe , J. 425
López, E. 413
López, F. J. 225
Lubosny, Z. 291

Macías, M. 225
Mantas, C. J. 77
Marín, R. 63
Martín, J. 355
Martín, M. J. 131
Martínez, H. 443
Martínez, J. I. 297

Martínez, L. 333
Mas, M. 163
Matía, F. 303
Mayor, G. 171,355
Miranda, P. 157
Mohedano, V. 85
Molina, J. M. 125,131
Montes, S. 279
Moraga, C. 21
Morales, R. 91

Navas-González, R. 217

Peláez, J. Y. 265
Pérez, J. L. 91
Pérez, R. 391
Pino, R. 211
Pradera, A. 137
Puente, J. 211

Recasens, J. 47,103,151,245
Requena, Y. 205,413
Rodríguez, A. 217
Ruiz, A. 367

Sala, A. 309
Salvador, A. 57,361
Salvador, L. de 327
Sánchez, D. 193,271
Sánchez, L. 315
Sánchez, S. 437
Sanchis, A. 131
Sancho-Royo, A. 179
Sobrino, A. 23,117
Soto, A. R. de 233,245
Suñer, J. 171
Szmidt, E. 143

Tiliouine, H. 291
Torra, V. 185,251
Torrens, J. 163
Triguero, F. 91
Trillas, E. 31,53,57,137,233,239

Uribe, J. P. 449

Velasco, M. 125
Verdegay, J. L. 179
Vidal, F. 217
Vila, M. A. 193

Zabala, J. 449
Zapico, A. 405
Zurita, J. M. 285,383