TECNOLOGIAS
Y
LOGICA FUZZY

Actas del Congreso ESTYLF'95

V Congreso Español sobre Tecnología y Lógica Fuzzy

estylf 95

Organiza :



Dpto. Informática y Sistemas
Facultad de Informática
Universidad de Murcia

Auspiciado por la Asociación Española de Tecnología
y Lógica Fuzzy, y la International Fuzzy Systems Association (IFSA)

IFSA

Tecnologías y Lógica Fuzzy

Editores:
Roque Marín Morales
Jose Manuel Cadenas Figueredo
Antonio F. Gómez Skarmeta
Alberto Ruíz García

Actas del V Congreso Español
sobre Tecnologías y Lógica Fuzzy
(ESTYLF'95)

20, 21, y 22 de Septiembre de 1995
Dpto. Informática y Sistemas.
Facultad de Informática
Universidad de Murcia
Espinardo - Murcia

Entidades Patrocinadoras

Instituto de Fomento de la R. de Murcia
Universidad de Murcia
DGICYT
General Electric Plastics, S.A.
Repsol Petróleo, S.A.
CajaMurcia
INTI, S.A.

Entidades Colaboradoras

Logic Murcia, S.L.
Excmo. Ayuntamiento de Murcia
Consejería de Turismo de la R. de Murcia
Cámara de Comercio de Murcia
Euronova Convenciones
Pedro Domecq, S.A.

Auspiciado por:

Asociación Española de Tecnologías y Lógica Fuzzy
International Fuzzy Systems Association

© Servicio de Publicaciones
Universidad de Murcia
I.S.B.N. 84-7684-625-8
ISSN: En trámite
Depósito Legal: MU-1215-1995

Indice

Prefacio v

Comité de Programa y Organizador vii

COMUNICACIONES

Aplicaciones

Contribuciones al procesado simbólico de segmentos en visión por computador 3
Cabrera, J.; Falcón, A.; Hernández, F.; Méndez, J.

Recuperación de errores de segmentación y clasificación utilizando sistemas deformados 9
Villadangos, J.; Sanz, A.; González de Mendivil, J.R.; Fariña, F.

Aplicación de la lógica borrosa al diagnóstico de maquinaria rotativa 15
Fernández, J.; De Miguel, L.J.; Perán, J.R.

Arquitectura cliente/servidor para bases de datos relacionales difusas 23
Medina, J.M.; Cubero, J.C.; Pons, O.; Vila, M.A.

Una aproximación sistemática al tratamiento de conceptos y cantidades difusas 29
Cadenas, J.M.; Gómez, A.; Camacho, A.; Fernández, L.; Martín, F.

Aplicación de operadores estructurales a la segmentación de imágenes aéreas 35
Puig, D.; Montseny, E.; Sobrevilla, P.

Aprendizaje y Optimización

Un enfoque aproximativo para la generación de reglas mediante análisis cluster 43
Delgado, M.; Gómez, A.; Martín, F.

Un modelo borroso de un sistema observado de forma inexacta 49
Sánchez, L.; Otero, A.; Tuya, J.

Métodos neuronales para la extracción de reglas difusas 55
Benítez, J.M.; Blanco, A.; Delgado, M.; Requena, I.

Aproximación global en la optimización de un proceso de información/aprendizaje sujeto a incertidumbre ... 61
Ferreiro, R.; Yañez, V.

Dualidad en programación lineal fuzzy 65
Cadenas, J.M.; Jiménez, F.

Resolución de un programa lineal difuso mediante la teoría de la posibilidad 71
Arenas, M.; Bilbao, A.; Rodrígez, M.; Jiménez, M.

Teoría

La desigualdad como medida de variación de una variable aleatoria difusa 79
Corral, N.; Gil, M A.; López, H.

Integración difusa sobre espacios producto 83
Suarez, E.

Experimentación numérica y comparación de integrales difusas 89
Bolaños, M.J.; Hernández, L.D.; Salmeron , A.

Las ϵ-particiones difusas 95
Montes, S.; López, M J.; Gil, P.

Indistingibilidad y funciones de analogía 99
Valverde, L.

Clases de subconjuntos borrosos con el mismo condicional material 103
Trillas, E.; De Soto, A.R.; Cubillo, S.

Funciones de agregación extendidas 107
Mayor, G.

Retículos de conceptos asociados a W-contextos L-fuzzy 111
Burusco, A.; Fuentes, R.

Relaciones extendidas por agregación y principio de extensión 117
González, L.; Marín, A.

Construcción de correspondencias borrosas entre subconjuntos 123
González, L.; Marín, A.

Gramáticas de estructuras de frase y expresividad de los conceptos vagos 129
Sobrino, A.; Olivas, J.A.

El lenguaje de programación borrosa XL. 135
Morales, R.; Pérez, J.L.; Conejo, R.; Clares, B.

Caracterización mediante programas de ciertas clases de funciones difusas 141
Pérez, J.L.; Morales, R.; Conejo, R.; Clares, B.

Métodos intuicionistas de inferencia: Comparación por compatibilidades 147
Burillo, P.; Bustince, H.; Mohedano, V.

Un análisis modal de la lógica de la similitud 153
Esteva, F.; García, P.; Godo, L.

Una contribución a la aproximación lingüística utilizando prototipos y supervaloraciones 159
Fernández, S.; Del Campo, C.; Sobrino, A.

Ideales T-S difusos. Análisis de sus propiedades 165
Garmendia, A.; Molero, M.; Salvador,A.

Ideal semiprimo T-S difuso. Nil-Radical T-S difuso 171
Garmendia, L.; Molero, M.; Salvador,A.

Comparación de definiciones de antisimetría fuzzy. Propiedades 177
Eraso, M.L.; Portilla, M.I.

Control

Fuzzy logic supervisory control of induction motor under full range of operating conditions 185
Brdys, M.A., Quevedo, J.; Sim, W.L.

Controlador borroso con aprendizaje mediante la evolución de su base de conocimiento 193
Magdalena, L.; Monasterio, F.

La lógica borrosa en control de sistemas autónomos inteligentes 199
Matía ,F.; Moraleda, E.; Puente, E.A.; Escriche, M.

Lógica difusa a bordo de un velero 205
Abril, J.; Salom, J.; Calvo, O.

Controlador fuzzy optimizado mediante algoritmos genéticos para el seguimiento de paredes en un robot móvil 211
Uribe, J.P.; Mujika, J.; Braunstingl, R.

Supervisor de la distancia de seguridad en un vehículo basado en lógica borrosa 217
Ferreiro, R.; Juanes, R.; Sánchez, J.C.; López, J.; Del Val, D.

Integración de controladores difusos en un robot autónomo 221
Matellán, V.; Molina, J.M.; Fernández, C.; Blázquez, J.M.

Control fuzzy de la velocidad en un sistema ferroviario a escala 227
Bandera, G.; Vicente, J.M.

Hardware

Una filosofía novedosa para la realización de circuitos integrados analógicos para lógica fuzzy 235
Castillo, F.; Puyo, I.

Aprendizaje supervisado de reglas en un controlador difuso analógico 239
Vidal, F.; Rodríguez, A

Memoria asociativa difusa para reconocimiento de modelos 245
Reyes, M.; Valderrama, E.; Ullastre,S.

Metodología para la materialización de algoritmos borrosos: de una especificación de alto nivel a un ASIC . 251
Deschamps, J.P.; Martínez, J.I.

Sobre las prestaciones de los procesadores de información borrosa 257
Gutiérrez, J.; De Salvador, L.

Síntesis automática de sistemas de control basados en lógica difusa 263
Jiménez, C.J.; Galán D.; Barriga, A.; Sánchez, S.

Metodología

Synthesis of information in KBS validation 271
Torra, V.

Revising and updating a knowledge base by means of fuzzy similarity relations 277
Rodríguez, R.; García, P; Godo, L.

Diseño de un ATMS genérico: casos multivaluado y difuso 281
Castro, J.L.; Zurita, J.M.

Representación de bases de conocimiento borroso sobre redes de Petri: Operación en el dominio de la verdad 287
Bugarín, A.; Barros, S.

Relaciones de igualdad borrosas: Aplicaciones a un sistema "fuzzy" para CAGD 295
Jacas, J.; Monreal, A.; Recasens, J.

Asignación básica de probabilidad difusa 299
Herencia, J.A.; Lamata, M. T.

Una aplicación del concepto de prototipo deformable a un método que gestiona incertidumbre en un SBC ..303
Olivas, J.A.; Sobrino, A.

Medidas lingüísticas para el consenso en grupo 309
Herrera, F.; Herrera Viedma, E.; Verdegay, J.L.

Obtención de soluciones en redes de restricciones borrosas 315
Cardenas, M A.; Marín, R.; Balsa, M.; Sánchez, J.L.; Navarrete, I.

Patrones temporales borrosos en la supervisión de pacientes 321
Félix, P.; Barros, S.; Marín, R.; Taboada, M.J.; Engel, A.

Las redes neuronales artificiales son sistemas difusos aditivos 327
Requena, I.; Castro, J.L.; Benítez, J.M.

Adquisición de conocimiento basado en redes de restricciones borrosas: Aplicación en agricultura 331
Túnez,S.; Marín,R.; Bosch,A.; Bienvenido,F.; Del águila,L.; Taboada,M.

INDICE DE AUTORES **337**

Prefacio

Este volumen recopila las comunicaciones presentadas al V Congreso Español sobre Tecnologías y Lógica Fuzzy. El número y la calidad científica de los trabajos aquí incluidos dan fe del interés que la lógica fuzzy y sus aplicaciones despiertan en nuestro país, y del papel relevante que los investigadores españoles desempeñan en el desarrollo del campo. Ciertamente, el estado actual de la I+D fuzzy en España es fruto del trabajo pionero que, desde la década de los 60, vienen realizando grupos españoles de reconocido prestigio.

Las sucesivas ediciones del Congreso han reflejado este crecimiento incesante, y han contribuido en gran medida a su aceleración y consolidación. Precisamente, la experiencia acumulada desde la primera celebración del Congreso, y el éxito cosechado por sus anteriores organizadores, constituyen la plataforma privilegiada de la que partimos. En la tradición de las anteriores ediciones, el V Congreso Español sobre Tecnologías y Lógica Fuzzy tiene por objetivos promover la investigación en estos temas y proporcionar un foro para la discusión y presentación de nuevas ideas y resultados, reuniendo a expertos en teoría y aplicaciones con jóvenes investigadores que inician su andadura en este campo. Como es habitual, el Congreso está auspiciado por la Asociación Española de Tecnologías y Lógica Fuzzy. Este año, además, el evento tiene el reconocimiento oficial de la International Fuzzy Systems Association (IFSA).

Asumimos también la continuidad en la labor de promoción de la interacción con el mundo empresarial e industrial, que ya se acometió en las últimas ediciones del Congreso. Así, un objetivo particular de esta nueva celebración es explorar el potencial existente en transferencia de tecnología fuzzy hacia la sociedad, analizando el impacto de la lógica fuzzy en el área industrial y las posibles formas de cooperación que se abren. Con este fin, se celebran este año las I Jornadas sobre Transferencia de Tecnología Fuzzy. Las actividades programadas incluyen conferencias, sesiones invitadas, mesas redondas y demostraciones de hardware y software. Las Jornadas pretenden proporcionar una vía de apertura hacia las empresas e industrias y facilitar la realización de contactos.

En 1995 se cumplen exactamente 30 años de la publicación del histórico trabajo del Prof. Zadeh alumbrando la teoría de los Fuzzy Sets. Desde entonces, el progreso en fundamentos teóricos y en metodología ha sido notable, con una muy digna contribución desde nuestro país. Notable ha sido también el profundo impacto de la tecnología y las aplicaciones fuzzy sobre el sector comercial, empresarial e industrial en paises como Japón o Estados Unidos, y en ciertos paises europeos. El potencial existente en nuestro país debería tener también su reflejo en los correspondientes sectores productivos españoles. Esperamos que el Congreso y las Jornadas contribuyan a acelerar este proceso y a preparar el terreno para iniciativas más ambiciosas.

Los editores queremos manifestar nuestro agradecimiento a todos aquellos que, con su contribución generosa y desinteresada han hecho posible la realización de este Congreso. Reciban nuestra más especial gratitud los conferenciantes y ponentes invitados, así como los investigadores que han contribuido con sus valiosos trabajos a que este libro sea una realidad.

Asímismo, queremos expresar nuestro agradecimiento al Comité de Programa por haber realizado las revisiones de los trabajos con extraordinaria eficacia, prontitud y rigor científico, y a los

miembros del Comité Organizador por su abnegada labor. Nuestra gratitud se extiende a la junta directiva de la Asociación Española de Tecnologías y Lógica Fuzzy, a la International Fuzzy Systems Association, y a los Vicerrectorados de Investigación, de Extensión Universitaria, y de Economía y Finanzas de la Universidad de Murcia. Es necesario agradecer también la colaboración de las siguientes entidades y empresas, que patrocinan el Congreso: Departamento de Informática y Sistemas, Facultad de Informática de la Universidad de Murcia, DGICYT, Excelentísimo Ayuntamiento de Murcia, Logic Murcia, S.L.,Repsol Petróleo, S.A., CajaMurcia, Consejería de Turismo de la Región de Murcia, Cámara de Comercio de Murcia, Pedro Domecq, S.A., INTI y Convenciones Euronova. Por último, agradecemos la colaboración de la Fundación Universidad-Empresa de Murcia en la organización de las Jornadas, y la generosa aportación de los patrocinadores de las Jornadas: Instituto de Fomento de la Región de Murcia, General Electric Plastics, S.A., Repsol Petróleo, S.A. y CajaMurcia.

Roque Marín
José Manuel Cadenas
Antonio Gómez Skarmeta
Alberto Ruíz

Comité de Programa

Dr. Josep Aguilar ..... *LAAS-CNRS*
Dr. Pedro Albertos ..... *U. P. de Valencia*
Dra. María C. García-Alegre ..... *IAI-CSIC*
Dr. Senén Barro ..... *U. de Santiago*
Dr. José Manuel Cadenas ..... *U. de Murcia*
Dr. Juan Luis Castro ..... *U. de Granada*
Dr. Miguel Delgado ..... *U. de Granada*
Dr. Francesc Esteva ..... *IIIA-CSIC*
Dr. Julio Gutiérrez ..... *U. P. de Madrid*
Dr. Jose Luis Huertas ..... *U. de Sevilla*
Dr. Joan Jacas ..... *U.P. de Catalunya*
Dr. Ramón López de Mántaras ..... *IIIA-CSIC*
Dr. Gaspar Mayor ..... *U. de les Illes Balears*
Dr. José Mira ..... *UNED*
Dr. Ramón Ruiz ..... *U. de Murcia*
Dra. María de los Angeles Gil ..... *U. de Oviedo*
Dr. Alejandro Sobrino ..... *U. de Santiago*
Dr. Enric Trillas ..... *U. P. de Madrid*
Dr. Llorenç Valverde ..... *U. de les Illes Balears*
Dr. José Luis Verdegay ..... *U. de Granada*
Dra. Amparo Vila ..... *U. de Granada*

Comité Organizador

D. Roque Marín (Coordinador)
D. Fernando Martín
D. Ramón Ruiz
D. José Manuel Cadenas
D. Antonio Gómez
D. Fernando Jiménez
D. Luis Daniel Hernández
D. Alberto Ruiz

Aplicaciones

Contribuciones al Procesado Simbólico de Segmentos en Visión por Computador*

Jorge Cabrera-Gámez †
e-mail: jcabrera@dis.ulpgc.es

Antonio Falcón Martel †
e-mail: afalcon@dis.ulpgc.es

Francisco M. Hernández Tejera †
e-mail: fhernandez@dis.ulpgc.es

Juan Méndez Rodríguez †
e-mail: jmendez@dis.ulpgc.es

Resumen

Se presenta una metodología de proceso a nivel de segmentos en el contexto de los sistemas de visión basados en conocimiento. Se muestra la utilización de conceptos de Lógica Difusa en las operaciones que tienen lugar en dicho procesador y para establecer el paso entre los dominios de representación numérico y simbólico.

Palabras clave: *Sistemas de Visión, Sistemas Basados en Conocimiento, Segmentación de Imágenes.*

1 Introducción.

La Visión por Computador está dedicada al estudio de teorías y al desarrollo de métodos y algoritmos en sistemas de percepción visual artificial. En los últimos años se ha producido un cambio sustantivo en las metodologías de diseño y desarrollo de tales sistemas, por un lado, debido a la integración de tecnologías y metodos de la Ingeniería del Conocimiento y, por otro lado, en la consideración básica de que un sistema de visión no es un elemento pasivo o aislado que recibe información y la procesa en algún sentido, sino mas bién que forma parte indisoluble con un entorno concreto y que, para su estudio, es necesario la consideración del sistema interactuando con el medio mediante un proceso de percepción-acción [1], [2]. Además, la evolución se manifiesta en los planteamientos de resolución del compromiso de la flexibilidad o adaptabilidad de los sistemas desarrollados respecto a las modificaciones del entorno frente a la calidad o robustez de los resultados. Por otra parte, asi como en muchas áreas de la Ciencia y de la Tecnología se tratan problemas para los que se dispone de modelos útiles y válidos que proporcionan resultados concretos, en Visión por Computador los problemas carecen de modelos plenamente válidos, salvo en circunstancias muy concretas. La resolución de tales problemas requiere en muchos casos de la utilización de procedimientos con múltiples heurísticas y la integración de resultados de hipótesis de validez parcial con distintos grados y tipos de incertidumbres Dicha integración puede hacerse de manera adecuada utilizando metodos de Inteligencia Artificial y Conjuntos Fuzzy para generar conclusiones y resultados.

Los sistemas de visión por computador involucran procesos que actúan sobre elementos de diferente tipología, esto es, a bajo nivel o pixels, a nivel intermedio o segmentos y a alto nivel u objetos. En cada uno de los niveles se definen categorías simbólicas de objetos y en ellas aparecen, en general incertidumbre. Particularmente, la segmentación de imágenes puede considerarse como un proceso de simbolización, entendida en el contexto de lo que asi se considera en Reconocimiento de Formas o en Visión por Ordenador [3]. En este caso, el grano de información que se maneja es el segmento o agregación de pixels, definiendo un determinado nivel de representación en intenso de mayor complejidad que el nivel de pixels pero de inferior del de objetos o estructuras relacionales. En este contexto, un segmento de una escena puede pasar de una descripción numérica a una simbólica con ciertos coeficientes de certeza y es aquí donde las técnicas de la Lógica Fuzzy permiten una aproximación práctica plausible [4]. Además en este nivel la definición de categorías visuales no es dura sino mas bien de variación suave [5]. De otro lado, las particularidades de los procesos de refinamiento que se producen en una segmentación, donde las condiciones que afectan a cada segmento juegan un papel importante en el resultado final, nos hacen plantearnos una via de estudio del proceso de evaluación de condiciones y de reglas de control dentro del marco de los Conjuntos Fuzzy. La incertidumbre se manifiesta en los diferentes niveles [6] y adquiere características específicas según sean estos (pixels, segmentos, objetos, ...) fundamentadas en las definiciones de clases y en las ambigüedades de las interpretaciones motivadas por preguntas mal planteadas desde niveles más altos. En este trabajo se expone la arquitectura que debería tener un procesador de segmentos que produzca particiones estables hacien-

* Este trabajo ha sido parcialmente financiado por la Consejería de Educación del Gobierno de Canarias (Proy. 93/069)
† Grupo de Inteligencia Artificial y Sistemas de la Universidad de Las Palmas de Gran Canaria

do uso de mecanismos de Ingeniería del Conocimiento y de Conjuntos Fuzzy, asi como el análisis de los elementos necesarios para el control en este nivel, todo ello dentro del marco de los Sistemas de Visión Basados en Conocimiento.

2 El Nivel de los Segmentos.

La organización en niveles es una herramienta que suele utilizarse con relativa frecuencia al tratar de analizar o sintetizar sistemas complejos. Asi, en el caso de análisis, el concepto de nivel responde a una forma de ordenar en jerarquías una realidad existente en hechos. Mientras en el caso de síntesis, constituye una forma de simplificar el proceso en base a construir capas de complejidad reducida. En un sentido no formal consideraremos a un nivel como un conjunto de símbolos y una serie de operadores sobre estos. En esencia, lo que define un nivel es la elección de un conjunto de símbolos que representan las entidades de su dominio y un álgebra o conjunto de formalismos u operadores que realizan las transformaciones de los símbolos.

En el nivel de los segmentos el objetivo es proporcionar la partición de la imagen en un conjunto de segmentos o agregaciones de pixels con sus respectivos grados de pertenencia a clases de segmentos ("cuadrado_rojo", "línea_recta", ...) y esto se establecerá desde un conjunto de transformaciones sobre los segmentos y sus símbolos asociados. Consideramos al segmento como una agregación conexa de pixels definida por criterios de forma o de uniformidad en propiedades. Para clarificar la terminología, denominaremos partición a la lista de segmentos que constituyen una teselación completa de la imagen y que incluye las asignaciones simbólicas de los segmentos a clases y las relaciones espaciales entre segmentos.

A partir de la aplicación de procedimientos de diferente naturaleza es posible hacer explícitas ciertas categorías o clases visuales relacionadas bien con los pixels o con agregaciones de estos. En general, la definición de estas categorías no es dura, sino más bien de variación suave o gradual [7],[5] y existe ambigüedad visual, bien a nivel de la definición geométrica de clases de segmentos o de la definición de categoría tonal de los mismos, o bien a nivel de la definición de las relaciones geométricas entre segmentos. Además de la citada aplicación directa de procedimientos, los diagnósticos simbólicos sobre los segmentos pueden obtenerse por combinación lógica de diagnósticos previos, donde también se manifiesta la incertidumbre.

Es necesario hacer notar que la utilización del segmento como grano de información en este nivel plantea dos dificultades que no se presentan en el nivel de los pixels. De una parte, la necesidad de obtener la definición espacial de los segmentos a partir de los mapas de diagnóstico o clases de pixels obtenidos desde niveles inferiores como paso previo a su descripción simbólica en términos de clases de segmentos. De otra, en el procesado de segmentos es muy interesante permitir que las entidades puedan sufrir cambios durante el proceso de refinamiento de la segmentación como consecuencia de la fusión o división de segmentos.

En este nivel, para los segmentos se consideran los siguientes elementos, que permiten el procesado simbólico:

a) *Propiedades* o Características, estas representan el valor numérico de una propiedad, $f_i(s)$, que se evalúa desde el segmento, s, en cuestión a través de los *Procedimientos* que son un tipo particular de *Operadores* de naturaleza numérica. Estas características definen el dominio numérico y reflejan ciertas cualidades del segmento en cuestión, bien en su aspecto morfológico (forma) o referida a propiedades intrínsecas (tonal) de los pixels que lo componen tales como brillo, color o textura.

b) *Operadores*, que además de los Procedimientos incluyen a los elementos destinados a la obtención de clases asociadas a cada segmento desde propiedades, o desde condiciones a partir de otras clases de segmentos. Los primeros son los *Clasificadores* que transforman en un dominio borroso las propiedades en diagnósticos o *Clases de Segmentos*. Existen otro tipo de operadores que están orientados a la definición de clases desde combinaciones, esto es las *Reglas*, que implican condiciones que verifican ciertas clases y cuya verificación en un dominio borroso se articula para la generación de nuevas clases.

c) *Clases de Segmentos*, que representan lo que podemos decir en un dominio simbólico acerca de la asignación de un segmento a una determinada clase o tipología de segmentos en un intervalo [0-100].

d) *Transformaciones*. Estas representan un tipo especial de operadores que afectan al dominio espacial más que al simbólico. Surgen de la necesidad de alterar la composición espacial de los segmentos al verificarse ciertas condiciones en un proceso de generación de la partición. Funcionalmente se articulan como *Reglas de Control* que, desde la verificación en un dominio borroso de ciertas condiciones, se producen acciones de control que modifican la partición actual para alcanzar otra que se acerque más a un cierto objetivo.

La estructura de clases de segmentos y su evaluación se define desde mapas de pixels expresados en forma simbólica, esto es imágenes simbólicas generadas por procesos de simbolización sobre mapas de características. En estos mapas de pixels el valor de cada pixel es el grado de pertenencia a una tipología o clase de pixel [3] representa la asignación de un pixel a una determinada clase de pixel. Con estos elementos, es posible realizar computación simbólica, dado que la representación se basa en clases y tipos que se pueden considerar como símbolos más que números. Como consecuencia se pueden utilizar operaciones relacionadas con jerarquías de clases y relaciones entre ellas tales como reglas o producciones. La utilización de grados de pertenencia a clases posibilita la utilización de clasificadores borrosos y coeficientes de certeza en las reglas.

Los procedimientos permiten la obtención de las características de los segmentos desde mapas de diag-

nósticos de pixels o desde otras características. En estos casos nos movemos en un dominio claramente numérico. Los clasificadores son básicamente funciones discriminantes y de decisión y aportan una mayor riqueza de significado. Utilizan características de tipo numérico para computar una de tipo simbólico en el rango[0-100]. En esencia este operador es una implementación de procedimientos de clasificación incluyendo decisiones borrosas mediante el cual se pueden implementar decisiones más complejas [8]. Sea $f_i(s)$, i=1,...,n, un conjunto de características evaluadas sobre el segmento s, $F(s)$ un funcional computado desde ellas y C(s) el resultado suministrado por el clasificador. Se tiene:

$$C(s) = D_M(F(s))$$

Donde $D_M(u)$ es una función de decisión. Esta definición queda abierta a distintos tipos de funcionales (unitarios, lineales, cuadráticos ...). Así por ejemplo el funcional cuadrático será de la forma:

$$F(s) = \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} \left[f_i(s) - a_i\right] b_{ij} \left[f_j(s) - a_j\right]$$

Donde $\{a_i\}$y $\{b_{ij}\}$son conjuntos de coeficientes numéricos. En este contexto son posibles distintos tipos de funciones de decisión, en general estos vendrán expresados en la forma $D_M(u; p_1,...,p_m)$, donde M representa el modelo (borroso, duro,...). Por ejemplo, si consideramos la función tipo-S de conjuntos borrosos definida por $S(u;\alpha,\beta)$, [9], la función de decisión $D_{SS}(u;\alpha,\beta)$, queda como:

$$D_{ss}(u;\alpha,\beta) = 100\, S(u;\alpha,\beta)$$

Donde los parámetros α y β definen el intervalo de borrosidad entre la decisión nula (0) y total (100). El esquema permite otros posibles modelos para las funciones de decisión [10].

La definición de una clase de segmentos puede establecerse desde clasificadores y características permitiendo la generación de símbolos asociados a los segmentos

Las reglas como operadores para obtener diagnósticos son construcciones lógicas, similares a las cláusulas de Horn. Una regla de este tipo define una contribución a una clase de segmentos basada en los valores de una serie de condiciones que pueden admitir cualificadores lógicos y operadores lingüísticos borrosos "very" y "more-or-less"[9],[11]. La incertidumbre se transforma y asigna desde las clases, clasificadores y reglas hacia otras clases. El valor intrínseco de cada clase se determina desde las fuentes de datos, tales como clasificadores y reglas. Si la clase posee particularizaciones, entonces el citado valor intrínseco debe normalizarse para obtener el valor final. Esta normalización está basada en considerar los casos particulares como conjuntos borrosos. La combinación de evidencias para definir una clase de segmentos desde clasificadores, reglas y particularizaciones utiliza el siguiente esquema:

Paso 1: Computar todos los clasificadores, casos particulares y clases incluidas en las condiciones de las reglas de una clase.

Paso 2: Computar todos los predicados en las condiciones de reglas.

Paso 3: Computar el resultado de la reglas, en base a los predicados y a la certeza de la misma. Si $C_{pk}(s)$, k=1,..., h, es la certeza de cada predicado, y C es la certeza de la regla, entonces el resultado de la regla, $C_r(s)$, es:

$$C_r(s) = \frac{C}{100} \underset{k=1..h}{Min}\left[C_{pk}(s)\right]$$

Paso 4: Combinar los resultados de clasificadores y reglas, proporcionando el valor intrínseco. El valor intrínseco $IC(s)$ se evalúa desde las certezas C_{Ik} (s), k=1,...,m, de los m clasificadores y reglas en conjunto. Esta evaluación se realiza incrementalmente en la forma siguiente:

$$IC(s) := 0; \text{for } k=1 \text{ to } m\ IC(s) := COM(IC(s), C_{Ik}(s));$$

La función $COM()$ se define como sigue según el tipo de combinación elegido, una propuesta es:

$$COM(u,v) = \begin{cases} u + v - \dfrac{uv}{100} & si\ \ Logic \\ 50\left[EV\left(\dfrac{u}{50} - 1, \dfrac{v}{50} - 1\right) + 1\right] & si\ \ Evidence \end{cases}$$

Donde $EV(a,b)$ es:

$$EV(a,b) = \begin{cases} a + b - ab & a \geq 0 \wedge b \geq 0 \\ a + b + ab & a \leq 0 \wedge b \leq 0 \\ \dfrac{a+b}{1 - min(|a|,|b|)} & (a > 0 \wedge b < 0) \vee (a < 0 \wedge b > 0) \end{cases}$$

La ley "Evidence" utiliza el dominio [-1,+1] y está basada en la Teoría de la Evidencia, en la forma en que es utilizada por algunos sistemas basados en conocimiento [14].

Paso 5: Normalizar el valor intrínseco utilizando los valores de los casos particulares. Sea $C(s)$ el valor de la clase, y $C_{Ik}(s)$, k=1,..,n, la certeza de los n casos particulares. La normalización es de la forma siguiente:

$$C(s) = Max\ (IC(s), \underset{k=1..n}{Max}[\,C_{Ik}(s)\,])$$

Con objeto de estudiar mejor como se realizan los procesos en el Nivel de Segmentos, se introduce el concepto de Procesador de Segmentos [13] que define la arquitectura de procesos en este nivel y que se incluye resumidamente a continuación. Sobre el se ha definido un lenguaje que permite el desarrollo de aplicaciones.

2.1 Arquitectura del Procesador

Desde el punto de vista funcional, la arquitectura del Procesador de Segmentos sobre la que hemos desarrollado el trabajo (figura 1), se organiza en tres grandes bloques con cometidos específicos. El primero está dedicado a la definición de la partición inicial y es el cometido del módulo de presegmentación integrado en el denominado procesador Bottom-Up (BU). Este módulo está concebido como una parte intercambiable dentro del Procesador de Segmentos, pues en principio son posibles diferentes aproximaciones a la hora de establecer una partición inicial o segmentación primaria de la imagen. El segundo tiene como cometido la definición de la estructura computacional de obtención de los diagnósticos de segmentos. Está basado en la distinción de un dominio numérico o de características y otro simbólico o de clases. El tercer bloque funcional se ubica en el denominado procesador Top-Down (TD) y gestiona la atención de peticiones de diagnósticos y el control del Procesador de Segmentos. Éste incluye la evaluación del estado de la partición desde un conjunto de reglas de control que pueden producir acciones que modifiquen la definición espacial de los segmentos.

La unidad BU del Procesador de Segmentos tiene dos cometidos fundamentales: la definición del nuevo conjunto de entidades de este nivel a partir de los símbolos definidos para el nivel anterior y la atribución simbólica de las entidades así definidas. El Procesador de Segmentos opera según dos fases diferenciadas. Una directa, ascendente, ejecutada por la unidad BU en la que se calculan los grados de pertenencia a clases de segmentos de aquellos que están presentes en la partición inicial. La otra queda bajo la supervisión del procesador TD y permite desarrollar estrategias que modifican la partición de una imagen desde un conjunto de reglas de control.

La segmentación inicial producida en el presegmentador se trata como un mapa de colores y se transforma posteriormente en una partición inicial mediante un análisis de componentes conectadas, que extrae del mapa de color información espacial exhaustiva que utiliza durante la descripción simbólica de los segmentos.

3 Las Condiciones.

Una *Condición* en el marco del Procesador de Segmentos se entiende como el encadenamiento de un conjunto de premisas conectadas por conjunciones lógicas tipo "And". Las condiciones así definidas se utilizan en el Procesador de Segmentos en los diversos tipos de reglas, tanto aquellas que definen la estructura de clases, como en reglas de control o en acciones de control. Las premisas utilizadas en este nivel se enriquecerán con la inclusión de relaciones espaciales. Para clarificar el ámbito de estas condiciones, detallamos a continuación la definición sintáctica de la premisa de una condición, que se establece simbólicamente como sigue:

premise::=prem_atom_true | prem_list
prem_atom_true::=Is (True, [qualifier1] x)
prem_list::= prem_atom (And prem_atom)
prem_atom::=Is(property (And property), [qualifier1] x)
property::=[qualifier2] [order] sclassname
qualifier1::=[neig][qualifier2][spatial]
neig::=All | Any
qualifier2::=[Not][linguistic]
linguistic::=Very | MoreLess
spatial::=Above|Below|Left|Right|Beside|InBeside|Neib|
ExNeib | InNeib | Contains | user_defined_localizer
order::= Gt (number) | Eq (number) | Lt (number)

Los cualificadores All y Any hacen referencia al entorno y operan en conjunción con los cualificadores de relaciones espaciales que analizaremos posteriormente. Se incluyen además relaciones de orden (Gt, Eq, Lt) que afectan al grado de la propiedad sobre la cual se establece el diagnóstico. Se permite asimismo la inclusión de listas de propiedades And-conectadas. La sintaxis de escritura de una premisa de esta naturaleza se fundamenta en la expresión elemental:

$$Is(\ property\ ,\ x)$$

Donde x denota el segmento foco de atención (cuya referencia puede omitirse, pero que mantendremos por claridad en la escritura) y property es el diagnóstico sobre el que se condiciona a x. Así la premisa:

$$Is(Green\ ,\ x)$$

Sería la expresión formal del hecho "El segmento foco de atención es verde", y la evaluación de esta premisa devolvería el grado en que el segmento foco de atención satisface el diagnóstico asociado a la clase "Green". El sentido de una premisa puede modificarse mediante el uso de los modificadores Not, Very y MoreLess. Es fundamental en el marco de este procesador establecer premisas sobre la vecindad del segmento foco de atención, para obtener una descripción del papel particular que juegan las relaciones espaciales. La sintaxis básica para incorporar relaciones espaciales es:

$$Is(\ property\ ,\ spatial\ \ x)$$

Donde spatial representa a un elemento del lenguaje que se asocia con una relación espacial de los tipos: Above, Below, Left, Right, Beside, Neib, InBeside, ExNeib, InNeib y Contains. Así la expresión :

$$Is(\ Green,\ \ Below\ x\ )$$

Se correspondería con la condición "Algún segmento de los que están debajo del foco de atención es verde". Por otra parte, algunos localizadores tienen acotado su ámbito de aplicación. Así, localizadores como Above, Below, Left, Right, Beside, ExNeib y Contains se evalúan sólo sobre los vecinos "externos" de un segmento dado. Por el contrario, los localizadores InBeside e InNeib producen la evaluación exclusivamente sobre los segmentos

contenidos dentro de uno dado, es decir, sobre sus vecinos "interiores". Finalmente, Neib hace que se evalúe la condición sobre cualquier vecino del segmento foco de atención. También resulta útil establecer la condición, no sobre algún vecino (opción de defecto) sino sobre todos los adyacentes, lo que se hará utilizando el modificador All . Así, si deseamos expresar condiciones como: "Todos los segmentos contenidos en el segmento foco de atención no son rojos" o " Todos los segmentos que no están debajo del foco de atención son muy cuadrados", tendríamos respectivamente las premisas:

Is(Not Red , All InNeib x)
Is(Very Square, All Not Bellow x)

La evaluación del predicado "A Above B" devuelve la medida, en el intervalo [0,100], en que el segmento A, vecino del B, se sitúa "por encima" de éste.

La utilización de predicados que incluyen relaciones espaciales entre segmentos permiten establecer también relaciones de orden, esto es comparaciones entre vecinos para una propiedad o diagnóstico dado. Las relaciones de orden permitidas son: Gt (Mayor), Eq (Igual) y Lt (Menor), cuya evaluación, establecida de manera difusa, trataremos a continuación.

Para ello, considérense dos segmentos A y B y sea SC un determinado diagnóstico en base al cual se desea establecer una relación de orden del tipo GT entre A y B, que designaremos por la expresión '$SC_A Gt SC_B$', siendo SC_A y SC_B los valores que toma el diagnóstico sobre A y B respectivamente. Sea u una variable definida como la diferencia entre estos dos valores y sea δ un parámetro indicativo de la semianchura del intervalo en u que separa un resultado verdadero (100) de otro falso (0). Entonces el juicio sobre "$SC_A Gt(\delta) SC_B$" se resuelve de acuerdo con:

$$R_{SC_A\, Gt(\delta)\, SC_B} = 100\, S(u;\ -\delta\ ,\ \delta)$$

El parámetro δ controla la incertidumbre en el juicio y es dependiente del diagnóstico utilizado para establecer la relación de orden. De esta forma será posible expresar hechos del tipo "Algún vecino del foco de atención es más negro con incertidumbre 10", como:

Is(Gt(10) Black , Neib x)

Donde *Black* es una clase o diagnóstico representativo de la clase segmentos negros. En análogo sentido se definen las evaluaciones de las relación del tipo "Eq" o "Lt".

Para la evaluación de las premisas se utiliza el esquema propuesto en [15], mientras la evaluación de una condición se realiza mediante una operación que devuelva el mínimo de los resultados obtenidos para cada una de las premisas que la integran.

Las condiciones también se utilizan en las reglas de control. Estas reglas de control, son diferentes de las utilizadas en la generación de clases de pixels o de segmentos y se estructuran a partir de una condición o lista de condiciones, una acción de control a ejecutar si se dispara la regla y, si se requiere, una condición adicional que actúa como selector de algún segmento vecino.

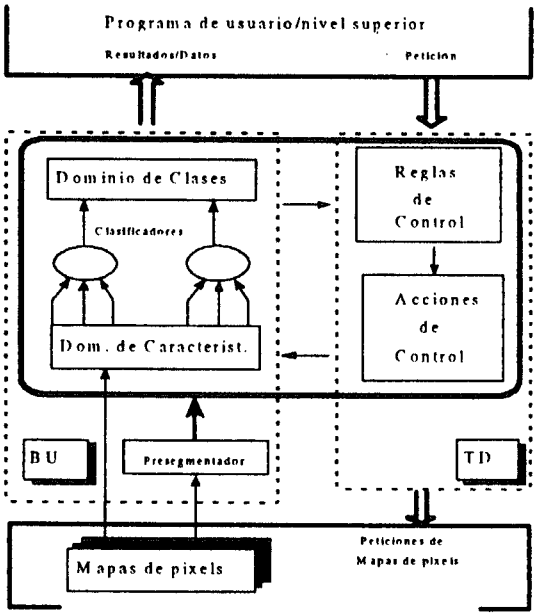


Figura 1. Estructura de procesadores y dominios en el nivel de los segmentos.

4 Aplicaciones.

Las aplicaciones han sido desarrolladas en el contexto del sistema SVEX [10]. Para este sistema se ha desarrollado un lenguaje de propósito específico orientado a objetos. Un programa escrito para este sistema se compone de módulos que afectan a cada uno de sus procesadores. La aplicación que se incluye está inspirada en un problema real tomado del dominio de la sedimentología. Este consiste en la obtención de la forma aproximada de las partículas presentes en una muestra como la que se presenta en la figura 2.a. La dificultad radica, en la adyacencia de partículas. La estrategia se articula en dos etapas. La primera y más sencilla está destinada a aislar el conjunto de partículas del fondo de la imagen. Una vez obtenida esta primera segmentación se convierte la partición en un nuevo mapa de pixels y se activa un segundo programa en el Procesador de Segmentos. En este caso el Presegmentador obtiene el perfil a partir del gradiente morfológico de las zonas que en la partición inicial corresponden a las partículas. Durante el proceso de inundación se hace crecer las partículas desde los mínimos del gradiente compitiendo con el fondo. Las zonas peor delimitadas son las que corresponden a la frontera entre dos partículas, donde evidentemente la incertidumbre acerca de la forma original es muy alta. Finalmente, (figura 2.c) se computan un conjunto de descriptores relativos a la forma geométrica y se seleccionan aquellas partículas que se clasifican como de aspecto redondeado..

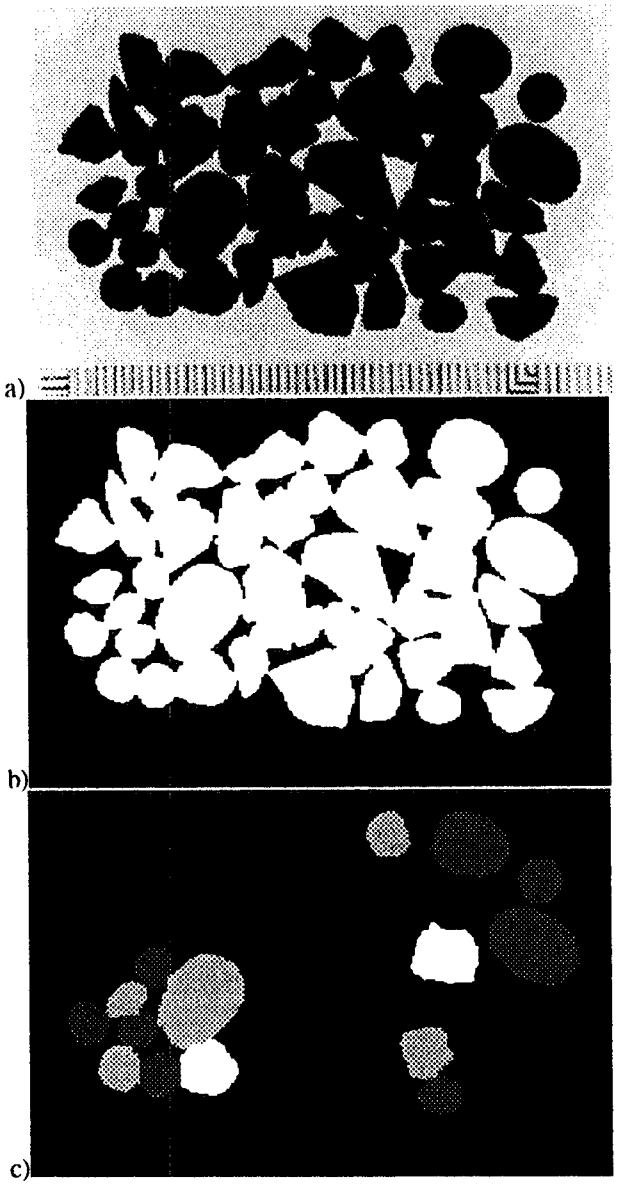


Figura 2. a) Imagen original del caso de estudio. b) Primera Partición. c) Partición final con identificación de partículas redondeadas

5 Conclusiones.

La naturaleza difusa presente en muchas de las tareas que desarrolla un sistema de visión (reconocimiento, clasificación, evaluación de relaciones espaciales, fusión de información, etc.) hacen que estos sean un marco adecuado para la utilización de técnicas de lógica difusa. Por ello, el objeto de los experimentos no ha sido el demostrar la superioridad del esquema propuesto frente a otros sistemas o algoritmos que trabajan a nivel de segmentos, sino el ilustrar aplicaciones de un nuevo esquema que permite resolver problemas de segmentación en contextos muy diferentes mediante mecanismos sencillos fundamentados en la utilización de conceptos de lógica borrosa y la utilidad de estos en el contexto de los Sistemas de Visión por Computador Basados en Conocimiento.

Referencias

[1] Aloimonos J., Weis I., Active Vision, *Int. J. of Comp. Vision*, 2, (1988), 333-356.

[2] Bajcsy R., Active Perception, *Proc of the IEEE*, 76, 8, (1988), 996-1005.

[3] Wilson R., Spann M., *Image Segmentation and Uncertainty*, Research Studies Press Ltd., 1.988.

[4] Niemann H., Brünig H., Salzbrunn R., Schröder S., A Knowledge-Based Vision System for Industrial Applications, *Machine Vision and Applications*, 3, (1990) 201-229.

[5] Dubois D., Prade H., Yager R.R., Introduction en *Reading in Fuzzy Sets for Intelligent Systems*, Dubois D., Prade H., Yager R.R. Eds, Morgan Kaufmann Pub. Inc., pp 1-20, 1.993.

[6] Krisnapuram R., Keller J.M., Fuzzy Set Theoretic Approach to Computer Vision: An Overview, en *Fuzzy Logic Tecnology an Applications*, IEEE Tech. Activities Board, Marks R.J. (ed), pp 25-32, 1.994.

[7] Huntsberger T.L., Rangarajan C., Jayaramamurthy S.N., Representation of Uncertainty in Computer Vision Using Fuzzy Sets, *IEEE Trans. Comput.*, 35, 2 (1986) 145-156.

[8] Riseman E.M., Hanson A.R., A Methodology for the Development of General Knowledge-Based Vision Systems, en *Vision, Brain and Cooperative Computation*, Ed. Arbib y Hanson, MIT Press., Cambridge Mass., pp. 285-328, 1987.

[9] Zadeh L.A., PRUF-a meaning representation language for natural languages, *en Fuzzy Reasoning and its Applications*, Mandani y Gaines (Eds.), Academic Press, pp. 1-66, London, pp. 1-39, 1981.

[10] Mendez J, Falcón A., Hernández F., Cabrera J., A Development Tool for Computer Vision Systems at Pixel Level, *Cybernetic & Systems* 25, 2 (1994) 289-316.

[11] Zadeh L.A., Commonsense Knowledge Representation based on Fuzzy Logic, *IEEE Computer*, 16, 10 (1983), 61-65.

[12] Gordon J., Shortliffe E.H., The Dempster-Shafer Theory of Evidence, en *Rule-Based Expert Systems*, Ed. Buchanan y Shortliffe, Addison-Wesley, pp. 272-292, 1984.

[13] Cabrera J., *Sistema Basado en Conocimiento para Segmentación de Imágenes. Desarrollos y Aplicaciones*, Tesis Doctoral, Universidad de Las Palmas de Gran Canaria, 1.994.

Recuperación de Errores de Segmentación y de Clasificación Utilizando Sistemas Deformados

J. Villadangos[1], A. Sanz[1], J.R. González de Mendívil[2], F. Fariña[1]

[1] Dept. Electricidad y Electrónica
Universidad del País Vasco
APTDO. 644
48080 - BILBAO
e-mail: fitxi@we.lc.ehu.es

[2] Dept. Automática, Electrónica e Ing. de Sistemas
Universidad Pública de Navarra
Campus Arrosadía s/n
31006 - PAMPLONA
e-mail: jrmendi@tsc.upna.es

Resumen

*Los sistemas deformados han mostrado su capacidad para mejorar el rendimiento de sistemas OCR donde el contexto lexicográfico se representa mediante un diccionario porque son capaces de modelar y corregir errores de sustitución (**change**) en las palabras.*

*Este trabajo presenta una extensión simple de estos sistemas para el tratamiento de los errores de inserción (**insert**) y borrado (**delete**) debidos al proceso de segmentación de los caracteres.*

La idea fundamental es modelar los errores de segmentación mediante la extensión del diccionario. Para comprobar la eficacia de los sistemas deformados. Tratando con todo tipo de errores se realiza un experimento cuyo objetivo es la clasificación segura de documentos bancarios.

Palabras Clave: *Sistemas OCR, Sistemas Deformados, Errores de Segmentación*

1. Introducción

El análisis de documentos y la clasificación que contienen siguen siendo hoy en día un problema de gran interés. A pesar de los avances en la construcción de sistemas OCR genéricos, algunos tipos de documentos presentan características peculiares que obligan a la construcción de OCRs particulares para estos casos.

En este trabajo se presentan los resultados obtenidos al trabajar con un sistema OCR basado en sistemas deformados sobre un tipo especial de documentos denominados letras de cambio. El proceso de análisis del documento sigue los siguientes pasos hasta completar la clasificación de la información: escaneado del documento, segmentación de los caracteres de cada zona de interés del documento y clasificación de cada campo o zona con información del documento.

Al analizar este tipo de documentos las dificultades se centran en dos puntos fundamentales: uno de ellos es llegar a obtener una clasificación segura, es decir, determinar que la información reconocida en cada documento sea la correcta, y el otro, evitar que la clasificación sea altamente dependiente del proceso de segmentación de caracteres. Teniendo en cuenta estos dos aspectos, el clasificador que se pretende obtener debe ser un clasificador fiable y tolerante a los fallos que se produzcan en la fase de segmentación y clasificación de caracteres.

La fiabilidad se consigue con la clasificación independiente de dos campos de estos documentos en estudio, que contienen la misma información, pero expresada de forma distinta. La obtención del mismo significado semántico en ambos campos permite determinar con alto grado de seguridad que el documento ha sido bien clasificado. En [7] se expone la forma de hacer uso de la redundancia de la información contenida en los documentos que se analizan con este trabajo (letras de cambio), para conseguir una clasificación segura. En nuestro caso, el significado de ambos campos es una cantidad: un campo la expresa mediante un número (campo de cantidad) y el otro campo mediante palabras organizadas en una frase siguiendo la gramática española para números (campo de importe). A este último campo vamos a prestar atención en los siguientes apartados porque es el más sensible a los errores del segmentador.

En un primer sistema completo de reconocimiento de estos documentos [9] se ha realizado una prueba sobre 140.627 imágenes, de las cuales se han reconocido correctamente el 45% de los documentos. Del 65% restante no se ha reconocido un 1% y el otro 64% contiene errores de segmentación: falsa localización de zonas de información (8%) y falso reconocimiento por mala segmentación (56%). Estos datos ponen de manifiesto la necesidad de que el sistema soporte los errores de la segmentación con el fin de reducir su influencia en la clasificación de los documentos.

El modelado de los errores de segmentación se va a realizar de una forma muy simple mediante la extensión del diccionario a a partir del estudio empírico de los posibles errores introducidos por el segmentador, lo que permite considerar como válidas palabras que han sido mal segmentadas.

Este diccionario extendido es utilizado por el sistema

deformado con objeto de introducir el contexto lexicográfico, ahora con la posibilidad de tratar con todos los posibles errores. En los experimentos realizados se obtiene una importante mejoría en la tasa de reconocimiento lo cual permite su posible incorporación a un sistema OCR. Si bien esta mejora de la tasa de reconocimiento es importante, el proceso está limitado por el aumento del tamaño del diccionario extendido (lo que afecta directamente a la velocidad de cómputo) y por los conflictos entre diferentes palabras cuyo modelo de error es el mismo.

El trabajo se organiza en tres secciones según el siguiente esquema: En la segunda sección se describe el problema con detalle. En la tercera sección se describe la extensión de los sistemas deformados realizada para el caso particular que nos ocupa. En la cuarta sección se mostrarán los resultados experimentales que se han obtenido utilizando tal sistema de clasificación. El trabajo finaliza con las conclusiones y las referencias.

2. Descripción del problema

Los documentos que se analizan tienen un formato específico en cuanto a la posición de la información sobre el documento. Esto facilita, durante el proceso de segmentación, la localización precisa de las zonas de información que se analizarán. Por el contrario, presentan la desventaja de que las imagenes obtenidas en dos niveles contienen mucho ruido. Los campos que son objeto de nuestro tratamiento son el campo de cantidad y el campo de importe que pasamos a describir.

El campo de cantidad está formado por números, caracteres de puntuación y caracteres especiales (#, /, %, $, -, =). Para el reconocimiento de este campo se utiliza un filtrado de la imagen que suministra en la mayoría de los casos imagenes donde los números están bien definidos. Esto facilita la labor de un segmentador que se basa en la localización de mínimos en el histograma vertical. Los caracteres así obtenidos son normalizados y sirven de entrada a un clasificador basado en una Red Neuronal [8,4]. En la información obtenida a la salida de la red se toma el máximo valor que coincidirá con la decisión del carácter elegido. En este caso no tenemos un contexto lexicográfico que nos pueda guiar en el reconocimiento, aunque si se utiliza el conocimiento de la longitud máxima en número de caracteres que el campo puede contener. La red neuronal utilizada para los números tiene una estructura 64/64/10 (neuronas por capa de entrada/interna/salida) obteniendose un 98% de aciertos para este campo del documento.

En este trabajo centramos la atención en el campo de importe, zona donde se escribe el importe en letra. Este campo está compuesto de caracteres mecanografiados en mayúsculas y minúsculas. También puede haber signos de puntuación u otros caracteres gráficos (#,@,|,/,\) sin función semántica, los cuales se identifican y anulan durante el proceso de segmentación, utilizando el histograma horizontal para obtener las lineas de texto (base/media/alta). Se considera que las palabras están correctamente organizadas según la gramática española para números a pesar de que esto no suele ser lo habitual en los documento.

El análisis de este campo tiene la finalidad de comparar su interpretación semántica con la obtenida para el campo de cantidad del mismo documento y poder así determinar cuando el documento ha sido bien clasificado, es decir, que las dos interpretaciones semánticas obtenidas sean las mismas.

El problema con esta zona, como se ve en las figuras 1.a y 1.b, es la gran cantidad de ruido que se capta en el proceso de escaneado de los documentos, previo a la segmentación. El ruido es muy variado y puede llegar a ser de tanta definición como uno de los caracteres del campo, por ejemplo, existen líneas que cruzan los caracteres del campo importe cuya anchura es semejante a los propios caracteres. Este ruido es inevitable, ya que existe en el documento, y filtrar dicho ruido sería filtrar los caracteres que componen el campo. Esto hace que el proceso de segmentación tenga gran dificultad y se tenga que invertir mucho tiempo de cálculo en realizar una segmentación muy precisa que no siempre resulta ser eficiente.

Una vez que un carácter es obtenido por el segmentador es normalizado y una Red Neuronal se encarga de su clasificación. El problema fundamental es conseguir una red que reconozca multitud de tipos de letras y tamaños.

Estructura de R.N.	Error aprendizaje	%Letras reconocidas	%Palabras reconocidas
64/64/27	0.1	68%	7%
64/64/27	0.01	72%	8%
100/64/27	0.01	86%	9%
64/100/27	0.01	65%	7%
64/52/27	0.01	58%	3%

Tabla 2.1: Resultados de reconocimiento para el clasificador de caracteres aislados

La red neuronal utilizada da a la salida del proceso de clasificación un carácter fuzzy compuesto por 27 salidas asociadas a cada carácter del alfabeto español. Cada salida adopta un valor entre 0 y 1 que se interpreta como el grado de reconocimiento del clasificador para ese carácter.

La red neuronal ha sido entrenada con patrones extraídos de los documentos a analizar. La tabla 2.1 presenta diferentes situaciones de entrenamiento, estructuras de red y sus resultados.

La estructura de la red neuronal elegida es de 64 neuronas de capa de entrada, 64 de capa intermedia y 27 de salida. Esta red se entrenó hasta alcanzar un error cuadrático de reconocimiento de 0.01 para 5x27x2

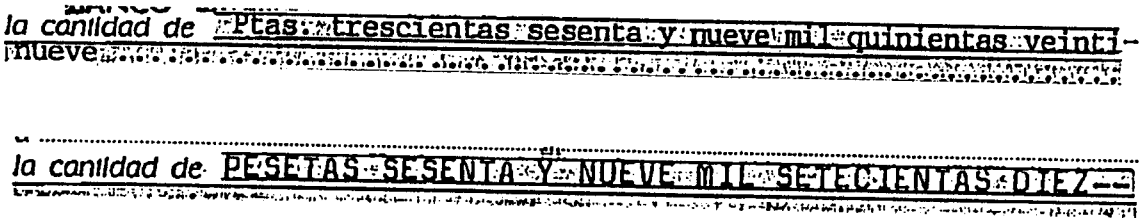


Figura 1 (a), (b) campo importe procedentes de dos documentos distintos

patrones (5 patrones por cada letra mayúscula y minúscula). La tabla 2.1 expresa los resultados de reconocimiento en tanto por ciento para un conjunto de pruebas compuestas por 9.000 letras aproximadamente (mayúsculas y minúsculas) que forman del orden de 1.580 palabras en castellano. En estos experimentos el conjunto de palabras de prueba no estuvo limitado a palabras del campo importe.

Como se puede observar las tasas de reconocimiento de caracteres individuales son bajas debido a que en la prueba se dieron tipos de letra para los que la red no había sido entrenada. Lo peor sucede con las palabras reconocidas al completo cuando no se utiliza el contexto.

La existencia de una gramática permite definir un traductor sintáctico (parser) que dirige el proceso de reconocimiento de la frase del campo del importe. El traductor sintáctico da el control al sistema deformado [5,3], el cual usa la red neuronal para clasificar los caracteres aislados procedentes del proceso de segmentación.

La inclusión del contexto se realiza de dos formas distintas. Una se fija en el contexto lexicográfico, haciendo uso de los sistemas deformados, y la otra se fija en el contexto sintáctico, atendiendo a la gramática del campo importe.

Los sistemas deformados representan un autómata de estados finitos basado en un diccionario. Se construye a partir de las palabras del diccionario, que son quienes establecen las transiciones en el autómata. El diccionario que se usa en este caso está compuesto por el conjunto de palabras que pueden aparecer en la frase del campo importe.

Estos sistemas tienen la particularidad de que las transiciones se realizan para todos los caracteres del alfabeto, ya que utiliza un carácter fuzzy para establecer cada transición, de esta forma se alcanzan diferentes estados con cada carácter fuzzy de salida del clasificador de caracteres aislados. Esto permite modelar los errores de sustitución automáticamente. El resultado al acabar los caracteres de una palabra dada serán, en principio, todas las palabras de la misma longitud. De las posibles salidas del sistema deformado, estados finales, se elige uno de ellos como el reconocido. Se pueden utilizar diferentes criterios para establecer la salida del sistema deformado, entre ellos se define la operación de max-min, max-prod, etc. [1].

Con el análisis sintáctico se tiene en cuenta la gramática que sigue el campo de importe. Éste análisis permite dar la interpretación semántica correcta de dicho campo del documento.

En el siguiente apartado se presenta la extensión de los sistemas deformados para incluir el contexto y los fallos de segmentación que se pueden producir durante el preprocesado de las imágenes a reconocer por el sistema OCR.

Esta extensión está basada en aumentar el diccionario para que sea capaz de aceptar las palabras que contengan los posibles errores de insertado, borrado y sustitución generados durante el proceso de segmentación previo a la clasificación de la información.

3. Extensión de los sistemas deformados

Los sistemas deformados son una herramienta que permiten la inclusión del contexto basado en un diccionario y con buenos resultados de rendimiento [7]. Debido a las buenas características de los sistemas deformados se aborda el problema de la segmentación de caracteres haciendo uso de los mismos.

Un sistema deformado basa su funcionamiento en un Autómata Finito Determinista cuya entrada será no un único caracter sino un conjunto fuzzy. Si el autómata determinista es dado en una forma Moore $A=(\Sigma, Q, q_0, \delta, \lambda, \Delta)$; donde Σ es el alfabeto, Q es el conjunto de estados, q_0 estado inicial, δ la función de transición de estados, λ la función de salida, Δ el alfabeto de salida; y la entrada es un conjunto fuzzy obtenido a la salida de la red neuronal $\tilde{y} = \{(\mu_{\tilde{y}}(\alpha), \alpha), \alpha \in \Sigma\}$ entonces el sistema deformado se expresa como la tupla $((\Sigma,\tilde{y}), (Q,S), q_0, \delta, \lambda, (\Delta,W))$ donde cada conjunto anterior está afectado por la borrosidad introducidad por la entrada. Así S es un conjunto fuzzy con universo Q y W con universo Δ.

Lo que hay que tener en cuenta es como se pasa de un estado borroso al siguiente.

Sea $(S_t, \tilde{y}_t, S_{t+1})$ un paso del sistema deformado, las siguientes ecuaciones establecen la transicion:

(1) dada $\delta(q_t, \alpha_t, q_{t+1})$ entonces $\mu_S(q_{t+1}) \geq \min(\mu_S(q_t), \mu_{\tilde{y}}(\alpha_t), \delta(q_t, \alpha_t, q_{t+1}))$.

(2) dado $\omega = \lambda(q_t)$ entonces $\mu_W(\omega) \geq \mu_S(q_t)$.

En el proceso de reconocimiento cuando el sistema deformado analiza una secuencia de entradas fuzzy dadas por la red neuronal entonces del ultimo estado fuzzy obtenido se calcula la salida correspondiente; en realidad un conjunto fuzzy con universo Δ (el cual representará al diccionario).De este conjunto se elige la palabra con mayor grado de pertenencia.

Una vez visto el concepto de sistema deformado podemos pensar en dos cosas: el error de sustitución es tratado implícitamente por el sistema deformado ya que la información del clasificador es dada al completo; para tratar con el resto de errores de segmentación basta en principio con extender el diccionario aunque otras posibilidades con mayor coste computacional también son posibles. A continuación se dan los detalles sobre la extensión del diccionario.

Ya que la segmentación es un proceso de alto coste computacional se requiere que la modelización de los errores sea lo más exacta posible para poder usar un único algoritmo de segmentación, evitando así implementar algoritmos particulares para los diferentes campos de información del documento.

Los errores modelados están descritos en [6] y se corresponden con posibles transformaciones de caracteres, por ejemplo m-> ni | iii | in, etc. Estas transformaciones se clasifican en errores de sustitución (mil→mii), errores de borrado (cien→cie), y errores de insertado (mil→niil).

La lista de los errores de segmentación propuestos en [6] y, a su vez, modelados en nuestras pruebas experimentales es la siguiente:

Errores de sustitución:

I→ L | l, L→ I, rm →nn, ru→ ni, ni→ ru, t→ f, f→ t, c→ e, e→ c, u→ v, v→ u, h→ b, b→ h.

Errores de borrado:

NI→ M, IN→ M, in→ m, ni→ m, li→ h, cl→ d.

Errores de insertado:

M→ NI, M→III, A→ fl, U→ ll, m→ ni, m→ iii, m→ rn, m→ ro, d→ cl.

El sistema deformado de partida, antes de la extensión, contiene el conjunto de palabras con las que se pueden escribir todos los números del castellano con valores comprendidos entre 1 y 999.999.999.999. El diccionario en cuestión contiene 64 palabras en total.

El sistema deformado extendido equivale a aumentar las palabras del diccionario sobre el que se basa dicho sistema. La extensión se realiza mediante adición de las palabras que se obtienen de las transformaciones indicadas en [6] que han sido listadas anteriormente.

Las palabras obtenidas aplicando las modificaciones que distingue [6] como habituales en los procesos de segmentación, mantienen su valor semántico respecto de la palabra de la que proceden. Por tanto, MIL se puede transformar en NIIL y el sistema deformado seguirá considerando que NIIL está asociado a MIL. Esto tiene el significado de que una palabra mal segmentada no pierde su valor semántico real.

La extensión, de la forma en que se realiza en este trabajo, es útil y no añade ninguna dificultad debido a que el diccionario en el que se basa el sistema deformado es bastante pequeño, 64 palabras.

La extensión del diccionario transforma éste en un diccionario de 11.191 palabras. Hay que tener en cuenta que se han hecho, en cada palabra, todas las combinaciones posibles de alteraciones por mala segmentación. Por tanto, se puede tener para la palabra MIL las siguientes posibles segmentaciones: MIL, NIIL, INIL, IIIIL, IML, IINL, MII, NIII, INII, IIII, IMI y IINI.

La extensión realizada del diccionario no produce ninguna doble definición de palabras, es decir, que las palabras existentes en el diccionario del sistema deformado extendido no poseen diferente valor semántico para diferentes palabras del campo de importe.

Este método de extensión del diccionario sobre el que se basa el sistema deformado incrementa los requerimientos de memoria y el coste computacional debido al aumento del número de palabras.

4. Resultados experimentales

Las pruebas experimentales se han realizado sobre un conjunto de prueba suficientemente significativo. Se ha preparado una prueba formada por 2000 importes correctos (sin errores de segmentación) y diferentes.

De cada uno se han obtenido distintos importes cuyo valor semántico es el mismo pero que han sido transformados en importes que contienen errores de segmentación de los listados anteriormente.

Las transformaciones se han realizado en función de diversos porcentajes de errores de sustitución, insertado y borrado. Estos porcentajes de error varian entre un 30% a un 90% para cada uno.

Los importes han sido preparados sin hacer uso del segmentador disponible para poder dar mayor variedad al conjunto de prueba. Es decir, se han construido los vectores de salida del clasificador de caracteres aislados para cada importe. Para cada caracter se obtiene un valor aleatorio en el intervalo [0.0,0.7] si no es el carácter que se considera como reconocido a la salida de la red. El valor aleatorio entre [0.7,1.0] se obtiene a la salida considerada como reconocida por la red.

También se han usasdo los mismos importes, aunque variándoles los valores de salida de la red neuronal, para considerar el caso de que las falsas segmentaciones también generan peores grados de reconocimiento en el clasificador de caracteres aislados. Los valores de salida de la red se encuentran en el intervalo [0.0,0.6], dando el mayor valor al carácter considerado como mejor reconocido por el clasificador de caracteres aislados.

Los parámetros que se han modificado durante las pruebas han sido los porcentajes de error de borrado, insertado y sustitución sobre el número de posibles borrados, inserciones y cambios en el conjunto de caracteres del importe. A su vez, se ha variando la influencia de cada uno de los porcentajes de manera aleatoria, es decir, se da prioridad a un tipo de errores sobre los demás aleatoriamente.

A continuación se presentan en la tabla 4.1 los resultados obtenidos. En las tres primeras columnas se presentan los porcentajes de error que se han incluido en los importes correctos. De estos importes se muestran en la cuarta columna de la misma tabla los importes que han sido reconocidos correctamente por el sitema deformado a pesar de los errores de segmentación que contienen.

%Errores insertado	%Errores sustitución	%Errores borrado	%Importes recuperados
40%	30%	35%	98.89%
50%	54%	62%	96.25%
84%	90%	85%	94.50%

Tabla 4.1: Resultados de reconocimiento para el clasificador de caracteres aislados

La tabla anterior (tabla 4.1) se ha obtenido considerando salidas de la red con valores entre [0.8,1.0] para los caracteres consierados como reconocidos por la red. En el otro caso explicado anteriormente, donde los valores de los caracteres reconocidos por el clasificador aislado pertenecian al intervalo [0.0,0.6], los resultados son semejantes. Se consigue, por lo tanto, recuperar casi el 100% (98.89% en el mejor de los casos) de los importes con errores de segmentación.

Conclusiones

Los sistemas deformados basados en diccionario se muestran como una herramienta útil para incluir el contexto lexicográfico en el reconocimiento de textos. Estos sistemas permiten, automáticamente, modelar los errores de sustitución para un diccionario.

Este trabajo muestra cómo los sistemas deformados y su extensión basada en los errores típicos de segmentación permiten implementar sistemas OCR altamente fiables y tolerantes a fallos de segmentación cuando el diccionario no es muy grande.

El sistema desarrollado permite recuperar casi el 100% de los importes con fallos debidos a la segmentación. Por tanto, el porcentaje de documentos mal clasifcados por el sistema OCR estará limitado por el porcentaje de documentos cuyo campo de cantidad esté bien clasificado.

Este trabajo, por otra parte, permite deducir que se puede desarrollar un sistema deformado específico para cada sistema OCR basando su diccionario extendido en los errores de segmentación del propio sistema OCR.

Agradecimientos

Este trabajo ha sido financiado por el proyecto P332 de la Fundación Euskoiker y la empresa FILOSOFT S.A. dentro del programa de colaboración Universidad-Empresa.

Referencias

[1] "Estudio de funciones de composición de evidencias en un problema de reconocimiento de textos". J. Echanove, R. Reina, J.R. Garitagoitia, J.R. González de Mendívil. IV Congreso de la Asociación Española sobre Tecnologías y Lógica Fuzzy, Sept. 1994 (Blanes), España.

[2] "Deformed Systems in Text Recognition". J. Echanove, J.R. González de Mendívil, J.R. Garitagoitia. International Conference on Artificial Neural Networks (ICANN'95), Mayo 1994 (Sorrento), Italia.

[3] "A Text Recognition System based on a Neural Network and on a Deformed System". J. Echanove, J.R. González de Mendívil, J.R. Garitagoitia. International Workshop on Artificial Neural Networks (IWANN'95), Junio 1995 (Torremolinos), España.

[4] "A Theory of Multiple Classifier Systems and Its Application to Visual Word Recognition". X. Ho. A dissertation submitted to the Faculty of the Graduate School of State University of New York at Buffalo in partial fulfillment of the requirements for the degree of Doctor of Philosophy, 1992.

[5] "Experiencias en el reconocimiento de caracteres mediante un sistema basado en una red neuronal y un autómata fuzzy". R. Reina, J.R. González de Mendívil, J. Echanove, J.R. Garitagoitia. II Congreso de la Asociación Española sobre Tecnologías y Lógica Fuzzy, Nov. 1992 (Boadilla del Monte), España.

[6] "Hybrid Contextual Text Recognition with String Matching" R. M. K. Sinha, B. Prasada, G. F. Houle, M. Sabourin IEEE Transactions on Pattern Analysis and Machine Intelligence" Vol 15, nº9, Sept. 1993.

[7] "An OCR System for Document Classification". J. Villadangos, F. Fariña, J.R. González de Mendívil, J. Echanove, J.L. Esteban. IASTED'95, Int. Conf. on APPLIED INFORMATICS, 21-23 Feb. 1995 Innsbruck (Austria).

[8] "Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition". L. Xu, A. Kryzak, Ch. Y. Suen. IEEE Transactions on Systems, Man and Cybernetics, Vol 22, nº3, May/June 1992.

[9] "Documentación proyecto P332. Fundación Euskoiker. J.R. González de Mendívil. 1994.

Aplicación de la Lógica Borrosa al Diagnóstico de Fallos en Maquinaria Rotativa

Jesús Fernández
Dpto. Ingeniería de Sistemas y Automática.
E.T.S.I.I.
Universidad de Valladolid
E-mail: davcar@dali.cis.uva.es

Luis J. de Miguel
Dpto. Ingeniería de Sistemas y Automática.
E.T.S.I.I.
Universidad de Valladolid
E-mail: luimig@dali.cis.uva.es

José R. Perán
Dpto. Ingeniería de Sistemas y Automática.
E.T.S.I.I.
Universidad de Valladolid
E-mail: peran@dali.cis.uva.es

Marzo 1995

Resumen

Este artículo describe un método de detección y diagnóstico de fallos en máquinas rotativas que combina el uso de técnicas de lógica difusa con el análisis de vibraciones. La información útil para el diagnóstico que el análisis de vibra- ciones genera está afectada por incertidumbre en las condiciones de medida o producida por perturbaciones. Sin embargo, esta información puede ser inter- pretada cualitativamente, y se presenta de forma redundante en los distintos síntomas de diagnóstico, de manera que la detección y diagnóstico de fallos se puede obtener fiablemente mediante un procesamiento difuso del conocimiento. La sensibilidad de los fallos a los distintos síntomas será el factor determinante para estable- cer un diagnóstico mediante el procesamiento difuso de estos síntomas.

Palabras clave: Detección de fallos y diagnóstico, Análisis de vibraciones, Maquinaria rotativa, Lógica difusa.

1 Introducción.

La importancia económica del mantenimiento industrial ha motivado en los últimos años un gran desarrollo de los sistemas de diagnóstico automático. Con mayor frecuencia, se usan técnicas basadas en la Inteligencia Artificial para representar el conocimiento del experto en diagnóstico de maquinaria. [1, 16].

El mantenimiento predictivo de maquinaria rotativa mediante las técnicas de análisis de vibraciones se utiliza desde hace al menos veinte años. [6, 13]. Una abundante información sobre el estado de una máquina rotativa puede ser inferida a partir de las vibraciones mecánicas medidas en sus diversos puntos críticos [4]. Para una máquina en correcto estado de funcionamiento, se observa una tendencia constante en las propiedades estadísticas y frecuenciales de estas vibraciones, mientras que variaciones importantes pueden ser observadas cuando en ésta se presenta algún tipo de fallo. Estos cambios permitirán detectar y diagnosticar los cuatros defectos más frecuentes: desequilibrio, desalineamiento en el eje, defecto en algún rodamiento y fallo de origen eléctrico. Teniendo en cuenta estas consideraciones, un sistema completo de mantenimiento predictivo deberá ser capaz de realizar las siguientes funciones:

1. Adquisición de las vibraciones y selección de la información necesaria para el diagnóstico de defectos: a partir de la señal temporal de la vibración medida y de su espectro de frecuencia, será posible seleccionar un conjunto de parámetros con los que el estado de los distintos fallos podrá ser determinado. Algunos de los parámetros normalmente analizados tienen una naturaleza estadística como la curtosis, mientras que otros vienen expresados en el dominio de la frecuencia, como las amplitudes de las frecuencias asociadas con la frecuencia de rotación de la máquina. Es necesaria una selección previa de aquellos parámetros que provean información relevante sobre los fallos que el sistema pretende diagnosticar. Los parámetros escogidos de este modo se denominarán parámetros de diagnóstico.

2. Diagnóstico del estado de desarrollo de los fallos estudiados, basado en la información previamente seleccionada: los expertos en análisis de vibraciones pueden evaluar la importancia de cada parámetro de diagnóstico y ofrecer de esta manera el diagnóstico más adecuado. El modelo de conocimiento descrito en este artículo intenta emular en términos lingüísticos el proceso de decisión realizado por el experto.

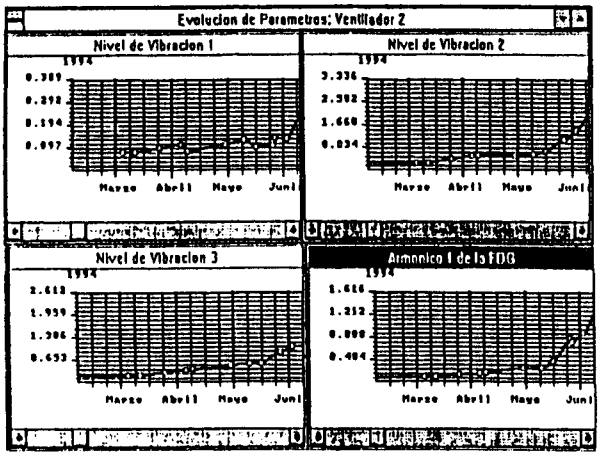


Figura 1: Evolución de diferentes parámetros de diagnóstico durante el desarrollo de un defecto de rodamiento

La figura 1 muestra la evolución real de algunos de los parámetros de diagnóstico tras la aparición de un defecto de rodamiento en un ventilador industrial de la empresa en la que se instaló el sistema de mantenimiento para el que este modelo de diagnóstico fue concebido. La misma conclusión alcanzada por el experto sobre la aparición de un fallo debe ser inferida por el modelo de diagnóstico desarrollado.

El uso de variables difusas se considera adecuado para implementar el diagnóstico de fallos por las dos siguientes características del problema:

1. Incertidumbre en los parámetros de diagnóstico originada por las condiciones de medida: la variación de estas condiciones puede provocar desviaciones pequeñas e incontroladas en los parámetros, que dificulta el uso de algoritmos deterministas para realizar el diagnóstico. Otros motivos para la incertidumbre son el ruido y las perturbaciones con las que se ve afectado el proceso de medida, así como las imprecisiones de computación.

2. Redundancia de la información: la aparición real de un fallo en la maquinaria produce cambios significativos en algunos parámetros de diagnóstico; por lo tanto, este fallo podría ser identificado por un experto al aplicar distintos criterios, cada uno de ellos basado en el cambio experimentado por algún síntoma indicador de ese fallo. No obstante, cada criterio puede ofrecer una fiabilidad diferente de cara al diagnóstico final. Un diagnóstico completo debe tener en cuenta la importancia y fiabilidad de cada uno de los criterios asociados a los distintos síntomas.

El método de diagnóstico desarrollado intenta detectar y aislar los cuatro fallos mencionados en su estado incipiente, a partir del análisis de los distintos parámetros. Para conseguir esto, el proceso se divide en tres etapas[8]:

1. *Detección del fallo*: constatar que algo está mal en el sistema.

2. *Aislamiento del fallo*: localización del fallo.

3. *Identificación del fallo*: medida del tamaño del fallo.

Las primeras dos etapas son fundamentales para el diagnóstico, y para su tratamiento se ha desarrollado el sistema basado en el conocimiento propuesto. La detección y aislamiento de fallos las realiza el experto en términos lingüísticos; por este motivo el uso de variables borrosas está justificado. La identificación del fallo se estimará numéricamente.

Una breve descripción de cada una de estas tres etapas se desarrollará en este artículo, dando especial importancia al tratamiento difuso de síntomas y fallos.

2 Detección de fallos usando información redundante con incertidumbre. Borrosificación de los síntomas.

Los fallos que el método de diagnóstico pretende detectar son: desequilibrio, desalineamiento del eje, defecto de rodamiento y fallos con origen eléctrico. Éstos son los defectos que más frecuentemente se presentan en las máquinas rotativas.

El comienzo de cada uno de estos fallos provocará el incremento de algunos parámetros de diagnóstico. Por ejemplo, un aumento significativo del segundo armónico de la frecuencia de giro puede ser un síntoma claro del desa- lineamiento del eje. Por lo tanto, una labor previa a la detección de fallos sería la selección del conjunto de aquellos parámetros de diagnóstico P, under study. que incluyan información importante sobre el conjunto de fallos a estudiar F. Un modelo de diagnóstico ha sido implementado informáticamente con el método descrito en este artículo. Para este modelo se han utilizado 36 parámetros de diagnóstico.

Con el análisis de vibraciones se pueden obtener también algunos parámetros, como los niveles de vibración, cuyo valor aumentará con la aparición de cualquiera de los fallos estudiados. Como se explicará posteriormente, la detección de fallos estará basada en este subconjunto de parámetros, $\mathbf{P_D} \subset \mathbf{P}$, cuyos elementos se denominarán parámetros de detección.

La información relevante para el diagnóstico y que se incluye en los parámetros de diagnóstico, no se encuentra en su valor absoluto, sino en la variación respecto a un valor de referencia. Sea P_j la medida del parámetro j, y P_j^{ref} su valor de referencia. El síntoma numérico de fallo [1] puede ser definido como:

[1] En este artículo las variables que representan síntomas y fallos pueden tener un formato numérico o difuso. El superíndice * indicará que el síntoma o fallo pertinente se expresa en formato numérico.

A menudo, no se dispone de suficiente información experimental para poder aplicar el criterio estadístico, y el valor final otorgado a estos límites estará determinado por la opinión del experto.

Se ha descrito la forma de expresar la información contenida en los parámetros de diagnóstico como un síntoma numérico, que muestra el cambio relativo del parámetro respecto a un valor de referencia. Los síntomas son transformados posteriormente en variables difusas, definidas como una cuádrupla de posibilidades de pertenencia. Denominaremos S al conjunto de síntomas difusos. Los elementos de este conjunto correspondientes a parámetros de detección serán los síntomas de detección y compondrán el subconjunto $S_D \subset S$.

Para una mejor comprensión del procedimiento de detección, se considerará inicialmente que el subconjunto de síntomas de detección contiene un único elemento, S_j^D. Se utilizará una nueva variable difusa para describir en términos lingüísticos los resultados de la detección basada en el síntoma S_j^D. Esta variable se denominará F_D y está expresado como un conjunto de posibilidades de pertenencia a las siguientes etiquetas lingüísticas:

I *No se ha detectado ningún fallo.*

II *Se ha detectado un fallo incipiente.*

III *Se ha detectado un fallo parcialmente desarrollado.*

IV *Se ha detectado un fallo totalmente desarrollado.*

Al disponerse de un único síntoma de detección S_j^D, será éste el único criterio utilizado para la detección de fallos. Por lo tanto, las posibilidades de pertenencia de F_D a las etiquetas lingüísticas I, II, III and IV tendrán los valores correspondientes de S_j^D a través de la implicación difusa:

$$IF\ S_j^D\ THEN\ F_D \tag{9}$$

En el caso de que se disponga de más de un síntoma de detección, lo que será más normal, la detección de fallos será procesada mediante los operadores difusos de la siguiente ecuación:

$$IF\ (S_1^D \wedge ... \wedge S_{n_d}^D)\ THEN\ F_D \tag{10}$$

donde n_d representa el número de síntomas de detección.

La detección de fallos debe ser realizada en términos lingüísticos, y, por esta razón, la variable F_D debe sufrir un proceso de desborrosificación para decidir qué etiqueta lingüística será seleccionada como resultado de la detección. Para esta tarea, se podrían utilizar diversos algoritmos de desborrosificación. En el modelo desarrollado, la etiqueta lingüística con una mayor posibilidad de pertenencia será la que determine el estado final del proceso de detección

3 Aislamiento de fallos y diagnóstico. Evaluación difusa de los síntomas.

Si un fallo ha sido detectado, es decir, F_D ha adoptado la etiqueta lingüística I, II, III or IV, se deberá llevar a cabo un diagnóstico para determinar qué fallos han causado la alarma de detección.

El diagnóstico está basado en la dependencia que cada uno de los fallos estudiados tiene con cada uno de los síntomas, que pueden ser clasificados en tres clases según esta relación:

1. *Síntomas sensibles*: un síntoma es sensible a un fallo si un aumento significativo del síntoma es un indicio de desarrollo del fallo.

2. *Síntomas insensibles*: un síntoma es insensible a un fallo si dicho fallo no produce un aumento significativo del síntoma. Una consecuencia de este hecho será que la aparición de este síntoma debe orientar el diagnóstico hacia otro posible fallo.

3. *Síntomas inciertos*: un síntoma es incierto para un fallo si no puede ser considerado como sensible o insensible a este fallo.

Una primera información básica sobre estas relaciones puede ser representada por medio de la matriz de dependencias D, con dimensions (n_f x n_s). donde n_f representa el número de fallos estudiados, y n_s el número de síntomas considerados. Cada elemento d_{ij} de la matriz D adoptará un valor de 0, -1 or 1. El valor que se le otorgue significará la siguiente relación entre el fallo i y el síntoma j:

-d_{ij} es 1 si el síntoma i es sensible al fallo j.
-d_{ij} es -1 si el síntoma i es insensible al fallo j.
-d_{ij} es 0 si el síntoma i es incierto para el fallo j.

Los valores finales asignados a cada fila de la matriz D deben ser tales que los cuatro fallos diferentes sean aislables. Para obtener una medida de la separabilidad del sistema de diagnóstico, se definirá la distancia entre dos vectores filas de la matriz D como el número de elementos diferentes en ambos vectores. Los siguientes resultados son evidentes:

a) El modelo de diagnóstico definido por la estructura de la matriz D será suficientemente separable si la distancia mínima entre cada par de vectores fila es 1.

b) Si la distancia entre dos vectores filas es mayor que 1, la separabilidad de los fallos representados por estas filas será más sencilla, y su diagnóstico, más robusto.

Una medida más precisa de la diagnosticabilidad del modelo de diagnóstico debe considerar la sensibilidad de los síntomas a los fallos, que se definirá a continuación.

$$S_j^* = \frac{P_j - Pj^{ref}}{P_j^{ref}} \quad (1)$$

donde $P_j^{ref} \geq 0$, puede ser obtenido empíricamente como la media de un conjunto de muestras del parámetro P_j, medidas bajo condiciones nominales y sin presencia de fallo.

Si se aplica la ecuación 1, se podrá obtener un valor numérico para el síntoma, que estará afectado por un alto grado de incertidumbre por razones ya explicadas. Por tanto, un síntoma aislado no podrá ser lo suficientemente fiable para obtener un diagnóstico robusto.

Una primera solución para este problema consistiría en la elección de un umbral elevado H_j, de tal manera que sólo se declarase un síntoma si

$$S_j^* \geq H_j \quad (2)$$

Esta solución aumentaría la fiabilidad, pero la capacidad del sistema para detectar fallos se vería mermada. Por lo tanto, la detección de fallos ocurriría cuando sus efectos fuesen considerables, lo que dificultaría su corrección.

Por otro lado, la elección de un umbral bajo h_j significaría que la declaración del síntoma

$$Sj^* \geq h_j \quad (3)$$

podría estar motivada por ruido y perturbaciones, lo que provocaría falsas alarmas.

La determinación de un umbral adecuado h'_j tal que

$$h_j \leq h'_j \leq H_j \quad (4)$$

de manera que el coste asociado con la declaración de falsas alarmas se compense con el coste del retraso en la detección, es un problema de difícil resolución.

El método adoptado para superar estos inconvenientes está basado en la teoría de conjuntos borrosos [17], y en el hecho de que el uso de información redundante permite evaluar la posibilidad de fallo a través de distintos síntomas. Esta solución aumenta la fiabilidad del diagnóstico.

El proceso de borrosificación de cada síntoma definido a través de la ecuación 1 consiste en la asignación de una variable borrosa a cada valor numérico. La correspondiente variable borrosa estará expresada en forma de etiquetas lingüísticas con sus correspondientes posibilidades de pertenencia.

En nuestro caso, se han definido cuatro etiquetas lingüísticas:

I *No hay síntoma de fallo.*

II *Síntoma pequeño de fallo.*

III *Síntoma mediano de fallo.*

IV *Síntoma grande de fallo.*

De esta manera, un síntoma numérico de fallo, S_j^* será borrosificado e identificado por estos valores simbólicos, con sus correspondientes posibilidades de pertenencia. El síntoma difuso resultante se denominará S_j. Sus posibilidades de pertenencia se definen mediante las funciones de pertenencia representadas en la figura 2.

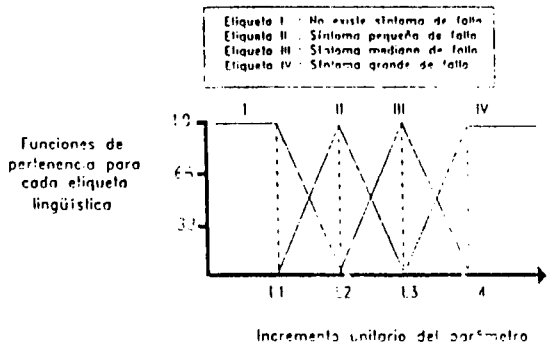


Figura 2: Funciones de pertenencia para síntomas difusos

Los cuatro límites representados en la figura son necesarios para definir funciones de pertenencia: L_1, L_2, L_3 and L_4. Este conjunto de límites representa los distintos umbrales para los que un incremento relativo lo suficientemente significativo de un parámetro de diagnóstico puede ser considerado como síntoma de fallo.

L_1 y L_4 son los límites más arriesgados (L_1) y conservadores c(L_4) del problema. Serán establecidos siguiendo el criterio del experto en vibraciones o mediante un análisis estadístico de las medidas experimentales. Un método estadístico para establecer estos límites para el parámetro P_j puede ser:

$$L_1 = \sigma_j \quad (5)$$
$$L_2 = 2\sigma_j \quad (6)$$
$$L_3 = 3\sigma_j \quad (7)$$
$$L_4 = 4\sigma_j \quad (8)$$

donde σ_j representa la desviación estándar del parámetro j cuando no se presenta ningún fallo.

Esta técnica para asignar valores a los límites L_1, L_2, L_3 y L_4 está basada en el siguiente hecho estadístico: si se considera que, en ausencia de fallos, los valores del parámetro P_j siguen una distribución normal de probabilidad la probabilidad de que una medida P_j pertenezca al intervalo de confianza $[-\sigma_j, \sigma_j]$ sería 0.68268, mientras que la probabilidad de que se encuentre en el intervalo de confianza $[-4\sigma_j, 4\sigma_j]$ sería 0.99993. Por tanto, L_4 sólo será superado por muestras P_j con grandes desviaciones de su valor de referencia.

La evaluación del estado de los fallos debe realizarse teniendo en cuenta que cada síntoma tendrá una relevancia distinta para cada fallo. Algunos síntomas serán un indicativo claro para un fallo, mientras que otros no serán significativos para el desarrollo de éste. Esta propiedad puede ser definida como la sensibilidad de un síntoma a un fallo.

La sensibilidad del síntoma j al fallo i se denominará T_{ij} y puede ser expresada simbólicamente como:

$$T_{ij} = \frac{\text{Tamaño del } SINTOMA_j}{\text{Tamaño del } FALLO_i} \qquad (11)$$

Los tamaño de síntomas y fallos se expresarán numéricamente, como el incremento del síntoma o fallo respecto a su valor de referencia, que han sido medidos bajo condiciones nominales y sin presencia de fallo. Ambos valores de referencia serán nulos bajo estas condiciones. Si se aplica esta definición, la sensibilidad síntoma-fallo puede ser descrita como:

$$T_{ij} = \frac{S_j^*}{F_i^*} \qquad (12)$$

Si se dispone de suficientes medidas experimentales, los valores de las sensibilidades podrán ser calculados. En caso contrario, deberán ser definidos cuantitativamente por el experto.

Los valores de las sensibilidades para el fallo i serán mayores para los síntomas sensibles ($d_{ij} = 1$ en la matriz de dependencias). Un método para el diagnóstico del fallo i será la ponderación de este tipo de síntomas. Este tipo de diagnóstico se denominará *diagnóstico basado en síntomas sensibles*

El criterio utilizado para obtener el vector de pesos necesarios para la ponderación de los síntomas sensibles a un fallo, será la normalización de sus correspondientes sensibilidades.

La matriz de los diferentes vectores de pesos para los n_f fallos formará una matriz de pesos para síntomas sensibles, $\mathbf{W^S}$, con dimensiones (n_f x n_s). Cada elemento w_{ij}^S de esta matriz con un valor no nulo representa la influencia del síntoma sensible j en el diagnóstico del fallo i. Se calculará mediante la siguiente ecuación:

$$w_{ij}^S = \begin{cases} \dfrac{T_{ij}}{\sum_{k=1}^{n_i^S} T_{ik}} & \text{if } d_{ij} = 1 \\ 0 & \text{if } d_{ij} \neq 1 \end{cases} \qquad (13)$$

donde n_i^S es el número de síntomas sensibles para el fallo i.

Para este diagnóstico, un concepto importante debe ser considerado: cada componente del vector de síntomas ha sido definido como una variable difusa, expresada como un conjunto de posibilidades de pertenencia a cuatro etiquetas lingüísticas. La ponderación de este tipo de variables resultará en otra variable difusa con cuatro etiquetas lingüísticas. En este caso, la variable obtenida será F_i^S, con $i \in [1, n_d]$, que representa el estado difuso del fallo i deducido a partir del conjunto de síntomas sensibles a este fallo.

F_i^S vendrá expresado por las posibilidades de pertenencia a los conjuntos borrosos identificados por las siguientes etiquetas lingüísticas:

I *No hay fallo.*

II *Fallo incipiente.*

III *Fallo parcialmente desarrollado.*

IV *Fallo plenamente desarrollado.*

Las etiquetas son similares a las asociadas a las variables de detección F_D. Sin embargo, F_D representa el estado de un fallo genérico durante el proceso de detección, mientras que F_i^S representa un fallo específico obtenido como resultado del diagnóstico basado en síntomas sensibles.

Para la obtención de una expresión matemática válida para el diagnóstico, se considerará que F_i^S y S_i están expresados como vectores fila, cuyos componentes son las cuatro posibilidades de pertenencia a sus correspondientes etiquetas lingüísticas:

$$F_i^S = (F_{i_I}^S, F_{i_{II}}^S, F_{i_{III}}^S, F_{i_{IV}}^S) \qquad (14)$$

$$S_j = (S_{i_I}, S_{i_{II}}, S_{i_{III}}, S_{i_{IV}}) \qquad (15)$$

donde $i \in [1, n_f]$, y $j \in [1, n_s]$.

El diagnóstico para el conjunto de n_f fallos se calcularía usando la matriz $\mathbf{W^S}$:

$$\begin{pmatrix} F_1^S \\ \vdots \\ F_i^S \\ \vdots \\ F_{n_f} \end{pmatrix} = \begin{pmatrix} w_{11}^S & \cdots & w_{1n_s}^S \\ \vdots & \vdots & \vdots \\ w_{i1}^S & \cdots & w_{in_s}^S \\ \vdots & \vdots & \vdots \\ w_{n_f1}^S & \cdots & w_{n_fn_s}^S \end{pmatrix} \begin{pmatrix} S_1 \\ \vdots \\ S_i \\ \vdots \\ S_{n_s} \end{pmatrix} \qquad (16)$$

El vector que contiene las variables difusas F_i^S será denominado $\mathbf{F}$, y tendrá dimensiones (n_f x 4). De la misma manera, los síntomas difusos pueden formar $\mathbf{S}$, con dimensiones (n_s x 4). Si se introduce esta notación en la ecuación 16, el diagnóstico puede expresarse como:

$$\mathbf{F^S} = \mathbf{W^S} \times \mathbf{S} \qquad (17)$$

Mediante este procedimiento, se ha obtenido un conjunto de posibilidades de pertenencia para cada variable difusa que representa uno de los fallos estudiados. Los síntomas sensibles para cada fallo han sido seleccionados como datos de diagnóstico.

Hasta el momento, los síntomas insensibles no han sido considerados. Sin embargo, también aportan información relevante para el aislamiento y diagnóstico:

si un síntoma insensible para el fallo i sufre un incremento significativo (adopta posibilidades de pertenencia altas para las etiquetas lingüísticas III o IV), se podría interpretar como signo de que es otro fallo, y no el fallo i, el que ha sido detectado. Esta evidencia puede ser utilizada para establecer un diagnóstico paralelo como una ponderación de los síntomas insensibles a través de una nueva matriz de pesos $\mathbf{W^I}$, con dimensiones $(n_f \times n_s)$, definida como:

$$w_{ij}^I = \begin{cases} 1 - \dfrac{T_{ij}}{\sum_{k=1}^{n_i^I} T_{ik}} & \text{if } d_{ij} = -1 \\ 0 & \text{if } d_{ij} \neq -1 \end{cases} \tag{18}$$

donde n_i^I es el número de síntomas insensibles para el fallo i.

Los elementos no nulos w_{ij}^I de la matriz $\mathbf{W^I}$ contienen las sensibilidades normalizadas del síntoma i al fallo j.

Para obtener un diagnóstico *basado en síntomas insensibles*, se aplicará el mismo método:

$$\mathbf{F^I} = \mathbf{W^I} \times \mathbf{S} \tag{19}$$

donde $\mathbf{F^I}$ contiene el estado de los n_f fallos difusos, inferido a partir de la ponderación de síntomas insensibles, y $\mathbf{W^I}$ es la matriz de pesos para los síntomas insensibles.

Para obtener un diagnóstico para el fallo i basado tanto en síntomas sensibles como en síntomas insensibles, se utilizará un operador Y difuso:

$$F_i = F_i^S \wedge F_i^I \tag{20}$$

F_i representa finalmente el diagnóstico difuso para el fallo i. Como se pretende establecer este diagnóstico en términos lingüísticos, se deberá llevar a cabo el proceso de desborrosificación, que consistirá en la selección de la etiqueta lingüística con mayor posibilidad de pertenencia.

En la figura 3 se puede observar un esquema con el proceso completo de detección y diagnóstico difuso.

4 Identificación del fallo.

La identificación del fallo, o estimación del tamaño, es un complemento a la detección y aislamiento de fallos. Si un fallo ha sido considerado detectado y diagnosticado, la estimación del tamaño consiste en establecer un valor numérico que exprese el grado de desarrollo del fallo respecto a un nivel de alarma. Por tanto, la estimación del tamaño es el único aspecto del proceso de diagnóstico total que no es realizado en términos lingüísticos, pero será descrita superficialmente para lograra una mejor comprensión del proceso global.

Para justificar el método de estimación del tamaño, se supondrá en principio que el criterio asociado al síntoma sensible j es el único relevante para el aislamiento del fallo i. Una primera estimación absoluta

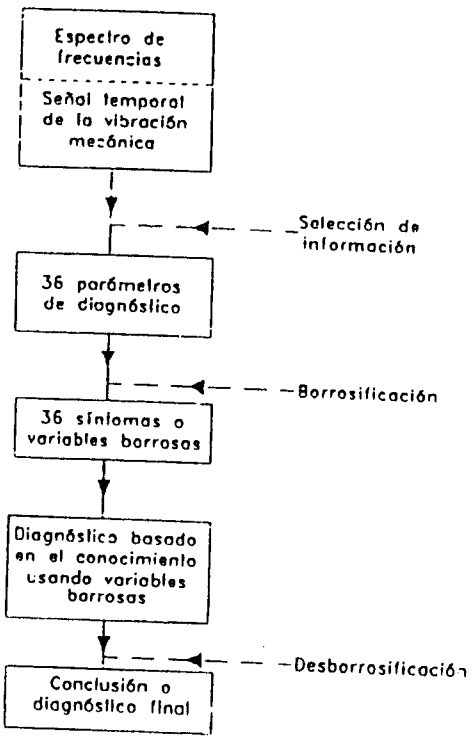


Figura 3: Procesos de detección y diagnóstico.

del tamaño de fallo puede ser deducida a partir de la definición de sensibilidad dada en la ecuación 12:

$$F_i^* = \frac{S_j^* - S_j^{*ref}}{T_{ij}} \tag{21}$$

Sin embargo, un valor numérico para F_i^* será más descriptivo si viene expresado en términos relativos a un nivel de alarma, F_i^{*A}. Entonces, se volverá a definir F_i^* de la siguiente manera:

$$F_i^* = \frac{1}{F_i^{*A}} \frac{S_j^* - S_j^{*ref}}{T_{ij}} \tag{22}$$

En general, el tamaño de un fallo no será estimado por medio de un único síntoma, sino a través de todos aquellos síntomas que sean sensibles a ese fallo. Por lo tanto, será lógico considerar el tamaño de fallo como la media obtenida a partir de los tamaños de fallo estimados para cada uno de los n_i^S síntomas sensibles:

$$F_i^* = \frac{1}{n_i^S} \frac{1}{F_i^{*A}} \sum_{k=1}^{n_i^S} \frac{S_k^*}{T_{ik}}, \tag{23}$$

Se definirá la matriz V, donde cada elemento v_{ij} podrá calcularse por medio de la siguiente ecuación:

$$v_{ij} = \begin{cases} \frac{1}{n_j} \frac{1}{F_i^{*A}} \frac{1}{T_{ij}} & \text{if} \quad d_{ij} = 1 \\ 0 & \text{if} \quad d_{ij} \neq 1 \end{cases} \tag{24}$$

Si F^* y S^* son vectores columna, con dimensiones (n_f x 1) y (n_s x 1), que contienen los valores numéricos para síntomas y fallos, un cálculo directo del vector tamaño de fallo puede realizarse a través de la ecuación:

$$F^* = V \, x \, S^* \tag{25}$$

5 CONCLUSIONES.

Se ha propuesto un método de diagnóstico difuso para maquinaria rotativa. Los datos de entrada a ser procesados se obtienen por medio del análisis de vibraciones, aunque también pueden ser incluidos síntomas de otras fuentes de conocimiento. La incertidumbre en la estimación de los parámetros ha sido tratada por medio de la representación de los síntomas de fallo utilizando variables borrosas. La formalización de las relaciones de dependencia entre síntomas y fallos a través de la estimación de sensibilidades posibilita la definición de un grado de diagnosticabilidad para el modelo. El uso de información redundante y de variables difusas permite el diseño de un método robusto de diagnóstico. El método puede ser extendido a otros problemas de diagnóstico en los que la sensibilidad entre síntomas y fallos pueda ser definida.

Referencias

[1] D. Barschdorf, "Artificial Intelligence: Diagnostic Expert System, Fuzzy Logic and Neural Networks.", *Proceedings of the 8th International IMEKO Symposium on Technical Diagnostics*, Dresden, Alemania, 23-25 Septiembre, 1992.

[2] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes. Theory and Application*, Prentice-Hall, Inc., 1993.

[3] F. Bastani, R. Chen and T.Tsao "Reliability of Systems with Fuzzy-Failure Criterion", *1994 Proceedings Annual Reliability and Maintainability Symposium*

[4] J.S. Bendat and A.G. Piersol, *Engineering Applications of Correlation and Spectral Analysis*, John Wiley & Sons, 1980.

[5] J. C. Bezdek, "Computing with uncertainty", *IEEE Communications Magazine*, vol. 30, no. 9, pp. 24-26, 1992.

[6] S. Braun, *Mechanical Signature Analysis: Theory and Applications*, Academic Press, 1986.

[7] P.A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice Hall International, Inc., Londres, 1982.

[8] J. Gertler, "Survey of Model-Based Failure Detection and Isolation in Complex Plants", *IEEE Control Systems Magazine*, December 1988, pp. 3-11, 1988.

[9] A. Kaufmann and M.M. Gupta, *Introduction to Fuzzy Arithmetic Theory and Applications* Van Nostrand Reinhold, Nueva York 1991.

[10] R. Keith, *An Introduction to Predictive Maintenance* Van Nostrand Reinhold, Nueva York 1991.

[11] J. Knezevic and D. Verma, "Application of Fuzzy Logic in the Assurance Sciences" *1994 Proceedings Annual Reliability and Maintainability Symposium*

[12] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Approach to Machine Intelligence* Prentice Hall, Englewood Cliffs NJ, 1991.

[13] R.H. Lyon, *Machinery Noise and Diagnostic*, Butterworth Publishers, 1987.

[14] R. Patton, P. Frank and R. Clark (Eds.), *Fault Diagnosis in Dynamic Systems. Theory and Applications*, Prentice Hall Int., 1989.

[15] E. Sánchez Jacob, *Aplicación del análisis de vibraciones a un sistema de mantenimiento industrial* Proyecto fin de carrera, Escuela de Ingenieros Industriales, Valladolid, España.

[16] S. Tzafestas, M. Singh and G. Schmidt (Eds.), *System Fault Diagnostics, Reliability and Related Knowledge-base Approaches*, D. Reidel Publishing Company, Dordrecht, Holanda, 1987.

[17] L.A. Zadeh, "Fuzzy sets" *Information and Control*, vol. 8, pp. 338-352, 1965.

[18] L.A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes" *IEE Transactions on Systems, Man and Cybernetics*, SMC-1, 1973.

Arquitectura Cliente/Servidor para Bases de Datos Relacionales Difusas*

J. M. Medina, J. C. Cubero, O. Pons, M. A. Vila
Dpto. de Ciencias de la Computación e Inteligencia Artificial
E.T.S. de Ingeniería Informática.
Universidad de Granada 18071. Granada (Spain)
email: {medina,carlos,opc,vila}@robinson.ugr.es
http://pirata.ugr.es/difuso/difuso.html

Resumen

En esta comunicación se propone e ilustra una arquitectura de SGBDRD cuyas principales características son: 1) se implementa completamente sobre SGBDR clásicos, con los que se integra perfectamente utilizando unicamente los recursos que estos proporcionan, 2) respeta todas las operaciones y características del SGBDR anfitrión y las potencia con nuevas capacidades para tratar información "difusa" y 3) proporciona un entorno para el desarrollo de aplicaciones que exploten información difusa.

Palabras clave: *Bases de datos relacionales, sistemas de bases de datos difusas.*

1 Introducción

Tomando como punto de partida el modelo de Bases de Datos Relacionales, propuesto por Codd, han surgido diversas aproximaciones, que tratan de proporcionar un marco teórico para la representación y tratamiento de información de naturaleza difusa. Los enfoques aparecidos, que se basan en el empleo de la teoría de subconjuntos difusos como herramienta de representación y manipulación, se agrupan principalmente en torno a dos tendencias: *Modelos de Unificación Mediante Relaciones de Similitud* y *Modelos Relacionales sobre Distribuciones de Posibilidad.*

La primera línea se encuentra recogida en la propuesta de Buckles & Petry [1]. Dentro del segundo enfoque existen varias aproximaciones segun las propuestas de Umano, [11], la de Prade y Testemale, [10] y la contribución de Zemankova en [12].

En [5, 6], presentamos un modelo, denominado GEFRED, que pretende aglutinar los aspectos más relevantes de los anteriores enfoques en un marco teórico común.

Los Sistemas Gestores de Bases de Datos Relacionales Difusas, (SGBDRD en lo sucesivo), extienden las capacidades de los SGBDR clásicos a fin de que puedan tratar información difusa. Se basan en un modelo teórico y en unas directrices de construcción con las que se pueda abordar una implementación operativa. Existen varias propuestas en este sentido, en su mayoria defendidas por los propursores de los diferentes modelos teóricos [2, 3, 9, 10, 11, 12].

En [4] se proponen una serie de directrices para representar e implementar datos y operaciones difusas en SGBDR clásicos dentro del marco teórico establecido en GEFRED.

En este trabajo se presentan algunas ideas para integrar los elementos del tratamiento de información imprecisa en el ámbito de las modernas arquitecturas de gestión de bases de datos y de sistemas de información en general. En concreto, se propone una arquitectura de SGBDRD basada en el enfoque Cliente/Servidor y se plantean los requisitos que deben reunir los módulos la componen para resolver los diferentes aspectos envueltos en el tratamiento de la información (difusa o no): acceso en línea a los datos, desarrollo de aplicaciones de explotación, soporte a la decisión, accesibilidad de datos desde otras aplicaciones, etc.

Basándonos en la arquitectura introducida en [8] describiremos en el próximo apartado los elementos que la componen. En el apartado tercero estableceremos los requisitos mínimos que debe reunir un SGBDR para que pueda ser empleado en la implementación de los componentes Servidor y Cliente del SGBDRD propuesto, daremos algunas ideas para la construcción de los mismos y estudiaremos técnicas de comunicación entre ambos componentes en entornos de red. En el apartado cuarto ilustraremos algunos aspectos de la construcción de la arquitectura propuesta sobre el SGBDR Oracle7 y su entorno de desarrollo. Por último analizaremos las ventajas que presenta nuestra aproximación y estudiaremos futuros trabajos que la desarrollen.

2 Descripción General de la Arquitectura

La figura 1 muestra el esquema general del SGBDRD propuesto bajo la denominación de FIRST (Fuzzy Interface for Relational SysTems) [8]. Es preciso aclarar que consideramos clientes a las aplicaciones que explotan los datos del servidor y no

*Este trabajo ha sido financiado por la CICYT con cargo al Proyecto TIC94-1347

a las herramientas de desarrollo que proporciona dicho servidor. Los elementos principales de la arquitectura de FIRST son el Servidor de FSQL y los interfaces de programación que éste proporciona. Ambos se construyen mediante los recursos que proporciona el SGBDR que actúa como "host" y deben estar fuertemente acoplados con el servidor de SQL y el interfaz de programación de éste. En torno al servidor de FIRST se construyen las aplicaciones clientes que se encargan de la explotación de los datos que éste gestiona. A continuación describiremos los elementos que integran FIRST atendiendo a su funcionalidad, estructura e implementación:

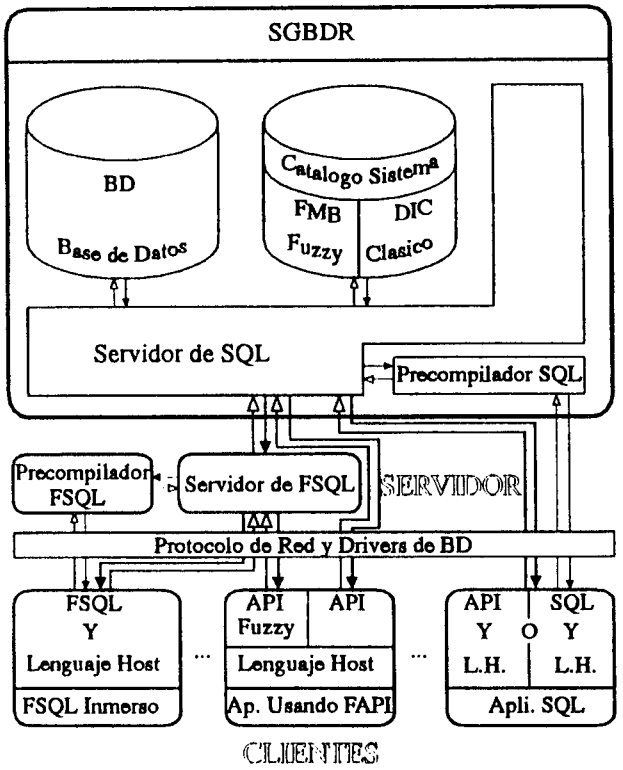


Figura 1: Esquema General de FIRST

SGBDR Anfitrión Es el sistema usado como base para la implementación de FIRST. Proporciona los recursos para la gestión de BD relacionales y un conjunto de herramientas básicas para la programación de aplicaciones SQL con lenguajes "host", (mediante un API o una especificación SQL inmerso). A la sintaxis SQL que incorpora se le dota de una extensión difusa y se la utiliza en la implementación de las rutinas de conversión de sentencias FSQL a SQL. En el próximo apartado analizaremos algunas características adicionales que debe poseer el SGBDR para facilitar la implementación de un servidor FSQL adecuado.

BD Almacena toda la información permanente, "difusa" o no, mediante el empleo del esquema relacional. Se adoptan determinados criterios de representación para implementar los datos "difusos" en dicho esquema de datos.

FMB La Base de Metaconocimiento Difuso es una extensión del catálogo del SGBDR anfitrión cuya misión es la de proporcionar información a FIRST acerca de todas las estructuras, datos y definiciones difusas que alberga la BD. Toda esta información se organiza de acuerdo con el esquema del catálogo del SGBDR anfitrión.

Servidor de FSQL Este módulo extiende la capacidad del Servidor de SQL anfitrión para que pueda procesar peticiones difusas al sistema expresadas en FSQL. Para llevar a cabo esta taréa dispone de un procesador para la sintaxis FSQL adoptada y de un protocolo para la comunicación con las aplicaciones clientes que incorporen código FSQL. Dicho protocolo puede estar basado en una extensión del API proporcionado por el SGBDR anfitrión o en una sintaxis de FSQL inmerso para la que proporciona diferentes precompiladores. El Servidor de FSQL contiene los siguientes módulos:

- **Procesador de FSQL** Transforma sentencias expresadas en FSQL en sentencias expresadas en el SQL del SGBDR anfitrión, las ejecuta y entrega los resultados a la aplicación cliente.
- **FCL** (Fuzzy Call Library) es una librería de funciones que contiene las rutinas que permiten realizar las tareas de interfase soportadas por el Servidor de FSQL.

Precompiladores de FSQL Son un conjunto de programas que, en interacción con el Servidor de FSQL, permiten transformar aplicaciones escritas en FSQL inmerso un lenguaje anfitrión en aplicaciones pueden ser compiladas con el compilador de este lenguaje anfitrión y enlazadas con la biblioteca basada en el FCL.

Protocolo de Red y "Drivers" de BD FIRST puede ser portado a cualquier plataforma a la que lo esté el SGBDR anfitrión y debe soportar los protocolos de comunicación a nivel de red y de "drivers" de BD que éste soporte. En el apartado 3 incideremos sobre estos aspectos.

Los clientes de FIRST pueden acceder al sistema empleando una o varias de las siguientes alternativas:

FSQL Inmerso Aplicaciones escritas en un lenguaje anfitrión que incorporan sentencias FSQL inmerso. FIRST proporciona dicha sintaxis y precompiladores para los lenguajes más importantes.

FAPI Fuzzy Application Programming Interface (Interfaz de programación de aplicaciones difusas) para incorporar en las aplicaciones clientes acceso al servidor de FSQL. Es conveniente que este interfaz esté lo más fuertemente acoplado posible al API del SGBDR anfitrión.

Acceso no difuso Aplicaciones que precisan un acceso convencional al SGBDR pueden emplear directamente los interfaces proporcionados por éste, (API o/y SQL inmerso).

3 Implementación de FIRST

La especificación de FIRST se asienta sobre cuatro pilares: a) el modelo teórico adoptado, GEFRED [5], b) los criterios de representación para la información difusa (datos y operaciones) [4, 6], c) la descripción de un sublenguaje de datos (FSQL) [7] y d) la implementación de todo esto empleando los recursos proporcionados por un SGBDR convencional. En este apartado abordaremos algunos aspectos concernientes al punto d), para ello plantearemos, en primer lugar, las especificaciones a satisfacer por los componentes de FIRST y, despues estableceremos los requisitos mínimos que debe reunir un SGBDR anfitrión para facilitar la implementación de los mismos.

3.1 Especificaciones de FIRST

Los elementos más importantes de la arquitectura propuesta son: el Servidor de FSQL, el interfaz de programación para aplicaciones "difusas" y los protocolos de comunicación entre clientes y el Servidor de FSQL en entornos de red. Estableceremos las especificaciones de estos elementos y en función de esto y de las técnicas de implementación introducidas en [4, 6, 7, 8] plantearemos los recursos a proporcionar por el SGBDR Anfitrión. Veamos las especificaciones:

- **Servidor de FSQL**

1. Proporcionará una extensión de la sintaxis de SQL que permitirá procesar sentencias SQL anfitrión combinadas o no con cláusulas para el tratamiento de información difusa.
2. Incorporará un procesador para esta sintaxis FSQL.
3. Este procesador transformará una sentencia FSQL en un conjunto de sentencias ejecutables por el SGBDR anfitrión.
4. Almacenará en el Gestor anfitrión los datos difusos con que opere, así como la información que precise para operar con ellos.
5. Es conveniente que tanto el procesador de FSQL como las rutinas empleadas en su construcción esten implementadas en un lenguaje proporcionado por el Gestor anfitrión y que estén almacenadas en el servidor del mismo.

- **FAPI**

1. Estará constituido por una especificación para abordar la construcción de aplicaciones que tengan acceso al Servidor de FSQL.
2. La especificación será un superconjunto del API que proporcione el Gestor anfitrión. Las operaciones adicionales que incorporen vendrán determinadas principalmente por el tratamiento de los datos difusos en su transferencia del Servidor a la aplicación cliente.
3. Se compondrá de una biblioteca basada en la FCL que habrá que incorporar a estas aplicaciones. Esta biblioteca podrá estar desarrollada en todos los lenguajes que soporte el API anfitrión e incorporará de forma transparente el acceso a través de la red.
4. La biblioteca del FAPI estará desarrollada utilizando el API anfitrión a través de la biblioteca que éste proporcione.
5. Por este motivo, resulta interesante que el API del Gestor anfitrión esté desarrollado en un lenguaje que permita la reutilización del código en la mayor medida posible. Un lenguaje orientado a objetos con su biblioteca de clases facilitará en gran medida el desarrollo.

- **Protocolo de Comunicación**

1. La comunicación entre el Servidor de FSQL y los clientes se realizará sobre los protocolos de red estándar soportados por el SGBDR anfitrión.
2. Esta comunicación debe poder hacerse mediante los "drivers" proporcionados por el Gestor anfitrión. Estos "drivers" pueden ser "nativos" (sólo permiten la comunicación entre clientes de bases de datos con servidores de un único fabricante), o pueden estar basados en un protocolo de comunicación abierto que permite el acceso a servidores de diferentes fabricantes, por ejemplo ODBC (Open Data Base Connectivity).
3. El empleo del protocolo de comunicación será transparente para el usuario.

3.2 Requisitos del SGBDR Anfitrión

Para satisfacer estas especificaciones el SGBDR anfitrión debe reunir los siguientes requisitos:

- **Mínimos**

1. Soporte de una sintaxis de SQL lo más potente posible.
2. Arquitectura Cliente/Servidor.
3. Disposición de alguna alternativa para desarrollar aplicaciones en lenguajes "host" que permitan el acceso a los datos del servidor mediante SQL (API o SQL inmerso).
4. Implementación de algun protocolo de comunicación de datos entre Servidor y Cliente desde distintas plataformas.

La práctica totalidad de los SGBDR actuales cumplen sobradamente estos requisitos lo que permite la implementación de la arquitectura propuesta sobre una variedad de gestores, en [6] puede encontrarse un ejemplo. Sin embargo, si perseguimos un perfecto acoplamiento de FIRST con el SGBDR anfitrión resulta conveniente que éste reuna la mayor parte de los siguientes requisitos:

- **Adicionales**

1. Que disponga de una extensión procedural de SQL. Esta extensión acopla el lenguaje de datos con características propias de los lenguajes de 3ª Generación (C, Pascal, etc).

2. Que presente la capacidad de almacenar en la base de datos procedimientos y paquetes escritos en esa extensión procedural.

3. Que proporcione un interfaz de programación con el que se puedan construir aplicaciones que accedan y ejecuten estos procedimientos almacenados.

4. Que dicho interfaz soporte al menos el empleo de un protocolo "nativo" para el acceso a los datos y procedimientos almacenados en el servidor. Sería deseable que también aceptase un protocolo abierto.

La evolución actual de los SGBDR asegura la presencia de estas características en la mayoría de ellos a través de múltiples plataformas. Todo esto garantiza la construcción de versiones de FIRST fuertemente acopladas a los SGBDR anfitrión para todas las combinaciones posibles de máquinas, sistemas operativos, SGBDRs y entornos de red, permitiendo a su vez la transferencia de información de una plataforma a otra.

4 Ejemplo de Implemetación sobre un SGBDR concreto

Para ilustrar las posiblidades de implementación que presenta FIRST vamos a mostrar la arquitectura completa de una solución que proporciona acceso al servidor de FSQL en una máquina remota desde una aplicación escrita sobre plataforma Windows™ local. Del lado del servidor vamos a emplear el SGBDR Oracle7™ sobre SunSPARC™ bajo Solaris™ 2.x y del lado del cliente utilizaremos los Oracle Objects™ para Windows. Para acceder a los datos del Servidor Oracle7 a través de la red emplearemos SQL*Net V1™ para TCP/IP (el protocolo de conexión "nativo" de Oracle). La figura 2 muestra la arquitectura adoptada en el ejemplo.

En primer lugar describiremos los elementos que conforman la arquitectura y la forma en que empleamos las herramientas de desarrollo proporcionadas por Oracle7 para construirlos, por último, estudiaremos el proceso de resolución de una consulta FSQL genérica a través de los diferentes módulos.

4.1 Elementos de la Arquitectura

Procesador en línea de FSQL Esta aplicación cliente acepta las peticiones en FSQL del usuario las transforma en sentencias PL/SQL (la extensión procedural de SQL proporcionada por Oracle) y las envía al componente servidor de OLE desarrollado por Oracle para aceptar accesos a bases de datos desde Windows. Esta aplicación puede estar desarrollada en Visual Basic, en Borland C++ o en Visual C++, o en cualquier combinación de las mismas. Oracle Objects para OLE proporciona las DLL, controles VBX y las bibliotecas de clases C++ precisas para ello.

Oracle Object Server Es un servidor OLE 2.0 desarrollado sobre el servidor OLE 2.0 para Windows y resuelve todas las peticiones de acceso a un servidor Oracle efectuadas desde una aplicación.

Oracle Call Interface Los servicios interceptados por el módulo anterior son transformadas por él a llamadas soportadas por este interfaz de programación. Éste módulo se encarga de establecer la comunicación a través la red mediante el empleo de los drivers nativos de Oracle para el entorno de que se trate.

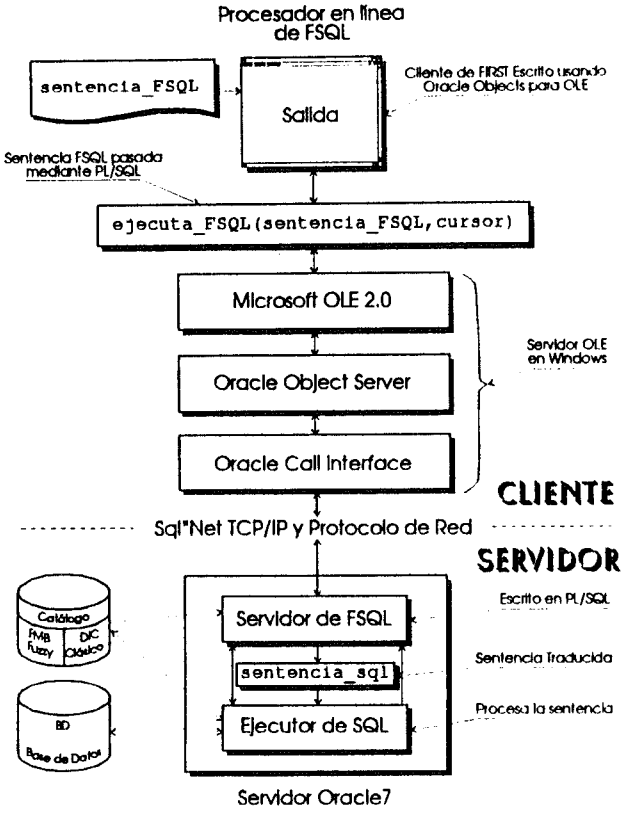


Figura 2: Implementación de FIRST sobre Oracle7

Sql*Net Éste está construido sobre el protocolo de red e intercepta, de forma transparente, las peticiones dirigidas al servidor de Oracle, liberando al desarrollador de la taréa de implementar a bajo nivel en las aplicaciones el acceso a los datos del servidor a través de red.

Servidor de FSQL Se encarga de resolver las peticiones en FSQL y traducirlas a sentencias SQL ejecutables por Oracle. Está desarrollado en PL/SQL e incorpora un procesador sintáctico y semántico para la sintaxis FSQL adoptada. Una vez traducida la sentencia, la envía al ejecutor de SQL del cual obtiene los resultados los cuales, debidamente procesados, son enviados al cliente a través de un mecanismo de comunicación soportado por el concepto de cursor. El servidor de FSQL puede requerir en diversas ocasiones de los servivios del ejecutor de SQL y, por supuesto, de los servicios de la máquina de PL/SQL incorporada en el servidor Oracle (no mostrada en la figura 2). El servidor de FSQL no sólo está escrito en PL/SQL, sino que también se encuentra

precompilado y almacenado en el servidor de Oracle.

Es preciso decir que de todos los elementos presentes, sólo es preciso implementar los módulos correspondientes al procesador en línea de FSQL y al Servidor de FSQL. Los demás se encuentran disponibles como elementos de soporte y de desarrollo de la arquitectura de FIRST.

4.2 Procesado de una sentencia FSQL

A continuación, apoyándonos en la figura 2, seguiremos la evolución a lo largo del esquema del procesamiento de una consulta expresada en FSQL.

1. El Procesador en línea de FSQL tras recibir una entrada, realiza un análisis sintáctico para comprobar que es correcta y transforma la entrada en uno o varios procediminetos en PL/SQL de Oracle.

2. Seguidamente, solicita su ejecución mediante el Oracle Objects Server implementado en el cliente Windows.

3. Éste intercepta la llamada y convierte la petición en un conjunto de rutinas del Oracle Call Interface.

4. Estas rutinas detectan de forma automática la presencia del soporte de red SQL*Net y utilizan este driver para enviar, a través del protocolo de la red, la petición al SQL*Net disponible en la máquina que contiene el Servidor de Oracle7.

5. Éste recibe la petición y la envía al Servidor Oracle7 para su procesamiento.

6. Como esta petición se compone de rutinas PL/SQL que constituyen el paquete que implementa el Servidor de FSQL, será éste el que se encargue de ejecutar la petición.

7. El procedimiento invocado desde el procesador cliente tiene por misión analizar la sentencia FSQL que lleva por argumento y, caso de ser correcta ella y los datos que referencia, proceder a la traducción de la misma a una sentencia SQL, que luego será enviada al ejecutor de SQL proporcionado por el Servidor Oracle7. Este proceso de análisis y traducción hace uso de la máquina PL/SQL, del ejecutor SQL, de la base de datos y del catálogo "difuso" implementado en el diccionario del sistema.

8. La ejecución de la sentencia traducida por parte del ejecutor de SQL puede ocasionar la recuperación de una serie de tuplas que deben ser devueltas al Procesador en Línea de FSQL para su edición y presentación al usuario.

9. Este proceso sigue un camino inverso al descrito y se muestra a dicha aplicación cliente como un conjunto de tuplas almacenado mediante una estructura cuyo empleo viene determinado a través del protocolo que proporcionan los Oracle Objects para OLE.

10. La integración de este protocolo con las características visuales de Windows permitirá la presentación de los datos en una forma adecuada a cada necesidad.

5 Conclusiones

La arquitectura presentada en este trabajo proporciona un buen punto de partida para la construcción de SGBDRD. Esto es así porque aprovecha los recursos disponibles en los SGBDR actuales y consigue incorporar en ellos mecanismos de para el tratamiento de información difusa de una forma transparente para el usuario. Cara a éste, la arquitectura propuesta proporciona una extensión difusa de SQL y unas herramientas de desarrollo con las que puede construir aplicaciones que exploten información de naturaleza difusa o/y clásica. De esta forma conseguimos SGBDRD que preservan todas las operaciones y características de los actuales, (enfoque Cliente/Servidor, gestión distribuida de la información, integridad y seguridad de los datos, etc.) potenciándolos con la posiblidad de manejar información imprecisa.

Como planes de trabajo inmediato, nos planteamos el desarrollo de prototipos sobre plataformas concretas para los elementos más importantes de esta arquitectura: el Servidor de FSQL, procesador en línea de FSQL, una versión visual para procesar consultas imprecisas y herramientas de desarrollo de aplicaciones FSQL.

Referencias

[1] Buckles B.P., Petry F.E. "A Fuzzy Representation of Data for Relational Databases". Fuzzy Sets and Systems, 7. 213-226. (1982)

[2] Hamon Guy. "Extension d'un language d'interrogation de Base de Donnees en vue vue de l'utilisation de questions imprecises", Tesis Doctoral. (1986).

[3] Li D., Liu. "A Fuzzy Prolog Database System", ed. John Wiley & Sons. New York. (1990)

[4] Medina J. M., Vila M.A., Cubero J.C., Pons O. "Towards the Implementation of a Generalized Fuzzy Relational Database Model". Por aparecer en Fuzzy Sets & Systems. (1995)

[5] Medina J. M., Pons O., Vila M.A. "GEFRED. A Generalized Model of Fuzzy Relational Data Bases". Information Sciences,76,1-2, pp 87-109. (1994)

[6] Medina J. M. "Bases de Datos Relacionales Difusas: Modelo Teórico y Aspectos de su Implementación". Tesis Doctoral. Universidad de Granada. (1994)

[7] Medina J. M., Pons O., Vila M.A., "Un Procesador Elemental de Fuzzy SQL". 4º Congreso Español sobre Tecnologías y Lógica Fuzzy. Blanes (Gerona).(1994)

[8] Medina J. M., Pons O., Vila M.A. "FIRST. A Fuzzy Interface for Relational SysTems". Proceedings del 6º Congreso la International Fuzzy Systems Association IFSA 95. (1995)

[9] Nakajima Hiroshi, Sogoh Taiji, Arao Masaki. "Fuzzy Database Language and Library - Fuzzy Extension to SQL -" Proceedings del Second IEEE International Conference on Fuzzy Systems, pp. 477-482. (1993)

[10] Prade H., Testemale C. "Generalizing Database Relational Algebra for the Treatment of Incomplete/Uncertain Information and Vague Queries". Information Sciences, 34. 115-143. (1984)

[11] Umano M. "Freedom−0 : *A* Fuzzy Database System". Fuzzy Information and Decision Processes. Gupta-Sanchez edit. North-Holland Pub. Comp. (1982)

[12] Zemankova M., Kandel A. "Fuzzy Relational Data Bases - A Key to Expert Systems". Verlag TUV Rheinland. (1984)

Una Aproximación Sistemática al Tratamiento de Conceptos y Cantidades Difusas

J.M. Cadenas, A. Gómez-Skarmeta, A.M. Camacho, L. Fernández, F. Martín
Dpto. de Informática y Sistemas
Facultad de Informática. Universidad de Murcia
30071-Espinardo. Murcia

Resumen

El diseño de sistemas difusos se ha realizado en general de una forma específica en relación a la aplicación, ya que una aproximación de tipo general depende de la determinación de los conjuntos difusos que intervienen. Por tanto de cara a conseguir que los sistemas difusos sean más genéricos, es necesario el desarrollo de un método sistemático para la caracterización de los conjuntos difusos implicados. En este trabajo proponemos el diseño de una herramienta de adquisición de dicho conocimiento impreciso.

Palabras Clave: adquisición del conocimiento, medidas de similaridad, comparación de conjuntos difusos.

1 Introducción

Los procesos son muchas veces controlados por operadores humanos los cuales tienen en general mucho conocimiento intuitivo acerca del comportamiento del sistema. El operador humano posiblemente pueda dar una descripción aceptable para controlar el sistema, aunque quizás no de forma precisa. Dichas descripciones se realizan en base a conceptos lingüísticos o bien cantidades de naturaleza imprecisa, que necesitan ser descritos para poder ser utilizados en un sistema difuso. La determinación preliminar de dichos conjuntos difusos puede realizarse en base a información suministrada por el usuario o el colectivo implicado, para posteriormente realizar sobre ella un proceso de ajuste (tuning) con objeto de adaptarla al sistema difuso concreto.

El problema de la adquisición tiene implicaciones también en una fase posterior del modelado difuso, ya que si se quiere realizar una aproximación lingüística existen múltiples medidas que pueden utilizarse para establecer la similaridad entre conjuntos difusos, y será deseable por tanto tener un sistema capaz de que de una forma dependiente del proceso en cuestión, sepa ayudar a la decisión del tipo de medida más adecuada.

Tenemos pues tres componentes a considerar:

1. Determinación de los conjuntos difusos.
2. Ajuste de los conjuntos en relación al problema en cuestión.
3. Determinación de la validez de las medidas existentes para capturar el modo de comparación entre cantidades difusas del o los individuos, y/o la similaridad entre conceptos difusos para ser utilizados en una aproximación lingüística.

En este trabajo, se describe una herramienta para el proceso sistemático de conceptos y cantidades difusas, que nos permite adquirir las expresiones de los conjuntos difusos, así como utilizarlos en la evaluación de diferentes métodos de comparación y ordenación de dichas cantidades, y para la caracterización en cada problema tanto de los conjuntos difusos implicados como de las medidas que mejor se adecuen en su tratamiento.

Para ello, en la sección 2 se planteará el enfoque general de problema, presentándose el esquema general del sistema de adquisición que planteamos en este trabajo. Posteriormente en la sección 3 se indica la Arquitectura del sistema que presentamos, para continuar con los resultados que se han obtenido al aplicar el sistema a colectivo de individuos. Concluimos en la sección 5 con algunas aspectos sobre las líneas de continuación de este trabajo.

2 Esquema de Adquisición del Conocimiento

2.1 Determinación del Problema

El conjunto de términos de una variable lingüística no es cerrado respecto a las operaciones conjuntistas o aritméticas (unión, intersección, suma, producto, ...) propias de los conjuntos difusos. Como consecuencia de ello la resolución de problemas con planteamientos imprecisos como sucede en el razonamiento aproximado, cuando las entradas son valores de variables lingüísticas, proporcionan una solución que es un subconjunto difuso de un cierto universo pero puede no ser un valor lingüístico de alguna variable. Parece razonable exigir que la salida sea una etiqueta y para conseguirlo hay que llevar a cabo lo que se conoce como Aproximación Lingüística que puede formularse como encontrar una etiqueta cuyo significado sea al menos lo más próximo posible al subconjunto difuso encontrado por el modelo.

En este sentido uno de lo enfoques mas utilizados en la literatura consiste en lo que se denomina la Mejor Aproximación, en la que se define una distancia entre subconjuntos difusos, la cual se utiliza para calcular la distancia entre el subconjuto difuso encontrado y cada una de las etiquetas del conjunto de términos

para finalmente asignar la etiqueta que minimiza la distancia.

Para poder determinar que medida se ajusta mejor al colectivo concreto de usuarios de un sistema parece adecuado estudiar las diferentes medidas de distancia (o similaridad) entre el conjunto de términos de la variable para de esta forma, cuando aparezca un subconjunto difuso cualquiera, saber que medida aplicar para calcular su distancia con el conjunto de términos.

Con respecto a la comparación de los números difusos, no existe un consenso tan amplio como puede aparecer con otros conceptos de la teoría de conjuntos Difusos. Diversos autores han realizado numerosos estudios. En [2, 9] se recogen algunos de ellos, a la vez que realizan un estudio comparativo de los mismos.

Cuando los soportes de los números difusos no intersecan, no existe ningún problema, y de hecho todos los métodos conducen a la misma solución. En cambio, la decisión al comparar no es evidente cuando efectivamente los soportes intersecan, y podemos encontrar métodos de comparación que aplicados sobre determinados ejemplos parecen contradecir la intuición de algún decisor.

Este es un problema complejo porque, dado el carácter impreciso de las cantidades, por ejemplo A y B, de antemano no puede garantizarse el que $A \leq B$, o el que $B \leq A$, sino que, ambas propiedades van a verificarse simultáneamente con ciertos grados de cumplimiento.

Esta situación respecto a conceptos y cantidades difusas ha conllevado hasta ahora la necesidad de tratar cada sistema de forma especial y única, lo cual nos ha planteado la necesidad de establecer un marco más general donde poder tratar el problema de la adquisición. Para ello hemos planteado la necesidad de proponer un sistema de entrevistas o pruebas a realizar sobre el colectivo de individuos del cual se desea obtener conocimiento. Dichas entrevistas o pruebas estarán enfocadas a adquirir el conocimiento relativo a la interpretación del concepto o cantidad difusa, así como del valor que el individuo atribuye a la comparación de cantidades o conceptos difusos.

Del catálogo de medidas de similaridad y de comparación existentes hemos seleccionado aquellos que, estando suficientemente contrastados en la literatura, nos han parecido más apropiados para el tema concreto que estudiamos. En los apartados siguientes introducimos brevemente estas medidas.

2.2 Proceso de Adquisición

La primera labor a realizar consiste en extraer la información sobre las funciones de pertenencia que las personas tenemos en mente cuando hablamos en términos lingüísticos (mucho, poco, bastante, etc.). Para ello hemos planteado un sistema automatizado de encuestas a los usuarios donde mediante un símil gráfico, mostramos al sujeto unas determinadas intensidades en el significado de la variable a captar y obtenemos los índices que asocia a cada uno de los grados, para así construir la función de pertenencia del concepto tratado. La función elegida para ajustar los puntos registrados, es una aproximación a una gaussiana compuesta por cuatro segmentos de parábola, con la siguiente expresión analítica y representación gráfica:

$$\pi(x; u, \underline{\beta}, \overline{\beta}, \underline{\Gamma}, \overline{\Gamma}) = \begin{cases} S(x; \underline{\Gamma}, \underline{\beta}, u) & \text{si } x \leq u \\ 1 - S(x; u, \overline{\beta}, \overline{\Gamma}) & \text{si } x > u \end{cases}$$

$$S(x; a, b, c) = \begin{cases} 0 & \text{si } x \leq a \\ \frac{1}{2}\left(\frac{x-a}{b-a}\right)^2 & \text{si } a < x \leq b \\ 1 - \frac{1}{2}\left(\frac{x-c}{b-c}\right)^2 & \text{si } b < x \leq c \\ 1 & \text{en otro caso} \end{cases}$$

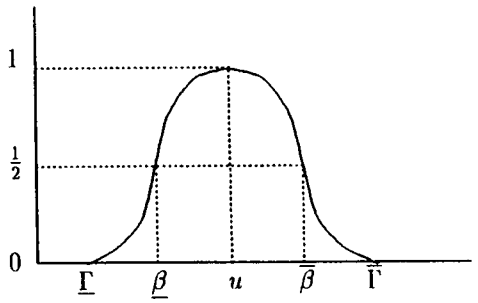


Figura 1: Función de pertenencia

Una vez realizados los ajustes de todas las funciones de pertenencia obtenidas, hemos pasado al cálculo de funciones de pertenencia generales, para llegar a la función de pertenencia de todo un concepto para el grupo entero.

Para sacar conclusiones sobre la validez o bondad de los estimadores teóricos de similaridad, hemos aplicado los índices más comunes que se encuentran en la literatura y que se muestran más adelante, realizando un estudio sobre lo adecuado de cada índice para cada concepto. La forma de obtener estas conclusiones ha sido mostrar a los entrevistados dos conceptos y registrar el grado de similitud entre ambos, comparando posteriormente con los resultados teóricos.

Con respecto a las cantidades difusas, nos basamos en una función de pertenencia triangular (aunque se podía haber extendido a otras funciones) y presentamos, al sujeto, dos de dichas cantidades, solicitándole que indique la relación existente entre ellas, es decir, si una es mayor que la otra, menor, o indiferentes. Una vez indicada dicha relación, se solicita que mueva una de ellas hasta que se cumpla la condición contraria, de forma que podamos sacar conclusiones sobre lo que ha hecho que se incline hacia uno u otro lado el individuo. De entre todos los índices teóricos que se muestran más adelante se elegirá aquél que mejor se comporte con lo indicado por el entrevistado en ambas respuestas.

2.3 Métodos para el cálculo de la similitud.

El análisis teórico de relaciones de similaridad ha sido dominado por los modelos geométricos. Estos modelos representan los objetos como puntos en algún espacio de coordenadas tal que la disimilitud observada entre objetos corresponde a la distancia métrica

entre los respectivos puntos. Esto extrapolado a los conjuntos difusos nos proporciona medidas de la disimilitud entre dichos conjuntos y si tenemos en cuenta que cada uno de ellos puede ir asociado a un valor lingüístico, obtenemos formas de comprobar cómo de cercanos en significado están dos variables lingüísticas.

A continuación mostramos las métricas que hemos utilizado.

-Métrica-r de Minkowski.

$$d_r(A,B) = \left(\int_{-\infty}^{+\infty} |\mu_A(x_i) - \mu_B(x_i)|^r \right)^{\frac{1}{r}} , r \geq 1$$

que representa un conjunto de distancias geométricas aplicadas a conjuntos difusos, siendo sus elementos más característicos aquellos con $r = 1, r = 2$, y $r = \infty$, resultando con éste último valor de r, la siguiente medida

$$d_\infty(A,B) = \sup_x |\mu_A(x_i) - \mu_B(x_i)|$$

-Métrica de Hausdorff.

$$q_1(A,B) = \int_{\alpha=0}^{1} q(A_\alpha, B_\alpha) d\alpha$$

donde

$$q(U,V) = \max \left\{ \sup_{v \in V} \inf_{u \in U} d_2(u,v), \sup_{u \in U} \inf_{v \in V} d_2(u,v) \right\}$$

y A_α es el α-corte del conjunto difuso A.

$$q_\infty(A,B) = \sup_{\alpha \geq 0} q(A_\alpha, B_\alpha)$$

También utilizaremos la distancia de Hausdorff entre los intervalos con mayor grado de pertenencia, es decir:

$$q_*(A,B) = q(A_{1.0}, B_{1.0})$$

-Indice de disimilitud.

Kaufman y Gupta definieron

$$\Delta_1(A,B) = \int_{\alpha=0}^{1} \Delta(A_\alpha, B_\alpha) d\alpha$$

donde

$\Delta(A,B) = \frac{(|a_1-b_1|+|a_2-b_2|)}{2(\beta_2-\beta_1)}$ con $A = [a_1, a_2], B = [b_1, b_2]$ y $[\beta_1, \beta_2]$ un intervalo que contiene tanto a A como a B.

Además utilizaremos las analogías que obviamente aparecen con la medida anterior:

$$\Delta_\infty(A,B) = \sup_\alpha \Delta(A_\alpha, B_\alpha)$$

$$\Delta_*(A,B) = \Delta(A_{1.0}, B_{1.0})$$

2.4 Métodos para comparación de cantidades difusas.

De la misma forma que comparamos la similaridad de dos conceptos difusos o variables ligüísticas, podemos comparar cantidades difusas representadas por un conjunto base y una función de pertenencia. Un problema de constante actualidad es el de la ordenación de cantidades imprecisas y por tanto el de la comparación de números difusos.

Hemos utilizado distintas formas de comparar números difusos para ver cuál de ellos se adecua mejor a la forma de actuar del usuario en un problema dado. Por tanto, del catálogo de métodos existentes hemos seleccionado algunos como botón de muestra para el tema que estudiamos. A continuación mostramos las medidas que vamos a utilizar:

Métodos sin participación del decisor.

-Medidas de Yager.

$$F_1(\tilde{u}_i) = \frac{\int_0^1 g(z)\mu_{\tilde{u}_i}(z)dz}{\int_0^1 \mu_{\tilde{u}_i}(z)dz}$$

$$F_2(\tilde{u}_i) = \max_{z \in S_i} \min(z, \mu_{\tilde{u}_i}(z))$$

$$F_3(\tilde{u}_i) = \int_0^1 M(U_i^\alpha) d\alpha$$

- Medida de Chang.

$$F(\tilde{u}_i) = \int_{S_i} z\mu_{\tilde{u}_i}(z)dz$$

Métodos con participación del decisor.

-Medida de Adamo.

$$F_\alpha(\tilde{u}_i) = \max \left\{ z \mid \mu_{\tilde{u}_i}(z) \geq \alpha \right\}$$

3 Arquitectura del sistema 'TRUCO' (Tratamiento Unificado de Conjuntos Difusos).

3.1 Esquema General

El sistema **TRUCO** (Tratamiento Unificado de Conjuntos difusos) que nos ha permitido obtener los resultados, consta de tres módulos. El primero es el encargado de la adquisición del conocimiento, realizando lo indicado en la sección anterior, es decir, mostrando al individuo diversas situaciones e interrogándole sobre la función de pertenencia en cada caso.

El segundo recibe los datos sobre las funciones de pertenencia obtenidos por el primer módulo, y realiza los cálculos necesarios para obtener una función genérica para cada concepto, que sea representativa de todas las funciones obtenidas de los sujetos para el mismo concepto.

También proporciona las medidas de similitud de conceptos y comparación de cantidades mostradas anteriormente para los tipos de funciones de pertenencia que se contemplan, así como los cálculos necesarios para, a partir de los datos, obtener la función de pertenencia que mejor se ajuste a la nube de puntos y utilizarla en las medidas.

El tercer y último módulo es una puerta abierta a las posibles aplicaciones de los otros dos módulos en diversos campos, y proporciona la facilidad de añadir a cualquier proyecto que precise un conocimiento difuso, la posibilidad de obtener las funciones de pertenencia que mejor que ajusten al tema de que se trate.

El esquema del sistema descrito es el siguiente:

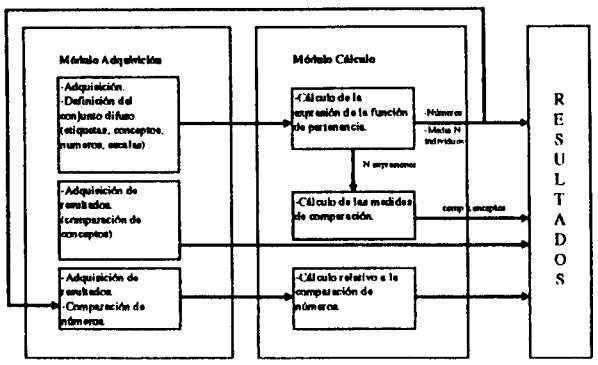


Figura 2: Arquitectura del Sistema

3.2 Descripción de la herramienta

La herramienta ha sido diseñada teniendo en cuenta que debe ser amigable desde el punto de vista del usuario, y lo suficientemente explícita como para que el entrevistado comprenda a simple vista la información que se pretende extraer con las pruebas a las que se está sometiendo, para ello hemos programado bajo windows utilizando las facilidades para diseño de interfaces que éste brinda.

Por otra parte, también hemos considerado que las pantallas que se muestren no deben influir en medida alguna en las decisiones que tome el individuo, con la finalidad de que la información extraída refleje realmente el punto de vista del mismo. Por ejemplo, hemos cuidado en evitar mencionar números en la graduación de cada medida.

En la pantalla principal se ofrece la posibilidad de escoger cualquiera de las tres pruebas que hemos preparado, importar para regenerar los ficheros a partir de datos tabulados, exportar los datos que permite hacerlos accesibles a hojas de cálculo y paquetes estadísticos y, por último, utilizando la opción resultados se genera un fichero de texto con las funciones de pertenencia introducidas por los usuarios, agregadas por las funciones media y máximo.

En la primera prueba aparece en la pantalla una etiqueta. Basándose en la idea que el entrevistado tiene de dicha etiqueta, debe señalar a través de la barra de desplazamiento horizontal en qué grado se adecua la figura al concepto indicado, dicha barra tiene un rango de desplazamiento de cien posiciones distintas para permitir una graduación flexible. También hemos indicado el sentido de la barra por medio de dos adverbios "completamente" y "nada" para no dar lugar a confusión. Una vez que ha grabado su opinión, automáticamente se le va planteando la misma cuestión con la figura un poco más cubierta (ver figura(3)).

En la segunda prueba pasamos a evaluar la idea que los entrevistados poseen sobre el grado de similitud entre todos los pares de etiquetas sobre los que anteriormente se les ha encuestado (ver figura(4)).

En la tercera prueba hemos intentado extraer información sobre comparación de números difusos pero sin especificar los números concretos de que se trata, para lo cual hemos utilizado las etiquetas A y B.

Se han escogido las configuraciones que poseen algún tipo de conflictividad, ya que en los casos triviales todos los métodos están de acuerdo entre si y normalmente, salvo excepciones, con los entrevistados.

Cuando se le presentan los dos números difusos al entrevistado, debe decidir qué relación cumplen, y además debe mover uno de ellos hasta que dicha relación deje de cumplirse (ver figura(5)).

La herramienta también tiene la posibilidad de mostrar cómo queda la función de pertenencia para un concepto y un sujeto, una vez ajustada en base a su nube de puntos (ver figura(6)).

En el momento en que los sujetos han pasado los tests que nos han servido para extraer la información sobre las etiquetas que pretendemos manejar, pasamos a obtener resultados a partir de los datos de entrada.

4 Utilizando el sistema de adquisición TRUCO: Análisis de resultados

Debido a que se trabaja con un grupo de individuos que representa a una población y que deseamos que los resultados sean extrapolables, debemos enfocar nuestro esfuerzo a realizar un estudio estadístico del comportamiento de los datos recogidos para poder llegar a conclusiones correctas. En nuestro caso la muestra consiste en 83 personas que se prestaron a pasar la entrevista.

4.1 Resultados de la selección de métodos teóricos para comparar conceptos difusos.

Para evaluar la herramienta la hemos aplicado a un colectivo de 83 alumnos de informática, y a un rango de siete conceptos difusos relativos a capacidad (lleno, medio lleno,..., vacío). Una vez analizada la forma y adecuación de las funciones de pertenencia asociadas a los conceptos que estabamos tratando, se procedió a analizar las respuestas dadas en relación a la similaridad de los conceptos.

Dado que las respuestas de los individuos hay errores u opiniones que modifican los resultados reales, por lo que decidimos que la mejor medida para evitar esto es considerar como representante de la opinión general la media de las respuestas que a cada comparación indicó cada individuo.

Cada uno de los métodos de comparación obtuvo un coeficiente de regresión que da una medida de su bondad. El orden que tomaron los métodos de mejor a peor fue $(q_1, q_\infty, \Delta_1, \Delta_\infty, \Delta_*, q_*, d_1, d_2, d_\infty)$ donde

hemos tomado como función de agregación la media de las funciones de pertenencia.

Se aprecia que los métodos que han dado mejor resultado son aquellos que tienen en cuenta el eje x (sobre el que se encuentra el soporte), utilizando los α-cortes. Esto es bastante razonable, ya que cuando tenemos dos conceptos con soportes prácticamente disjuntos, los métodos que no usan información sobre la coordenada x pierden la capacidad de distinguir grados de disimilitud.

El método q_∞ es el que presenta el segundo mejor funcionamiento, a pesar de concentrar toda la información en un único valor (α-corte donde la distancia x es mayor entre los dos conceptos) en lugar de realizar algún tipo de media o integración. El que este método haya obtenido tan buen resultado reafirma la opinión de que la forma de pensar de los sujetos puede ser algo tan simple como reducir las complicadas funciones de pertenencia a un único corte que se practica intuitivamente.

Este resultado coincide con lo afirmado en [Zwi87], donde las medidas sugeridas como buenas son $(q_\infty, q_*, \Delta_\infty, \Delta_*)$, dentro de las cuales se encuentra q_∞ que es una de las que mejor resultado ha producido. Por el contrario, medidas como q_* y Δ_∞ que en el mencionado artículo muestran un buen comportamiento, en nuestro caso no lo tienen.

El que los métodos q_* y Δ_* no tengan un rendimiento demasiado bueno puede ser debido a lo simple de su forma. A pesar de ser computacionalmente triviales, no parecen proporcionar información suficiente como para distinguir de una forma adecuada los conceptos, o por lo menos aquellos que tengan asociada una función de pertenencia como la que nos ocupa.

Se ha estudiado también la función de agregación máximo, ésta tiene un comportamiento muy sensible a las respuestas de individuos aislados, que la hacen poco utilizable cuando se pretende extraer información de una muestra en cuanto el número de elementos haga bastante probable que haya sujetos que provoquen una desviación en los resultados. Comprobamos que al reducir la muestra a los 22 elementos menos conflictivos, los coeficientes de regresión se hacen mejores, dando como resultado que los tres métodos que mejor funcionan con este tipo de agregación son: q_∞, Δ_∞, Δ_1. Comparando con los tres que obtuvimos con la función de agregación media $(q_1, q_\infty, \Delta_1)$ observamos que tanto q_∞ como Δ_1 se encuentran en los dos resultados. Esto nos muestar que ambas funciones de agregación se comportan bastante bien, pero la función máximo es bastante inestable. Además advertimos que q_∞ aparece de nuevo, confirmando su buen comportamiento.

4.2 Resultados de la selección de métodos teóricos para comparar cantidades difusas.

Para el estudio de los resultados obtenidos en las pruebas de comparación de números difusos, se consideró que la respuesta correcta era la moda muestral, ya que corresponde con lo que piensa la mayor parte de los sujetos.

Después de analizar las respuestas y compararlas con los métodos teóricos, obtuvimos las siguientes tablas, donde se compara el método con el número de aciertos en relación a los 21 pares:

Método	Ac.
Yager 3	8
Yager 1	7
Adamo 0.5	7
Adamo 0.75	7
Yager 2	7
Adamo 0.25	6
Chang	6
Adamo 0.0	5
Adamo 1.0	4

Método	Ac.
Yager 3	7
Adamo 0.75	7
Adamo 0.5	7
Yager 1	6
Adamo 0.0	6
Adamo 0.25	6
Adamo 1.0	6
Chang	6
Yager 2	6

Con respecto a las cantidades difusas, debemos destacar de entre los métodos utlizados los siguientes: Tercera medida de Yager, Adamo con $\alpha = 0.5$ y Adamo con $\alpha = 0.75$. Estos tres métodos obtuvieron la mayor cantidad de aciertos en la ordenación de los pares de cantidades mostrados, tanto en los pares originales como en los complemenetarios.

Por tanto podemos afirmar que la mayoría de los sujetos realizaron sus cálculos basándose en el área de la función de pertenencia, ya que es así como trabaja la tercera medida de Yager (calcula el área de la función formada por el punto medio de los α-cortes).

El siguiente método que mejor modela la forma de pensar de la gente es Adamo con $\alpha = 0.5$, consistente en tomar el máximo valor del eje x donde la función de pertenencia supera al 0.5-corte. A continuación se encuentra Adamo con $\alpha = 0.75$.

Por el contrario, Adamo con $\alpha = 0$, Adamo con $\alpha = 0.25$ y Adamo con $\alpha = 1$ no han producido buenos resultados, por lo que se podría pensar que la muestra se ha fijado en algún punto cercano al centro del flanco derecho del triángulo que forma la función de pertenencia de las cantidades difusas.

Al reducir la muestra a 22 sujetos poco conflictivos, hemos obtenido los mismos resultados, lo que nos da más confianza en los mismos.

5 Conclusiones y Desarrollos Futuros

Frente a los enfoques clásicos de especificación de los conjuntos difusos que intervienen en un sistema difuso de una forma ad-hoc, nosotros planteamos abordar el problema de una forma sistemática, con el diseño de una herramienta de ayuda al modelado. Dicha herramienta nos permite en primer lugar afrontar la captura del conocimiento intuitivo tanto de las funciones de pertenencia de los conceptos y cantidades difusas que intervienen en el sistema, como de las adecuación de las diferentes medidas teóricas existentes de comparación, en relación al problema concreto. En segundo lugar la herramienta nos permite también el disponer de unos conjuntos difusos inciales sobre los que trabajar posteriormente en un proceso de ajuste, utilizando técnicas como redes neuronales, o algoritmos genéticos.

Se han dejado varias puertas abiertas a un perfeccionamiento en el desarrollo del sistema TRUCO. No se ha implementado el tratamiento de escalas ni la realimentación de la salida del módulo de cálculo para poder realizar ajustes desde la herramienta.

Actualmente se han utilizado paquetes estadísticos para tratar los datos obtenidos de la herramienta y

llegar a conclusiones, pero lo idóneo sería que la herramienta contemplara la posiblidad de enlazar con dichos paquetes desde el menú o bien incluyera los módulos necesarios para fines de análisis. Esto permitiría cerrar el ciclo mostrado en la arquitectura del sistema, permitiendo realizar, gracias a la interacción conseguida, análisis preliminares para detectar sujetos conflictivos, eliminarlos y ver cómo ha variado el comportamiento de las salidas.

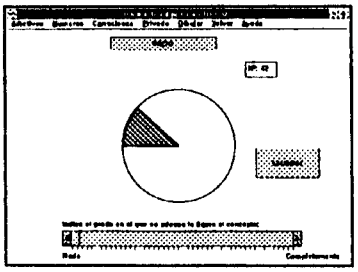


Figura 3: Adquisición de las funciones de pertenencia



Figura 4: Comparación de Conceptos

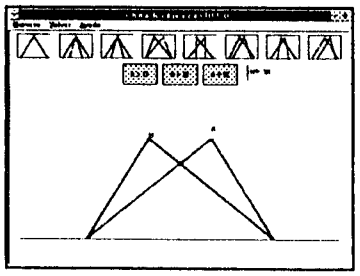


Figura 5: Comparación de Cantidades

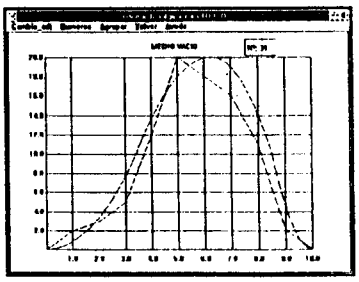


Figura 6: Función de Pertenencia adquirida

Referencias

[1] **P.P. Bonissone**, A Pattern Recognition Approach to the problem of Linguistic Approximation in System Analysis. *Proceedings of the IEEE Int. Conference on Cybernetics and Society (1979), 793-798.*

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

[3] **J.M. Cadenas**, Diseño e implementación de un Sistema Interactivo para resolver problemas de Optimización Difusa, *Tesis Doctoral, (1993) Universidad de Granada.*

[4] **D.P. Mandal, C.A. Morthy, S.K. Pal**, Formulation of a Multivaluated Recognition System. *IEEE Transaction on Systems, Man, and Cybernetics 22 (4) (1992), 607-620.*

[5] **V.S. Manoranjan, A. de Sam Lazaro, D. Edwards, A. Athalye**, A Systematic Approach to Obtaining Fuzzy Sets for Control Systems. *IEEE Transaction on Systems, Man, and Cybernetics 25 (1) (1995), 206-213.*

[6] **R. Zwick, E. Carlstein and D.V. Budescu**, Measures of similarity among fuzzy concepts: A comparative analysis. *International Journal of Aproximate Reasoning 1 (1987), 221-242.*

[7] **I. Requena**, Redes Neuronales en problemas de decisión con ambiente difuso. *Tesis doctoral (1992). Universidad de Granada.*

[8] **L.A. Zadeh**, The Concept of Linguistic Variable and its Application to Approximate Reasoning, Part 1, 2, 3. *Info. Sci. 8,9 (1975), 199-249; 301-357; 43-98.*

[9] **Q. Zhu, E.S. Lee**, Comparison and Ranking of Fuzzy Numbers. *Int. J. Kacprzyk and M. Fedrizzi, Eds. Fuzzy Regression Analysis. Onmitech Press Warsan and Physica-Verlag (1992), 21-44.*

Aplicación de Operadores Estructurales a la Segmentación de Imágenes Aéreas

Domènec Puig , Eduard Montseny

Dept. d'Enginyeria Informàtica
Universitat Rovira i Virgili
Carretera Salou s/n, 43006 Tarragona, España
e-mail: dpuig@etse.urv.es

Pilar Sobrevilla

Dept. Matemàtica Aplicada II
Universitat Politècnica de Catalunya
Pau Gargallo 5, 08028 Barcelona, España

Resumen

En este trabajo se presenta una aplicación del análisis posibilístico de texturas a la segmentación de imágenes aéreas. El objetivo es identificar las zonas de imágenes aéreas donde aparecen plantaciones de árboles (árboles de cultivo).

En el análisis de imágenes aéreas de entornos naturales resulta particularmente difícil automatizar la extracción de las características relevantes de las texturas que aparecen en ellas. Esto es debido a la gran dispersión de las intensidades entre los elementos de una misma textura, a la existencia de texturas con características variables, así como a la pérdida de precisión como consecuencia de la distancia a la que se toman las imágenes. Tradicionalmente, para resolver este tipo de problemas se han desarrollado analizadores de texturas que utilizan operadores estadísticos realizando su evaluación en el campo probabilístico. Aquí se propone un análisis de texturas naturales, basado en la utilización de operadores estructurales y su modelización en el terreno posibilístico mediante técnicas Difusas, lo que permite realizar un tratamiento más adecuado de los factores de incertidumbre y mejorar los resultados de la segmentación.

Palabras clave: *análisis de texturas, operador estructural, segmentación, conjunto difuso, regla difusa.*

1 Introducción

El análisis de imágenes aéreas se ha convertido en un tema de creciente actualidad en los últimos años, ya que gracias a la sofisticación de la tecnología utilizada para la captación de imágenes aéreas (aviones, satélites, sondas, óptica, etc.), cada vez es mayor la cantidad, calidad y resolución de la información contenida en dichas imágenes, esto ha suscitado el interés por su estudio desde diferentes áreas, como por ejemplo en el campo de la cartografía [14],en segmentación o clasificación de territorios poblados por distintas especies vegetales [3], [19], [22], en el estudio de superficies agrícolas [13], en geología [6], en el estudio de especies vegetales en dependencia con la climatología [17], en planimetría [7], [9], etc.

Por otra parte, las nuevas tecnologías de captación de imágenes han hecho aumentar de forma exponencial el volumen de datos a tratar en cada imagen. En contraposición, el interés por su análisis hace que su procesado deba efectuarse en periodos de tiempo cada vez más cortos. Este factor ha originado la necesidad de automatizar el análisis de imágenes para incrementar la velocidad de procesado, lo cual se contrapone con la escasez de técnicas globales para la interpretación de las imágenes. Los sistemas que se han diseñado hasta ahora, para automatizar el análisis de imágenes, realizan operaciones muy específicas y tan solo aportan datos intermedios que son analizados, generalmente, por expertos humanos que se encargan finalmente de la interpretación para la identificación de los elementos de interés que aparecen en las imágenes.

El análisis de imágenes aéreas pretende, en último extremo, identificar los distintos tipos de elementos que aparecen en ellas, para lo que es necesario un primer procesado consistente en la segmentación de la imagen, cuya automatización constituye una línea de investigación abierta debido a la dificultad que presenta el tratamiento de la variabilidad de las texturas propias de las imágenes aéreas de entornos naturales. Por esto, si bien la existencia de diferentes texturas en una imagen resulta evidente para el hombre, el proceso de automatizar la separación de distintos tipos de texturas resulta mucho más complejo de conseguir (ver [20]).

Teniendo en cuenta lo expuesto, nuestro interés se ha centrado en la elaboración de operadores que permitan segmentar un determinado tipo de textura natural con la mayor precisión posible, presentándose en este artículo un algoritmo para la segmentación de este tipo de

texturas. En concreto se analiza la textura "plantación de árboles" (ver figs. 1 y 2), es decir, alineaciones de árboles de cultivo con una separación variable entre si. Debido tanto a la variabilidad de las texturas naturales como a la vaguedad introducida por los procesos de captación y digitalización de las imágenes, en el algoritmo que aquí se presenta, se realiza un análisis posibilístico de la información presente en ellas. Se definen un conjunto de reglas que indican, para cada pixel de la imagen, su posible pertenencia a la textura analizada, para lo que se formalizan las reglas en el campo Difuso, efectuando su evaluación por medio de operadores estructurales basados en la utilización del vector gradiente.



Figura 1: Imagen original (en el recuadro aparece la textura a segmentar).



Figura 2: Imagen original (en los recuadros aparecen las texturas a segmentar).

2 Interpretación del problema

Para lograr la caracterización de la textura "plantación de árboles" se realiza un análisis ascendente, en el que se estudian los elementos que la componen (árboles), y la presencia de una estructura en la distribución espacial de los mismos.

En las figuras 1 y 2 se muestran dos imágenes representativas del tipo de imágenes analizadas. En ellas aparecen enmarcadas las zonas que corresponden a la textura estudiada, así mismo pueden observarse los elementos bàsicos que la componen (árboles), la distinta variabilidad de dicha composición, así como la dificultad que supone distinguir la textura "plantación de árboles" de otras parecidas, y la dificultad para establecer límites precisos entre las diversas zonas con diferente textura.

2.1 Descriptores de la geometría local

La caracterización de cada microtextura (componente de la textura árbol) se realiza a partir del estudio local de la direccionalidad en un entorno de cada pixel. Este estudio de la direccionalidad se lleva a cabo a partir del cálculo local del vector gradiente y de la información cuantitativa y cualitativa que nos ofrecen, respectivamente, el módulo y el argumento de dicho vector. Es decir, se analiza la repetición de patrones direccionales que aportan características geométricas que permiten distinguir las microtexturas que componen un árbol. Concretamente se identifica la textura "árbol" mediante la circunferencia mínima que lo engloba. Posteriormente, con el objetivo de definir la textura "árbol" con mayor precisión, se analiza su descomposicón en cuatro elementos, para lo que se considera el cuadrado mínimo que circunscribe a la circunferencia anterior, y se analiza la direccionalidad en un entorno de los pixels correspondientes a cada uno de los cuatro vértices del cuadrado.

2.2 Descripción de los operadores.

Para el cálculo del vector gradiente se utiliza el operador de Sobel que se basa en la convolución de la imagen con máscaras cuyo tamaño és de 3x3 pixels. Es conocido el problema de que la utilización de ventanas pequeñas conlleva una mayor sensibilidad respecto al ruido y por lo tanto una capacidad de detección peor, mientras que la utilización de ventanas grandes aumenta la capacidad de detección pero disminuye la precisión con que se realizará la posterior localización de las zonas de interés. La elección del operador de Sobel supone un compromiso intermedio con el objetivo de conseguir una capacidad de detección y de localización aceptables (ver [5]).

Una vez calculado el vector gradiente, se aplica un segundo nivel de operadores con el fin de localizar las microtexturas estudiadas. En concreto se utilizan ventanas de 3x3 y 5x5, ya que como puede observarse en las figuras 3 y 4, dichos tamaños delimitan exactamente los vértices de un cuadrado y su entorno.

3 El contexto difuso

Para segmentar la textura "plantación de árboles" se estudia la disposición, dentro de la misma, de los elementos que la componen ("árboles") y su número. Para ello se definen unas reglas, que atendiendo a criterios de emplazamiento de dichos elementos, proporcionan una medida para evaluar la semejanza de la textura de la imagen con la textura "plantación de árboles".

Asimismo, la textura tipo "árbol" se estudia a partir de su composición en cuatro microtexturas. Para ello se definen unas reglas que, atendiendo a criterios de ubicación de estas microtexturas, permiten la obtención de una medida de semejanza del elemento definido por estas microtexturas y el elemento "árbol".

3.1 Definición de reglas y conjuntos difusos

La existencia de una textura tipo "plantación de árboles" vendrá condicionada por la existencia de los elementos básicos que la componen o "árboles". A su vez, la textura "árbol" se define en función de cuatro microtexturas que la componen. Por esto, el análisis se ha dividido en tres niveles, definiéndose en cada uno de ellos un conjunto difuso que se identifica con el tipo de componente del nivel en cuestión. En concreto se consideran los siguientes conjuntos difusos: *plantación de árboles*, *árbol*, *subárbol*$_i$ ($i \in \{1,2,3,4\}$).

Para cada nivel, el grado de pertenencia de cada pixel a cada conjunto difuso del nivel se definirá en función del grado de pertenencia del mismo pixel a los conjuntos difusos de los niveles inferiores y de acuerdo con las reglas que se enuncian para cada nivel.

La pertenencia de cada pixel de la imagen a cada uno de los cuatro conjuntos difusos *subárbol*$_i$ ($i \in \{1,2,3,4\}$) se obtiene mediante el estudio de la estructura direccional en un entorno de cada pixel. Concretamente, cada conjunto difuso *subárbol*$_i$ está compuesto por los pixels que forman parte de cada uno de los cuatro vértices del cuadrado que circunscribre a la textura "árbol". Se definen varias reglas que proporcionan el grado de pertenencia de cada pixel a los conjuntos difusos, en función de la comparación entre la estructura direccional encontrada y el patrón direccional teórico esperado. Para la definición de estas reglas se estudia un entorno alrededor de cada pixel y se subdivide en dos zonas, en función de la distancia al pixel en cuestión, para su análisis más detallado: una zona de máxima influencia y una zona de influencia secundaria alrededor de la primera. Para cada una de estas zonas se definen dos reglas difusas, una prioritaria en función del argumento del vector gradiente y otra complementaria en función del módulo del vector gradiente. Finalmente, se realiza la agregación de las reglas dentro de cada zona y entre las dos zonas para obtener el grado de pertenencia de cada pixel a cada conjunto difuso *subárbol*$_i$.

La información que aportan las reglas de pertenencia a un cierto nivel constituye un factor fundamental en la definición de las reglas de pertenencia de cada pixel a los conjuntos difusos de los niveles superiores al mismo.

A continuación se definen las reglas difusas en función de los niveles en los que se ha desglosado el análisis de la textura "plantación de árboles":

- Nivel de microtextura (subárbol)

  Se definen las reglas que proporcionan el grado de pertenencia de cada pixel al conjunto difuso *subárbol*$_i$. (El conjunto difuso *subárbol*$_1$ está referido al vértice superior izquierdo del cuadrado; el conjunto difuso *subárbol*$_2$ está referido al vértice inferior izquierdo del cuadrado; el conjunto difuso *subárbol*$_3$ está referido al vértice superior derecho del cuadrado; el conjunto difuso *subárbol*$_4$ está referido al vértice inferior derecho del cuadrado).

  1. Vértice Superior Izquierdo:
     - Regla 1.1: el grado de pertenencia de un pixel (i,j) al conjunto difuso *subárbol*$_1$ es mayor si el vector gradiente en dicho pixel y en sus dos pixels vecinos derecho y inferior (ver fig. 3a) se encuentra fuertemente polarizado en dirección noroeste.
     - Regla 1.2: el grado de pertenencia de un pixel (i,j) al conjunto difuso *subárbol*$_1$ es mayor si el vector gradiente en los cinco pixels vecinos de los tres pixels considerados en la regla anterior, situados geográficamente en la parte superior y en la parte izquierda de aquellos (ver fig. 4a), se encuentra fuertemente polarizado en dirección noroeste.

  2. Vértice Inferior Izquierdo:
     - Regla 2.1: el grado de pertenencia de un pixel (i,j) al conjunto difuso *subárbol*$_2$ es mayor si el vector gradiente en dicho pixel y en sus dos pixels vecinos derecho y superior (ver fig. 3b) se encuentra fuertemente polarizado en dirección suroeste.
     - Regla 2.2: el grado de pertenencia de un pixel (i,j) al conjunto difuso *subárbol*$_2$ es mayor si el vector gradiente en los cinco pixels vecinos de los tres pixels considerados en la regla anterior, situados geográficamente en la parte inferior y en la parte izquierda de aquellos (ver fig. 4b), se encuentra fuertemente polarizado en dirección suroeste.

  3. Vértice Superior Derecho:
     - Regla 3.1: el grado de pertenencia de un pixel (i,j) al conjunto difuso *subárbol*$_3$ es mayor si el vector gradiente en dicho pixel y en sus dos pixels vecinos izquierdo y inferior (ver fig. 3c) se encuentra fuertemente polarizado en dirección noreste.
     - Regla 3.2: el grado de pertenencia de un pixel (i,j) al conjunto difuso *subárbol*$_3$ es

mayor si el vector gradiente en los cinco pixels vecinos de los tres pixels considerados en la regla anterior, situados geográficamente en la parte superior y en la parte derecha de aquellos (ver fig. 4c), se encuentra fuertemente polarizado en dirección noreste.

4. Vértice Inferior Derecho:
   - Regla 4.1: el grado de pertenencia de un pixel (i,j) al conjunto difuso *subárbol4* es mayor si el vector gradiente en dicho pixel y en sus dos pixels vecinos izquierdo y superior (ver fig. 3d) se encuentra fuertemente polarizado en dirección sureste.
   - Regla 4.2: el grado de pertenencia de un pixel (i,j) al conjunto difuso *subárbol4* es mayor si el vector gradiente en los cinco pixels vecinos de los tres pixels considerados en la regla anterior, situados geográficamente en la parte inferior y en la parte derecha de aquellos (ver fig. 4d), se encuentra fuertemente polarizado en dirección sureste.

- Nivel de elemento básico de la textura (árbol)

  A partir de las reglas del nivel anterior, que indican la posibilidad de la existencia de una microtextura, se estudia la ubicación espacial de cada una de ellas con respecto a las demás y se define una regla en función de dicha ubicación, que proporciona el grado de pertenencia de cada pixel al conjunto difuso *árbol*.

- Nivel de textura (plantación de árboles)

  A partir de las reglas definidas para los niveles anteriores que indican la posibilidad de la existencia de cada textura tipo "árbol", se estudia la distribución espacial (atendiendo a criterios de linealidad) entre las distintas texturas "árbol" obtenidas, y se define una regla en función de dicha distribución, que proporciona el grado de pertenencia de cada pixel al conjunto difuso *plantación de árboles*.

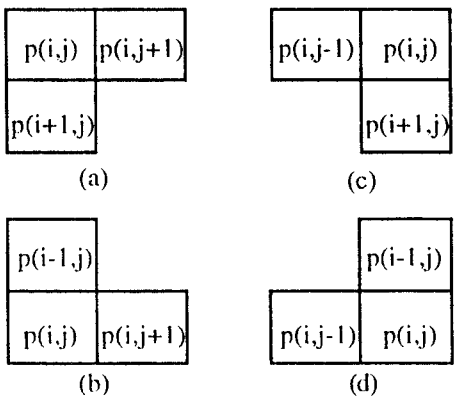


Figura 3: Estructuras estudiadas para la construcción de los conjuntos difusos *subárbol*$_i$ (i$\in$ {1,2,3,4}).

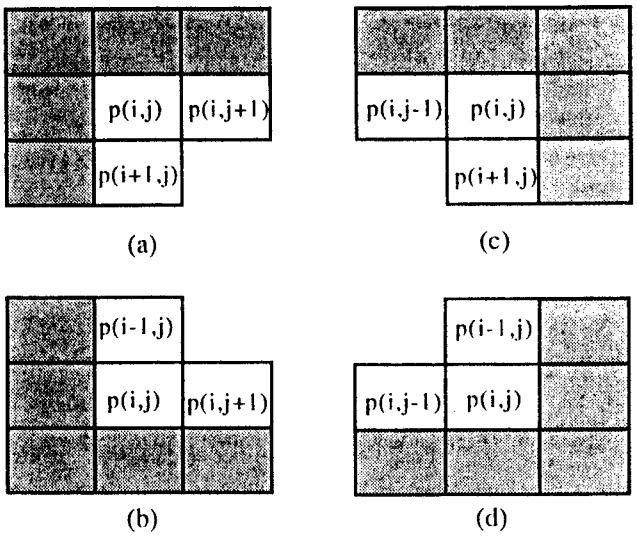


Figura 4: Estructuras auxiliares estudiadas para la construcción de los conjuntos difusos *subárbol*$_i$ (i$\in$ {1,2,3,4}).

4 Resultados

En las figuras 5 y 6 pueden verse los resultados obtenidos a partir de las imágenes originales (figs. 1 y 2). Dichos resultados se presentan en imágenes binarias como consecuencia de aplicar un α-corte sobre las funciones de pertenencia difusas. En ellas pueden observarse las texturas tipo "árbol" detectadas (los árboles aparecen en blanco sobre el fondo negro). Cabe destacar el buen comportamiento del algoritmo ante algunas dificultades que se desprenden del análisis de las texturas estudiadas, como son: la distinta densidad de árboles en cada área de cultivo, su diferencia de tamaño, su alineación irregular y la presencia de otros elementos que distorsionan la estructura teórica esperada.

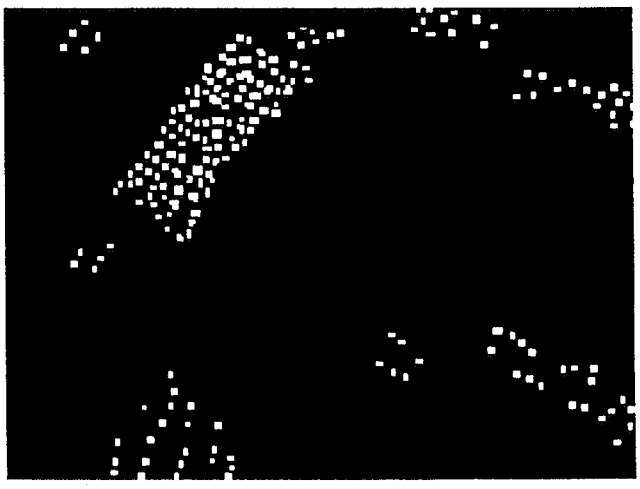


Figura 5: Imagen resultado (correspondiente a la imagen de la fig. 1).

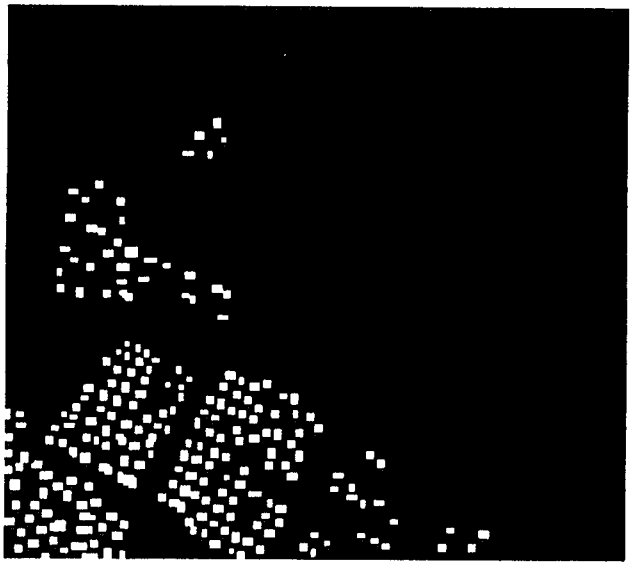


Figura 6: Imagen resultado (correspondiente a la imagen de la fig. 2).

5 Conclusiones

En este trabajo se han presentado descriptores estructurales de texturas naturales para la segmentación de imágenes aéreas.

A diferencia de los enfoques clásicos (ver [8],[10],[12], [22]), en el estudio que aquí se ha presentado se plantea la dificultad de clasificar un determinado tipo de textura, a partir del análisis de imágenes aéreas, en las que aparece junto con otros tipos de texturas naturales. Para ello se propone la necesidad de realizar un estudio estructural a nivel de microtextura, que permita discriminar la textura estudiada del resto de texturas que aparecen en la imagen. Aunque tradicionalmente se ha venido afrontando el estudio de microtexturas desde un punto de vista estadístico, la robustez de este enfoque queda en entredicho cuando se trabaja a nivel de pixel, o en entornos muy pequeños alrededor de un pixel, debido al ruido inherente a toda imagen y a la imprecisión motivada por la distancia. Por tanto, en aras a incrementar la robustez y fiabilidad de los algoritmos de segmentación, resulta más adecuado diseñar operadores estructurales basados en la información que aporta el vector gradiente, ya que en diversos estudios de visión por computador (ver [11],[15],[19],[21]) se ha demostrado que para la realización de análisis locales la utilización del vector gradiente es una de las mejores opciones (ver [4],[19]). Bajo estas premisas, en nuestro trabajo se ha formalizado la información que aporta el vector gradiente mediante técnicas Difusas (ver [1],[2], [16],[18],[23]) con el propósito de conseguir una mayor adecuación de la información que proporcionan los operadores a las características de las texturas naturales.

Referencias

[1] Alsina,C., Fank,M.J., Schweizer,B., *Associative Functions on Intervals*, 1987.

[2] Alsina,C., Trillas,E., Valverde,L., On Some Logical Connectives for Fuzzy Sets Theory. *Journal of Mathematical Analysis and Aplications*, Vol. 93, nº 1, pp. 15-26, 1983.

[3] Alvarez,S., Puig,D., Sobrevilla,P., Montseny,E., Análisis de texturas naturales mediante técnicas difusas. *Proc. IV Congreso Asociación Española Tecnologías Lógica Fuzzy*. Blanes, pp 339-343, 1994.

[4] Bajcsy,R., Lieberman,L., Texture gradient as a depth cue. *Computer Graphics and Image Processing*, Vol 5, nº 1, pp 52-67, 1976.

[5] Canny,J.F., Finding edges and lines in images. *Technical Report 720*, MIT Cambridge, 1983.

[6] Conese,C., Maselli,F., De Filippis,T., Romani,M., Statistical integration of ancillary data into an advanced maximum likelihood classifier, *Proc. ISPRS 3th Symposium Spatial Information from Digital Photogrammetry and Computer Vision*. Munich (Germany), pp 152-157, 1994.

[7] Fritsch,D., Sester,M., Test on image understanding, *Proc. ISPRS 3th Symposium Spatial Information from Digital Photogrammetry and Computer Vision*. Munich (Germany), pp 243-248, 1994.

[8] Galloway,R.M., Texture analysis using gray-level run lengths, *Computer Graphics and Image Processing*, Vol 4, pp 172-179, 1975.

[9] Gruen,A., Li,H., Semi-automatic road extraction by dynamic programming, *Proc. ISPRS 3th Symposium Spatial Information from Digital Photogrammetry and Computer Vision*. Munich (Germany), pp 324-332, 1994.

[10] Haralick,R.M., Shanmugam,K., Dinstein,I., Textural Features for Image Classification, *IEEE Trans SMC-3*, nº 6, 1973.

[11] Larré,A., Montseny,E., A step edge detector algorithm based on symbolic analysis. *Proc. 12th IAPR, Inter. Conference On Pattern Recognition*. Jerusalem (Israel), Vol 1, pp 6-10, 1994.

[12] Laws,K.I., Textured image segmentation, *Ph.D. dissertation*, Univ. Southern California, Los Angeles, Rept. 940, 1980.

[13] Löcherbach,T., Fusion of multi-sensor images and digital map data for the reconstruction and interpretation of agricultural land-use units, *Proc. ISPRS 3th Symposium Spatial Information from Digital Photogrammetry and Computer Vision.* Munich (Germany), pp 505-511, 1994.

[14] Lotti,J.L., Giraudon,G., Adaptive window algorithm for aerial image stereo, *Proc. ISPRS 3th Symposium Spatial Information from Digital Photogrammetry and Computer Vision.* Munich (Germany), pp. 517-524, 1994.

[15] Montseny,E., Sobrevilla,P., García-Barroso,C., Detección de contornos escalón mediante algoritmos difusos. *Proc. IV Congreso Asociación Española Tecnologías Lógica Fuzzy.* Blanes, pp 321-328, 1994.

[16] Pal,K. et al., Image Description and Primitive Extraction Using Fuzzy Sets, *IEEE Trans. SCM 13,* January 1983.

[17] Palubinskas,G., A comparison of texture-based and fuzzy classification approaches for regenerating tropical forest mapping using Landsat TM, *Proc. ISPRS 3th Symposium Spatial Information from Digital Photogrammetry and Computer Vision.* Munich (Germany), pp 640-647, 1994.

[18] Pao,Y.H., Vague Features and Vague Decision Rules: The Fuzzy-Set Approach, Adaptative Pattern Recognition and Neural Networks. *Ed. Adison Wesley,* 1989.

[19] Puig,D., Alvarez,S., Sobrevilla,P., Montseny,E., Analysis of natural textures using fuzzy techniques. *Proc. VI Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes.* Córdoba, pp 112-120, 1995.

[20] Tomita,F., Tsuji,S., Computer Analysis of Visual Textures, *Kluwer Academic Publishers,* USA, 1990.

[21] Vilnrotter,F.M., Nevatia,R., Price,K.E., Structural analysis of natural textures, *IEEE Trans. PAMI-8,* N° 1, pp 76-89, 1986.

[22] Weszka,J.S., Dyer,C.R., Rosenfeld,A., A Comparative Study of Texture Measures for Terrain Classification, *IEEE Trans. on SMC-6,* n° 4, April, 1976.

[23] Zadeh,L.A., Fuzzy Sets, *Information and Control,* n°8, pp. 338-353, Academic Press, New York,1965.

Aprendizaje y Optimización

Un Enfoque Aproximativo para la Generación de Reglas mediante Análisis Cluster *

M. Delgado
Dept. Ciencias de la Computación e Inteligencia Artificial. Universidad de Granada
A. F. Gómez-Skarmeta y F. Martín
Dept. Informática y Sistemas. Universidad de Murcia

Resumen

En este trabajo presentamos diferentes aproximaciones al problema de la extracción de reglas difusos a partir de conjuntos de ejemplos de funcionamiento de un sistema. Estos métodos intentan obtener una primera aproximación al modelo del sistema, sin realizar ninguna consideración previa sobre la estructura presenta en los datos.

Palabras Claves: *clustering difuso; aprendizaje no supervisado; modelado difuso.*

1 Introducción

A lo largo de la literatura encontramos diferentes métodos para la generación de reglas que buscan asignar un valor de certeza o posibilidad al conjunto de reglas que se generan sobre un conjunto de etiquetas predefinidas (o conjuntos difusos referencia)[9][13][10]. Si bien este método ha sido muy aplicado mediante diferentes definiciones del concepto peso, factor de certeza o posibilidad de las reglas, tiene un inconveniente importante cuando el antecedente de las reglas está formado por un número elevado de variables. El problema estriba en que se produce una explosión combinatoria respecto al número de reglas posibles.

En este trabajo proponemos un enfoque alternativo, mediante la consideración de un conjunto mucho más reducido de reglas, inducidas a partir de los clusters obtenidos sobre el espacio producto de las entradas y las salidas. Se presentan diferentes métodos que utilizan la información que suministra el clustering difuso para obtener una primera aproximación a las reglas difusas que describirían el comportamiento del sistema. Debido a que lo que se busca es una primera aproximación, los métodos propuestos son simples pero con gran capacidad aproximativa, al permitirnos trabajar con conjuntos difusos directamente en el espacio producto de las variables de entrada.

En cualquier caso y tal como veremos en la última sección, siempre podemos pasar de este enfoque aproximativo a un enfoque de tipo descriptivo al intentar expresar el sistema en base a un conjunto de etiquetas linguísticas predefinidas de cada dominio de las variables.

Lo que es importante destacar es que los clusters difusos sobre los que trabajaremos pueden ser obtenidos directamente al aplicarse el análisis cluster difuso sobre los datos, o bien tras haberse realizado un preproceso utilizando un clustering jerárquico crisp, tal y como ya se ha estudiado en otros trabajos[4].

*Este trabajo ha sido parcialmente financiado por la CICYT con cargo al Proyecto TIC95-1019

2 Preliminares

En principio y sin perdida de generalidad vamos a considerar un sistema del tipo MISO (múltiple entradas, salida única). Para alcanzar el objetivo del modelado del sistema partiremos de la existencia de datos de ejemplos del comportamiento del sistema $\Omega = ((x_{t1}, x_{t2}, ..., x_{tp}), y_t)\ t = 1, 2, ..., n$ en el espacio producto de las entradas y las salidas $(X^p \times Y)$, donde $X_1, X_2, ..., X_p$ notan el dominio o universo de discurso de las variables de entradas, e Y el dominio de la variable de salida. Concretamente nuestro objetivo consiste en intentar aproximar la función $\varphi : X^p \rightarrow Y$ que modela el sistema.

Partimos de un conjunto de k clusters en el espacio $(X^p \times Y)$ obtenidos a través de un proceso de clustering difuso que reflejan diferentes tendencias o comportamiento del sistema, y lo que nos interesa es caracterizar dichas tendencias a través de k reglas del tipo:

$$R_h : \text{Si } x \text{ es } A_h \text{ entonces } y \text{ es } B_h \quad h = 1, ..., k$$

donde A_h y B_h son respectivamente los conjuntos difusos en X^p e Y obtenidos como representantes en esos espacio del cluster $h-$ésimo C_h del espacio producto $(X^p \times Y)$. Una vez generado este conjunto de reglas, podemos inferir valores de salida para una nueva entrada utilizando cualquier método de razonamiento aproximado que haga uso de este conjunto de k reglas.

En particular debido a su sencillez nosotros hemos elegido el método de razonamiento simplificado[8], el cual permite inferir un valor de salida para este conjunto de reglas mediante la expresión:

$$\hat{y} = \frac{\sum_{h=1}^{k} A_h(x) * (v_h)}{\sum_{h=1}^{k} A_h(x)} = \frac{\sum_{h=1}^{k} \tau_h * (v_h)}{\sum_{h=1}^{k} \tau_h}$$

donde v_h es un consecuente puntual que asociamos a cada regla. De esta forma lo que obtenemos es un

conjunto de k reglas del tipo consecuente puntual con expresión:

$$R_h: \text{ Si } x \text{ es } A_h \text{ entonces } y \text{ es } v_h \; h = 1, ..., k$$

Por tanto nuestro problema está en caracterizar a estos conjuntos A_h y B_h asociables al cluster $h-$ésimo de $(X^p \times Y)$. Una de las formas más directas de obtener los conjuntos difusos A_h y B_h asociados al cluster C_h de $(X^p \times Y)$ es realizar la proyección del cluster en los espacios X^p e Y. En el caso de que el clustering sea difuso, esto es relativamente sencillo y se ha aplicado en algunos enfoques como en los de Sugeno[12], Yager[14], etc.

Por contra en los métodos que proponemos asociados con el clustering difuso no nos interesará tanto el obtener las proyecciones como ver formas alternativas de construcción de dichos conjuntos difusos, así como estudiar y analizar formas de determinar valores de salida puntual v_h para cada regla asociada a los cluster de $(X^p \times Y)$ que permitan caracterizar de la mejor forma posible el comportamiento del sistema.

3 Generación de reglas a partir de Clustering Difuso

En esta sección proponemos un conjunto de métodos para trabajar con la información que se puede obtener a partir de clustering difusos sobre los datos de aprendizaje. Un clustering difuso del tipo particional (es decir que genera una partición para el conjunto de datos) como el de las $K-$medias (FCM), genera un conjunto de k clusters tales que $\forall x \in X$ $\sum_{h=1}^{k} \mu_{C_h}(x) = 1.0$. El uso del clustering difuso de tipo particional nos introduce tres importantes características:

1. Cuando se realiza un clustering difuso de un determinado espacio tendremos garantizado por la propiedad anterior, la obtención de una partición difusa. Esto nos abre una vía para superar uno de los aspectos problemáticos como es la posibilidad de obtener bases de reglas dispersas.

2. Al emplear una técnica de clustering difuso del tipo $K-$medias o alguna de sus variantes, resulta que la función de pertenencia de los conjuntos difusos asociados a los clusters tiene una expresión funcional definida en términos de las distancia de los datos a los centroides. Esto trae consigo, entre otros aspectos positivos, que en un dominio formado por el producto cartesiano de los dominios de varias variables, es posible obtener un conjunto difuso directamente en dicho dominio producto cartesiano.

3. Además debido a la forma de la expresión de la función de pertenencia, ésta depende sólo del valor de los centroides, con lo que en cualquier dominio los conjuntos difusos que forman los clusters, están pefectamente caracterizados para cualquier dato del dominio, sólo en virtud del valor de los centroides de los clusters generados.

3.1 Generación de reglas a partir de clusters difusos en $(X^p \times Y)$

Un aspecto importante que debemos destacar es el significado de la formación de agrupamientos representados por medio de un centroide y unos grados de similitud de los ejemplos con respecto a él. Este puede interpretarsecomo una tendencia que viene caracteriza por el centroide, el cual puede considerarse como el representante "ideal" del comportamiento del grupo. Bajo esta interpretación por tanto, consideramos que los centroides y el comportamiento asignable mediante el conjunto difuso que tendrán asociado a través del cluster difuso van a ser los elementos fundamentales en la caracterización de las reglas a generar.

Presentamos dos enfoques alternativos, para los cuales vamos a considerar que tras un proceso de clustering difuso de tipo particional como puede ser el realizado utilizando el algoritmo FCM o alguna de sus variantes, hemos obtenido sobre un conjunto de n datos en $(X^p \times Y)$, k clusters difusos con centroides c^h_{XY} $h = 1, 2, ..., k$. y con funciones de pertenencia asociadas:

$$\mu_{C^h_{XY}}(x_t, y_t) = 1/\left\{ \frac{\sum_{l=1}^{k} \left(\left\| (x_t, y_t) - c^h_{XY} \right\|^2 \right)}{\left\| (x_t, y_t) - c^l_{XY} \right\|^2} \right\}^{1/(m-1)}$$

donde $\mu_{C^h_{XY}}(x_t, y_t)$ es la pertenencia al cluster $h-$ésimo encontrado $(X^p \times Y)$ del elemento (x_t, y_t).

3.2 Primera aproximación

Una primera aproximación al problema de la generación de reglas sería el pasar el agrupamiento en el espacio $(X^p \times Y)$ directamente a los espacio X^p e Y, sin más que considerar sobre esos espacio los conjuntos difusos generados por los componentes en X^p e Y de los centroides c^h_{XY} que notaremos como c^h_X y c^h_Y respectivamente, y que tendrá como función de pertenencia la misma expresión funcional de los clusters difusos.

En este sentido podemos hablar de clusters inducidos ya que dichos clusters no se obtienen por un proceso de clustering. No se trata de una proyección global, ya que no se preserva para cada ejemplo su grado de pertenencia en $(X^p \times Y)$, sino que estamos proyectando sólo los centroides y considerando los conjuntos difusos que ellos generan a través de la misma métrica del espacio original. Así tenemos k conjuntos difusos en X^p y en Y, con funciones de pertenencia dadas por:

$$\mu_{C^h_X}(x_t) = 1/\left\{ \sum_{l=1}^{k} \left(\left\| x_t - c^h_X \right\|^2 \right) / \left\| x_t - c^l_X \right\|^2 \right\}^{1/(m-1)}$$

$$\mu_{C^h_Y}(y_t) = 1/\left\{ \sum_{l=1}^{k} \left(\left\| y_t - c^h_Y \right\|^2 \right) / \left\| y_t - c^l_Y \right\|^2 \right\}^{1/(m-1)}$$

Con estos conjuntos difusos inducidos podemos por

tanto construir k reglas del tipo:

$$R_h: \text{ Si } x \text{ es } \mu_{C_X^h}() \\ \text{entonces } y \text{ es } \mu_{C_Y^h}() \; h = 1, ..., k$$

Por la forma en que han sido construídos, es obvio que estos clusters generan sendas particiones difusas sobre X^p e Y, por lo que garantizan la cobertura de los dominios. La idea básica por tanto está en mantener la relación de agrupamiento de los datos con respecto a los centroides para las proyecciones de éstos. Una ventaja importante de este método consiste en la posibilidad de inducir un conjunto difuso directamente en X^p en virtud del mantenimiento de la expresión funcional de los clusters, lo cual conlleva también una escasa complejidad asociada con el método. Debido a que los centroides dan mucha información sobre las tendencias en el comportamiento, podemos generar un conjunto de reglas de consecuente puntual de forma:

$$R_h: \text{ Si } x \text{ es } \mu_{C_X^h}() \text{ entonces } y \text{ es } c_Y^h \; h = 1, ..., k$$

y por tanto realizar un proceso de inferencia ante una nueva entrada x mediante el método simplificado de la forma:

$$EST1 = \hat{y} = \frac{\sum_{h=1}^{k} \mu_{C_X^h}(x) * c_Y^h}{\sum_{h=1}^{k} \mu_{C_X^h}(x)}$$

siendo c_Y^h la componente en Y del centroide del cluster $h-$ésimo en $(X^p \times Y)$.

Siguiendo este mismo enfoque se puede desarrollar un método que utilice directamente los clusters obtenidos en $(X^p \times Y)$, sin necesidad de hacer proyecciones. Con este objetivo empleamos una técnica similar a la propuesta por Sin y deFigueredo ([11]). En ella una vez han sido detectados los diferentes agrupamientos en $(X^p \times Y)$ se les asocia una valor de salida, tal que éste no sólo es el consecuente puntual correspondiente a la regla, sino que se empleará además como un valor fijo que se asociará a cada nueva entrada para determinar su grado de adecuación a los clusters de $(X^p \times Y)$.

Con esta idea como punto de partida, en nuestro caso podríamos utilizar el consecuente puntual c_Y^h de cada regla como dicho valor, y por tanto inferir una salida ante una nueva entrada x de la forma:

$$EST2 = \hat{y} = \frac{\sum_{h=1}^{k} \mu_{C_{XY}^h}(x, c_Y^h) * c_Y^h}{\sum_{h=1}^{k} \mu_{C_{XY}^h}(x, c_Y^h)}$$

siendo $\mu_{C_{XY}^h}(x, c_Y^h)$ el grado de pertenencia al cluster $h-ésimo$ de $(X^p \times Y)$ del par (x, c_Y^h). De esta forma no hay necesidad de crear proyecciones de los clusters de $(X^p \times Y)$.

También hay que hacer notar que son posibles otras expresiones de las funciones de pertenencia de los conjuntos difusos. Dado que la fuente principal de información son los centroides así como el comportamiento a su alrededor, sería pensable por tanto el uso de funciones de tipo exponencial que tengan como argumentos las coordenadas de los centroides y la varianza difusa asociada a los agrupamientos[6][7].

Así una forma simplificada de utilizar estas ideas sería utilizando los componentes en X^p de los centroides encontrados en $(X^p \times Y)$ y calcular las covarianzas para cada cluster inducido en X^p teniendo en cuenta las componentes de los centroides. De esta forma llegamos a un método de inferencia del tipo consecuente puntual:

$$EST3 = \hat{y} = \frac{\sum_{h=1}^{k} \exp\left(-\frac{\|x - c_X^h\|^2_{A^h}}{2}\right) * c_Y^h}{\sum_{h=1}^{k} \exp\left(-\frac{\|x - c_X^h\|^2_{A^h}}{2}\right)}$$

siendo A^h la matriz de covarianzas para el cluster $h-$ésimo.

Dado que es posible que los clusters inducidos en X^p e Y no se ajusten exactamente al agrupamiento descubierto a través de los clusters en $(X^p \times Y)$, un posible refinamiento del modelo se obtiene asociando a cada conjunto difuso inducido en X^p a partir de un cluster de $(X^p \times Y)$, un valor en el espacio de salida Y teniendo en cuenta la relación entre el cluster de $(X^p \times Y)$ y su conjunto asociado en X^p . Dado que tenemos el mismo número de clusters en $(X^p \times Y)$ que conjuntos en X^p, podemos generar diferentes valores para el consecuente puntual por ejemplo:

$$CP1 = y_h^* = \frac{\sum_{t=1}^{n} \mu_{C_X^h}(x_t) * \mu_{C_{XY}^h}(x_t, y_t) * y_t}{\sum_{t=1}^{n} \mu_{C_X^h}(x_t) * \mu_{C_{XY}^h}(x_t, y_t)} \qquad (1)$$

o bien:

$$CP2 = y_h^* = \frac{\sum_{t=1}^{n} \left[\mu_{C_X^h}(x_t) * \mu_{C_{XY}^h}(x_t, y_t)\right]^m * y_t}{\sum_{t=1}^{n} \left[\mu_{C_X^h}(x_t) * \mu_{C_{XY}^h}(x_t, y_t)\right]^m} \qquad (2)$$

siendo $\mu_{C_{XY}^h}()$ y $\mu_{C_X^h}()$ los grados de pertenencia de los ejemplos a los conjuntos difusos correspondiente al cluster h en $(X^p \times Y)$ y a su inducido en X^p respectivamente.

Los valores que se calculan en el caso de $CP1$ corresponden al valor inferido de salida para cada cluster siguiendo un proceso que sería análogo a una desfuzzificación por el método del Center of Area. En cuanto a $CP2$ puede interpretarse como la proyección en el dominio Y de los centroides de los clusters que se obtendrían considerando los clusters difusos de $(X^p \times Y)$ combinándolos o condicionándolos por los conjuntos difusos inducidos por dichos clusters en X^p.

De esta forma tendríamos un conjunto de k reglas $R_h \; h = 1, 2, ..., k$ del tipo consecuente puntual con expresión:

$$R_h: \text{ Si } x \text{ es } \mu_{C_X^h}() \text{ entonces } y \text{ es } y_h^*$$

Estos métodos de inferencia con consecuente puntual serán denominados $EST4$ y $EST5$ según que

utilicen el consecuente puntual $CP2$ o $CP1$ respectivamente.

Hay que hacer notar que este tipo de salida inferida puede obtenerse alternativamente empleando los conjuntos difusos inducidos en Y con función de pertenencia $\mu_{C_Y^h}$ $h = 1, 2, ..., k$, y considerando el grado de aproximación del valor inferido respecto a los centroides de los clusters proyectados sobre el dominio Y, es decir:

$$\hat{y}' = \frac{\sum_{h=1}^{k} \mu_{C_Y^h}(\hat{y}) * c_Y^h}{\sum_{h=1}^{k} \mu_{C_Y^h}(\hat{y})}$$

siendo c_Y^h la proyección del centroide c_{XY}^h sobre Y .

3.3 Segunda aproximación

Un segundo enfoque alternativo a la hora de buscar los conjuntos difusos inducidos consiste en realizar las proyecciones de los clusters sobre cada uno de los dominios que forman X^p. Se trata por tanto de una adaptación del método propuesto por Sugeno y Yasukawa [12], o del propuesto por Babuska et al. [1], con la diferencia de que nosotros partimos del clustering difuso de $(X^p \times Y)$, y a partir de él se generan las proyecciones difusas.

Tras obtener las proyecciones se procede a la construcción del cierre extensional de los conjuntos difusos generados lo que nos garantizaría la obtención de conjuntos difusos normales y convexos. La semántica de estos conjuntos difusos puede ser compleja por lo que posteriormente se les puede asociar una aproximación mediante un conjunto difuso triangular o trapezoidal que sea más manejable.

Este enfoque, por tanto, obliga a generar conjuntos difusos en los dominios individuales de las variables, perdiéndose parte de la potencia conseguida al considerar directamente conjuntos difusos en X^p (como en el caso anterior). Además por la forma de construcción de los conjuntos difusos en principio no existe garantía en cuanto a la cobertura de los datos en los diferentes dominios, salvo que a la hora de aproximar los conjuntos difusos mediante unos triangulares o trapezoidales se tenga en cuanta este requisito y se busque cumplirlo mediante algún proceso de ajuste. Tenemos así un conjunto de k reglas de la forma:

R_h : Si x_1 es A_{1h} y x_2 es A_{2h} y ... y x_p es A_{ph} entonces y es B_h $h = 1, ...k$

Asociado a estas reglas podemos tener un mecanismo de inferencia del tipo simplificado con consecuente puntual de forma que ante una nueva entrada $x = (x_1, x_2, ..., x_p)$ obtengamos una salida:

$$\hat{y} = \frac{\sum_{h=1}^{k} T(A_{1h}(x_1), A_{2h}(x_2), ..., A_{ph}(x_p)) * c_Y^h}{\sum_{h=1}^{k} T(A_{1h}(x_1), A_{2h}(x_2), ..., A_{ph}(x_p))}$$

siendo T una t-norma, y c_Y^h el componente en Y de los centroides de los clusters difusos de $(X^p \times Y)$. En particular dos métodos posibles variando la $t-norma$ elegida serían:

$$EST6 = \frac{\sum_{h=1}^{k} \left[\min_{q=1}^{p}(A_{qh}(x_q))\right] * c_Y^h}{\sum_{h=1}^{k} \left[\min_{q=1}^{p}(A_{qh}(x_q))\right]}$$

$$EST7 = \frac{\sum_{h=1}^{k} \left[\prod_{q=1}^{p}(A_{qh}(x_q))\right] * c_Y^h}{\sum_{h=1}^{k} \left[\prod_{q=1}^{p}(A_{qh}(x_q))\right]}$$

3.4 Ejemplos

En esta sección presentamos los resultados de las diferentes medidas para dos ejemplos. El primero consiste en un sistema estático y no lineal con dos variables de entrada y una de salida, propuesto en el trabajo de Sugeno y Yasukawa como ejemplo de su método. El sistema corresponde a la expresión:

$$z = \left(1 + x^{-2} + y^{-1.5}\right)^2 \quad 1 \leq x, y \leq 5$$

Partiendo de esta ecuación, Sugeno et al. obtienen 50 datos de aprendizaje de las entradas y salida del sistema introduciendo cierta componente de ruido. Además generan otro conjunto de 25 datos de test con lo que comprobar la eficacia del sistema obtenido mediante su técnica.

El segundo ejemplo esta tomado de los trabajos de Babuska et al.[2], en donde se estudian diferentes métodos para identificar el modelo difuso para la dinámica de la presión de un fermentador de laboratorio del tipo fed-batch.

En la tabla (1) se señalan los diferentes errores detectados por nuestros métodos así como los de referencia de la literatura.

	Literatura	EST1	EST2
Sistema No Lineal	0.56	0.51	0.50
Fermentador	0.021	0.044	0.037

EST3	EST4	EST5	EST6	EST7
0.55	0.49	0.50	0.66	0.64
0.032	0.043	0.044	0.050	0.052

Tabla 1: Ejemplos/Error

4 Conclusiones

Una vez caracterizados los conjuntos difusos de las reglas, nos podemos plantear la posibilidad de relacionar este enfoque con los usuales en la literatura. La respuesta es que siempre es posible reexpresar el modelo aquí obtenido a través de un conjunto de etiquetas predefinidas y de una matriz de pesos que indiquen la certeza que se tiene en cada posible combinación de etiquetas.

Para ello podemos hacer uso de la noción de distribución de posibilidad de un conjunto difuso A respecto de otro de referencia A_i definido como:

$$Poss\,(A_i/A) = \sup_{x_t \in X} \min\left[A_i(x_t), A(x_t)\right]$$

(supuesto en este caso que el dominio de la variable en cuestión es finito). En caso de que sólo se dispusiera de los valores de los ejemplos de aprendizaje, entonces si A_h es el conjunto difuso inducido por el cluster $h-ésimo$ en un dominio de las variables, y A_i es uno de los conjuntos difusos referencias de dicho dominio, tendremos:

$$Poss\,(A_i/A_h) = \sup_{x_t \in C_h} \min\left[A_i(x_t), A_h(x_t)\right]$$

y de esta forma cada conjunto difuso inducido generaría en cada dominio un vector $z_q = \left(z_{q1}, z_{q2}, ..., z_{qn_q}\right)$ con $q = 1, 2, .., p, p+1$ ($|(X^p \times Y)| = p+1$) y siendo n_q el número de conjuntos difusos referencias o etiquetas linguísticas del dominio $q-$ésimo $n_q = |\{A_{ql}\}|\ q = 1, ..., p\ l = 1, ..., n_q$ y con $n_q = |\{B_j\}|\ j = 1, ..., s$ para $q = p+1$. A partir de estos vectores podemos construir para cada cluster una matriz $W_h\ h = 1, 2, ..., k$ de forma:

$$w_{ij}^h = \Lambda gg\left(z_{1s_1}, z_{2s_2}, ..., z_{ps_p}, z_{(p+1)s_{(p+1)}}\right)$$

siendo $s_1, s_2, ..., s_p, s_{(p+1)}$ la combinación de etiquetas correspondiente a la entrada $ij-$ésima de la matriz W_h y Agg una función de agregación como la correspondiente a una Composición $S-T$, Resolución de Ecuaciones Relaciones o Frecuencias Difusas

Finalmente podemos aplicando una agregación de los elementos de las matrices parciales$W = F(W_1, W_2, ..., W_k)$ mediante diferentes funciones como:

$$w_{ij} = F_1(w_{ij}^k, w_{ij}^k, ..., w_{ij}^k) = \max_{h=1}^k w_{ij}^h$$
$$w_{ij} = F_2(w_{ij}^k, w_{ij}^k, ..., w_{ij}^k) = \frac{\sum_{h=1}^k w_{ij}^h}{k}$$

Referencias

[1] Babuska, R.; Verbruggen, H.B., "A new identification method for linguistic fuzzy models", *Open Meeting of the FALCON working group, Aachen, 1994.*

[2] Babuska, R.; Verbruggen, H.B., "Applied fuzzy modeling", *Proceedings IFAC congress AIRTC,* 1994.

[3] Bezdek, J.C., "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenun, New York, 1981.

[4] Delgado, M.; Vila, A.; Gomez-Skarmeta, A.F., "An Unsupervised Learning Procedure to Obtain a Classification from a Hierarchical Clustering", *EUFIT94,* pp. 528-533, Aachen, Germany, 1994.

[5] Delgado, M.; González, A., "A Frequency Model in a fuzzy Environment", *International Jorunal of Approximate Reasoning,* 11, pp. 159-174, 1994.

[6] Gath, I.; Geva, A., "Unsupervised Optimal Fuzzy Clustering", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 773-781, 1989.

[7] Karayiannis, N.B., " MECA: Maximum Entropy Clustering Algorithm", *Proceedings of the IEEE,* pp. 630-1648, 1994.

[8] Mizumoto, M., "Method of Fuzzy Inference suitable for Fuzzy Control",*J. Soc. Instrument and Control Engrs*, 58, pp. 959-963, 1989

[9] Pedrycz, W., "An identificaction algorithm in fuzzy relational systems", *Fuzzy Stes and Systems,* 13, pp. 153-167, 1984.

[10] Peng, X.; Wang, P., "On Generating linguistic rules for fuzzy models", *Lectures Notes in Computer Science* 313, Uncertainty and Intelligente Systems (B. Bouchon, L. Saitta, R.R. Yager (Eds.), pp. 185-192, 1988.

[11] Sin, S.K.; deFigueiredo R.J.P., "Fuzzy System Desing Through Fuzzy Clustering and Optimal Predefuzzification", *Proceedings of the IEEE,* pp. 190-195,1993.

[12] Sugeno, M.; Yasukawa T., "A Fuzzy-Logic-Based Approach to Qualitative Modeling", *IEEE Trans. Fuzzy Systems,* vol 1, n.1, pp. 7-31, 1993.

[13] Wang, L.; Mendel, J. M.; "Generating Fuzzy Rules by Learning from Examples", *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 22, n.6, pp. 1414-1427, 1992.

[14] Yager, R. R.; Filew, D. P., "Generation of fuzzy rules by mountain Clustering", *Tech. Report* #MII-1318, 1993.

Un modelo borroso de un sistema observado de forma inexacta

Luciano Sánchez, Adolfo Otero, Javier Tuya
Area de Lenguajes y Sistemas Informáticos
ETSII Gijón. Viesques, s/n
e-mail: luciano@etsiig.uniovi.es

Junio 15, 1995

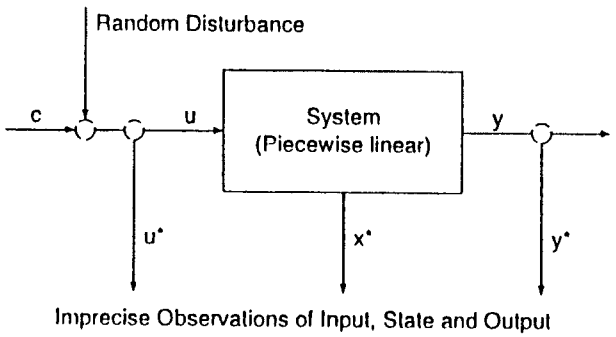


Figura 1: Sistema físico

Resumen

La naturaleza de las fuentes de interferencias en el proceso de señales es un problema clave en varias disciplinas aplicadas. Estas interferencias son modeladas mayoritariamente mediante procesos aleatorios. No obstante, se ha mostrado que algunos modelos pueden modificarse favorablemente si algunas fuentes se asimilan a experimentos borrosos.

Este trabajo se centra en este punto. Su objetivo es la construcción de un modelo matemático que explique un conjunto de medidas imprecisas tomadas en un sistema físico. El método propuesto, en combinación con otros criterios planteados en trabajos anteriores, se emplea para construir un modelo simple y que se mantenga dentro de unas tolerancias dadas. Las hipótesis realizadas consisten en que el efecto de las entradas no modeladas se traduce en una perturbación de naturaleza aleatoria y que el efecto de la imprecisión en las observaciones se explica mediante una distribución de posibilidad.

Palabras clave: *Modelo borroso, Variable aleatoria borrosa, Conjunto Aleatorio.*

1 Introducción

Suponga que se dispone de una secuencia de mediciones imprecisas de las entradas, salidas y variables de estado que gobiernan un sistema físico como el reproducido en la figura 1

$$E = \{c_k^*\}, \quad \text{donde } c_k^* = (x_k^*, u_k^*, y_k^*); \quad c_k^* \in \mathrm{R}^q, \ k = 1, 2 \ldots, n \qquad (1)$$

—donde y_k queda determinado por x_k y u_k. x_k^*, u_k^* e y_k^* son observaciones imprecisas de estos valores— y que se desea realizar un modelo matemático del comportamiento de este sistema.

Las técnicas más habituales se basan en suponer que el conjunto E está compuesto por instancias de un proceso aleatorio puntual ξ, por lo que tiene sentido hablar de una distribución de probabilidad conjunta de entrada, estado y salida

$$\mathrm{p}(x) = \mathrm{p}(\xi = x) \quad \text{para todo } x \in \mathrm{R}^q \qquad (2)$$

que mida la facilidad relativa de que se dé determinada combinación de estas tres variables.

De forma muy similar, los modelos borrosos se basan en la construcción del "grafo borroso" [16] de una función que relacione de nuevo valores compatibles de entrada, salida y estado. Es muy frecuente que se elija *a priori* una parametrización para este grafo, y que el problema se reduzca a una optimización no lineal de estos parámetros con respecto a un funcional del error entre las previsiones del modelo y las muestras inexactas. Existen numerosos trabajos a este respecto. Por ejemplo, pueden consultarse [1][6][12][13][14][15].

2 Modelo borroso-aleatorio

Ambos modelos —borroso y aleatorio— realizan un tratamiento homogéneo de todas las fuentes de incertidumbre involucradas. Pero es fácil constatar que las dos causas principales de incertidumbre —las entradas no modeladas y los errores de observación— son de naturaleza distinta. El error de observación se deriva de la imposibilidad de contestar inequívocamente a la ocurrencia del evento

"La entrada es u_k, y el estado es x_k, y la salida es y_k" (3)

que, a su vez, podría considerarse como borroso o como aleatorio.

En este sentido, obsérvese que es habitual indicar las tolerancias en las mediciones mediante la acotación de los máximos errores admitidos —"La velocidad es de 30 kph más/menos 1 kph"—. Esta afirmación se traduce en que la fracción de medidas para las que el error cae fuera de las tolerancias es muy baja; esto

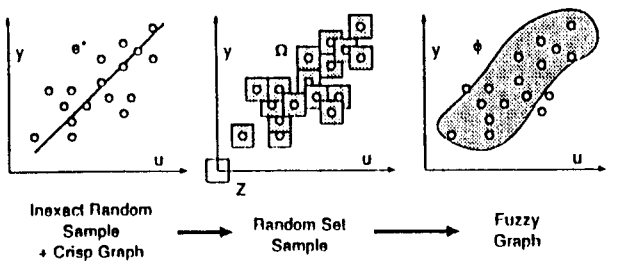


Figura 2: Relaciones entre los modelos probabilístico, basado en conjuntos aleatorios y basado en conjuntos borrosos

es, que la distribución de probabilidad del error de medida podría ser cualquiera de las que cumplen que

$$\mathrm{p}(e_k - e_k^* \notin Z) \leq \epsilon \tag{4}$$

donde Z es el intervalo de errores admisibles y ϵ es un número real pequeño. Es fácil ver que esta familia determina una distribución de posibilidad [10], cuyo conjunto borroso asociado es

$$\mu(x) = \begin{cases} 1 & \text{if } x - e_k \in Z \\ \epsilon & \text{en otro caso.} \end{cases} \tag{5}$$

Admitiendo, como en el modelo clásico, que el evento (3) es aleatorio, el que una tupla de medidas se extienda a un conjunto borroso conlleva que el experimento sea modelable mediante una variable aleatoria borrosa $\mathcal{Z}$ [4]. Es interesante constatar que el modelo que se deducirá está más relacionado con un conjunto aleatorio que con un proceso aleatorio. Para mostrar esto, en primer lugar —por simplicidad— se define la operación

$$\begin{gathered} x \oplus Z = \{x + z : z \in Z\} \\ \text{para cualquier } x \in \mathbf{R}^q,\ Z \subset \mathbf{R}^q. \end{gathered} \tag{6}$$

La variable $\mathcal{Z}$, al ser escrita en función de sus α-cortes —que son conjuntos aleatorios— queda de la siguiente forma:

$$\mathcal{Z}_\alpha = \begin{cases} \mathbf{R}^q & \text{if } \alpha \leq \epsilon \\ \xi \oplus Z & \text{en otro caso.} \end{cases} \tag{7}$$

Como todos los conjuntos aleatorios obtenidos cuando $\alpha > \epsilon$ son idénticos, y los demás son irrelevantes, se considerará que el conjunto aleatorio

$$\Phi = \xi \oplus Z \tag{8}$$

es un modelo del sistema.

Además, la función de cubrimiento a un punto de este conjunto

$$\phi(x) = \mathrm{p}(x \in \Phi) \tag{9}$$

cumple que $\phi = p$ cuando $Z = \{\tilde{0}\}$, luego este modelo es una generalización de (2), como por otra parte cabía esperar. Y cuando la medida de Lebesgue de Z es mayor que cero, siempre existe un conjunto borroso S' [8] que cumple

$$S'(x) = \phi(x) \quad \text{para todo } x \in \mathbf{R}^q \tag{10}$$

y que determina el grado de pertenencia de cualquier punto de $\mathbf{R}^q$ al conjunto formado por las tuplas de valores de entrada, salida y estado que son compatibles entre sí; es decir, un grafo borroso de la función de transferencia del sistema, o simplemente un modelo borroso del sistema. Asignando el valor 1 a un punto característico de este conjunto, el modelo propuesto será

$$S(x) = \frac{\phi(x)}{\sup_{\mathbf{R}^q}\{\phi(x)\}} \quad \text{para cualquier } x \in \mathbf{R}^q \tag{11}$$

2.1 Particularización del método

Supongase que las entradas no modeladas tienen un efecto asimilable al de una perturbación aleatoria gaussiana y que el sistema es lineal por tramos. Entonces, el conjunto E estará compuesto por observaciones inexactas de varias poblaciones normales distintas. Si estas observaciones se separan previamente en varias clases, el modelo borroso del sistema estará formado por la unión de varios conjuntos borrosos, cada uno de los cuales se ajustará a una de las clases obtenidas.

La forma de realizar esta clasificación ya ha sido introducida en trabajos anteriores [11]; se comienza definiendo un modelo muy complejo, en que cada conjunto borroso se estima a partir de una única muestra, y se simplifica mediante fusiones sucesivas de los conjuntos que lo componen, de acuerdo con una medida de entropía que pondera en cada caso la bondad de la unión. Al final, la misma entropía se emplea para decidir el mejor de los modelos así formados.

A continuación se detalla la construcción de uno de estos conjuntos. Sea F la función de distribución de una variable normal unidimensional, de media 0 y varianza 1. Sea además ξ otra variable aleatoria normal, de media c y covarianza C, que representará a una de las poblaciones normales a las que se ha hecho referencia.

C es simétrica y definida positiva, luego existe una matriz ortogonal B tal que

$$C = B^t V B \tag{12}$$

donde V es diagonal, y sus elementos son los autovalores de C. Sea

$$C^{-1/2} = B^t V^{-1/2} B \tag{13}$$

y

$$A = V^{-1/2} B \tag{14}$$

entonces, llamando $T(x) = A(x - c)$,

$$T(\xi) \to N(\tilde{0}, I) \tag{15}$$

donde I es la matriz identidad.

Sea ahora λ un valor tal que

$$\lambda = \inf\{t : T(x) \in [-t, t]^q \text{ para todo } x \in c \oplus Z\} \tag{16}$$

entonces, se define el modelo aleatorio del sistema como

$$\Phi = \xi \oplus \{z : T(z) \in [-\lambda, \lambda]^q\}. \tag{17}$$

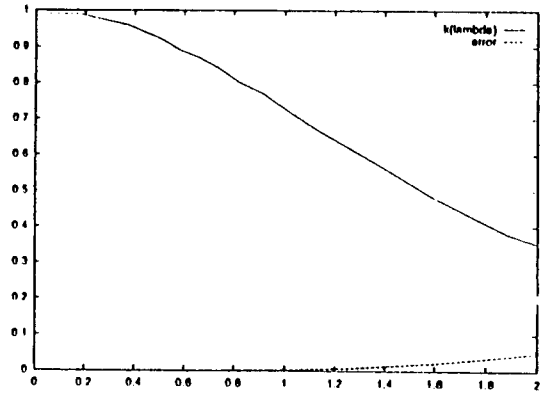


Figura 3: Factor de aproximación para procesos gaussianos

Haciendo $T(x) = (t_1, \cdots, t_n)^t$,

$$\begin{aligned} & \mathrm{p}(x \in \Phi) \\ & = \mathrm{p}(A(x-\xi) \in [-\lambda, \lambda]^q) \\ & = \mathrm{p}(A(x-c) + A(c-\xi) \in [-\lambda, \lambda]^q) \\ & = \mathrm{p}(N(\vec{0}, I) \in [-\lambda + t_1, \lambda + t_1] \times \\ & \ldots \times [-\lambda + t_q, \lambda + t_q]) \\ & = \prod_{i=1}^{q} (F(t_i + \lambda) - F(t_i - \lambda)) \end{aligned} \tag{18}$$

de donde se deduce que la función de pertenencia del conjunto S es

$$S(x) = \frac{1}{(2F(\lambda) - 1)^q} \prod_{i=1}^{q} (F(t_i + \lambda) - F(t_i - \lambda)). \tag{19}$$

Observe que existe un valor $k(\lambda)$ que hace

$$\begin{aligned} err(\lambda) = \sup_{x \in \mathbf{R}} & |F(x+\lambda) - F(x-\lambda) - \\ & (2F(\lambda) - 1)\mathrm{e}^{-\frac{1}{2}k(\lambda)x^2}| < 0.05 \end{aligned} \tag{20}$$

cuando $\lambda \in [0, 2]$ —los valores de $k(\lambda)$ y el máximo error de la aproximación se han dibujado en la figura 3.— Entonces,

$$S(x) \approx \frac{1}{(2F(\lambda) - 1)^q} \prod_{i=1}^{q} (2F(\lambda) - 1)\mathrm{e}^{-\frac{1}{2}k(\lambda)t_i^2} \tag{21}$$

pero

$$\begin{aligned} & \sum_{i=1}^{q} t_i^2 = t^t t = (A(x-c))^t (A(x-c)) = \\ & (x-c)^t (A^t A)(x-c) = (x-c)^t C^{-1} (x-c) \end{aligned} \tag{22}$$

luego

$$S(x) \approx \mathrm{e}^{-\frac{1}{2}k(\lambda)(x-c)^t C^{-1}(x-c)} \tag{23}$$

Este resutado corrobora que la imprecisión de las medidas conlleva que el grafo borroso del modelo sea menos específico. Pero, además, la fórmula (23) sugiere una relación entre la varianza aparente y la real:

$$C^* = \frac{1}{k(\lambda)} C. \tag{24}$$

Note también que, para valores altos de λ, S tiende a parecerse al conjunto

$$\{x : T(x) \in [-\lambda, \lambda]^q\}. \tag{25}$$

De hecho, λ expresa el balance entre borrosidad y aleatoriedad en la variable aleatoria borrosa.

3 Ejemplo

Se dispone del siguiente conjunto de mediciones, tomadas en un sistema físico:

$e_1 = (0.619, 0.695)$	$e_2 = (1.392, 1.183)$
$e_3 = (1.276, 0.901)$	$e_4 = (0.764, 1.398)$
$e_5 = (0.704, 0.947)$	$e_6 = (0.885, 0.646)$
$e_7 = (1.472, 1.329)$	$e_8 = (0.607, 0.675)$
$e_9 = (0.867, 0.589)$	$e_{10} = (1.476, 0.792)$
$e_{11} = (-0.83, -0.912)$	$e_{12} = (-1.413, -1.265)$
$e_{13} = (-1.469, -1.129)$	$e_{14} = (-1.128, -1.179)$
$e_{15} = (-1.106, -0.829)$	$e_{16} = (-0.509, -1.191)$
$e_{17} = (-0.763, -0.755)$	$e_{18} = (-1.324, -0.665)$
$e_{19} = (-1.088, -1.259)$	$e_{20} = (-0.762, -1.28)$
$e_{21} = (2.225, 2.474)$	$e_{22} = (2.345, 1.916)$
$e_{23} = (1.924, 2.118)$	$e_{24} = (2.061, 1.874)$
$e_{25} = (1.658, 2.362)$	$e_{26} = (2.405, 2.334)$
$e_{27} = (1.822, 1.583)$	$e_{28} = (1.513, 1.596)$
$e_{29} = (1.559, 1.679)$	$e_{30} = (2.056, 1.533)$

El instrumento empleado para tomar las medidas garantiza que el 99% de las veces los valores de las variables se encuentran en un círculo de diámetro 0.5 unidades, centrado en la medida inexacta. Se desea estimar un modelo borroso del sistema.

3.1 Solución

Se considerará que los valores tabulados son observaciones imprecisas de varias variables aleatorias gaussianas bivariantes, medidas con una precisión que se caracteriza mediante el conjunto Z

$$Z = \{x : x^t \mathbf{Z} x \leq 1\} \tag{26}$$

donde

$$\mathbf{Z} = \begin{pmatrix} 16 & 0 \\ 0 & 16 \end{pmatrix} \tag{27}$$

Si la varianza del proceso aleatorio es C, $T(Z)$ está dentro de una celda $[-\lambda, \lambda]^2$, de modo que

$$\lambda = \min\{t : T(x) \in [-t, t]^2,\ x \in Z\}. \tag{28}$$

Dado que

$$(T(x))^t \mathrm{W} T(x) \leq 1, \quad \text{para } \mathrm{W} = (A^{-1})^t \mathrm{Z} A^{-1} \tag{29}$$

entonces

$$\lambda = \max\{\sqrt{u_{ii}} : \mathbf{W}^{-1} = (u_{ij})\}. \quad (30)$$

Cuando $\lambda < 2$, la pertenencia del conjunto se aproximará mediante

$$\mathcal{S}(x) \sim \mathrm{e}^{-0.5k(\lambda)x^t(C)^{-1}x} \sim \mathrm{e}^{-0.5x^t(C^*)^{-1}x} \quad (31)$$

donde C^* es la varianza del la muestra inexacta. De otra forma, $\mathcal{S}$ degenera en el conjunto

$$\mathcal{S} = \{x : T(x) \in [-\lambda, \lambda]^2\} \quad (32)$$

A continuación, se realiza una partición del conjunto de ejemplos y se ajusta un modelo borroso a cada subconjunto. En este caso, la partición se ha realizado mediante un clustering jerárquico aglomerativo, empleando como similaridad la ganancia de longitud de descripción borrosa definida en [11]; la disimilaridad entre dos subconjuntos disjuntos de E, E_1 y E_2 es

$$\begin{aligned}\Delta(\mathcal{S}_1, \mathcal{S}_2) = \#(E_1 \cup E_2)U(\mathcal{S}_{1+2}) - \\ \#(E_1)U(\mathcal{S}_1) - \#(E_2)U(\mathcal{S}_2)\end{aligned} \quad (33)$$

y el mejor modelo se determina por el valor mínimo de

$$\mathrm{LD} = \log \#(E) + \sum_{i=1}^{N} \frac{\#(E_i)}{\#(E)} U(\mathcal{S}_i) \quad (34)$$

donde U es la noespecificidad de Klir y N el número de regiones que componen el modelo. La noespecificidad se ha calculado a partir de las covarianzas estimadas con (24).

Los α cortes al nivel 0.1 de los conjuntos ajustados a cada elemento de la partición se muestran en la figura 4. El mejor modelo, de acuerdo con el mismo criterio, está formado por 2 conjuntos borrosos:

$$\mathcal{S}_1(x,y) = \mathrm{e}^{0.5(x,y)^t \begin{pmatrix} 0.33 & 0.29 \\ 0.29 & 0.36 \end{pmatrix} (x,y)} \quad (35)$$

$$\mathcal{S}_2(x,y) = \mathrm{e}^{0.5(x,y)^t \begin{pmatrix} 0.09 & -0.02 \\ -0.02 & 0.05 \end{pmatrix} (x,y)} \quad (36)$$

donde

$$\lambda_1 = 1.02 \quad C_1 \sim 0.73 C_1^*$$
$$\lambda_2 = 1.13 \quad C_2 \sim 0.67 C_2^*$$

4 Conclusión

Se ha comprobado que, cuando las observaciones de las variables de un sistema físico son inexactas, estos experimientos son modelables mediante variables aleatorias borrosas. En este caso particular sólo son significativos ciertos cortes de estas variables, que son conjuntos aleatorios. Aprovechando que las funciones de cubrimiento a un punto de un conjunto aleatorio son compatibles con pertenencias borrosas, se ha diseñado un método objetivo para determinar una función de pertenencia a partir de información numérica imprecisa.

El algoritmo final, en combinación con trabajos anteriores, ha sido empleado para modelar cualitativamente un sistema físico no lineal.

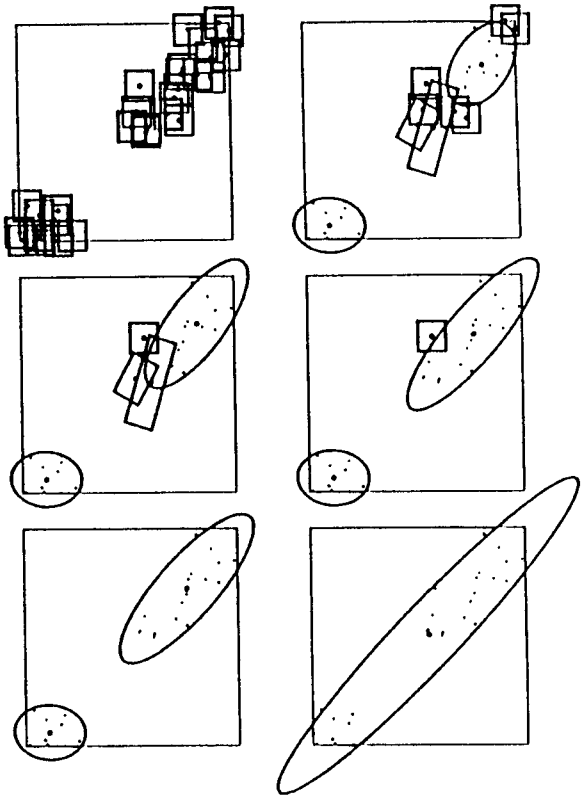


Figura 4: α-cuts ($\alpha = 0.1$) of M_{30}, M_{10}, M_5, M_3, M_2 y M_1

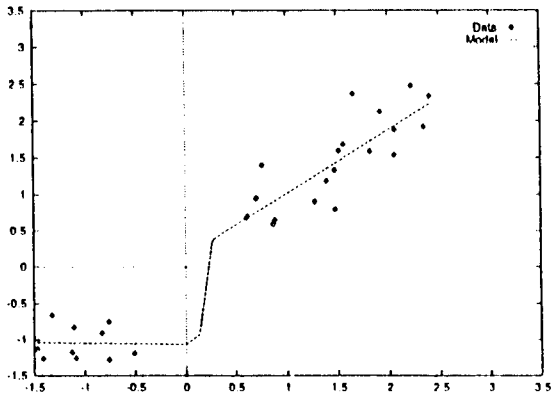


Figura 5: Estimación puntual basada en este método

Referencias

[1] Berenji, H. R. (1994) "Fuzzy Systems that can Learn". In *Computational Intelligence: Imitating Life*. IEEE Press. pp 23-30

[2] Gil, M. A. (1992) "Sufficiency and fuzzyness in random experiments". Ann. Inst. Statist. Math. 44, No. 3, pp 451-462

[3] Gupta, M.M. and Qi, J. (1991) "Theory of t-norms and Fuzzy Inference Methods". Fuzzy Sets and Systems 40, pp 431-450.

[4] Kwakernaak, H. (1978) "Fuzzy Random Variables: I. Definitions and theorems" Information Sciences, 15, pp 1-29.

[5] Klir, G., Folger, T. (1988) *Fuzzy Sets, Uncertainty, and Information*. Prentice Hall.

[6] Khedkar, P. S. (1994) "Learning as Adaptive Interpolation in Neural Fuzzy Systems". In *Computational Intelligence: Imitating Life*. IEEE Press. pp 31-42

[7] Laviolette, M. "A probabilistic and statistical view of fuzzy methods". To appear in Technometrics.

[8] Nguyen, Goodman (1985) *Uncertainty models for Knowledge-based systems*. North Holland.

[9] Procyk, T.J., Mamdani, E.H. (1979) "A linguistic self-organizing process controller". Automatica, Vol 15, pp 15-30. Pergamon Press

[10] Shafer, G. (1976) *A mathematical theory of evidence*. Princeton University Press.

[11] Sánchez, L. (1994) "Modelado borroso de sistemas mediante un algoritmo de simplificación". IV Congreso Español de Tecnologías y Lógica Fuzzy. Blanes.

[12] Sugeno, M. , Tanaka, K. (1991) "Succesive identification of a fuzzy model and its applications to prediction of a complex system". Fuzzy Sets and Systems 42. pp 315-334

[13] Sugeno, M., Yasukawa, T. (1993) "A fuzzy logic based approach to qualitative modelling". IEEE Transactions on Fuzzy Systems, Vol 4, No. 4. pp 7-31

[14] Tagaki, T. Sugeno, M. (1985) "Fuzzy identification of systems and its application to modeling and control". IEEE Trans on Systems, Man and-Cibernetics, 15. pp 116-132.

[15] Wang, Mendel (1992) "Generating Fuzzy Rules by Learning from Examples". IEEE Transactions on S.M.C., vol 22. No. 6.

[16] Zadeh, L.A. (1993) "The Evolution of Fuzzy Set Theory and its Applications". IFCA. Terrassa.

Métodos Neuronales para la Extracción de Reglas Difusas

José M. Benítez, Armando Blanco, Miguel Delgado e Ignacio Requena
Depto. Ciencias de la Computación e Inteligencia Artificial
E.T.S. Ingeniería Informática
Avda. Andalucía, 38
Universidad de Granada
18071 - GRANADA
e-mail: jmbs, requena@robinson.ugr.es

Resumen

En trabajos anteriores, presentamos una metodología empírica basada en Redes Neuronales, para obtener reglas difusas que permiten describir un sistema, a partir de un conjunto de ejemplos con entradas y las salidas correspondientes del mismo. Ahora se describen los resultados obtenidos, y se presenta otro procedimiento, también basado en Redes Neuronales con Backpropagation, para obtener reglas difusas, sin necesidad de establecer previamente las etiquetas o valores de las variables que gobiernan el sistema.

Palabras clave: *Obtención de reglas difusas, Aprendizaje, Redes Neuronales Artificiales.*

1 Introducción.

Un problema básico y de múltiples aplicaciones en Inteligencia Artificial es el de la identificación y reproducción del comportamiento de sistemas. Entre sus grandes áreas de aplicación se encuentran: la adquisición del conocimiento y la de determinar relaciones causales.

Entre los métodos de representación del conocimiento el más usado es el basado en reglas. En muchos problemas reales, la descripción del sistema mediante reglas difusas resulta ventajosa, bien porque la descripción es más acorde con la situación real, o bien porque la información subyacente en el sistema es imprecisa o difusa, usando como una herramienta eficaz para el tratamiento de la incertidumbre, técnicas difusas.

En este trabajo se exponen métodos para realizar la extracción automática del conocimiento, empleando Redes Neuronales Artificiales (RNA). Por un lado, una metodología ya presentada en [1, 2], y por otro lado, otra metodología que utiliza en la primera fase las ideas propuestas en [4] para extraer reglas de producción clásicas. Además, un método de este tipo puede ayudar en el estudio de las RNA, tratando de ofrecer una interpretación del funcionamiento de las mismas.

Las características de las RNA, como su capacidad de aprendizaje y generalización a partir de ejemplos, y su robustez frente al ruido, las hacen particularmente eficaces para recoger la información contenida en un conjunto de datos y reproducirla de forma fiel. Se puede así, expresar, en forma de reglas difusas, el conocimiento captado por una RNA mediante aprendizaje de un conjunto de muestras.

2 RNA para adquisición de conocimiento.

En [4] se propone un método para obtener reglas de producción clásicas de un sistema, del tipo

$$R_i: \text{ If } a_1^i \wedge a_2^i \wedge \ldots \wedge a_{n_i}^i \text{ then } b_i$$

Esta regla indica que si en un ejemplo se presentan los atributos a_j^i entonces se trata de un objeto de clase b_i (consideran variables binarias). Las reglas deben ser lo más simple posible, es decir, ser suficientes para caracterizar al objeto y con el menor número de atributos en el antecedente. La idea clave es entrenar una RNA multicapa, incorporando las propias salidas como entradas adicionales, y estudiar la relación entre los pesos de las entradas auténticas y los de las salidas (entradas adicionales), hacia la capa oculta.

En esencia, el método se puede resumir en:

1. Se entrena una RNA con una sola capa oculta, donde las salidas se usan además como entradas adicionales. Es decir, si los datos del sistema tienen n entradas y m salidas, los ejemplos de entrenamiento tienen $n+m$ entradas y m salidas. Se usa el algoritmo de retropropagación de errores (BackPropagation).

2. Para cada par (entrada i, salida j) se calcula una medida de la concordancia o proximidad entre los pesos de estas neuronas hacia la capa oculta, que será menor cuanto más estrecha sea la relación:

$$\mathrm{SDCP}_{ij} = \sum_{k=1}^{h} (w_{ik} - w_{jk})^2$$

3. Se obtiene un nuevo conjunto de muestras a partir de los ejemplos, complementando las entradas. Se entrena una RNA sin capas ocultas con este conjunto de muestras y empleando la regla de Hebb. Así se obtienen relaciones inhibitorias entre entradas y salidas. La medida de correlación entre una entrada y una salida será el propio peso V_{ij} en la red entrenada.

4. Se mide la relación entre cada entrada y salida, mediante

$$prod_{ij} = \mathrm{SDCP}_{ij} \cdot V_{ij}$$

5. Para cada salida, se ordenan las entradas según el sentido creciente de los valores de estos productos. Se establece un punto de corte adecuado y se consideran relevantes los atributos que quedan por debajo del punto de corte. La regla se construye conjuntando en el antecedente los atributos seleccionados como relevantes y poniendo en el consecuente la salida.

El método es fácilmente extensible para el caso en que se consideran atributos con un número finito de posibles valores. En [4] señalan algunas posibilidades para el caso de que no exista un corte adecuado, lo que indicaría más de una regla para esa salida.

En [4] se aplica este método al ejemplo del LED de 7 segmentos, mostrando buenos resultados.

En [3] dan también indicaciones para extender el método al caso con atributos continuos. Las reglas de producción son en este caso de la forma:

$$R_i: \text{ If } (v_{min}{}_1^i \le a_1^i \le v_{max}{}_1^i) \wedge \ldots \wedge \\ \wedge (v_{min}{}_{n_i}^i \le a_{n_i}^i \le v_{max}{}_{n_i}^i) \text{ then } b_i$$

En esta versión del método se realizan todos los pasos del caso anterior. Pero en el paso 5, sólo se obtiene la estructura de cada regla (que llamaremos pre-regla), indicando la presencia de atributos relevantes. Los rangos adecuados para estos atributos se establecen siguiendo los pasos:

1. Determinar qué ejemplos son cubiertos por cada pre-regla. Para ello se emplea una medida de suma de diferencias al cuadrado y se asigna a cada regla una cota c_i. Esta cota puede ser distinta para distintas reglas, y de hecho es modificada dinámicamente hasta conseguir que todos los ejemplos estén cubiertos por alguna regla.

2. Para cada atributo incluido en la regla, se establece su rango de validez como el intervalo cuyos extremos son el mínimo y el máximo de los valores de ese atributo para los ejemplos cubiertos por la regla.

3 Extracción de reglas difusas.

En muchos problemas reales, la descripción del sistema mediante reglas difusas es necesaria (p.e., los datos son vagos y se representan mejor con números difusos) o proporciona ganancia porque las reglas difusas representan más adecuadamente el sistema.

Exponemos dos metodologías sobre RNA. Una se había presentado en un trabajo anterior [1], basada en la construcción previa de todas las reglas posibles, seleccionando entonces la mejor combinación de ellas. Aquí comentamos con más detalle los resultados obtenidos con esta técnica y, nos vale como punto de comparación y contraste frente a la segunda. Esta última, basada en parte en el método indicado en la sección 2, construye las reglas a partir de la RNA entrenada con los ejemplos.

3.1 Metodología de Selección (descendente).

En [1], presentamos una metodología para obtener las reglas difusas que describen un sistema, a partir de un conjunto de ejemplos de su comportamiento, y que básicamente consiste en:

1. Suponemos que tenemos identificadas las variables relevantes en la descripción del comportamiento del sistema, cuyos valores son etiquetas. Las etiquetas se representan con números difusos trapezoidales, que se procesan en la RNA como se indica en [11].

   Entrenamos una RNA (RE) con un conjunto de ejemplos del comportamiento del sistema. Los valores reales en los ejemplos se procesan como números difusos *singleton*.

2. Construimos todas las reglas posibles, según las etiquetas (valores) de las variables identificadas.

3. Colocamos los antecedentes de cada regla en la red y comparamos la salida de ésta con el consecuente. Descartamos las reglas que claramente no se adaptan al sistema (RE), y obtenemos una primera selección de reglas relevantes en la descripción del mismo.

4. De entre todas las posibles combinaciones de reglas preseleccionadas, se toma como conjunto definitivo de reglas la combinación que mejor se adapta a la RNA (RE) y menor cardinal tiene.

Para reducir el coste computacional del paso 4, en [2], empleamos un algoritmo tipo *greedy*. La idea subyacente es sencilla: se trata de partir de un conjunto de reglas vacío y en cada paso añadir la mejor regla de entre las que quedan, hasta que el conjunto alcance el criterio deseado.

Sea el conjunto de muestras $M = \{e_j\}$, y $R = \{R_i\}$ el conjunto de reglas (paso 3). La componente l-ésima de entrada del ejemplo j-ésimo es x_j^l y la k-ésima de salida y_j^k. Las reglas se representan en la forma

$$\text{If } X_1 \text{ is } A_1 \text{ and} \ldots \text{and } X_n \text{ is } A_n \text{ then } Y \text{ is } B$$

Definiciones.

Grado de Cobertura[1] de la regla R_i sobre la muestra e_j:

$$Gc(R_i, e_j) = 1 - \frac{\sqrt{\sum_{l=1}^{t}(IP(A_i^l) - x_j^l)^2 + \sum_{k=1}^{s}(IP(B_i^k) - y_j^k)^2}}{MaxD},$$

donde IP(D) es el valor del Índice Promedio[2] del conjunto difuso D [6, 7]; y $MaxD$ es el valor máximo para la expresión en el numerador. $Gc(R_i, e_j)$ indica el grado en que R_i representa a e_j.

Grado Medio de cobertura de una regla R_i, (o un conjunto de reglas):

$$GMc(R_i) = \frac{\sum_{j=1}^{|M|} Gc(R_i, e_j)}{|M|}.$$

Grado Parcial de cobertura de un conjunto de reglas C y una regla R_j:

$$GPc(C, R_j) = \sum_{k=1}^{|M|} \begin{cases} Z_{C,j,k}, & \text{si } Z_{C,j,k} > 0 \\ 0, & \text{en otro caso} \end{cases}$$

con

$$Z_{C,j,k} = Gc(R_j, e_k) - Gc(C, e_k)$$

El GPc indica en cuánto se mejora la cobertura del conjunto de reglas cuando se le añade R_j. $GPc(C, R_j)$ es cero cuando una regla no mejora en nada la cobertura del conjunto.

El algoritmo greedy selecciona en cada paso la regla que hace que la cobertura del conjunto sea mayor, o sea, la regla con mayor grado parcial de cobertura. El proceso culmina cuando todas las muestras (o un porcentaje elevado de ellas, β) son cubiertas con un grado superior a cierto nivel α. Podemos expresar el criterio de parada como que el grado medio de cobertura sea mayor o igual a $\alpha\beta$. Con estos dos parámetros podemos regular la precisión que le exigimos a la descripción que devuelve el algoritmo. El algoritmo queda:

1. $C \leftarrow R_i$ tal que $GMc(R_i)$ es máximo.
2. Mientras $((GMc(C) < \alpha\beta)$ y $(|C| < |R|))$
   $C \leftarrow C \cup R_j$ tal que $R_j \notin C$ y $GPc(C, R_j)$ es máximo.
3. El conjunto de reglas es C.

El algoritmo produce un resultado casi-optimal que puede mejorarse modificando el paso 1, para construir varios conjuntos de reglas partiendo de una regla distinta cada vez y, al final quedarse con el mejor de todos los obtenidos.

Ninguna de las dos versiones asegura obtener el conjunto de reglas óptimo pero sí son bastante rápidos. Sus complejidades algorítmicas son $O(|M| \cdot |R|^2)$ para la primera versión y $O(|M| \cdot |R|^3)$ para la segunda, que son bastante pequeñas con respecto a una búsqueda exhaustiva que supondría $O(|M| \cdot 2^{|R|})$.

3.1.1 Resultados Experimentales.

Para comprobar el rendimiento del método de extracción de reglas lo aplicamos al problema del frenado de un coche [13]. Se trata de determinar la fuerza que hay que aplicar a un coche para fijarlo en un punto considerando la posición y la velocidad. Además, hay que considerar la acción de fuerzas aleatorias con módulo y sentido cambiantes. El problema se puede resolver con un controlador difuso que emplea una base de siete reglas con la siguiente estructura:

$$R_i\text{: If } p \text{ is } A_i \text{ and } v \text{ is } B_i \text{ then } f \text{ is } C_i$$

donde p es la posición, v es la velocidad y f la fuerza. Son tres variables lingüísticas que toman valores en conjuntos con granularidades 5, 3 y 5, respectivamente.

Para aplicar el paso 1 del método, usamos 250 ejemplos de control, en que los valores reales se representan como números difusos *singleton*. Con estos ejemplos y con el algoritmo del Backpropagation entrenamos una red con 8 neuronas de entrada, 3 neuronas ocultas y 4 neuronas de salida.

[1] Otra definición de grado de cobertura ajustada al sistema de inferencia o control en estudio, puede ser:

$$Gc(R_i, e_j) = 1 - \frac{d(y_j, S(R_i, x_j))}{MaxD}$$

donde d es una métrica en el espacio de las salidas y $S(R_i, x_j)$ representa la salida del sistema de difuso para x_j cuando su única regla es R_i.

[2] Se podría utilizar otra función índice sobre números difusos en vez del Índice Promedio.

La aplicación del paso 2 produce las 75 posibles reglas, que tras la ejecución del paso 3 se reducen a 15.

Para obtener el conjunto definitivo de reglas usamos el algoritmo greedy utilizando diversos valores para el grado de cobertura. Conforme se disminuye el grado medio de cobertura requerido disminuye el número de reglas necesario. Variando el valor de este grado entre 1 y 0'77 hemos obtenido conjuntos de reglas de cardinales entre 11 y 2. Como casos representativos: para un grado de 0'812 el conjunto tiene 7 reglas y para un grado de 0'8 el conjunto lo forman 4.

Modificando debidamente el grado medio de cobertura podemos establecer el nivel de precisión que le exigimos a la descripción, lo que se puede traducir en el grado de efectividad en el caso de reglas difusas para control.

De acuerdo con los resultados obtenidos en los experimentos se pueden señalar los siguientes comentarios y conclusiones:

- El conjunto de 7 reglas obtenido con un grado medio de 0'812 es distinto al obtenido en [13]. Sin embargo, el control del sistema es igualmente perfecto. Con los conjuntos de 5 y 4 reglas aún se consigue un buen control, aunque, disminuye ligeramente la fuerza externa máxima que se puede controlar.

- El algoritmo explota las propiedades de generalización y robustez de las redes neuronales para extraer reglas difusas a partir de un conjunto de ejemplos. El principal objetivo en la elaboración de las reglas es el de obtener una descripción del comportamiento del sistema, pero también son hábiles para control difuso. La precisión de la descripción ofrecida se puede regular mediante dos parámetros.

- El método es independiente del modo en que se establecen las etiquetas lingüísticas. Considerando que el principal objetivo es obtener una descripción de un proceso que sea significativa para un experto humano, el conjunto de etiquetas debe tener un cardinal limitado por la granularidad detectable y la forma de sus funciones de pertenencia debe cumplir algunos requerimientos como los indicados en [8].

- En el paso final se ha empleado un algoritmo greedy que supone una solución rápida y simple. Como alternativa se puede emplear alguna técnica de optimización combinatoria tal como el Simulated Annealing, la Máquina de Boltzmann o los Algoritmos Genéticos.

- En casos reales donde la intervención de los expertos humanos es importante en las definiciones de cobertura se utilizaría un índice de decisión personal asociado al experto, como se indica en [11].

3.2 Metodología Constructiva (ascendente).

Una descripción del conocimiento incluido en un conjunto de datos muy cercana e inteligible para un ser humano es la regla de producción. Los sistemas basados en reglas difusas son, al igual que las RNA, aproximadores universales [5] y, además pueden justificar las respuestas que dan apoyándose en su base de reglas.

El caso de datos en que los atributos son continuos es harto común, sin embargo, la representación mediante las reglas que se generan con el método indicado en [3] no es apropiada. Entendemos que una aproximación más acorde con la realidad sería utilizar reglas difusas, considerando cada atributo como una variable cuyos posibles valores son conjuntos difusos sobre el intervalo de definición. Por otro lado, una vez establecidas las reglas, los conjuntos pueden ser etiquetados dando a las reglas un aspecto más adecuado para su uso y comprensión por parte humana. Este aspecto resulta de importancia capital, si tratamos de crear un sistema que nos ofrezca respuestas justificadas.

Con esta nueva perspectiva, se ha diseñado un método constructivo que expresa el conocimiento contenido en un conjunto de datos en forma de reglas difusas.

En primer lugar, consideramos que los datos obtenidos del sistema, y que utilizaremos en el entrenamiento de la RNA, están expresados como números reales (crisp), normalizados a [0,1]. No obstante, las variables que intervienen en las reglas que describirán al sistema, van a ser difusas y valoradas con etiquetas lingüísticas expresadas como números difusos trapezoidales.

La primera aproximación que planteamos es la siguiente:

1. Se transforma el conjunto de datos normalizando las entradas y salidas. Se entrena una RNA con una sola capa oculta, añadiendo las salidas como entradas adicionales.

2. Para cada entrada i y salida j (en la entrada de la red), se calcula

$$S_{ij} = \sum_{k=1}^{h} (w_{ik} - w_{jk})^2$$

   donde k recorre las neuronas de la capa oculta

3. El conjunto de datos se transforma normalizando entradas y salidas. Además las entradas se cambian por su complemento a 1. Se entrena una

RNA, sin capas ocultas, empleando la regla de Hebb. Los pesos de la red entrenada son ahora V_{ij}.

4. Se mide la relación entre cada entrada y salida, mediante

$$prod_{ij} = S_{ij} \cdot V_{ij}$$

Para cada salida j, se ordenan estos productos de menor a mayor.

5. Buscamos un punto de corte adecuado en estos productos ordenados, y consideramos las entradas i (variables) que quedan por debajo del punto de corte, como contributivas a la salida j. Obtenemos así, lo que podemos considerar como la estructura de una regla o pre-regla, para cada salida j, expresada en términos de las variables como:

$$P_{R_i} : X_p^i, X_q^i, X_r^i \rightarrow Y_j^i$$

6. Para obtener las etiquetas asociadas a las variables en la pre-regla, hacemos lo siguiente:

   (a) Consideramos en los ejemplos de entrenamiento, sólo las componentes de las variables que están en la pre-regla. Con estos subvectores, hacemos un clustering difuso, basado en la proximidad de los valores de Y_j (o que además tengan también valores próximos en las X_i de la pre-regla). Desestimamos para esta regla los ejemplos que no clasifica bien.

   (b) Con los ejemplos de cada clase, construimos la función de pertenencia de la etiqueta para cada variable. Obtenemos por tanto, tanta reglas efectivas como clases en el clustering. Una forma sencilla, es considerar números triangulares o trapezoidales. Para una variable, el soporte vendría dado por sus valores mínimo y máximo en los ejemplos de esa clase. La moda vendría dada por la mediana (cuartiles 1 y 3, para trapezoidales) de los valores de esa variable en los ejemplos de la clase). Otra forma es obtener una aproximación lineal de la distribución de frecuencias en esos valores de la variable.

En segundo lugar, consideramos que los ejemplos del sistema vienen expresados por etiquetas o números difusos (trapezoidales (a, b, c, d), sin pérdida de generalidad). En este caso, el método se puede aplicar también, pero con modificaciones en los siguientes pasos:

- En el paso 1, utilizamos el procesamiento de los números difusos en la red, como se indica en [10], pero considerando las salidas también como entradas.
- En el paso 2, para calcular S_{ij}, cada sumando se obtiene ahora como

$$\frac{(\mathrm{a}(e_i) - \mathrm{a}(s_j))^2 + (\mathrm{b}(e_i) - \mathrm{b}(s_j))^2}{4} + \frac{(\mathrm{c}(e_i) - \mathrm{c}(s_j))^2 + (\mathrm{d}(e_i) - \mathrm{d}(s_j))^2}{4}$$

  donde $\mathrm{x}(e_i)$ representan los pesos del número difuso de entrada i-ésimo, para las correspondientes entradas (a, b, c, d) que representan al difuso, y $\mathrm{x}(s_j)$ indica lo mismo para el número difuso de salida j-ésimo, puesto como entrada.

- En el paso 3, los V_{ij} se obtienen de la misma forma que en el punto anterior.

Los demás pasos del proceso se realizan de la misma forma.

Estamos aplicando estas metodologías sobre sistemas concretos. Los resultados iniciales parecen prometedores. Los resultados definitivos serán presentados en el congreso.

4 Conclusiones y comentarios finales.

Se han expuesto dos metodologías para obtener reglas difusas de un sistema, del que se conoce un conjunto de ejemplos de su comportamiento, en base a variables o atributos posiblemente relevantes en su comportamiento.

Ambas metodologías consideran en primer lugar, el entrenamiento de una RNA, usando el algoritmo de retropropagación de errores (Backpropagation) y los ejemplos disponibles del sistema. Ambas metodologías se pueden usar, tanto si los ejemplos del sistema vienen dados como valores crisp, o como números difusos (etiquetas, p.e.).

La primera metodología (inicialmente presentada en Blanes'94 [1] parte de la construcción de todas las reglas posibles, y selecciona las relevantes en la descripción del sistema; por eso la señalamos como metodología selectiva o descendente. Hemos presentado resultados experimentales obtenidos sobre un ejemplo de control difuso y, son bastante satisfactorios.

La segunda metodología extiende un método para obtener reglas de producción con atributos binarios [4] a la extracción de reglas difusas que permitan describir un sistema. Determina las variables de entrada contributivas a cada posible salida, y construye las reglas difusas; por eso la indicamos como metodología constructiva. Los resultados iniciales parecen prometedores.

Estamos considerando otras formas de obtener las etiquetas asociadas a las variables (paso 6) como:

Con los datos del sistema (ejemplos), se hace previamente un clustering difuso según criterios de los expertos y/o del propio problema a resolver. Para cada posible variable en la salida del sistema, usando el grado de cobertura definido en [2] (ver sección 3.1), obtenemos los ejemplos cubiertos por la pre-regla en cada clase (etiqueta) de la variable de salida Y, por encima de un nivel (prefijado) α, y la obtención de las etiquetas se hace igual que antes.

Otra forma es obtener para cada ejemplo, el nivel mínimo α con que cada ejemplo es cubierto por la pre-regla, para cada clase (etiqueta) de la variable Y, y construir el número difuso en base a los ejemplos, y estos grados de pertenencia. En este caso, podemos considerar dos posibilidades:

- Con un grado de precisión α, cada pre-regla cubre a un determinado conjunto de ejemplos. Con estos ejemplos construimos la función de pertenencia de cada variable (su etiqueta) en la regla. Esto da un conjunto global bueno de reglas, al nivel α.

- Con un grado de precisión α, cada ejemplo estará cubierto por una regla o no.

  - El grado de pertenencia del ejemplo (i.e., del valor del atributo correspondiente) es el menor de estos α.

  - Con estos grados de pertenencia, construir la función de pertenencia

  Esto daría reglas más precisas cada una.

Referencias

[1] Benítez, J.M., Blanco, A., and Requena, I. *Un procedimiento empírico con Redes Neuronales para obtener reglas difusas.* IV Congreso de la Asociación Española de Tecnologías y Lógica Fuzzy. Blanes, Septiembre, 1994.

[2] Benítez, J.M., Blanco, A., and Requena, I. *An Empirical Procedure to Obtain Fuzzy Rules using Neural Networks.* 6th IFSA Congress. Brasil, July, 1995.

[3] Sestito S. and Dillon, T. *Automated Knowledge Acquisition of Rules with Continuously Valued Attributes.* Proc. Avignon'92, 1992.

[4] Sestito S. and Dillon, T. *Knowledge Acquisition of Conjunctive Rules Using Multilayered Neural Networks.* Inter. J. Intelligent Systems, 8, 779–805, 1993.

[5] Castro, J.L. *Fuzzy Logic Controllers are Universal Approximators.* IEEE Trans. on Syst. Man and Cybern. **25**(4), April, 1995.

[6] Campos, L.M. and González, A. *A Subjective Approach for Ranking Fuzzy Numbers.* Fuzzy Sets and Systems, **29**, 145–153, 1989.

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

[8] Lee, C.C. *Fuzzy Logic in Control Systems: Fuzzy Logic Controller.* IEEE Trans. on Systems, Man, and Cybernetics. **20**(2), Part I, 404–419, Part II, 419–435, 1990.

[9] Requena, I. *Tesis doctoral: Redes Neuronales en Problemas de Decisión con Ambiente Difuso.* Depto. Ciencias de la Computacin e I.A. Universidad de Granada, 1992.

[10] Requena, I., Delgado, M., and Verdegay, J.L. *Automatic Ranking of Fuzzy Numbers with the Criteria of a Decision-maker Learnt by an Artificial Neural Network.* Fuzzy Sets and Systems, **64**, 1–19, 1994.

[11] Requena, I., Blanco, A., Delgado, M., and Verdegay, J.L. *A Decision Personal Inder to Fuzzy Numbers based on Artificial Neural Networks.* Fuzzy Sets and Systems. (To appear)

[12] Simpson, P.K. *Artificial Neural Systems.* Pergamon Press, 1989.

[13] Wolf, T. *Das Fuzzy-Mobil.* mc 3/91. pp. 50–63, 1991.

[14] Yamakawa, T. *Stabilization of an Inverted Pendulum by a High-Speed Fuzzy Logic Controller Hardware System.* Fuzzy Sets and Systems, **32**, 161–180, 1989.

[15] Zadeh, L.A. *The Concept of a Linguistic Variable and its Applications to Approximate Reasoning.* Part I, II, and III. Information Sciences, **8**, pp. 199–249, pp. 301–357 and **9**, pp. 43–80, 1975.

APROXIMACION GLOBAL EN LA OPTIMIZACION DE UN PROCESO DE FORMACION/APRENDIZAJE SUJETO A INCERTIDUMBRE

Ramón Ferreiro García y Vicente Yañez Gonzalez
Universidad de La Coruña, Dep. Electrónica y Sistemas
E.S.Marina Civil, Paseo de Ronda, 51- 15011
Tel: 34. 81. 25 67 00. Fax: 34 81 25 15 68
email:ferreiro @ udc.es

Resumen

En este trabajo se desarrolla el procedimiento que, utilizando datos experimentales difusos, es aplicable a la planificación de proyectos de formación/aprendizaje que tiene como objetivo, minimizar los costes de formación de una competencia profesional predefinida. El algoritmo que desarrollamos decidirá el número de alumnos que son capaces de alcanzar la competencia deseada, con un coste mínimo, en las condiciones de incertidumbre que un proceso de aprendizaje conlleva. El problema de optimización se resuelve mediante el método Hooke & Jeeves, aplicando la información que se extrae de la base de datos difusos creada. Esta información (tiempo óptimo) es usado recursivamente en un proceso PERT para establecer una nueva planificación en función de la variación del entorno problema.

Palabras clave: Información difusa, optimización con incertidumbre, formación/aprendizaje.

1. Introducción

Los proyectos pedagógicos/didácticos curriculares detallan fundamentalmente las capacidades que constituyen la competencia profesional de un título o cualificación, así como el camino que debe seguir el proceso de formación para que los alumnos alcancen el logro del aprendizaje.

Estos proyectos están caracterizados por una serie de inexactitudes e imprecisiones derivadas de los difuso que es en sí mismo el "hecho" de aprender en el ser humano.

Esta peculiaridad del proceso es lo que nos ha estimulado en el desarrollo de un algoritmo de control fuzzy.

1.1. Identificación de las variables que configuran la base de datos

La información de entrada es sencilla, en tanto que el tiempo que se necesita para alcanzar la competencia depende del grado de capacidad deseada y del número de alumnos que forman el grupo.

La información de salida será el tiempo que se necesita para alcanzar la competencia.

La mayor parte de los artilugios que existen en el mundo real pueden ser manejados, dentro de ciertas incertidumbres, utilizando datos "crisp" [5].

Los valores de entrada/salida de la base de datos abarcarán la amplia gama de soluciones que se pueden dar y dependerán de las características dinámicas de cada subfunción.

Todos serán valores reales positivos. Un ejemplo se muestra en la figura 2(a), donde el mismo rango de valores puede estar definido por varias soluciones fuzzy.

1.2. Elección de las funciones

Conseguir el número, amplitud y forma de las funciones que conforman la competencia es algo que se asemeja a un arte. La experiencia y el conocimiento del proceso son la guía que más y mejor puede ayudar a definirlas.

Las capacidades profesionales se miden por sus resultados o logros.

El individuo aprende mediante un mecanismo de "asimilación"; es decir, que toda respuesta nueva está integrada en un esquematismo o en una estructura anterior: entonces hay que considerar la actividad organizadora del sujeto, tan importante como las relaciones inherentes a los estímulos exteriores, porque el sujeto no se hace sensible a éstos, sino en la medida en que son asimilables a las estructuras ya construidas, que modificarán y enriquecerán en función de las nuevas asimilaciones [7].

Por ello consideramos necesarias dos base de datos diferentes para cada capacidad o función definida: conocimientos y aplicaciones.

1.3. Definición de la base de reglas fuzzy

Creadas las subfunciones, entradas y salidas del problema propuesto, el paso siguiente es definir la serie de reglas IF-THEN que determinarán, según las datos de entrada elegidos, los datos base de salida.

Cada base de reglas describe una porción del conocimiento bajo la forma: Si la cantidad de destreza es

A y el grupo formado por B alumnos THEN el tiempo necesario será C.

La figura 1 muestra la variable I/O para la base de reglas de una función.

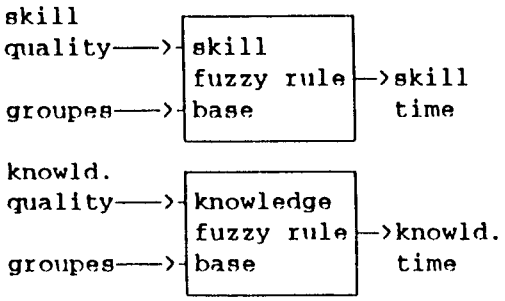


Fig. 1. Rule-base structure

2. Formulación del problema

Describiremos el procedimiento que se sigue para computar el tiempo global que se precisa para completar la tarea de formación, ajustando los tiempos deseados (M.L.T.) de las subfunciones a la competencia deseada en el proyecto. En nuestra propuesta (Ref. fig. 3) obtenemos el tiempo total deseado (M.L.T.) en función del número de alumnos de los grupos y del grado de competencia de las subfunciones (destrezas y conocimientos) con una base de reglas fuzzy tal como se indica en la figura 2.

Esta sección ilustra el procedimiento para completar el estudio preliminar del plan de formación/aprendizaje. Para lo cual debemos seguir los siguientes pasos.

knowledge competence

groupes	L=3	M=6	H=9
L=10	5	7	9
M=20	7	9	11
H=40	9	11	13

(a)

skill competence

groupes	L=4	M=6	H=8
L=5	2	3	4
M=10	3	5	7
H=15	4	7	10

(b)

Fig. 2. Subtask rulbase:(a) knowledge. (b) skill

a) Conseguir la base de datos experimental fuzzy en donde la salida o tiempo que se precisa, es una función de los datos de entrada, el número de alumnos de los grupos y la calificación esperada. La figura 2 muestra la base de datos de una configuración típica de una subfunción

b) Consecución de la función de coste. El coste de la función es un importante asunto a tratar para la solución del problema. Depende del tiempo que se necesita para alcanzar el conocimiento esperado y de las características de las instalaciones, como el tiempo depende de la extensión de los grupos, y de la competencia esperada, tal función es el proceso de defuzzificación por un método analítico (least squares) [1]. Tal subfunción es añadida a la subfunción que describe el costeglobal de la instalación.

$$Ck = (NP/X1).(K1.X1+K2.Y1)+KO.X1$$
$$Cs = (NP/X2).(K3.X2+K4.Y2)+KP.X2$$
$$C = Ck+Cs \qquad (1)$$

en donde NP el número de alumnos, X1 es el tamaño de un grupo de la subfunción "conocimientos", Y1 es la competencia esperada, X2 es el tamaño del grupo de prácticas e Y2 es la respectiva competencia esperada.

La solución se obtiene aplicando el método de optimización a la función coste. El resultado, X1 y X2, son los gruupos óptimos bajo el criterio de coste mínimo del proyecto de formación.

c) Entrando los valores óptimos computados, X1 y X2 y la competencia deseada en la base de reglas se obtiene el tiempo deseado (M.L.T. most likely time).

d) Aplicando el resultado de los tiempos deseados (M.L.T.) en un PERT, se completa el proyecto globalmente.

e) La aplicación de los estudios preliminares permite de una forma recursiva actualzar la regla de bases al repetir la experiencia, y así sucesivamente hasta que no se produzcan, de una forma general, variaciones en la base de reglas.

En la figura 3 se muestra el diagrama de bloque de las secuencias como estrategia de solución del problema.

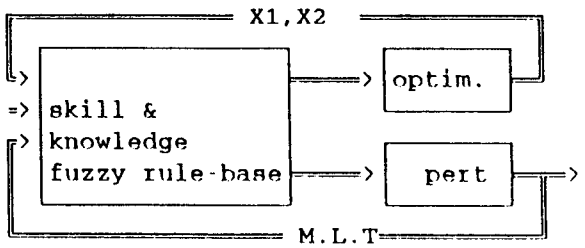


Fig.3. Problem solving task strategy

3. Aplicación de la tarea

Para ilustrar el problema propuesto se muestra una secuencia de la aplicación de la tarea en la cual el objetivo es encontrar el tamaño del grupo así que, la aproximación a un mínimo global bajo incertidumbres.

El tipo de problema relacionado concierne a la formación en seguridad y puede ser repartido en cuatro actividades de las cuales, dos son lecciones teóricas y las otras dos, ejercicios en el laboratorio. Cada par de tareas de formación teórica y práctica puede ser definida experimentalmente en términos de competencia esperada, extensión del grupo y tiempo deseado y necesario para completar la subtarea de formación. Asumiento un tiempo de formación corto en la extinción del fuego, las principales actividades se identifican como siguen:

A1 = Teoría del fuego. Principios de seguridad
A2 = Laboratorio de seguridad contra incendios.
A3 = Teoría sobre los sistemas de lucha contra incendios.
A4 = Ejercicios de extinción.

La figure 4 muestra el diagrama de flujo del juego de actividades en la tarea de formación/aprendizaje

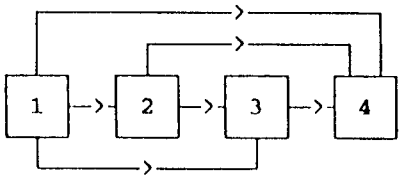


Fig. 4. Flow-chart of the training/learning task

competence Y1

X1	3	6	9
30	21	27	33
60	36	42	48
120	66	72	78

(a)

competence Y2

X2	3	6	9
10	16	22	28
20	26	32	38
30	36	42	48

(b)

Fig. 5. Subtask rulbase. (a) knowledge. (b) skill

En la inicialización del proyecto, las actividades A1 y A2 pueden ser definidas por ambas bases de reglas, entrando con la información experimental. Ambas bases de reglas se muestran en la figura 5.

Aplicando un procedimiento de defuzificación (p.e. el basado en mínimos cuadrados), [1], se consigue una función polinómoica que define cada base de reglas; esto es, .5*X1 + 2*Y1 y 1*X2 + 2*Y2. Siendo K_0 = 0.5 y Kp= 1, la función de coste está descrita a partir de las definiciones básicas dadas en (1) como,

$$Ck = (NP/X1).(.5.X1+2.Y1)+.5.X1$$
$$Cs = (NP/X2).(1.X2+2.Y2)+1.X2$$
$$C = Ck+Cs \qquad (2)$$

La minimización de la función de coste C, es llevada a cabo aplicando el método de búsqueda (Hooke & Jeeves) [4] . El resultado después de la minimización proporciona grupos de 92 y 24 alumnos respectivamente a partir de la primera etapa de optimización y que comprende las actividades A1 y A2. Volviendo de nuevo (recurriendo) con la información obtenida, y bajo la restricción dada por la competencia deseada,(donde se asume el valor 6 para conocimienntos y destrezas), el tiempo habitual (MLT), es el resultado del procedimiento de defuzificación, el cual proporciona los valores de MLTk = 57 y MLTs= 37, el cual ha sido hallado recursivamente al actuar sobre la base de reglas.

Aplicando la misma metodología al resto de la base de reglas, se completa el conjunto del proyecto, para que finalmente sea aplicado al PERT descrito en la figura 4.

4. Conclusiones

La aplicación propuesta permite extraer una conclusión relevante: este método aparece como un procedimiento útil en el intento de lograr el nivel esperado de comètencia profesional y acadeémica, por medio de la metodología clara inteligible y práctica, con resultados dsatisfactorios por el bajo coste de la aplicación.

Agradecimientos: Esta trabajo ha sido subvencionado por Ministerio español de educación y Ciencia bajo en código: TIC92-0267-C02-02 de la C.I.C.Y.T

Referencias

[1] Ruckdeschel, F. R., (1981)
Basic Scientific Subroutines Vol. II. McGraw-Hill Publications Co. New-York 10020. pp 16-45

[2] Practical Basic Problems-APLE II Edition (1981) Osborne/MaGraw-Hill. Berkeley, California 94710. U.S.A. pp 80-85

[3] Training, Enterprise and Education Directorate, (ED.TEED) The Place of Knowledge and understanding in the Development and Delivery of N.V.Q. and S.V.Q., Knowledge Guidance Note-Draft, 17 September 1.992. pp. 35-45.

[4] Brian D. Bunday , (1984)
Basic optimisation methods. Ed Edward Arnold.
Chap 6. pp. 93-97

[5] Bart kosko (1992). Neural Networks and Fuzzy Systems. Prentice Hall Int. Inc. Chap 8, pp 299-335

[6] Ferreiro Garcia. R. Yañez Gonzalez V. (1994). IEEE Catalog number 94CH3571-5. IEEE Systems,Man and Cybernetics Society. Fuzzy rule-based Pert applied on training/learning planning Proceedings of the 1994 IEEE International Conference on Systems, Man and Cybernetics October 2-5, 1994. San Antonio.Texas. U.S.A. Vol 2 of 3. pp. 1815-1821

[7] J. Piaget - Psicología del niño, pag. 17. Ediciones Morata

Dualidad en Programación Lineal Fuzzy

Jose M. Cadenas, Fernando Jiménez
Dpto. de Informática y Sistemas
Facultad de Informática. Universidad de Murcia
30071-Espinardo, Murcia
e-mail: jcadenas@dif.um.es, fernan@dif.um.es

Resumen

En este trabajo se presenta una relación de dualidad fuzzy. Para llegar a definir esta relación de dualidad, se utiliza el teorema de Representación de números fuzzy y algunos resultados sobre Programación Lineal con coeficientes Intervalares. Así, dado un problema de programación lineal con costos fuzzy representados por funciones de pertenencia de tipo $L-R$, puede resolverse por medio de dos problemas duales (problemas de programación lineal con restricciones fuzzy).

Además, estos resultados puede aplicarse a problemas multiobjetivo donde los coeficientes de las distintas funciones objetivo son números fuzzy definidos por funciones de pertenencia lineales.

Palabras clave: Programación Lineal Fuzzy, Programación Lineal Multiobjetivo, Dualidad.

1 Introducción

En la literatura aparecen diferentes puntos de vista para el caso de la teoría de la Dualidad en Programación Lineal Fuzzy (PLF).

Proponemos una relación de dualidad entre problemas de PLF. Daremos una definición de problema dual para un problema de Programación Lineal con costos fuzzy. Esta relación esta basada en la simetría existente entre el concepto de objetivo/restricción fuzzy y en la aritmética intervalar.

Desde este punto de vista, consideramos dos clases de problemas de PLF:

a) Problemas, $\mathbf{P}_1$, para los que se supone que el conjunto de restricciones está nítidamente definido, pero los costos son números fuzzy, es decir, problemas como:

$$\begin{array}{ll} \max & \{\tilde{c}_1 x_1 + \ldots \tilde{c}_n x_n\} \\ s.a: & \\ & Ax \leq b \\ x \geq 0 & \end{array}$$

donde $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$ y para cada costo existe una función de pertenencia, $\mu_{c_j} : \mathbb{R} \to [0,1]$, $j \in N = \{1,\ldots,n\}$, definida de la forma

$$\mu_{c_j}(u) = \begin{cases} 0 & \text{si } R_j \leq u \text{ o } u \leq r_j \\ h_j(u) & \text{si } r_j \leq u \leq \underline{c}_j \\ g_j(u) & \text{si } \bar{c}_j \leq u \leq R_j \\ 1 & \text{si } \underline{c}_j \leq u \leq \bar{c}_j \end{cases} \qquad (1)$$

con $[r_j, R_j]$ el soporte del costo c_j y $h_j(\cdot)$ y $g_j(\cdot)$, $j \in N$ funciones continuas estrictamente crecientes y decrecientes, respectivamente.

b) Problemas, $\mathbf{P}_2$, para los que se supone que las restricciones son fuzzy, es decir, problemas como:

$$\begin{array}{ll} \max & cx \\ s.a: & \\ & Ax \lesssim b \\ x \geq 0 & \end{array}$$

donde $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$ y para cada restricción tenemos una función de pertenencia $\mu_i : \mathbb{R}^n \to [0,1]$, $i \in M = \{1,\ldots,m\}$, que nos da el grado por el cual $x \in \mathbb{R}^n$ satisface la i-ésima restricción fuzzy.

$$\mu_i(x) = \begin{cases} 1 & \text{si } (Ax)_i \leq b_i \\ f_i[(Ax)_i] & \text{si } b_i \leq (Ax)_i \leq b_i + t_i \\ 0 & \text{si } (Ax)_i \geq b_i + t_i \end{cases} \qquad (2)$$

en el que se supone que $f_i(\cdot) \in [0,1]$ es tal que a mayor violación de las restricciones, menor valor de f_i.

Para obtener los resultados y objetivos que nos hemos propuesto en este trabajo, debemos de introducir una serie de definiciones y proposiciones que presentaremos en la siguiente sección (estas definiciones y proposiciones son casos particulares de las obtenidas en [12]).

2 Programación Lineal con Parámetros Intervalares

Sea el problema intervalar siguiente

$$\begin{array}{ll} \max & z = c^{int} x \\ s.a: & \\ & Ax \leq b \\ & x \geq 0 \end{array} \qquad (3)$$

donde $c^{int} = (c^{int,1}, \ldots, c^{int,n})$ es un vector de coeficientes intervalares ($c^{int,j} = [c^j_{\min}, c^j_{\max}]$, $j \in N$).

Definición 1 *Para un problema de Programación Lineal Intervalar (3), se define un problema de PL α-aproximado como*

$$\begin{array}{ll} \max & z = c(\alpha)x \\ s.a: & \\ & Ax \leq b \\ & x \geq 0 \end{array}$$

donde $\alpha \in [0,1]$ *y*

$$c(\alpha) = c_{\max} - \alpha(c_{\max} - c_{\min}) \quad \blacksquare$$

Proposición 1 *Supongamos que el parámetro de verdad del sistema es* $c^* \in [c_{\min}, c_{\max}]$ *y* x^*_{opt} *es la solución optimal del problema*

$$\begin{array}{ll} \max & z = c^*x \\ s.a: & \\ & X = \{Ax \leq b,\ x \geq 0\} \end{array}$$

entonces

$$c_{\min} x^{\min}_{opt} \leq c^* x^*_{opt} \leq c_{max} x^{\max}_{opt} \tag{4}$$

$$c_{\min} x^{\min}_{opt} \leq c(\alpha) x_{opt}(\alpha) \leq c_{\max} x^{\max}_{opt} \tag{5}$$

donde $x^{\min}_{opt}$ *es la solución optimal del problema* $\max_{x \in X} c_{\min} x$ *y* $x^{\max}_{opt}$ *es la solución optimal del problema* $\max_{x \in X} c_{\max} x$.

Demostración

Puesto que $\max_{x \in X} c^*x = c^*x^*_{opt} = \max_{x \in X}(c_{\min}x + (c^* - c_{\min})x)$, $x \geq 0$ y como $c^* - c_{\min} \geq 0$, se sigue que

$$c^* x^*_{opt} \geq \max_{x \in X} c_{min} x = c_{\min} x^{\min}_{opt}$$

De $c_{\max} x^{\max}_{opt} = \max_{x \in X} c_{max} x = \max_{x \in X}(c^*x + (c_{\max} - c^*)x)$, $x \geq 0$, como $c_{\max} - c^* \geq 0$ entonces

$$c_{\max} x^{\max}_{opt} \geq c^* x^*_{opt}$$

Por lo tanto, queda probada la desigualdad (4).

Una demostración similar a la realizada se puede hacer para probar la desigualdad (5). $\blacksquare$

Sea, ahora, el siguiente problema intervalar

$$\begin{array}{ll} \max & z = cx \\ s.a: & \\ & Ax \leq b^{int} \\ & x \geq 0 \end{array} \tag{6}$$

donde $b^{int} = (b^{int,1}, \ldots, b^{int,m})^t$ es un vector de coeficientes intervalares ($b^{int,i} = [b^i_{\min}, b^i_{\max}]$, $i \in M$).

Definición 2 *Para un problema de Programación Lineal Intervalar (6), se define un problema de PL α-aproximado como*

$$\begin{array}{ll} \max & z = cx \\ s.a: & \\ & Ax \leq b(\alpha) \\ & x \geq 0 \end{array}$$

donde $\alpha \in [0,1]$ *y*

$$b(\alpha) = b_{\max} - \alpha(b_{\max} - b_{\min}) \quad \blacksquare$$

Proposición 2 *Sean los conjuntos* $K = \{x \ / \ Ax \leq b,\ x \geq 0\}$, $K_\alpha = \{x \ / \ Ax \leq b(\alpha),\ x \geq 0\}$, $K_{\min} = \{x \ / \ Ax \leq b_{\min},\ x \geq 0\}$, $K_{\max} = \{x \ / \ Ax \leq b_{\max},\ x \geq 0\}$ *donde* $b \in [b_{\min}, b_{\max}]$, *entonces*

$$K_{\min} \subset K \subset K_{\max} \tag{7}$$

$$K_{\min} \subset K_\alpha \subset K_{\max} \tag{8}$$

Demostración

Si $x \in K_{\min}$ entonces $Ax \leq b_{\min} \leq b$ como $x \geq 0$ se sigue que $x \in K$. Si $x \in K$ entonces $Ax \leq b \leq b_{\max}$ y como $x \geq 0$ se sigue que $x \in K_{\max}$. Asi queda probado (7).

Similarmente a este razonamiento se prueba (8). $\blacksquare$

Proposición 3 *Supongamos que el parámetro de verdad del sistema es* $b^* \in [b_{\min}, b_{\max}]$ *y* x^*_{opt} *es la solución optimal del problema*

$$\begin{array}{ll} \max & z = cx \\ s.a: & \\ & Ax \leq b^* \\ & x \geq 0 \end{array}$$

entonces

$$cx^{min}_{opt} \leq cx^*_{opt} \leq cx^{max}_{opt} \tag{9}$$

$$cx^{min}_{opt} \leq cx_{opt}(\alpha) \leq cx^{max}_{opt} \tag{10}$$

donde x^{min}_{opt} *es la solución optimal del problema* $\max_{x \in K_{min}} cx$ *y* x^{max}_{opt} *es la solución optimal del problema* $\max_{x \in K_{max}} cx$.

Demostración

$\max_{x \in K} cx = cx^*_{opt} \geq \max_{x \in K_{min}} cx = cx^{\min}_{opt} \Rightarrow cx^*_{opt} \geq cx^{\min}_{opt}$

$\max_{x \in K_{max}} cx = cx^{\max}_{opt} \geq \max_{x \in K} cx = cx^*_{opt} \Rightarrow cx^{\max}_{opt} \geq cx^*_{opt}$ con lo que queda probado (9).

Haciendo una demostración similar se prueba (10). $\blacksquare$

3 Dualidad en Programación Lineal Fuzzy

En esta sección trataremos de encontrar las condiciones por las cuales un problema de PLF de tipo $\mathbf{P}_1$ puede resolverse por medio de un problema de PLF del tipo $\mathbf{P}_2$.

En PLF los conceptos de objetivo y restricción son simétricos, [2]. Esto hace que pueda construirse una Teoría de Dualidad, [11], en la que el papel que juega el objetivo, puede cambiarse con el de las restricciones, y viceversa.

El siguiente resultado nos proporciona la forma de obtener el dual de un problema de PLF.

Proposición 4 *Dado un problema de PLF del tipo* P_1 *como*

$$\begin{array}{ll} \max & \tilde{c}_1 x_1 + \ldots + \tilde{c}_n x_n \\ s.a: & \\ & Ax \leq b \\ & x \geq \tilde{0} \end{array} \qquad (11)$$

donde los coeficientes de la función objetivo están dados por funciones de pertenencia lineales $\mu_{c_j} : \mathbb{R} \to [0,1]$, $j \in N$ *como*

$$\mu_{c_j}(u) = \begin{cases} 0 & si\ u \leq \underline{c}_j - d_j^1\ o\ u \geq \overline{c}_j + d_j^2 \\ \dfrac{u - (\underline{c}_j - d_j^1)}{d_j^1} & si\ \underline{c}_j - d_j^1 \leq u \leq \underline{c}_j \\ \dfrac{\overline{c}_j + d_j^2 - u}{d_j^2} & si\ \overline{c}_j \leq u \leq \overline{c}_j + d_j^2 \\ 1 & si\ \underline{c}_j \leq u \leq \overline{c}_j \end{cases}$$

existen siempre dos problemas de tipo P_2*, con funciones de pertenencia lineales, que son duales al dado y éstos definen la solución difusa del problema (11), y viceversa.*

Demostración

Sea por tanto el problema de PL con costos fuzzy (11). Si resolvemos utilizando el Teorema de Representación de conjuntos fuzzy, obtenemos el siguiente problema

$$\begin{array}{ll} \max & cx \\ s.a: & \\ & \mu_{c_j}(u) \geq 1 - \alpha,\ j \in N \\ & Ax \leq b,\ x \geq 0,\ \alpha \in [0,1] \end{array}$$

que es equivalente al problema con costos intervalares

$$\begin{array}{ll} \max & [\underline{c} - d^1\alpha, \overline{c} + d^2\alpha]x \\ s.a: & \\ & Ax \leq b \\ & x \geq 0,\ \alpha \in [0,1] \end{array} \qquad (12)$$

donde $\underline{c} = (\underline{c}_1, \ldots, \underline{c}_n)$, $d^1 = (d_1^1, \ldots, d_n^1)$, $\overline{c} = (\overline{c}_1, \ldots, \overline{c}_n)$, $d^2 = (d_1^2, \ldots, d_n^2)$.

Así, para cada uno de los vectores que tomemos de los costos, obtenemos problemas clásicos de programación lineal. Para cada uno de estos problemas podemos obtener su problema dual.

Por tanto, el problema dual del problema (12) es el siguiente,

$$\begin{array}{ll} \min & ub \\ s.a: & \\ & uA \geq [\underline{c} - d^1\alpha, \overline{c} + d^2\alpha] \\ & u \geq 0,\ \alpha \in [0,1] \end{array} \qquad (13)$$

Haciendo uso de la proposición 3 podremos acotar la solución del problema (13), es decir, la solución vendrá restringida por las soluciones de los siguientes problemas auxiliares:

$$\begin{array}{ll} \min & ub \\ s.a: & \\ & uA \geq \underline{c} - d^1\alpha \\ & u \geq 0,\ \alpha \in [0,1] \end{array} \qquad (14)$$

y

$$\begin{array}{ll} \min & ub \\ s.a: & \\ & uA \geq \overline{c} + d^2\alpha \\ & u \geq 0,\ \alpha \in [0,1] \end{array} \qquad (15)$$

A) Del conjunto restricción del problema (14) tenemos que:

$$uA \geq \underline{c} - d^1\alpha \Leftrightarrow uA \geq \underline{c} - d^1 + d^1(1-\alpha)$$

Tomando $\beta = 1 - \alpha$, obtenemos que $uA \geq \underline{c} - d^1 + d^1\beta$ con $\beta \in [0,1]$.

$$uA \geq \underline{c} - d^1 + d^1\beta \Leftrightarrow \frac{(uA - \underline{c} + d^1)}{d^1} \geq \beta \Leftrightarrow$$

$$\Leftrightarrow \mu(uA, \underline{c}) \geq \beta$$

con $\mu = (\mu_1, \ldots, \mu_n)$ tal que $\forall v \in \mathbb{R}$, $j \in N$,

$$\mu_j(v) = \begin{cases} 1 & v \geq \underline{c}_j \\ \dfrac{v - (\underline{c}_j - d_j^1)}{d_j^1} & \underline{c}_j - d_j^1 \leq v \leq \underline{c}_j \\ 0 & \text{otro caso} \end{cases} \qquad (16)$$

e interpretando que las $d_j^1 \in \mathbb{R}$, $j \in N$ expresan las violaciones que se permiten al acoplamiento de las restricciones lineales.

El problema (14) se transforma en el siguiente problema:

$$\begin{array}{ll} \min & ub \\ s.a: & \\ & \mu(uA, \underline{c}) \geq \beta \\ & u \geq 0,\ \beta \in [0,1] \end{array}$$

Es decir,

$$\begin{array}{ll} \min & ub \\ s.a: & \\ & uA \geq \underline{c} \\ & u \geq \tilde{0} \end{array} \qquad (17)$$

donde la vaguedad del conjunto restricción esta dada por un m-vector de funciones de pertenencia $\mu = (\mu_1, \ldots, \mu_n)$ como (16).

NOTA: Si tomamos $\beta = 1 - \alpha$, se obtiene la solución del problema (14). Si $u^*(\beta)$ es la solución de (17), la solución de (14) es $u^*(1-\alpha)$.

B) Del conjunto restricción del problema (15) tenemos que

$$uA \geq \overline{c} + d^2\alpha \Leftrightarrow (uA - \overline{c})/d^2 \geq \alpha \Leftrightarrow$$

$$\Leftrightarrow \mu(uA, \overline{c} + d^2) \geq \alpha$$

con $\mu = (\mu_1, \ldots, \mu_n)$ tal que $\forall v \in \mathbb{R}$, $j \in N$,

$$\mu_j(v) = \begin{cases} 1 & v \geq \overline{c}_j + d_j^2 \\ \dfrac{v - \overline{c}_j}{d_j^2} & \overline{c}_j \leq v \leq \overline{c}_j + d_j^2 \\ 0 & \text{otro caso} \end{cases} \qquad (18)$$

e interpretando que las $d_j^2 \in \mathbb{R}$, $j \in N$ expresan las violaciones que se permite al acoplamiento de las restricciones lineales.

El problema (15) se transforma en el siguiente problema:

$$\begin{array}{ll} \min & ub \\ s.a: & \\ & \mu(uA, \overline{c} + d^2) \geq \alpha \\ & u \geq 0,\ \alpha \in [0,1] \end{array}$$

Es decir

$$\begin{array}{ll} \min & ub \\ s.a: & \\ & uA \geq \overline{c} + d^2 \\ & u \geq \tilde{0} \end{array} \qquad (19)$$

donde la vaguedad del conjunto restricción esta dada por un m-vector de funciones de pertenencia $\mu = (\mu_1, \ldots, \mu_n)$ como (18).

Resueltos los problemas auxiliares (17) y (19), con soluciones optimales paramétricas $u^0(\beta)$ y $u^1(\alpha)$, respectivamente, el valor optimal de la función objetivo de (12) está comprendido en el intervalo:

$$u^0(1-\alpha)b \leq cx^* \leq u^1(\alpha)b,\ \alpha \in [0,1]$$

Con lo cual, la solución del problema (11) es el conjunto fuzzy:

$$u^0(1-\alpha)b \leq cx^* \leq u^1(\alpha)b,\ \forall \alpha \in [0,1]$$

dado por sus α-cortes. ∎

Proposición 5 *Dado un problema de PLF del tipo* $\boldsymbol{P}_1$ *con funciones de pertenencia como las definidas en (1), existen dos problemas de PLF del tipo* $\boldsymbol{P}_2$, *con funciones de pertenencia como las definidas en (2), que son duales al problema del tipo* $\boldsymbol{P}_1$, *proporcionándonos su solución fuzzy, y viceversa.*

Demostración

Sean $\mu_{c_j} : \mathbb{R} \to [0,1]$, $j \in N$ las funciones de pertenencia, para los costos del problema de tipo $\mathbf{P}_1$, definidas como en (1), donde $h_j(\cdot)$ y $g_j(\cdot)$ son funciones continuas estrictamente crecientes y decrecientes, respectivamente.

Realizando los α-cortes en el objetivo, obtenemos el problema siguiente,

$$\begin{array}{ll} \max & [\phi(1-\alpha), \psi(1-\alpha)] \\ s.a: & \\ & Ax \leq b \\ & x \geq 0,\ \alpha \in [0,1] \end{array}$$

donde $\phi(\cdot) = [\phi_1(\cdot), \ldots, \phi_n(\cdot)]$ y $\phi(\cdot) = [\psi_1(\cdot), \ldots, \psi_n(\cdot)]$, $\phi_j \equiv h_j^{-1}$, $\psi_j \equiv g_j^{-1}$, $j \in N$.

La demostración se sigue de un modo similar a la seguida en la proposición (4). ∎

El resultado obtenido, generaliza al obtenido en [11].

En [11] se demostró que dado un problema del tipo (11), donde cada costo esta dado por una función de pertenencia $\mu_{c_j} : \mathbb{R} \to [0,1]$, $j \in N$ del tipo: $\forall v \in \mathbb{R}$

$$\mu_{c_j}(v) = \begin{cases} h_j(v) & \text{si } c_j - d_j \leq v \leq c_j \\ 1 & \text{si } v > c_j \\ 0 & \text{si } v < c_j - d_j \end{cases} \qquad (20)$$

con $h_j(\cdot)$ una función continua creciente, su problema dual es el siguiente

$$\begin{array}{ll} \min & ub \\ s.a: & \\ & uA > c \\ & u \geq \tilde{0} \end{array}$$

donde las restricciones fuzzy están dadas por funciones de pertenencia como en (20).

Si en la proposición 5 se supone que los costos están dados por funciones de pertenencia como en (20), obtenemos los resultados obtenidos en [11].

4 Generalización: Dualidad en Programación Multiobjetivo Fuzzy

Sea el problema de Programación Multiobjetivo Fuzzy

$$\begin{array}{ll} \max & \tilde{C}x \\ s.a: & \\ & Ax \leq b \\ & x \geq \tilde{0} \end{array} \qquad (21)$$

donde $\tilde{C} = \tilde{C}_{q \times n}$ matriz de números fuzzy de tipo trapezoidal, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ y $x \in \mathbb{R}^n$.

Si resolvemos utilizando el Teorema de Representación de conjuntos fuzzy, obtenemos el siguiente problema multiobjetivo con costos intervalares

$$\begin{array}{ll} \max & [\underline{C} - D^1\alpha, \overline{C} + D^2\alpha]x \\ s.a: & \\ & Ax \leq b \\ & x \geq 0,\ \alpha \in [0,1] \end{array} \qquad (22)$$

donde $\underline{C} \in \mathbb{R}^{q \times n}$, $D^1 \in \mathbb{R}^{q \times n}$, $\overline{C} \in \mathbb{R}^{q \times n}$, $D^2 \in \mathbb{R}^{q \times n}$.

El problema dual del problema (22) es el siguiente, [13],

$$\begin{array}{ll} \min & ub \\ s.a: & \\ & uA \geq [\underline{C} - D^1\alpha, \overline{C} + D^2\alpha] \\ & u \geq 0,\ \alpha \in [0,1] \end{array}$$

Haciendo uso de la proposición 3, (generalizada al caso multiobjetivo), podremos acotar la solución del problema (21), obteniendo para ello los siguientes problemas:

$$\begin{array}{ll} \min & ub \\ s.a: & \\ & uA \geq \underline{C} - D^1\alpha \\ & u \geq 0,\ \alpha \in [0,1] \end{array} \qquad (23)$$

y

$$\begin{array}{ll} \min & ub \\ s.a: & \\ & uA \geq \overline{C} + D^2\alpha \\ & u \geq 0,\ \alpha \in [0,1] \end{array} \qquad (24)$$

Es decir,

$$\begin{array}{ll} \min & ub \\ s.a: & \\ & uA \geq \underline{C} \\ & u \geq \tilde{0} \end{array}$$

y

$$\begin{array}{ll} \min & ub \\ s.a: & \\ & uA \geq \overline{C} + D^2 \\ & u \geq \tilde{0} \end{array}$$

donde las restricciones fuzzy están representadas por funciones de pertenencia como (16) y (18), respectivamente.

NOTA: Los problemas paramétricos (23) y (24) pueden ser resueltos por los métodos simplex MC y MC^2, [13].

5 Ejemplo

Consideremos el problema siguiente,

$$\begin{array}{ll} \max & z = \tilde{c}_1 x_1 + \tilde{c}_2 x_2 \\ s.a: & \\ & x_1 + 3x_2 \leq 6 \\ & 2x_1 + 2x_2 \leq 8 \\ & x_j \geq 0,\ j = 1,2 \end{array}$$

donde los costos son números fuzzy con funciones de pertenencia dadas por:

$$\mu_{c_1}(u) = \begin{cases} 0 & si\ u \leq 0\ o\ u \geq 3 \\ \dfrac{u^2}{4} & si\ 0 \leq u \leq 2 \\ 3 - u & si\ 2 \leq u \leq 3 \end{cases}$$

$$\mu_{c_2}(u) = \begin{cases} 0 & si\ u \leq 1\ o\ u \geq 5 \\ \dfrac{(u-1)^2}{4} & si\ 1 \leq u \leq 3 \\ 1 & si\ 3 \leq u \leq 4 \\ (5-u)^2 & si\ 4 \leq u \leq 5 \end{cases}$$

Obtenemos por la proposición 5 los siguientes problemas duales.

Problema Dual 1:

$$\begin{array}{ll} \min & t_1 = 6u_1 + 8u_2 \\ s.a: & \\ & u_1 + 2u_2 \geq 2 \\ & 3u_1 + 2u_2 \tilde{\geq} 3 \\ & u_i \geq 0,\ i \tilde{=} 1,2 \end{array}$$

donde las restricciones son fuzzy y vienen dadas por las siguientes funciones de pertenencia,

$$\mu_{11}((uA)_1, 2) = \begin{cases} 0 & , (uA)_1 \leq 0 \\ \dfrac{((uA)_1)^2}{4} & , 0 \leq (uA)_1 \leq 2 \\ 1 & , (uA)_1 \geq 2 \end{cases}$$

$$\mu_{12}((uA)_2, 3) = \begin{cases} 0 & , (uA)_2 \leq 1 \\ \dfrac{((uA)_2 - 1)^2}{4} & , 1 \leq (uA)_2 \leq 3 \\ 1 & , (uA)_2 \geq 3 \end{cases}$$

La solución de este problema es:

$$(\frac{1}{3}, 0) \Rightarrow t_1^* = 2\ ,\ \beta \in [0, \frac{1}{16}]$$

$$(\frac{1}{2}, -\frac{1}{4} + \sqrt{\beta}) \Rightarrow t_1^* = 1 + 8\sqrt{\beta}\ ,\ \beta \in [\frac{1}{16}, 1]$$

Problema Dual 2:

$$\begin{array}{ll} \min & t_2 = 6u_1 + 8u_2 \\ s.a: & \\ & u_1 + 2u_2 \geq 3 \\ & 3u_1 + 2u_2 \tilde{\geq} 5 \\ & u_i \geq 0,\ i \tilde{=} 1,2 \end{array}$$

donde las restricciones son fuzzy y vienen dadas por las siguientes funciones de pertenencia,

$$\mu_{21}((uA)_1, 3) = \begin{cases} 0 & , (uA)_1 \leq 2 \\ (uA)_1 - 2 & , 2 \leq (uA)_1 \leq 3 \\ 1 & , (uA)_1 \geq 3 \end{cases}$$

$$\mu_{22}((uA)_2, 5) = \begin{cases} 0 & , (uA)_2 \leq 4 \\ 1 - (5 - (uA)_2)^2 & , 4 \leq (uA)_2 \leq 5 \\ 1 & , (uA)_2 \geq 5 \end{cases}$$

La solución de este problema es:

$$\begin{array}{l} (\dfrac{3}{2} - \dfrac{1}{2}\alpha - \dfrac{1}{2}\sqrt{1-\alpha}, \dfrac{1}{4} + \dfrac{3}{4}\alpha + \dfrac{1}{4}\sqrt{1-\alpha}) \\ \Rightarrow t_2^* = 11 + 3\alpha - \sqrt{1-\alpha}\ ,\ \alpha \in [0,1] \end{array}$$

La solución del problema propuesto es el conjunto fuzzy definido por los α-cortes (tomando la solución del problema Dual 1, para $\beta = 1 - \alpha$) .

$$1 + 8\sqrt{1-\alpha} \leq z^* \leq 11 + 3\alpha - \sqrt{1-\alpha}\quad ,\ \alpha \in [0, \tfrac{15}{16}]$$

$$2 \leq z^* \leq 11 + 3\alpha - \sqrt{1-\alpha}\quad ,\ \alpha \in [\tfrac{15}{16}, 1]$$

6 Conclusiones

Hemos propuesto una relación de dualidad entre problemas de PLF (dualidad fuzzy). Esta relación está basada en la simetría existente entre el concepto de objetivo/restricción fuzzy y en la aritmética intervalar. Se ha definido como el dual de un objetivo fuzzy, un conjunto de restricciones fuzzy y viceversa, obteniéndose como problema dual de un problema fuzzy de tipo $\mathbf{P}_1$ dos problemas fuzzy de tipo $\mathbf{P}_2$, y viceversa.

Un aspecto importante de la relación de dualidad, que ha sido mostrada en este trabajo, es la facilidad para resolver problemas con costos fuzzy dados por números fuzzy de tipo $L-R$ por medio de problemas de restricciones fuzzy, estando éstos perfectamente resueltos por la programación paramétrica.

Además, estos mismos resultados son más importantes cuando se aplican a problemas multiobjetivo con costos fuzzy dados por números fuzzy con funciones de pertenencia lineales. La solución de este problema multiobjetivo se calcula con ayuda de 2 problemas duales. Estos problemas son uniobjetivo con restricciones fuzzy donde los términos de la derecha son matrices de números reales. Utilizamos como herramienta de resolución el método simplex MC.

Referencias

[1] **G. Alefeld, J. Herzberger**, Introduction to interval computations (Translated by J. Rokne). *Academic Press (1983). New York.*

[2] **R.E. Bellman, L.A. Zadeh**, Decisión Making in a Fuzzy Environment. *Management Science 17 (B) 4 (1970), 141-164.*

[3] **G.R. Bitran**, Linear multiple objective Problems with Interval Coefficients. *Man. Sci., 26 (1985), 7.*

[4] **J.M. Cadenas**, Diseño e implementación de un sistema interactivo para resolver problemas de optimización fuzzy, *Tesis Doctoral, Universidad de Granada, 1993.*

[5] **M. Delgado, J.L. Verdegay, M.A. Vila**, Imprecise costs in mathematical programming problems. *Control and Cybernet 16 (2) (1987), 113-121.*

[6] **H. Hanacher, H. Leberling and H. -J. Zimmermann**, Sensitivity analysis in fuzzy Linear Programming. *Fuzzy Sets and Systems 1 (1978), 269-281.*

[7] **J. Llena**, On Fuzzy Linear Programming, *European Journal of Operational Research 32 (1985), 216-223.*

[8] **S.A. Orlovski**, On Programming with Fuzzy Constraint Sets. *Kybernetes 6 (1977), 197-201.*

[9] **W. Rodder and H. -J. Zimmermann**, Duality in fuzzy programming, *Internat. Symp. on Extremal Methods and Systems Analysis, University of Texas Austin, TX (1977).*

[10] **J.L. Verdegay**, Fuzzy Mathematical Programming. *In Fuzzy Information and Decisión Processes. M.M. Gupta and E. Sanchez. (eds.) (1982), 231-237.*

[11] **J.L Verdegay**, A dual approach to solve the fuzzy lineal programming problem. *Fuzzy sets and systems 14 (1984), 131-141.*

[12] **Y. Yang**, A new approach to uncertain parameter linear programming. *European Journal of Operational Research 54 (1991), 95-114.*

[13] **P.L. Yu**, Multiple-criteria decision making. *Mathematical concepts and methods in science and engineering. Vol. 30, Plenum Press (1985), New York.*

[14] **H.J. Zimmerman**, Optimization in fuzzy environments. *Presentado al XXI Int. TIMS and 46th ORSA Conference, San Juan, Puerto Rico.*

Resolución de un programa lineal difuso mediante la teoría de posibilidad.

M.MAR ARENAS
AMELIA BILBAO
M.VICTORIA RODRIGUEZ
Departamento de Matemáticas
Universidad de Oviedo*

MARIANO JIMENEZ
Departamento de Econ. Aplicada I
Universidad del País Vasco†

Abstract

In conventional decision making problems, the estimation of the parameters of the model is often a problematic task. Normally they are either given by the decision maker (DM) who has imprecise information and/or expresses his considerations subjectively, or by statistical inference from the past data and their stability is doubtful. Therefore, it is reasonable to construct a model reflecting imprecise data or ambiguity in terms of fuzzy sets and a lot of fuzzy approaches to linear programming have been developed.

This paper presents a fuzzy solution to a LP problem when all the parameters may be fuzzy numbers. The method relies on α-cuts of the fuzzy solution to generate the possibility distribution of the objective function. Ideas are illustrated with a numerical example.

Keywords: *Linear Programming, decision making, fuzzy programming, possibility distribution, fuzzy number.*

1 Introducción.

La lógica difusa proporciona instrumentos que permiten comparar magnitudes de cuantía incierta. En este trabajo, construiremos la distribución de posibilidad del óptimo del problema de programación lineal (PL) en términos de sus α-cortes. Utilizaremos una relación de preferencia difusa [7] que nos permitirá estimar, de modo sencillo, en qué grado se alcanza el óptimo deseado y en qué medida se cumplen las restriciones del PL objeto de estudio.

Entre los enfoques existentes para el tratamiento de la incertidumbre no probabilística en un PL hemos elegido, en nuestro trabajo, aquel que considera que los parámetros del problema están representados por números difusos. Se puede admitir que esta hipótesis de trabajo está próxima a aquellas que consideran que "los parámetros imprecisos están dentro de intervalos especificados" (problema que Yang [10] denomina GLP). Sin embargo, respecto al trabajo de Yang [10], la información de la que parte (intervalo de variación del parámetro) es menor que la usada en nuestro trabajo pues, además de saber que el parámetro "varía" en un intervalo, conocemos su distribución de posibilidad.

Esta diferencia en la información –a nuestro favor– nos conduce a objetivos diferentes de los del profesor Yang. El ha demostrado que el óptimo de su problema GLP es un parámetro "grey" y, según su definición 1, en su modelo la distribución del óptimo no puede caracterizarse; justamente, encontrar la distribución de posibilidad del óptimo es una de nuestras metas.

Las analogías que existen entre ambos trabajos –por otra parte inevitables– son debidas a la necesidad de tratar un problema con parámetros "variando" -en nuestro caso- y "estando" -en el de Yang- en intervalos. Por tanto, las caracterícticas de monotonía respecto de los extremos inferior y superior de los intervalos son exactamente las mismas. Hemos mantenido nuestra propuesta porque es más general que la de Yang: los lemas 1 y 2 de su trabajo son corolarios de nuestro teorema 4.

2 Determinación de la distribución de posibilidad del óptimo de un programa lineal difuso.

Cuando se plantea un programa lineal en ambiente de incertidumbre no probabilística y el decisor ha estimado los parámetros que lo definen se puede expresar el problema en términos difusos:

$$\left.\begin{array}{ll} \text{Optimizar} & \tilde{z} = \tilde{c}x \\ \text{sujeto a} & \\ & \tilde{A}x * \tilde{b} \\ & x \geq 0 \end{array}\right\} (PLF)$$

donde $x = [x_j]$ es un vector de decisión $n \times 1$ no difuso, $\tilde{c} = [\tilde{c}_j]$ es un vector $1 \times n$, $\tilde{A} = [\tilde{a}_{ij}]$ es la matriz tecnológica de dimensión $m \times n$ y $\tilde{b} = [\tilde{b}_i]$ es un vector $m \times 1$, estos últimos formados por números difusos.

La solución del problema será también difusa y puede ser caracterizada mediante la distribución de posibilidad del óptimo del mismo.

Representaremos por $\tilde{z}^*$ al conjunto difuso de los valores óptimos del PLF, de acuerdo con el principio de extensión y siguiendo a Buckley [2] la posibilidad de que la solución del PLF sea un determinado valor

*e-mail:VRODRI@hp&45.ECONO.UNIVI.ES
†e-mail:eupjilom@se.ehu.es

no difuso z^* es:

$$\Pi(\tilde{z}^* = z^*) = \sup_{A,b,c} \{\Pi(A,b,c) \text{ tal que } z \text{ es el óptimo del PL(A,b,c)}\} \quad (1)$$

donde la distribución de posibilidad conjunta $\Pi(A,b,c)$ es el mínimo de las funciones de pertenencia de los parámetros difusos del problema y PL(A,b,c) hace referencia al problema no difuso siguiente:

$$\left.\begin{array}{ll} \text{Optimizar} & z = cx \\ \text{sujeto a} & \\ & Ax * b \\ & x \geq 0 \end{array}\right\} (PL(A,b,c))$$

A lo largo del trabajo se tratará de definir $\tilde{z}^*$ mediante sus α-cortes, $z^*_\alpha = [z^{*L}_\alpha, z^{*R}_\alpha]$ con $\alpha \in [0,1]$, siendo el superíndice L indicativo del extremo inferior del intervalo y el superíndice R del extremo superior[1].

2.1 Definición 1

$$\Omega(\alpha) = \{ z \,/\, z \text{ es solución del PL(A,b,c) con } A \in A_\alpha,\ b \in b_\alpha,\ c \in c_\alpha\}$$

donde A_α, b_α, c_α son los α-cortes de $\tilde{A}$, $\tilde{b}$, $\tilde{c}$.

2.2 Definición 2

Definimos la función

$$f(A,b,c) = z^*$$

como aquella que a cada terna de parámetros (A,b,c) le hace corresponder el óptimo del problema PL(A,b,c).

A partir de esta definición se tiene:

$$\Omega(\alpha) = f(A_\alpha \times b_\alpha \times c_\alpha).$$

2.3 Teorema 1

f es continua en su dominio, $\mathcal{D}$.

2.3.1 Demostración

Sea (A_0, b_0, c_0) un elemento cualquiera del dominio $\mathcal{D}$ y sea $z_0 = f(A_0, b_0, c_0)$. Vamos a probar que

$$\forall \epsilon > 0\ \exists \delta > 0 \,/\, z = f(A,b,c) \in B(z_0,\epsilon)$$
$$\forall (A,b,c) \in B((A_0,b_0,c_0),\delta) \quad (2)$$

Como $z_0 = f(A_0,b_0,c_0)$ entonces $\exists x_0 \geq 0$ básico tal que $A_0 x_0 \leq b_0$, $A_{0_B} x_{0_B} = b_0$ y $z_0 = c_0 x_0$ y como $z = f(A,b,c)$ entonces $\exists x \geq 0$ básico que mantiene las mismas variables básicas que x_0 tal que $Ax \leq b$, $A_B x_B = by$ $z = cx$.

[1] Esta notación sería habitual a lo largo de este trabajo.

Por tanto:

$$\| z - z_0 \| = \| cx - c_0 x + c_0 x - c_0 x_0 \| \leq \leq \| c - c_0 \| \, \| x \| + \| c_0 \| \, \| x - x_0 \| \quad (3)$$

Como $x_B = A_B^{-1} b$ y $A_{0_B} x_{0_B} = b_0$ entonces

$$\begin{aligned} x_B &= A_B^{-1} b = A_B^{-1}(b_0 + b - b_0) = \\ &= A_B^{-1} b_0 + A_B^{-1}(b - b_0) = \\ &= A_B^{-1}(A_{0_B} x_{0_B}) + A_B^{-1}(b - b_0) = \\ &= A_B^{-1}(A_B + A_{0_B} - A_B) x_{0_B} + A_B^{-1}(b - b_0) = \\ &= x_{0_B} + A_B^{-1}(A_{0_B} - A_B) x_0 + A_B^{-1}(b - b_0) \end{aligned}$$

y por tanto

$$x_B - x_{0_B} = A_B^{-1}(A_{0_B} - A_B) x_{0_B} + A_B^{-1}(b - b_0)$$

Si $A \in B(A_0, \delta')$, es decir, $\| A - A_0 \| < \delta'$, utilizando la norma que da convergencia elemento a elemento entonces $\| A_{0_B} - A_B \| < \delta'$,

Si $b \in B(b_0, \delta'')$, es decir, $\| b - b_0 \| < \delta''$, y

Si $c \in B(c_0, \delta''')$, es decir, $\| c - c_0 \| < \delta'''$.

Tomando:

$$\delta' < \frac{\epsilon}{4\|A_B^{-1}\| \, \|x_0\| \, \|c_0\|}$$
$$\delta'' < \frac{\epsilon}{4\|A_B^{-1}\| \, \|c_0\|}$$
$$\delta''' < \frac{\epsilon}{2\|x\|}$$

y $\delta = min(\delta', \delta'', \delta''')$ de (3) obtendremos finalmente:

$$\|z - z_0\| < \frac{\epsilon}{2} + \|c_0\| \left[\frac{\epsilon}{4\|c_0\|} + \frac{\epsilon}{4\|c_0\|}\right] = \epsilon \qquad \square$$

2.4 Teorema 2

El conjunto $\Omega(\alpha)$ es un intervalo cerrado.

Este resultado se obtiene teniendo en cuenta que $\Omega(\alpha)$ es la imagen de un conexo por una función continua.

2.5 Teorema 3

$$\Omega(\alpha) = z^*_\alpha$$

2.5.1 Demostración

(i) La inclusión $\Omega(\alpha) \subset z_\alpha$ es evidente.

(ii) Sea $z \in z^*_\alpha$ entonces $\mu_{\tilde{z}^*}(z) \geq \alpha$ de aquí:

$$sup_{A,b,c} \{\Pi(A,b,c)/z = f(A,b,c)\} = \alpha^* \geq \alpha \quad (4)$$

Si denotamos por

$$\Upsilon_z = \{\Pi(A,b,c)/z = f(A,b,c)\}$$

por definición de supremo existe una sucesión

$$(\alpha_n) \subset \Upsilon_z \quad \text{convergente a} \quad \alpha^*$$

Puesto que $\alpha_n \in \Upsilon_z$ existen A_n, b_n, c_n tal que $z = f(A_n, b_n, c_n)$ y $\Pi(A_n, b_n, c_n) = \alpha_n$ entonces $A_n \in A_{\alpha_n}, b_n \in b_{\alpha_n}, c_n \in c_{\alpha_n}$, luego $z \in \Omega(\alpha_n)$.

Tenemos entonces

$$(A_n) \subset A_{\alpha=0}, \quad (b_n) \subset b_{\alpha=0}, \quad (c_n) \subset c_{\alpha=0}$$

como $A_{\alpha=0}, b_{\alpha=0}, c_{\alpha=0}$ son conjuntos compactos existen subsucesiones convergentes:

$$(A_{n_k}) \longrightarrow A^*, (b_{n_k}) \longrightarrow b^*, (c_{n_k}) \longrightarrow c^*$$

y como

$$\lim_{n\to\infty} \alpha_n = \alpha^*$$

entonces $A^* \in A_{\alpha^*}, b^* \in b_{\alpha^*}, c^* \in c_{\alpha^*}$. Por último, aplicando que f es continua tenemos que:

$$z = f(A_{n_k}, b_{n_k}, c_{n_k}) \longrightarrow f(A^*, b^*, c^*) = z$$

luego $z \in \Omega(\alpha^*)$ y como $\alpha \leq \alpha^*$ entonces

$$\Omega(\alpha^*) \subset \Omega(\alpha)$$

por lo que concluimos que $z \in \Omega(\alpha)$ □.

2.6 Teorema 4

Sean $z = f(A, b, c)$ y $z' = f(A', b', c')$ donde[2] $c < c'$, $b < b'$ y $A' < A$ entonces:

$$z = f(A, b, c) < z' = f(A', b', c')$$

2.6.1 Demostración

En primer lugar vamos a demostrar que si $\mathcal{X}_{(A,b)}$ y $\mathcal{X}_{(A',b')}$ son los conjuntos factibles de los problemas PL(A,b,c) y PL(A',b',c'),respectivamente, entonces $\mathcal{X}_{(A,b)} \subset \mathcal{X}_{(A',b')}$.

Si $x \in \mathcal{X}_{(A,b)}$ entonces $Ax \leq b, x \geq 0$ entonces:

$$A'x < Ax \leq b < b'$$

luego $x \in \mathcal{X}_{(A',b')}$.

Sean

$$x \in \mathcal{X}_{(A,b)} \text{ tal que } z = cx \quad \text{y}$$

$$x' \in \mathcal{X}_{(A',b')} \text{ tal que } z' = c'x'.$$

Como $x \in \mathcal{X}_{(A',b')}$ y $c < c'$ entonces

$$z = cx < c'x \leq c'x' = z' \qquad \square$$

2.7 Proposición 1

A partir del teorema anterior:

$$z_\alpha^{*L} = f(A_\alpha^R, b_\alpha^L, c_\alpha^L) \qquad z_\alpha^{*R} = f(A_\alpha^L, b_\alpha^R, c_\alpha^R)$$

[2]Consideramos ordenes estrictos.

2.8 Proposición 2

Por el teorema 4 y las propiedades de los α-cortes, tenemos que si $\alpha_1 < \alpha_2$ entonces:

$$(i) \quad z_{\alpha_1}^{*L} < z_{\alpha_2}^{*L}$$

$$(ii) \quad z_{\alpha_2}^{*R} < z_{\alpha_1}^{*R}$$

Teniendo en cuenta esta proposición los α-cortes de la distribución de posibilidad de z^* son intervalos cerrados encajados. Por tanto podemos concluir:

2.9 Teorema 5

z^* es un número difuso.

2.10 Ejemplo numérico.

Consideremos el siguiente programa lineal difuso:

$$\max \tilde{z} = L(24,2)x_1 + L(8,1)x_2$$

sujeto a

$$\left.\begin{array}{l} L(2,0.1)x_1 + L(5,0.2)x_2 \leq L(10,2) \\ L(4,0.2)x_1 + L(1,0.1)x_2 \leq L(20,1) \\ x_j \geq 0,\ j = 1,2 \end{array}\right\}$$

Donde todas las distribuciones de posibilidad de los parámetros vienen representadas por números difusos triangulares simétricos.

El extremo inferior del α-corte, z_α^{*L}, se determina resolviendo el programa lineal correspondiente a la previsión más desfavorable para cada $\alpha \in [0,1]$:

$$\max (22 + 2\alpha)x_1 + (7 + \alpha)x_2$$

sujeto a

$$\left.\begin{array}{l} (2.1 - 0.1\alpha)x_1 + (5.2 - 0.2\alpha)x_2 \leq 38 + 2\alpha \\ (4.2 - 0.2\alpha)x_1 + (1.1 - 0.1\alpha)x_2 \leq 19 + \alpha \\ x_j \geq 0,\ j = 1,2 \end{array}\right\}$$

Y el extremo superior, z_α^{*R}:

$$\max (26 - 2\alpha)x_1 + (9 - \alpha)x_2$$

sujeto a

$$\left.\begin{array}{l} (1.9 + 0.1\alpha)x_1 + (4.8 + 0.2\alpha)x_2 \leq 42 - 2\alpha \\ (3.8 + 0.2\alpha)x_1 + (0.9 + 0.1\alpha)x_2 \leq 21 - \alpha \\ x_j \geq 0,\ j = 1,2 \end{array}\right\}$$

La tabla 1 presenta las soluciones obtenidas resolviendo cada problema para los niveles de posibilidad 0, 0.2, 0.4, 0.6, 0.8 y 1:

α	z_α^{*L}	z_α^{*R}
0	107.11	164.26
0.2	112.01	160.93
0.4	117.08	151.27
0.6	122.32	145.09
0.8	127.73	139.11
1	133.33	133.33

Tabla 1: Soluciones.

3 Elección del valor a asignar a las variables de decisión.

Una vez obtenida la distribución de posibilidad de la solución difusa de la función objetivo, $\tilde{z}^*$, el decisor deberá optar por el z^* más satisfactorio teniendo en cuenta su aversión al riesgo. Se trata de obtener los valores $x_1, \ldots x_n$ de forma que el número difuso $\tilde{c}x$ sea mejor que $\tilde{z}$, sopesado esto con el riesgo de que se violen las restricciones.

Obsérvese que hemos de trabajar con una desigualdad entre números difusos por lo que es necesario elegir un criterio de comparación de los mismos. El método usado aquí está basado en una relación de preferencia difusa $\mu(\tilde{A}, \tilde{B})$ [7], que es una simplificación del método de Yuan [11].

El valor esperado de un número difuso $\tilde{A}$ se obtiene a partir de la siguiente expresión:

$$EI(\tilde{A}) = [E_1^A, E_2^A] = \left[\int_0^1 A_\alpha^L \, d\alpha, \int_0^1 A_\alpha^R \, d\alpha\right]$$

Teniendo en cuenta esto si $\tilde{A} = L(a, \sigma)$ es un número difuso triangular simétrico:

$$EI(\tilde{A}) = [E_1^A, E_2^A] = \left[a - \frac{1}{2}\sigma, a + \frac{1}{2}\sigma\right]$$

El grado en el que un número difuso $\tilde{A}$ es mayor que $\tilde{B}$ viene dado por:

$$\mu(\tilde{A}, \tilde{B}) = \begin{cases} 0 & E_2^A - E_1^B < 0 \\ \dfrac{E_2^A - E_1^B}{E_2^A - E_1^A + E_2^B - E_1^B}; & 0 \in [E_1^A - E_2^B, E_2^A - E_1^B] \\ 1 & E_1^A - E_2^B > 0 \end{cases}$$

Sea $\mu(\tilde{A}, \tilde{B}) = \lambda$, si $\lambda > \frac{1}{2}$ entonces $\tilde{A}$ es preferido a $\tilde{B}$ con grado λ, si $\lambda < \frac{1}{2}$ $\tilde{B}$ es preferido a $\tilde{A}$ con grado $1 - \lambda$ y si $\lambda = \frac{1}{2}$ $\tilde{A}$ y $\tilde{B}$ son indiferentes[11].

De acuerdo con lo anterior y para obtener x^* el decisor deberá especificar el grado de aversión al riesgo de que se viole cada restricción h_i, $i = 1, \ldots, m$ (a mayor valor de h_i mayor aversión). El PLF se transformará en:

$$\max \mu(\tilde{c}x, \tilde{z}^*)$$

sujeto a

$$\mu\left(\tilde{b}_i, \tilde{a}_i x\right) \geq h_i, \; i = 1, \ldots, m$$
$$x \geq 0$$

Hemos de destacar que el problema resultante podría ser no lineal.

En nuestro ejemplo, supongamos que el decisor está dispuesto a asumir un pequeño riesgo en que se viole la primera restricción, $h_1 = 0.8$, mientras que no tolera ningún riesgo de que se viole la segunda, $h_2 = 1$. De acuerdo con lo anterior el problema que tendremos que resolver será:

$$\max \mu(L(24,2)x_1 + L(8,1)x_2, \tilde{z}^*)$$

sujeto a

$$\left.\begin{array}{l} \mu(L(40,2), L(2,0.1)x_1 + L(5,0.2)x_2) \geq 0.8 \\ \mu(L(20,1), L(4,0.2)x_1 + L(1,0.1)x_2) \geq 1 \\ x_j \geq 0, \; j = 1,2 \end{array}\right\}$$

donde $\tilde{z}^*$ representa la distribución de posibilidad de la solución óptima del PLF. Teniendo en cuenta los resultados obtenidos en la primera parte del problema parece pausible aproximar $\tilde{z}^*$ a un número difuso triangular, en cuyo caso su intervalo esperado resulta ser $EI(\tilde{z}^*) = [120, 149]$. Con estos datos resolvemos el correspondiente problema[3] obteniendo los siguientes resultados:

$$x_1^* = 3.146$$

$$x_2^* = 6.524$$

$$\mu^* = \mu(\tilde{c}x, \tilde{z}^*) = 0.49$$

El valor de μ^* obtenido nos indica que $\tilde{c}x$ es algo más pequeño que la distribución de posibilidad $\tilde{z}^*$ que habíamos obtenido. A la vista de estos resultados podríamos preguntarnos si es posible aumentar el valor de μ^* asumiendo un mayor riesgo en la violación de las restricciones.

Para contestar a esta pregunta puede ser de gran ayuda hacer un análisis de sensibilidad respecto de los riesgos que está dispuesto a asumir el decisor, es decir, respecto de h_1 y h_2. En este caso dicho análisis nos proporciona la siguiente información aproximada: por cada décima en que disminuyamos h_1, μ^* aumenta en 0.004, mientras que por cada décima en que disminuyamos h_2, μ^* aumenta en 0.025. Por lo tanto si el decisor quiere mejorar los resultados es más conveniente que lo haga a costa de asumir un mayor riesgo respecto de la segunda restricción. Por ejemplo, si

[3] El problema resultante es lineal fraccional cuya resolución no presenta ninguna dificultad ya que puede resolverse con el software presente en el mercado, en nuestro caso hemos utilizado el programa GINO.

$h_1 = h_2 = 0.8$ planteado el correspondiente problema obtendríamos la siguiente solución:

$$x_1^* = 3.243$$

$$x_2^* = 6.485$$

$$\mu^* = \mu(\tilde{c}x, \tilde{z}^*) = 0.54$$

4 Conclusiones.

En este trabajo, partiendo de las ideas de Buckley[3] hemos estudiado como la distribución de posibilidad de los parámetros difusos puede obtenerse la distribución de posibilidad $\tilde{z}^*$ de la solución difusa de la función objetivo de un PLF.

Hemos contemplado este $\tilde{z}^*$ como un objetivo difuso y utilizando un criterio de comparación entre números difusos desarrollado en anteriores trabajos por uno de los autores, hemos determinado en qué grado se alcanza dicho objetivo, dependiendo del grado de aversión del decisor al riesgo de que se violen las restricciones. Como posible estudio para el futuro planteamos el análisis del caso en que las restricciones puedan ser violadas[5].

Referencias

[1] Bellman, R. E.; Zadeh, L.A. (1970): "Decision-making in a fuzzy environment", *Management Science 17*, B141-B164.

[2] Buckley, J.J. (1988): "Possibilistic linear programming with triangular fuzzy numbers", *Fuzzy Sets and Systems 26*, 135-138.

[3] Buckley, J.J. (1989): "Solving possibilistic linear programming problems", *Fuzzy Sets and Systems 31*, 329-341.

[4] Buckley, J.J.; Qu, Y. (1990): "On using α-cuts to evaluate fuzzy equations", *Fuzzy Sets and Systems 38*, 309-312.

[5] Delgado, M.; Verdegay, J.L.; Vila, M.A. (1989): "A general model for fuzzy linear programming", *Fuzzy Sets and Systems 29*, 21-29.

[6] Heilpern, S. (1992): "The expected value of a fuzzy number", *Fuzzy Sets and Systems 47*, 81-86.

[7] Jiménez, M. (1994): "Ordenación de números borrosos mediante la comparación de sus intervalos esperados", *Actas del IV congreso de la Asociación Española sobre Tecnologías y Lógica Fuzzy*, Blanes, septiembre.

[8] Julien, B. (1994): "An extension to possibilistic linear programming", *Fuzzy sets ans Systems 64*, 195-206.

[9] Tanaka, H.; Asai, K. (1984): "Fuzzy linear programming problems with fuzzy numbers", *Fuzzy Sets and Systems 13*, 1-10.

[10] Yang, Y. (1991): "An new approach to uncertain parameter linear programming", *European Journal Of Operational Research 54*, 95-114.

[11] Yuan, Y. (1991): "Criteria for evaluatong fuzzy ranking methods", *Fuzzy Sets and Systems 44*, 139-157.

[12] Zadeh, L.A. (1965): "Fuzzy sets". *Information and Control, 8*, 338-353.

Teoría

La desigualdad como medida de variación de una variable aleatoria difusa

Norberto Corral, María Angeles Gil y Hortensia López
Departamento de Matemáticas Universidad de Oviedo
33071 Oviedo
e-mail: angeles@pinon.ccu.uniovi.es

Resumen

En esta comunicación se presenta un primer acercamiento a la medición de la desigualdad de una población finita con respecto a un aspecto (económico, social, etc.) con valores difusos. Se propone una medida para la desigualdad y se examinan algunas de sus propiedades fundamentales.

1 Introducción

Existen aspectos con respecto a los cuales muchas poblaciones varían sustancialmente de unos individuos a otros. La *desigualdad* es una característica que evalúa dicha variación cuando el aspecto es cuantitativo, y en esa variación se tiene en cuenta si los valores del aspecto están más o menos próximos. El análisis de la desigualdad, su medición numérica y las propiedades de las medidas propuestas es un tema con múltiples aplicaciones en campos como la Economía (desigualdad de rentas, beneficios, etc.), Industria (concentración industrial, etc.) y otros.

Los estudios sobre la desigualdad suponen que los valores del aspecto considerado pertenecen al intervalo real $(0,\infty)$, puesto que el aspecto es habitualmente monetario o corresponde al tamaño de una subpoblación.

Sin embargo, existen diversos rasgos (prestigio, repercusión social, publicidad anadida, imprecisión en la información, etc.) intrínsecos al aspecto cuantitativo que pueden dar lugar a que la escala usual para el aspecto no refleje en cada caso el verdadero valor. Ejemplos de esta naturaleza motivaron la introducción del concepto de variable aleatoria difusa (Puri & Ralescu, 1986) y, más concretamente, el de utilidad difusa en los problemas de decisión (ver, por ejemplo, Freeling, 1980, Tong & Bonissone, 1980, Dubois & Prade, 1982, Whalen, 1984, Gil & Jain, 1992).

Por otra parte, en el estudio de aspectos numéricos exactos Basu (1987) subrayó que al comparar dos estados sociales es frecuente sentirse incapaz de dar una respuesta categórica sobre cuál de ellos presenta mayor desigualdad, puesto que a menudo la noción de desigualdad es imprecisa en sí misma.

El propósito de esta comunicación es presentar una primera aproximación a la medición de la desigualdad de una población finita con respecto a un "aspecto difuso", proponiendo una medida (difusa) y analizando la generalización de las propiedades deseables en ese tipo de medidas.

Para ello, hemos seguido las pautas de la definición de Puri & Ralescu (1986) del valor esperado difuso de una variable aleatoria difusa, ya que las medidas de desigualdad para un aspecto numérico suelen establecerse como promedios especiales; además, hemos adoptado la extensión de una medida que forma parte de la familia introducida por Gastwirth y colaboradores (ver, por ejemplo, Gastwirth *et al.*, 1986, para la función $g(x) = 1/x$ cuando $x \in (0,\infty)$), y que pese a no ser tan popular como el conocido índice de Theil (1967) es más adecuada en el muestreo y tratamiento de poblaciones finitas (según se ha señalado en trabajos previos, como el de Gil *et al.*, 1989); por último, para realizar comparaciones entre distintas situaciones o poblaciones a través de la desigualdad hemos recurrido a las relaciones de dominancia fuerte de Gonzalez & Vila (1992).

2 Desigualdad hiperbólica difusa de una variable aleatoria difusa

Sea $\mathcal{X}$ una variable aleatoria difusa (Puri & Ralescu, 1986) positiva, que en una población de N individuos toma como valores los números difusos $\tilde{x}_1,\ldots,\tilde{x}_N$ (distintos o no) tales que para $j = 1,\ldots,N$, satisfacen que:

1. $(\tilde{x}_j)_\alpha = \alpha$-corte de $\tilde{x}_j = \{x \in R / \mu_{\tilde{x}_j}(x) \geq \alpha\}$ es compacto, para cualquier $\alpha \in (0,1]$,
2. $sop(\tilde{x}_j) = \{x \in R / \mu_{\tilde{x}_j}(x) > 0\}$ es un conjunto acotado contenido en $(0,\infty)$,
3. $(\tilde{x}_j)_0 =$ adherencia de $sop(\tilde{x}_j)$,
4. $(\tilde{x}_j)_1 \neq \emptyset$.

La desigualdad de la población con respecto a esa variable puede "cuantificarse" mediante la medida siguiente:

Definición. *Llamaremos* **desigualdad hiperbólica difusa** *de la población con respecto a $\mathcal{X}$ al número difuso:*

$$\tilde{I}_H(\mathcal{X}) = N^{-1} \otimes \left[(\tilde{E}(\mathcal{X}) \oslash \tilde{x}_1) \oplus \cdots \oplus (\tilde{E}(\mathcal{X}) \oslash \tilde{x}_N) \right] \ominus 1$$

donde $\otimes$, $\oslash$, $\oplus$ y $\ominus$ son las extensiones de las operaciones algebraicas elementales (producto, cociente, suma

y diferencia, respectivamente) basadas en el principio de extensión de Zadeh (Zadeh, 1975, Dubois & Prade, 1978, 1987, y Kaufmann & Gupta, 1985) y $\widetilde{E}(\mathcal{X})$ representa el valor esperado difuso de la variable $\mathcal{X}$ en la población, que en este caso vendrá dada por:

$$\widetilde{E}(\mathcal{X}) = N^{-1} \otimes (\tilde{x}_1 \oplus \cdots \oplus \tilde{x}_N).$$

Si denotamos por $(\tilde{x}_j)_\alpha = [x_{j1}(\alpha), x_{j2}(\alpha)]$, para todo $\alpha \in [0,1]$ y $j = 1, \ldots, N$, la compatibilidad con los α-cortes de las operaciones extendidas permite concluir que:

$$(\widetilde{I}_H(\mathcal{X}))_\alpha = [I_{H1}(\alpha), I_{H2}(\alpha)]$$

con:

$$I_{H1}(\alpha) = N^{-1} \sum_{j=1}^{N} \frac{x_1(\alpha)}{x_{j2}(\alpha)} - 1,$$

$$I_{H2}(\alpha) = N^{-1} \sum_{j=1}^{N} \frac{x_2(\alpha)}{x_{j1}(\alpha)} - 1,$$

$$x_1(\alpha) = N^{-1} \sum_{j=1}^{N} x_{j1}(\alpha),$$

$$x_2(\alpha) = N^{-1} \sum_{j=1}^{N} x_{j2}(\alpha)$$

En esta comunicación no vamos a desarrollar un estudio pormenorizado de la medida que acabamos de introducir, pero sí queremos resaltar algunas de las generalizaciones más interesantes de las propiedades del caso clásico.

Propiedad 1 (No negatividad). *Cualquiera que sea la variable $\mathcal{X}$ y para cualquier $\alpha \in [0,1]$, se cumple que:*

$$I_{H2}(\alpha) \geq -I_{H1}(\alpha)$$

de modo que $\widetilde{I}_H(\mathcal{X})$ se considerará $\succeq 0$ con arreglo a la mayoría de los criterios de ordenación $\succeq$ de numeros difusos que puedan adoptarse (como, por ejemplo, la relación de dominancia fuerte de González & Vila, 1992, al menos para $\lambda = \mu \geq .5$, o la relación de preferencia difusa establecida por Yuan, 1991, entre otras).

Demostración: En efecto, si consideramos la función hiperbólica $g(x) = 1/x$ cuando $x \in (0, \infty)$, y aplicamos la desigualdad de Jensen y el hecho de que $x_1(\alpha) \leq x_2(\alpha)$, se tiene que:

$$I_{H2}(\alpha) = N^{-1} \sum_{j=1}^{N} g\left(\frac{x_{j1}(\alpha)}{x_2(\alpha)}\right) - 1$$

$$\geq g\left(\frac{x_1(\alpha)}{x_2(\alpha)}\right) - 1 \geq 1 - g\left(\frac{x_2(\alpha)}{x_1(\alpha)}\right)$$

$$\geq 1 - N^{-1} \sum_{j=1}^{N} g\left(\frac{x_{j2}(\alpha)}{x_1(\alpha)}\right) = -I_{H1}(\alpha)$$

$\triangle$

Propiedad 2 (Desigualdad mínima). $\widetilde{I}_H(\mathcal{X}) = 0$ *si la variable $\mathcal{X}$ toma valor real constante en toda la población.*

Demostración: En efecto, como $\widetilde{I}_H(\mathcal{X}) = 0$ si y sólo si para todo $\alpha \in [0,1]$ es $I_{H1}(\alpha) = I_{H2}(\alpha) = 0$, será condición necesaria y suficiente que las tres desigualdades de la demostración de la Propiedad 1 resulten igualdades, lo que ocurre, respectivamente, si y sólo si:

$$x_{j1}(\alpha) = x_{l1}(\alpha), \forall \alpha, \forall j, l$$

$$x_1(\alpha) = x_2(\alpha), \forall \alpha$$

$$x_{j2}(\alpha) = x_{l2}(\alpha), \forall \alpha, \forall j, l$$

En consecuencia, $\widetilde{I}_H(\mathcal{X}) = 0$ si y sólo si:

$$x_{j1}(\alpha) = x_1(\alpha) = x_2(\alpha) = x_{j2}(\alpha) \quad \forall \alpha, \forall j$$

lo que obliga a que $\mathcal{X}$ tome un único valor, que será real debido a que los α-cortes deben ser todos ellos independientes de α y se reducen a un mismo subconjunto real unitario. $\triangle$

Nota. La propiedad anterior puede formularse de manera aún más general, indicando que la función de pertenencia de $\widetilde{I}_H(\mathcal{X})$ será par (y, en consecuencia, indiferente con respecto a 0 según un gran número de criterios de ordenación de números difusos) si y sólo si la variable $\mathcal{X}$ toma valor real constante en toda la población.

Propiedad 3 (Descomposición de la desigualdad). *Cualquiera que sea la variable $\mathcal{X}$, se satisface que*

$$\widetilde{I}_H(\mathcal{X}) = N^{-2} \otimes [(N_1 \otimes \widetilde{I}_H(\{\tilde{x}_1\})) \oplus \cdots \oplus (N_N \otimes \widetilde{I}_H(\{\tilde{x}_N\}))] \oplus \widetilde{I}_H^v(\mathcal{X})$$

(con N_j = número de individuos de la población para los que $\mathcal{X}$ es igual a $\tilde{x}_j$) donde $\widetilde{I}_H^v(\mathcal{X}) = 0$ si la variable $\mathcal{X}$ toma un único valor (real o difuso) en toda la poblacion.

Demostración: En efecto:

$$(\widetilde{I}_H^v(\mathcal{X}))_\alpha = N^{-2} \left[\sum_{j=1}^{N} \sum_{l=1/\tilde{x}_l \neq \tilde{x}_j}^{N} \left(\frac{x_{l1}(\alpha)}{x_{j2}(\alpha)} - 1 \right), \sum_{j=1}^{N} \sum_{l=1/\tilde{x}_l \neq \tilde{x}_j}^{N} \left(\frac{x_{l2}(\alpha)}{x_{j1}(\alpha)} - 1 \right) \right]$$

Si todos los valores (reales o difusos) de $\mathcal{X}$ coinciden, se cumplirá que $N_1 = \cdots = N_N = N$ y $x_{j1}(\alpha) = x_1(\alpha)$, $x_{j2}(\alpha) = x_2(\alpha), \forall \alpha, \forall j$, de manera que $\widetilde{I}_H^v(\mathcal{X}) = 0$.

Por otro lado, si $\widetilde{I}_H^v(\mathcal{X}) = 0$ y existen $k \geq 2$ valores distintos de $\mathcal{X}$, $\tilde{x}_1^*, \ldots, \tilde{x}_k^*$, que aparecen en la población $n_1, \ldots, n_k$ veces, deberá ser:

$$x_{i1}^*(\alpha) x_{i'1}^*(\alpha) = x_{i2}^*(\alpha) x_{i'2}^*(\alpha), \forall \alpha, \forall i, i' \in \{1, \ldots, k\}$$

de donde se concluye que:

$$x_{i1}^*(\alpha) = x_{i'1}^*(\alpha) = x_{i2}^*(\alpha) = x_{i'2}^*(\alpha),$$

$$\forall \alpha, \forall i, i' \in \{1, \ldots, k\}$$

lo cual es absurdo. △

Nota. La propiedad precedente sugiere que la desigualdad difusa puede descomponerse como suma de cierta medida de la desigualdad difusa *intra*valores y de una medida de la desigualdad difusa *inter*valores, $\widetilde{I}_H^v$.

Propiedad 4 (Independencia de la media). *Cualquiera que sea la variable $\mathcal{X}$ y para cualquier constante $k > 0$, se satisface que*

$$\widetilde{I}_H(k \otimes \mathcal{X}) = \widetilde{I}_H(\mathcal{X})$$

donde $k \otimes \mathcal{X}$ es la variable aleatoria difusa que en la población considerada toma como valores los valores difusos $k \otimes \tilde{x}_1, \ldots, k \otimes \tilde{x}_N$. △

Propiedad 5 (Principio de transferencias). *Sea $\mathcal{X}$ una variable aleatoria difusa positiva que en una población de N individuos toma los valores difusos $\tilde{x}_1, \ldots, \tilde{x}_N$ y supongamos que existen $h, l \in \{1, \ldots, N\}$ tales que $\tilde{x}_h \succeq_S \tilde{x}_l$ de acuerdo con la relación de dominancia fuerte $\succeq_S$ de González & Vila (1992) (cualesquiera que sean λ y μ), es decir:*

$$x_{h1}(\alpha) - x_{l1}(\alpha) \geq 0, x_{h2}(\alpha) - x_{l2}(\alpha) \geq 0$$

para todo $\alpha \in [0,1]$. Sea $\varepsilon > 0$ tal que:

$$\varepsilon \leq inf_\alpha \{x_{h1}(\alpha) - x_{l1}(\alpha), x_{h2}(\alpha) - x_{l2}(\alpha)\}$$

(es decir, $\varepsilon \leq$ distancia mínima entre $\tilde{x}_h$ y $\tilde{x}_l$), y definamos una nueva variable aleatoria difusa positiva $\mathcal{Y}$ cuyos valores son:

$$\tilde{y}_h = \tilde{x}_h \ominus \varepsilon, \tilde{y}_l = \tilde{x}_l \oplus \varepsilon$$

$$\tilde{y}_j = \tilde{x}_j, j \neq h, l$$

Se cumple entonces que:

$$\widetilde{I}_H(\mathcal{X}) \succeq_S \widetilde{I}_H(\mathcal{Y}).$$

Demostración: En efecto, como $y_1(\alpha) = x_1(\alpha)$, $y_2(\alpha) = x_2(\alpha), \forall \alpha \in [0,1]$, de manera que tomando $\lambda = 1 - \varepsilon/(x_{h1}(\alpha) - x_{l1}(\alpha))$, se tiene que:

$$\begin{aligned} y_{h1}(\alpha) &= \lambda x_{h1}(\alpha) + (1-\lambda) x_{l1}(\alpha) \\ y_{h2}(\alpha) &= \lambda x_{h2}(\alpha) + (1-\lambda) x_{l2}(\alpha) \\ y_{l1}(\alpha) &= (1-\lambda) x_{h1}(\alpha) + \lambda x_{l1}(\alpha) \\ y_{l2}(\alpha) &= (1-\lambda) x_{h2}(\alpha) + \lambda x_{l2}(\alpha) \end{aligned}$$

Si consideramos la función $g(x) = 1/x$ en $(0, \infty)$ y aplicamos la desigualdad de Jensen, concluimos que:

$$\begin{aligned} g(y_{h1}(\alpha)) + g(y_{l1}(\alpha)) &\leq g(x_{h1}(\alpha)) + g(x_{l1}(\alpha)) \\ g(y_{h2}(\alpha)) + g(y_{l2}(\alpha)) &\leq g(x_{h2}(\alpha)) + g(x_{l2}(\alpha)) \end{aligned}$$

En consecuencia, para cada α, $I_{H1}(\alpha)$ e $I_{H2}(\alpha)$ son a lo sumo iguales para $\mathcal{Y}$ que los correspondientes para $\mathcal{X}$, de modo que:

$$\widetilde{I}_H(\mathcal{X}) \succeq_S \widetilde{I}_H(\mathcal{Y}).$$

△

Nota. La propiedad anterior formaliza la idea de que cuando se transfiere "parte" de un valor difuso superior a otro inferior conservando el valor esperado difuso se obtiene una reducción de la desigualdad.

3 Conclusiones finales

El estudio que hemos presentado es obviamente una primera toma de contacto con el problema de la medición de la desigualdad para variables difusas. Nuestro interés actual es profundizar en ese estudio desde varias perspectivas. Entre los objetivos más inmediatos podemos destacar el que corresponde a un análisis separado de la medida $\widetilde{I}_H^v$ y sus propiedades, y otros objetivos tambien a corto plazo son el de examinar otras propiedades diferentes a las recogidas en esta comunicación para las medidas $\widetilde{I}_H$ e $\widetilde{I}_H^v$, así como diferentes versiones de la Propiedad 5 de la sección precedente (según las relaciones de ordenación consideradas) y el de realizar aplicaciones de ambas medidas al tratamiento de datos agrupados, situación que aparece con bastante asiduidad en la práctica (ver, por ejemplo, Gastwirth *et al.*, 1986).

Agradecimientos

La investigación en este trabajo está subvencionada en parte a través del Proyecto DGICYT PB92-1014. Queremos agradecer al Ministerio de Educación y Ciencia dicha ayuda económica.

Referencias

[1] Basu, K. (1987). Axioms for a fuzzy measure of inequality. *Mathematical Social Sciences* 14, 275-288.

[2] Bourguignon, F. (1979). Decomposable Income Inequality Measures. *Econometrica* **47**, 901-920.

[3] Dubois, D. & Prade, H. (1978). Operations on fuzzy numbers. *International Journal of Systems Science* **9**, 613-626.

[4] Dubois, D. & Prade, H. (1982). The use of fuzzy numbers in Decision Analysis. In *Fuzzy Information and Decision Processes*. (Gupta, M.M. & Sanchez, E. eds.). Amsterdam: North-Holland, 309-321.

[5] Dubois, D. & Prade, H. (1987). Fuzzy numbers: an overview. In *Analysis of Fuzzy Information, Vol. I.* (Bezdek, J.C. ed.). Boca Raton: CRC, 3-39.

[6] Freeling, A.N.S. (1980). Fuzzy Sets and Decision Analysis. *IEEE Transactions on Systems, Man and Cybernetics* **10**, 341-354.

[7] Gastwirth, J.L., Nayak, T.K. & Krieger, A.M. (1986). Large sample theory for the bounds on the Gini and related indices of inequality estimated from grouped data. *Journal of Business and Economic Statistics* **4**, 269-273.

[8] Gil, M.A. & Jain, P. (1992). Comparison of experiments in statistical decision problems with fuzzy utilities. *IEEE Transactions on Systems, Man and Cybernetics* **22**, 662-670.

[9] Gil, M.A., Pérez, R. & Gil, P. (1989). A family of measures of uncertainty involving utilities: definitions, properties, applications and statistical inferences. *Metrika* **36**, 129-147.

[10] González, A. & Vila, M.A. (1992). Dominance relations on fuzzy numbers. *Information Sciences* **64**, 1-16.

[11] Kaufmann, A. & Gupta, M.M. (1985). *Introduction to Fuzzy Arithmetic*. New York: Van Nostrand Reinhold Co..

[12] Puri, M.L. & Ralescu, D.A. (1986). Fuzzy random variables. *Journal of Mathematical Analysis and Applications* **114**, 409-422.

[13] Tong, R.M.. & Bonissone, P.P. (1980). A linguistic approach to decisionmaking with fuzzy sets. *IEEE Transactions on Systems, Man and Cybernetics* **10**, 716-723.

[14] Whalen, T. (1984). Decisionmaking under uncertainty with various assumptions about available information. *IEEE Transactions on Systems, Man and Cybernetics* **14**, 888-900.

[15] Yuan, Y. (1991). Criteria for evaluating fuzzy ranking methods. *Fuzzy Sets and Systems* **44**, 139-157.

[16] Zadeh, L.A. (1975). The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences* Part 1 **8**, 199-249. Part 2 **8**, 301-353. Part 3 **9**, 43-80.

Integración Difusa sobre Espacios Producto

Emilio Suárez Díaz
Departamento de Matemáticas
E.T.S. Ingenieros Indistriales e Informáticos.
Universidad de Oviedo.
e-mail: meana@etsiig.uniovi.es

Junio 12, 1995

Resumen

Desde que en 1974 Sugeno introdujo el concepto de integral difusa y medida difusa [6], han aparecido en la literatura especializada numerosas definiciones sobre integración difusa [1], [5], [7].

En el presente trabajo, basándonos en la definición de Kruse [1], extendemos la posibilidad de utilizar estas integrales sobre el producto de dos espacios medibles, dando una versión del conocido teorema de Fubini para integrales difusas.

Palabras clave: *T-conorma, Medida Difusa, Integral Difusa*

1 Introducción

El objetivo principal del presente trabajo consiste en considerar, entre otros, el importante teorema de Fubini que aparece en la teoría clásica de funciones medibles respecto de medidas σ-aditivas finitas, y generalizarlo respecto de un tipo de medidas más generales: las medidas difusas σ-aditivas del tipo NSA, que además de incluir a las σ-aditivas, incluyen otros tipos interesantes de medidas ya muy estudiadas, tales como las medidas de Sugeno.

La generalización está basada en el concepto de t-conormas Arquimedianas en las que resulta fundamental el teorema de Ling, gracias al cual podemos convertir medidas NSA en medidas aditivas mediante una aplicación y así poder apoyarnos en la teoría clásica de la medida. El trabajo se desarrolla como sigue:

En la sección 2 se consideran los conceptos fundamentales necesarios para la introducción del tema tales como el concepto de t-conorma triangular y en espacial el del tipo arquimediano, así como teoremas sobre los mismos que se usarán en el desarrollo del mismo. Resulta fundamental aquí el teorema de C.H. Ling [2] mediante el cual, podemos expresar una t-conorma arquimediana T de la forma:

$$xTy = g^{(-1)}\left(g(x) + g(y)\right)$$

Siendo g una función creciente y continua de dominio $[0,1]$ y $g^{(-1)}$ su función pseudoinversa. También se incluyen algunos ejemplos de las t-conormas usuales que aparecen el los trabajos especializado.

En la sección 3 se consideran los conceptos de medidas T-descomponibles y σ-T-descomponibles que aparecen en trabajos de S. Weber [7] así como diversos teoremas sobre las mismas. Cuando T es una t-conorma arquimediana, se consideran los tres tipos fundamentales de este tipo m de medidas:

S: La composición $g \circ m$ es una medida aditiva infinita.

NSP: La composición $g \circ m$ es una medida pseudoaditiva finita.

NSA: La composición $g \circ m$ es una medida aditiva finita.

Estas últimas constituyen una parte fundamental del presente trabajo, ya que serán las que consideremos exclusivamente.

En la sección 4 aparece la parte fundamental del trabajo. R.Kruse [1] dio una definición de integrales de funciones medibles con respecto de una medida de Sugeno. Dado que las medidas $\lambda-$ aditivas difusas pueden considerarse un caso particular de medidas del tipo NSA, hemos considerado oportuno dar una generalización de la integral de Kruse [1] respecto de medidas NSA así como investigar algunas propiedades y teoremas que pudieran cumplirse.

La sección 5 es, en realidad, una continuación de las propiedades mencionadas en el apartado anterior, aunque por su extensión y necesidad de lemas previos se haya considerado por separado. Se trata de la demostración del teorema de Fubini respecto de medidas NSA. Ello conlleva la introducción de una medida del mismo tipo en un espacio producto.

2 Conceptos fundamentales.

2.1 t-conorma triangular

Una aplicación $T : [0,1] \times [0,1] \longrightarrow [0,1]$; $(x,y) \longrightarrow xTy$ que cumpla las propiedades:

1º. $xT0 = 0Tx = x \ ; \ \forall x \in [0,1]$

2º. La aplicación es no decreciente respecto de cada argumento:

$$x_1 \leq y_1 \ ; \ x_2 \leq y_2 \Longrightarrow x_1 T x_2 \leq y_1 T y_2$$

3º. Conmutatividad:

$$xTy = yTx \ ; \ \forall x, y \in [0,1]$$

4º. Asociatividad:

$$xT(yTz) = (xTy)Tz \ ; \ \forall x, y, z \in [0,1]$$

Diremos que es una *t- conorma triangular.*

2.2 t-conorma Arquimediana

Una t-conorma triangular diremos que es *Arquimediana*, si cumple las propiedades:

5º. T es continua en $[0,1]\times[0,1]$

6º. $xTx > x \ ; \ \forall x \in]0,1[$

Una t-conorma Arquimediana, diremos que es *estricta* cuando sea estrictamente creciente en $]0,1[\times]0,1[$.

Teorema 1 *Una función* $T : [0,1]\times[0,1] \longrightarrow [0,1]$ *es una t-conorma Arquimediana si, y sólo si, existe una función* $g : [0,1] \longrightarrow [0,\infty]$ *con* $g(0) = 0$, *creciente y continua, de tal forma que:*

$$xTy = g^{(-1)}\left(g(x) + g(y)\right) \ ; \ \forall x, y \in [0,1]$$

Donde $g^{(-1)}$ *denota la pseudoinversa de* g *definida de la forma:*

$$g^{(-1)}(y) = \begin{cases} g^{-1}(y) & si \ \ y \in [0, g(1)] \\ 1 & si \ \ y \in [g(1), \infty] \end{cases}$$

Además T *es estricta si, y sólo si,* $g(1) = \infty$

- La función g se dice que es un *generador aditivo de* T.

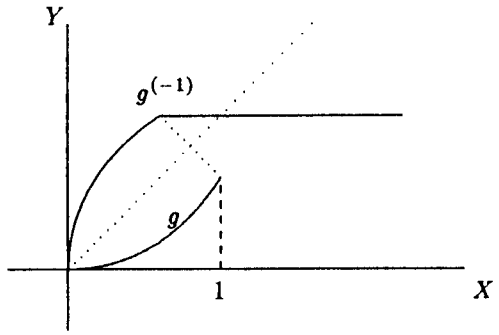


Figura 1: Función pseudoinversa

Son evidentes las propiedades:

- $g \circ g^{(-1)}(y) = \min\{ y, g(1) \} \ ; \ \forall y \in [0,\infty]$
- $g^{(-1)} \circ g(x) = x \ ; \ \forall x \in [0,1]$

Teorema 2 *Si* T *es una t-conorma Arquimediana y* g *un generador aditivo de la misma, entonces para cualquier* $k \in \mathbb{R} \ ; \ k > 0$, *la función* $f = kg$ *es otro generador aditivo de la misma.*

1

Ejemplo 1 *Sea* $\lambda \in]-1,\infty[$. *y sea la función:*

$$xTy = \min\{ x + y + \lambda xy \, , \, 1 \} \ ; \ x, y \in [0,1]. \quad (1)$$

Puede comprobarse que es una t-conorma Arquimediana (t-conorma de Sugeno). Una función generador aditivo es la siguiente:

$$g_\lambda(x) = \log_{1+\lambda}(1 + \lambda x) \ ; \ x \in [0,1]$$

Siendo $g(1) = 1$, *la t-conorma es no estricta. La función pseudoinversa es:*

$$g^{(-1)}(y) = \begin{cases} \frac{(1+\lambda)^y - 1}{\lambda} & si \ \ y \leq 1 \\ 1 & si \ \ y > 1 \end{cases}$$

3 Medidas descomponibles. Medidas σ-descomponibles

Definición 1 *Sea* $(\Omega, \mathcal{A})$ *un espacio medible y supongamos una función* $m : \mathcal{A} \longrightarrow [0,1]$ *tal que:*

$$m(\emptyset) = 0 \ ; \ m(\Omega) = 1$$

Sea T *una t-conorma. Entonces diremos que:*

a) m *es una medida T-descomponible si, y sólo si:*

$$m(A \uplus B) = m(A) \ T \ m(B) \ ; \ A, B \in \mathcal{A}$$

Donde $A \uplus B$ *denota* $A \cup B$ *para* $A \cap B = \emptyset$

b) m *es una medida* σ*-T-descomponible si, y sólo si:*

$$m\left(\biguplus_{k=1}^{\infty} A_k\right) = T_{k=1}^{\infty} \, m(A_k)$$

Teorema 3 *Se cumplen los siguientes apartados:*

1) m *es T-descomponible* $\Longrightarrow$ m *es monótona.*

2) m *es* σ*-T-descomponible* $\Longleftrightarrow$ m *es T-descomponible y continua inferiormente.*

La demostración de este teorema aparece en [7].

3.1 Transformación en medidas de probabilidad

Sea $(\Omega, \mathcal{A})$ un espacio medible y sea T una t-conorma Arquimediana con generador aditivo g . Consideremos una función $m : \mathcal{A} \longrightarrow [0,1]$. Podemos considerar la composición de funciones:

$$\begin{array}{ccccc} \mathcal{A} & \xrightarrow{m} & [0,1] & \xrightarrow{g} & [0,\infty] \\ A & \longrightarrow & m(A) & \longrightarrow & g(m(A)) \end{array}$$

Llamemos $g \circ m = m^*$

Teorema 4 *Se cumplen los apartados siguientes:*

1) m *es T-descomponible* $\Longrightarrow$ $m^*(A \uplus B) = \min\{ m^*(A) + m^*(B), g(1) \}$

2) m *es* σ*-T-descomponible* $\Longrightarrow$ $m^*\left(\biguplus_{k=1}^{\infty} A_k\right) = \min\{ \sum_{k=1}^{\infty} m^*(A_k), g(1) \}$

Supongamos que m es descomponible (σ-descomponible) respecto de la t-conorma Arquimediana T. A la vista de los resultados anteriores, podemos distinguir entre los tres siguientes casos:

S: Si T es estricta, entonces m^* es una *medida aditiva (σ-aditiva) infinita.*

NSP: Si T no es estricta y existe alguna familia finita o infinita numerable $\{ A_k \}$ de conjuntos disjuntos de $\mathcal{A}$ de forma que se cumple $\sum_k (g \circ m)(A_k) > g(1)$. Entonces m^* es un *medida pseudoaditiva (σ-aditiva).*

NSA: Si T no es estricta y para cualquier familia finita (infinita numerable) de subconjuntos disjuntos $\{A_k\}$ de $\mathcal{A}$, se cumple $\sum_k (g \circ m)(A_k) \leq g(1)$. Entonces es m^* *una medida aditiva (σ-aditiva) finita con* $m^*(\Omega) = g(m(\Omega)) = g(1)$. Podemos tomar un generador aditivo tal que $g(1) = 1$ de tal forma que para el caso σ-aditivo, la medida es de *probabilidad* ya que entonces $m^*(\Omega) = 1$. Este es el caso que nos interesa fundamentalmente para el resto del trabajo.

Ejemplo 2 *Medidas de Sugeno: Sea* $(\Omega, \mathcal{A})$ *un espacio medible. Una* g_λ*-medida difusa de Sugeno es una aplicación:*

$$g_\lambda : \mathcal{A} \longrightarrow [0,1] \ ; \lambda \in]-1, \infty[\quad \textit{cumpliéndose:}$$

1) $g_\lambda(\Omega) = 1 \ ; \ g_\lambda(\emptyset) = 0$

2) $g_\lambda(A \uplus B) = g_\lambda(A) + g_\lambda(B) + \lambda g_\lambda(A) g_\lambda(B)$

3) g_λ *es continua inferiormente respecto de sucesiones monótonas.*

Nótese que las medidas de Sugeno son un caso particular de medidas σ-descomponibles del tipo NSA cuando consideramos precisamente t-conormas de Sugeno (1).

En efecto, recordando que, en este caso, es $g(x) = \log_{1+\lambda}(1 + \lambda x)$, *se tiene:*

$$m^*_\lambda(A \cup B) = \log_{1+\lambda}(1 + \lambda m_\lambda(A \cup B)) =$$

$$= \log_{1+\lambda}(1 + \lambda(m_\lambda(A) + m_\lambda(B) + \lambda m_\lambda(A) m_\lambda(B))) =$$

$$= \log_{1+\lambda}(1 + \lambda m_\lambda(A))(1 + \lambda m_\lambda(B)) =$$

$$= \log_{1+\lambda}(1 + \lambda m_\lambda(A)) + \log_{1+\lambda}(1 + \lambda m_\lambda(B)) =$$

$$= m^*_\lambda(A) + m^*_\lambda(B)$$

Luego m^*_λ *es aditiva, y por la continuidad, será σ-aditiva.*

4 Integral de una función medible respecto de una medida del tipo NSA.

Sea $(\Omega, \mathcal{A})$ un espacio medible y referido al mismo, sea m una σ-T-medida descomponible del tipo NSA con generador aditivo g en el que suponemos $g(1) = 1$. La función $m^* = g \circ m$ es una medida de probabilidad en $(\Omega, \mathcal{A})$.

Supongamos ahora una función $f : \Omega \longrightarrow [0,1]$ que sea $\mathcal{A}$-medible. La función $f^* = g \circ f$ será también $\mathcal{A}$-medible al ser composición de una función medible y una función continua, con lo cual tiene sentido considerar la integral de Lebesgue-Stieljes:

$$\int_\Omega f^* \, dm^*$$

Es fácil ver que se verifica $0 \leq \int_\Omega f^* \, dm^* \leq g(1) = 1$.

Definición 2 *Definimos la integral de la función f respecto de la medida difusa m al valor:*

$$\boxed{\int_\Omega f \, dm = g^{-1}\left[\int_\Omega f^* \, dm^*\right]} \qquad (2)$$

Al igual que en la teoría clásica, si $A \in \mathcal{A}$, se define:

$$\int_A f \, dm = \int_\Omega f I_A \, dm$$

Se establece la misma relación que en (2) ya que se tiene:

$$\int_A f \, dm = \int_\Omega f I_A \, dm = g^{-1}\left[\int_\Omega (g \circ f I_A) \, dm^*\right] =$$

$$= g^{-1}\left[\int_\Omega (g \circ f) I_A \, dm^*\right] = g^{-1}\left[\int_\Omega f^* I_A \, dm^*\right] =$$

$$= g^{-1}\left[\int_A f^* \, dm^*\right]$$

Veamos ahora algunas propiedades de tal integral:

4.1 Monotonía

Sean $A, B \in \mathcal{A}$; entonces, es fácil ver que:

$$A \subset B \Longrightarrow \int_A f \, dm \leq \int_B f \, dm$$

4.2 T-descomponibilidad

Sean $A, B \in \mathcal{A}$ tales que $A \cap B = \emptyset$. Se demuestra:

$$\int_{A \cup B} f \, dm = \left(\int_A f \, dm\right) T \left(\int_B f \, dm\right)$$

4.3 Teorema de la convergencia monótona

Se puede demostrar el siguiente teorema:

Teorema 5 *Sea* $(\Omega, \mathcal{A})$ *un espacio medible y sea m una medida σ-T-descomponible NSA. Sea* $X_n : \Omega \longrightarrow [0,1]$ *una sucesión de funciones $\mathcal{A}$-medibles tales que* $\{X_n\} \uparrow X$ *puntualmente en* Ω. *Entonces se cumple:*

1) La función X tiene su recorrido en $[0,1]$.

2) $\{g \circ X_n\} \uparrow g \circ X$ *puntualmente en* Ω.

3) $\lim_{n \to \infty} \int_\Omega X_n \, dm = \int_\Omega X \, dm$

5 El Teorema de Fubini respecto de medidas NSA

5.1 Resultados de la teoría clásica

1) Sean $(\Omega_1, \mathcal{A}_1)$ y $(\Omega_2, \mathcal{A}_2)$ espacios medibles y sea $(\Omega_1 \times \Omega_2, \mathcal{A}_1 \times \mathcal{A}_2)$ el espacio medible producto. Sea $A \in \mathcal{A}_1 \times \mathcal{A}_2$. Para un $\omega_1 \in \Omega_1$ y un $\omega_2 \in \Omega_2$ arbitrarios consideremos las *secciones de A por ω_1 y ω_2*:

$$A_{\omega_1} = \{\omega_2 \in \Omega_2 \,/\, (\omega_1, \omega_2) \in A\}$$

$$A_{\omega_2} = \{\omega_1 \in \Omega_1 \,/\, (\omega_1, \omega_2) \in A\}$$

Proposición 1 *Los conjuntos* A_{ω_1} *y* A_{ω_2} *son* $\mathcal{A}_2$*-medibles y* $\mathcal{A}_1$*−medibles respectivamente.*

2) Sean $(\Omega_1, \mathcal{A}_1)$ y $(\Omega_2, \mathcal{A}_2)$ espacios medibles y sea $f : \Omega_1 \times \Omega_2 \longrightarrow \mathbb{R}$ una función medible respecto de $\mathcal{A}_1 \times \mathcal{A}_2$. Definimos *secciones de* f *por* $\omega_1 \in \Omega_1$ *y* $\omega_2 \in \Omega_2$ a las funciones:

$$f_{\omega_1} : \Omega_2 \longrightarrow \mathbb{R} / f_{\omega_1}(\omega_2) = f(\omega_1, \omega_2)$$
$$f_{\omega_2} : \Omega_1 \longrightarrow \mathbb{R} / f_{\omega_2}(\omega_1) = f(\omega_1, \omega_2)$$

Proposición 2 *Las secciones* f_{ω_1} *y* f_{ω_2} *son funciones* $\mathcal{A}_2$*-medibles y* $\mathcal{A}_1$*-medibles respectivamente.*

Proposición 3 *Sea* $f : \Omega_1 \times \Omega_2 \longrightarrow \mathbb{R}$ *función medible en el espacio medible* $(\Omega_1 \times \Omega_2, \mathcal{A}_1 \times \mathcal{A}_2)$ *y sean* μ_1 *y* μ_2 *medidas* σ*-finitas en* $(\Omega_1, \mathcal{A}_1)$ *y* $(\Omega_2\ \mathcal{A}_2)$ *respectivamente. Se tiene:*

$$s_1(\omega_1) = \int_{\Omega_2} f_{\omega_1}\, d\mu_2 \quad \textit{es } \mathcal{A}_1\textit{-medible.}$$
$$s_2(\omega_2) = \int_{\Omega_1} f_{\omega_2}\, d\mu_1 \quad \textit{es } \mathcal{A}_2\textit{-medible.}$$

Corolario 1 *Sean* μ_1 *y* μ_2 *medidas* σ*-finitas definidas en los espacios medibles* $(\Omega_1, \mathcal{A}_1)$ *y* $(\Omega_2, \mathcal{A}_2)$ *. Si consideramos el espacio medible producto* $(\Omega_1 \times \Omega_2, \mathcal{A}_1 \times \mathcal{A}_2)$ *y* $A \in \mathcal{A}_1 \times \mathcal{A}_2$ *, entonces las funciones:*

$$c_1(\omega_1) = \mu_2(A_{\omega_1}) \quad y \quad c_2(\omega_2) = \mu_1(A_{\omega_2})$$

Son $\mathcal{A}_1$*-medibles y* $\mathcal{A}_2$*-medibles respectivamente.*

Proposición 4 *Sean* $(\Omega_1, \mathcal{A}_1)$ *y* $(\Omega_2, \mathcal{A}_2)$ *espacios medibles y sean* μ_1 *y* μ_2 *medidas* σ*-finitas en* $\mathcal{A}_1$ *y* $\mathcal{A}_2$ *respectivamente. Definimos en* $\mathcal{A}_1 \times \mathcal{A}_2$ *la medida:*

$$\mu(A) = \int_{\Omega_1} \mu_2(A_{\omega_1})\, d\mu_1$$

La anterior medida es la única medida σ*-finita producto de* μ_1 *por* μ_2 *(o sea extensión de una función definida para los "cilindros" de la forma* $\mu(A_1 \times A_2) = \mu_1(A_1) \cdot \mu_2(A_2)$ *para* $A_1 \in \mathcal{A}_1$ *;* $A_2 \in \mathcal{A}_2$ *). Denotamos* $\mu = \mu_1 \times \mu_2$.

Lo mismo podemos afirmar para la medida:

$$\mu^*(A) = \int_{\Omega_2} \mu_1(A_{\omega_2})\, d\mu_2$$

Y por tanto, $\mu = \mu^*$

Teorema 6 (Fubini)
Consideremos las medidas μ_1 *;* μ_2 *en* $(\Omega_1, \mathcal{A}_1)$ *y* $(\Omega_2, \mathcal{A}_2)$ *así como la medida* $\mu = \mu_1 \times \mu_2$ *en* $(\Omega_1 \times \Omega_2, \mathcal{A}_1 \times \mathcal{A}_2)$ *indicada en la proposición anterior. Sea una función:*

$f : \Omega_1 \times \Omega_2 \longrightarrow \mathbb{R}$ *que suponemos es* $\mathcal{A}_1 \times \mathcal{A}_2$*-medible*

Entonces se verifica:

$$\int_{\Omega_1 \times \Omega_2} f\, d\mu = \int_{\Omega_1} \left[\int_{\Omega_2} f_{\omega_1}\, d\mu_2 \right] d\mu_1 =$$
$$= \int_{\Omega_2} \left[\int_{\Omega_1} f_{\omega_2}\, d\mu_1 \right] d\mu_2$$

5.2 Teoremas correspondientes para medidas NSA

Proposición 5 *Sean* m_1 *y* m_2 *medidas* σ*-T-descomponibles NSA en* $(\Omega_1, \mathcal{A}_1)$ *y* $(\Omega_2, \mathcal{A}_2)$ *respectivamente y sea:*

$f : \Omega_1 \times \Omega_2 \longrightarrow [0,1]$ *función* $\mathcal{A}_1 \times \mathcal{A}_2$*-medible*

Si son f_{ω_1} *y* f_{ω_2} *las secciones de* f *por* $\omega_1 \in \Omega_1$ *y por* $\omega_2 \in \Omega_2$ *, entonces las funciones:*

$$s_1(\omega_1) = \int_{\Omega_2} f_{\omega_1}\, dm_2 \;\; ; \;\; s_2(\omega_2) = \int_{\Omega_1} f_{\omega_2}\, dm_1$$

Son $\mathcal{A}_1$*-medibles y* $\mathcal{A}_2$*-medibles respectivamente.*

Demostración: Observemos que $(g \circ f)_{\omega_1} = g \circ f_{\omega_1}$ y, entonces, se tiene:

$$\int_{\Omega_1} f_{\omega_1}\, dm_2 = g^{-1}\left[\int_{\Omega_2} (g \circ f_{\omega_1})\, dm_2^* \right] =$$
$$= g^{-1}\left[\int_{\Omega_1} (g \circ f)_{\omega_1})\, dm_2^* \right]$$

Al ser $g \circ f$ $\mathcal{A}_1 \times \mathcal{A}_2$-medible, será $\int_{\Omega_1} (g \circ f)_{\omega_1}\, dm_2^*$ $\mathcal{A}_1$-medible, según la proposición (3). Por último, al ser g continua en el cerrado $[0,1]$, será g^{-1} continua en el cerrado $[0, g(1)] = [0,1]$ y por tanto, también es medible $g^{-1}\left[\int_{\Omega_1} (g \circ f)_{\omega_1}\, dm_2^* \right]$.

La demostración de que $s_2(\omega_2)$ es $\mathcal{A}_2$-medible es análoga. □

Corolario 2 *Sean* m_1 *y* m_2 *σ-T-medidas descomponibles tipo NSA definidas en los espacios medibles* $(\Omega_1, \mathcal{A}_1)$ *y* $(\Omega_2, \mathcal{A}_2)$ *. Si consideramos el espacio medible producto* $(\Omega_1 \times \Omega_2, \mathcal{A}_1 \times \mathcal{A}_2)$ *y* $A \in \mathcal{A}_1 \times \mathcal{A}_2$ *, entonces las funciones:*

$$c_1 : \Omega_1 \longrightarrow [0,1] \;/\; c_1(\omega_1) = m_2(A_{\omega_1})$$
$$c_2 : \Omega_2 \longrightarrow [0,1] \;/\; c_2(\omega_2) = m_1(A_{\omega_2})$$

Son $\mathcal{A}_1$*-medibles y* $\mathcal{A}_2$*-medibles respectivamente.*

Demostración: Tomemos en la proposición anterior $f = I_A$. Entonces la función:

$$\beta_1(\omega_1) = \int_{\Omega_2} (I_A)_{\omega_1}\, dm_2$$

Es una función $\mathcal{A}_1$-medible. Teniendo en cuenta las igualdades $(I_A)_{\omega_1} = I_{A_{\omega_1}}$ y que la composición de funciones $g \circ I_{A_{\omega_1}} = g(1) \cdot I_{A_{\omega_1}} = I_{A_{\omega_1}}$, podemos escribir el proceso:

$$\int_{\Omega_2} (I_A)_{\omega_1}\, dm_2 = g^{-1}\left[\int_{\Omega_2} (g \circ I_{A_{\omega_1}})\, dm_2^* \right] =$$
$$= g^{-1}\left[\int_{\Omega_2} I_{A_{\omega_1}}\, dm_2^* \right] = g^{-1}\left[m_2^*(A_{\omega_1}) \right] =$$

$$= g^{-1}[(g \circ m_2)(A_{\omega_1})] = m_2(A_{\omega_1})$$

Luego $m_2(A_{\omega_1}) = \beta_1(\omega_1)$ y por tanto, es una función $\mathcal{A}_1$-medible.

La demostración de que es $m_1(A_{\omega_2})$ función $\mathcal{A}_2$-medible es totalmente análoga. □

Teorema 7 *Sean m_1 y m_2 σ-T-medidas descomponibles NSA definidas respectivamente en los espacios medibles $(\Omega_1, \mathcal{A}_1)$ y $(\Omega_2, \mathcal{A}_2)$. Sean las funciones m y M definidas en $\mathcal{A}_1 \times \mathcal{A}_2$ de la forma:*

$$m(A) = \int_{\Omega_1} m_2(A_{\omega_1})\, dm_1 \;;\; A \in \mathcal{A}_1 \times \mathcal{A}_2$$

$$M(A) = \int_{\Omega_2} m_1(A_{\omega_2})\, dm_2 \;;\; A \in \mathcal{A}_1 \times \mathcal{A}_2$$

Entonces se verifica:

1) m y M son medidas σ-T-descomponibles NSA.

2) $m = M$

Demostración: Demostramos el teorema para la medida m. La demostración es idéntica para M. Por la definición de m, se tiene:

$$m(A) = \int_{\Omega_1} m_2(A_{\omega_1})\, dm_1 =$$

$$= g^{-1}\left[\int_{\Omega_1} g \circ m_2(A_{\omega_1})\, dm_1^*\right] =$$

$$= g^{-1}\left[\int_{\Omega_1} m_2^*(A_{\omega_1})\, dm_1^*\right]$$

Donde m_1^* y m_2^* son, como siempre, las medidas de probabilidad asociadas a m_1 y m_2 por la función g.

1) Veamos que se cumplen todas las propiedades:

$$m(\emptyset) = g^{-1}\left[\int_{\Omega_1} m_2^*(\emptyset_{\omega_1})\, dm_1^*\right] =$$

$$= g^{-1}\left[\int_{\Omega_1} 0\, dm_1^*\right] = g^{-1}(0) = 0$$

$$m(\Omega_1 \times \Omega_2) = g^{-1}\left[\int_{\Omega_1} m_2^*(\Omega_1 \times \Omega_2)_{\omega_1}\, dm_1^*\right] =$$

$$= g^{-1}\left[\int_{\Omega_1} m_2^*(\Omega_2)\, dm_1^*\right] =$$

$$= g^{-1}\left[\int_{\Omega_1} 1 \cdot dm_1^*\right] = g^{-1}(1) = 1$$

Veamos que m es T-descomponible:

Sean $A, B \in \mathcal{A}_1 \times \mathcal{A}_2$ siendo $A \cap B = \emptyset$. Se tiene:

$$m(A \cup B) = g^{-1}\left[\int_{\Omega_1} m_2^*(A \cup B)_{\omega_1}\, dm_1^*\right] =$$

$$= g^{-1}\left[\int_{\Omega_1} m_2^*(A_{\omega_1} \cup B_{\omega_1})\, dm_1^*\right] =$$

$$= g^{-1}\left[\int_{\Omega_1} m_2^*(A_{\omega_1})\, dm_1^* + \int_{\Omega_1} m_2^*(B_{\omega_1})\, dm_1^*\right] \tag{3}$$

Recordando que $xTy = g^{(-1)}[g(x) + g(y)]$, si p y q son tales que $\begin{cases} 0 \le p \le 1 \\ 0 \le q \le 1 \\ 0 \le p+q \le 1 \end{cases}$ entonces se tendrá:

$$g^{(-1)}(p+q) = g^{-1}(p+q) =$$

$$= g^{-1}[(g \circ g^{-1})(p) + (g \circ g^{-1})(p)] =$$

$$= g^{-1}(p)\, \mathrm{T}\, g^{-1}(q)$$

Entonces la expresión (3) podemos escribirla de la forma:

$$g^{-1}\left[\int_{\Omega_1} m_2^*(A_{\omega_1})\, dm_1^* + \int_{\Omega_1} m_2^*(B_{\omega_1})\, dm_1^*\right] =$$

$$= g^{-1}\left[\int_{\Omega_1} m_2^*(A_{\omega_1})\, dm_1^*\right] \mathrm{T}$$

$$g^{-1}\left[\int_{\Omega_1} m_2^*(B_{\omega_1})\, dm_1^*\right] =$$

$$= m(A)\, \mathrm{T}\, m(B)$$

Para demostrar que m es σ-T-descomponible, usamos el teorema 3 y demostramos que m es continua inferiormente. Sea una sucesión $(A_n) \uparrow A$ en $\mathcal{A}_1 \times \mathcal{A}_2$. Entonces se cumple $(A_n)_{\omega_1} \uparrow A_{\omega_1}$; $(A_n)_{\omega_2} \uparrow A_{\omega_2}$. Tendremos:

$$\lim_{n\to\infty} m(A_n) = \lim_{n\to\infty} g^{-1}\left[\int_{\Omega_1} m_2^*(A_n)_{\omega_1}\, dm_1^*\right] \tag{4}$$

Por el teorema de la convergencia monótona para medidas σ-aditivas, al ser $m_2^*(A_{\omega_1})$ una función $\mathcal{A}_1$-medible, se cumple:

$$\lim_{n\to\infty} \int_{\Omega_1} m_2^*(A_n)_{\omega_1}\, dm_1^* = \int_{\Omega_1} m_2^*(A_{\omega_1})\, dm_1^*$$

Por ser g^{-1} función continua, tendremos:

$$\lim_{n\to\infty} g^{-1}\left[\int_{\Omega_1} m_2^*(A_n)_{\omega_1}\, dm_1^*\right] =$$

$$= g^{-1}\left[\int_{\Omega_1} m_2^*(A_{\omega_1})\, dm_1^*\right] = m(A)$$

Luego m es continua inferiormente.

Por tanto, es m una medida σ-T-descomponible NSA.

2) La demostración de que $m = M$ es evidente pues si es $A \in \mathcal{A}_1 \times \mathcal{A}_2$, se tiene:

$$m(A) = g^{-1}\left[\int_{\Omega_1} m_2^*(A_{\omega_1})\, dm_1^*\right] =$$

$$= g^{-1}\left[\int_{\Omega_2} m_1^*(A_{\omega_2})\, dm_2^*\right] = M(A)$$

□

Definición 3 *Llamaremos* $m = m_1 \times m_2$

Corolario 3 *Si son* m_1^*, m_2^*, m^* *y* M^* *las medidas de probabilidad asociadas respectivamente a* m_1, m_2, m *y* M *del teorema anterior por la función* g, *se verifica:*

$$m^* = M^* = m_1^* \times m_2^*$$

Demostración: Sean $A \in \mathcal{A}_1$ y $B \in \mathcal{A}_2$, se tiene:

$$m(A \times B) = g^{-1}\left[\int_{\Omega_1} m_2^*(A \times B)_{\omega_1}\, dm_1^*\right] =$$

$$= g^{-1}\left[\int_{\Omega_1} I_A \cdot m_2^*(B)\, dm_1^*\right] = g^{-1}\left(m_1^*(A) \cdot m_2^*(B)\right)$$

Entonces $m^*(A \times B) = (g \circ m)(A \times B) = (g \circ g^{-1})\left(m_1^*(A) \cdot m_2^*(B)\right) = m_1^*(A) \cdot m_2^*(B)$.

Igualmente obtenemos, por un procedimiento análogo:

$$M(A \times B) = g^{-1}\left(m_1^*(A) \cdot m_2^*(B)\right)$$

Por tanto, también $M^*(A \times B) = m_1^*(A) \cdot m_2^*(B)$.

Entonces, $m^* = M^* = m_1^* \times m_2^*$ □

5.3 Teorema de Fubini para medidas σ-T-descomponibles

Teorema 8 (Fubini) *Sean* m_1 *y* m_2 *medidas* σ-*T-descomponibles NSA en los espacios* $(\Omega_1, \mathcal{A}_1)$ *y* $(\Omega_2, \mathcal{A}_2)$ *respectivamente y sea:*

$$f : \Omega_1 \times \Omega_2 \longrightarrow [0,1] \quad \textit{función } \mathcal{A}_1 \times \mathcal{A}_2\textit{-medible}$$

Entonces si es $m = m_1 \times m_2$ *la medida* σ-*T-descomponible definida anteriormente, se cumple:*

$$\int_{\Omega_1 \times \Omega_2} f\, dm = \int_{\Omega_1}\left[\int_{\Omega_2} f_{\omega_1}\, dm_2\right] dm_1 =$$

$$= \int_{\Omega_2}\left[\int_{\Omega_1} f_{\omega_2}\, dm_1\right] dm_2$$

Demostración: Se tiene:

$$\int_{\Omega_1 \times \Omega_2} f\, dm = g^{-1}\left[\int_{\Omega_1 \times \Omega_2} f^*\, dm^*\right] \qquad (5)$$

Aplicando el teorema de Fubini para medidas aditivas en la integral que aparece en la expresión anterior, obtenemos:

$$g^{-1}\left[\int_{\Omega_1 \times \Omega_2} f^*\, dm^*\right] =$$

$$= g^{-1}\left[\int_{\Omega_1}\left[\int_{\Omega_2} f_{\omega_1}^*\, dm_2^*\right] dm_1^*\right] =$$

$$= g^{-1}\left[\int_{\Omega_1} (g \circ g^{-1})\left[\int_{\Omega_2} f_{\omega_1}^*\, dm_2^*\right] dm_1^*\right] =$$

$$= g^{-1}\left[\int_{\Omega_1} g\left[\int_{\Omega_2} f_{\omega_1}\, dm_2\right] dm_1^*\right] =$$

$$= \int_{\Omega_1}\left[\int_{\Omega_2} f_{\omega_1}\, dm_2\right] dm_1$$

También si en la expresión (5) aplicamos el teorema de Fubini de la forma:

$$\int_{\Omega_1 \times \Omega_2} f^*\, dm^* = \int_{\Omega_2}\left[\int_{\Omega_1} f_{\omega_2}^*\, dm_1^*\right] dm_2^*$$

Mediante un proceso similar al anterior, obtenemos:

$$\int_{\Omega_1 \times \Omega_2} f\, dm = \int_{\Omega_2}\left[\int_{\Omega_1} f_{\omega_2}\, dm_1\right] dm_2$$

□

Referencias

[1] Kruse, Rudolf: *Fuzzy Integrals and Conditional Fuzzy Measures.* Fuzzy sets and Systems 10 (1983) 309-313.

[2] Ling, C.H.: *Representation of associative functions.* Publ. Math. Debrecen 12, 189-212.

[3] Loeve, Michel: *Teoría de la Probabilidad.*Ed. Tecnos. Madrid.

[4] Schweizer, B. and Sklar, A.: *Associative functions and statistical triangle inequalities.* Pub. Math. Debrecen 8, 169-186 (1961).

[5] Suárez García, F. y Gil Alvarez, P.: *Two Familys of fuzzy integrals.* Fuzzy sets and Systems 18 (1) (1986) 67-81.

[6] Sugeno, M: *Theory of fuzzy integrals and its applications.* Thesis, Tokyo Institute of Tecnology (1974)

[7] Weber, Siegfried: $\perp$-*Descomposable Measures and Integrals for Archimedean t-Conorms* $\perp$ J. Math. Anal. Appl. 101 (1984) 114-138.

Experimentación Numérica y Comparación de Integrales Difusas

Manuel Jorge Bolaños Carmona
Dpto. Ciencias de la Computación e I.A.
E.T.S. Informática
Universidad de Granada
e-mail: mjbc@robinson.ugr.es

Luis Daniel Hernández Molinero
Dpto. Informática y Sistemas
Facultad de Informática
Universidad de Murcia
e-mail: ldaniel@dif.um.es

Antonio Salmerón Cerdán
Dpto. Estadística y Matemática Aplicada
Facultad de Ciencias Experimentales
Universidad de Almería
e-mail: asalmero@ualm.es

Resumen

Las integrales de Sugeno y de Choquet han sido ampliamente estudiadas en la literatura desde un punto de vista teórico. Sin embargo, del comportamiento de estos funcionales sólo se tiene una idea general. En este trabajo se reflejan los resultados de una comparación numérica que pretende servir como referencia en la mejor comprensión y utilización de ambas integrales.

Palabras claves: *integral de Choquet, integral de Sugeno, medida difusa*

1 Introducción

La integral difusa de Sugeno y la integral difusa de Choquet, como funcionales que permiten valorar una función acotada sobre una medida difusa, han sido estudiadas ampliamente en la literatura. En efecto, desde su definición en 1974, la integral de Sugeno (también llamada Fuzzy Expected Value FEV cuando se define sobre probabilidades) ha sido estudiada en diferentes contextos [8, 6], generalizada [11, 10, 5] o caracterizada [9, 4, 12] por numerosos autores. Así mismo, y a partir de la definición original de Choquet en 1953 [3] la esperanza monótona ME ha sido estudiada en el contexto de las medidas difusas [1, 7] como una generalización de la esperanza matemática habitual sobre medidas de probabilidad. También se dispone de estudios que comparan y relacionan los dos funcionales citados [2] (un exhaustivo número de trabajos puede encontrarse en la bibliografía de los citados).

Sin embargo, y a pesar de la abundancia de trabajos teóricos, del comportamiento de estos funcionales sólo se tiene una idea general que interpretaría la integral de Sugeno como una "mediana ponderada" y a la integral de Choquet como una "media deformada"; en la práctica, la comparación de ambos funcionales está aún limitada a la llamada cota de Sugeno para probabilidades, que fue generalizada por Bolaños, de Campos y González para cualquier medida difusa [2]. En este trabajo se realiza una comparación numérica que sirva de referencia en la utilización de ambas integrales y en futuros desarrollos.

2 Definiciones y resultados conocidos

Sobre un conjunto finito $X = \{x_1, x_2, \ldots, x_n\}$, se define como medida difusa una función [1] $g : \mathcal{P}(X) \longrightarrow [0,1]$ que verifique:

1. $g(\phi) = 0$
2. $g(X) = 1$
3. $A \subset B \Rightarrow g(A) \leq g(B) \quad A, B \in \mathcal{P}(X)$

Dado un espacio de medida $(X, \mathcal{P}(X), g)$, se define la integral de Sugeno de una función $h : X \longrightarrow [0,1]$ como:

$$S(h) = \int h \circ g = \bigvee_{i=1}^{n} [a_i \wedge g(H_{a_i})]$$

donde H_a es el a-corte de h $(H_a = \{x \in X | h(x) \geq a\})$ y $a_i = h(x_i)$ para cada $x_i \in X$.

En las mismas condiciones, la integral de Choquet de h se define como:

$$C(h) = \int h \circ g = \int g(H_a)da = \sum_{i=1}^{n} (a_i - a_{i-1}) g(H_{a_i})$$

con $a_0 = 0$.

La relación entre ambos funcionales viene dada por la desigualdad [2]:

$$|S(h) - C(h)| \leq \frac{1}{4}$$

válida para todo g y para todo h.

[1] $\mathcal{P}(X)$ denota el conjunto de las partes de X

3 Experimentación

Para comparar y relacionar el valor de las dos integrales sobre diversas medidas difusas, se ha procedido de la siguiente forma:

1. Se considera un conjunto X con 10 elementos, $X = \{x_1, \ldots, x_{10}\}$
2. Se define una medida difusa g fija.
3. Se genera aleatoriamente una muestra de 1000 funciones $h : X \longrightarrow [0, 1]$ utilizando un generador congruencial lineal de números aleatorios.
4. Para cada función h se obtienen los valores de $S(h)$ y $C(h)$.
5. Se realiza un análisis comparativo y de correlación entre las 1000 parejas de valores obtenidas.

Este proceso se ha realizado para diversas medidas difusas:

1. Una medida de probabilidad uniforme ($g_1(x_i) = \frac{1}{10}$ para cada $x_i \in X$).
2. Una medida de probabilidad g_2 no uniforme (con dos valores 0.25, tres valores 0.10 y cinco valores 0.04).
3. Una medida difusa g_3 expansiva cercana a una probabilidad (las medidas de los conjuntos aumentan un 10 por ciento sobre las medidas del punto 1). Más detalladamente:
   Sea g_1 la medida considerada en el experimento 1. Dada una función h, para cualquier a-corte de h, $H_a = \{x_{a_1}, \ldots, x_{a_m}\}$, se calcula la medida $g_3(H_a)$ de la siguiente manera:

$$t = \sum_{i=1}^{m} g_1(x_{a_i}) + \left(\sum_{i=1}^{m} g_1(x_{a_i})\right) \times \frac{1}{10}$$

$$g_3(H_a) = \begin{cases} t & si \quad t \leq 1 \\ 1 & si \quad t > 1 \end{cases}$$

   Puede comprobarse que $g_3(\phi) = 0$ y $g_3(X) = 1$.

4. Una medida difusa g_4 lejana a una probabilidad (las medidas de los conjuntos aumentan un 50 por ciento sobre las medidas del punto 1). Es decir:
   Sea g_1 la medida considerada en el experimento 1. Dada una función h, para cualquier a-corte de h, $H_a = \{x_{a_1}, \ldots, x_{a_m}\}$, se calcula la medida $g_4(H_a)$ de la siguiente manera:

$$t = \sum_{i=1}^{m} g_1(x_{a_i}) + \left(\sum_{i=1}^{m} g_1(x_{a_i})\right) \times \frac{5}{10}$$

$$g_4(H_a) = \begin{cases} t & si \quad t \leq 1 \\ 1 & si \quad t > 1 \end{cases}$$

   Puede comprobarse que $g_4(\phi) = 0$ y $g_4(X) = 1$.

5. Una medida difusa g_5 restrictiva cercana a una probabilidad (las medidas de los conjuntos disminuyen un 10 por ciento sobre las medidas del punto 1). Es decir:
   Sea g_1 la medida considerada en el experimento 1. Dada cualquier función h, para cualquier a-corte de h, $H_a = \{x_{a_1}, \ldots, x_{a_m}\}$, se calcula la medida $g_5(H_a)$ de la siguiente manera:

$$t = \sum_{i=1}^{m} g_1(x_{a_i}) - \left(\sum_{i=1}^{m} g_1(x_{a_i})\right) \times \frac{1}{10}$$

$$g_5(H_a) = \begin{cases} 0 & si \quad H_a = \phi \\ 1 & si \quad H_a = X \\ t & en \quad otro \quad caso \end{cases}$$

6. Una medida difusa g_6 restrictiva lejana a una probabilidad (las medidas de los conjuntos disminuyen en un 50 por ciento sobre las medidas del punto 1). Es decir:
   Sea g_1 la medida considerada en el experimento 1. Dada cualquier función h, para cualquier a-corte de h, $H_a = \{x_{a_1}, \ldots, x_{a_m}\}$, se calcula la medida $g_6(H_a)$ de la siguiente manera:

$$t = \sum_{i=1}^{m} g_1(x_{a_i}) - \left(\sum_{i=1}^{m} g_1(x_{a_i})\right) \times \frac{5}{10}$$

$$g_6(H_a) = \begin{cases} 0 & si \quad H_a = \phi \\ 1 & si \quad H_a = X \\ t & en \quad otro \quad caso \end{cases}$$

Además se han considerado cuatro λ-medidas de Sugeno (g_7, g_8, g_9, g_{10}) dadas por la siguiente expresión genérica:

Dados $A, B \subset X$, $A \cap B = \phi$ y $\lambda > -1$,

$$g_\lambda(A \cup B) = g_\lambda(A) + g_\lambda(B) + \lambda g_\lambda(A) g_\lambda(B)$$

con $g_7(x_i) = g_8(x_i) = g_9(x_i) = g_{10}(x_i) = \frac{1}{10}$ para todo $x_i \in X$, y tomando $\lambda = -0.9$ para g_7, $\lambda = -0.5$ para g_8, $\lambda = 2$ para g_9 y $\lambda = 5$ para g_{10}.

4 Resultados

Para la primera medida (Tabla 1) se observa que ambas integrales proporcionan un promedio muy similar (diferencias no significativas) con mayor variabilidad de la integral de Choquet.

En cuanto a la segunda medida (Tabla 2), al igual que en el caso anterior, ambas integrales proporcionan un promedio muy similar con mayor variabilidad de la integral de Choquet.

Para la tercera medida (Tabla 3) el resultado muestra una media para la integral de Choquet significativamente superior ($p < 0.001$) a la correspondiente a la integral de Sugeno. Se mantiene una variabilidad ligeramente mayor en la integral de Choquet.

Respecto a la cuarta medida (Tabla 4) el resultado muestra una media para la integral de Choquet ampliamente superior ($p < 0.001$)a la correspondiente a la integral de Sugeno. Se mantiene una variabilidad ligeramente mayor en la integral de Choquet.

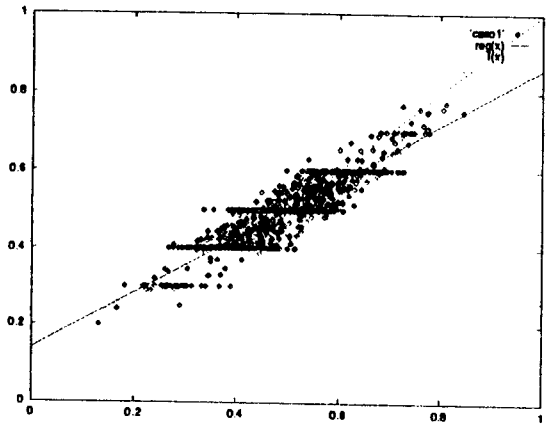


Figura 1: Integral de Sugeno (ordenadas) frente a la de Choquet (abcisas) para g_1

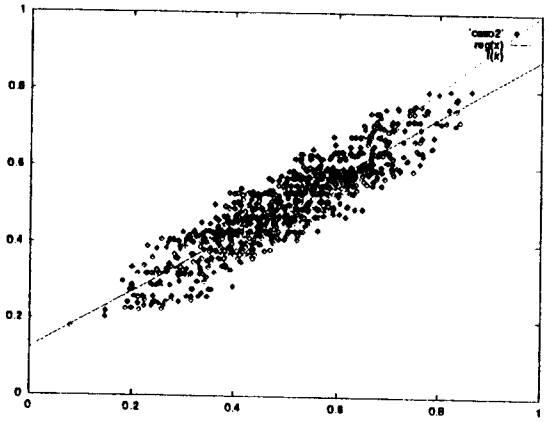


Figura 2: Integral de Sugeno (ordenadas) frente a la de Choquet (abcisas) para g_2

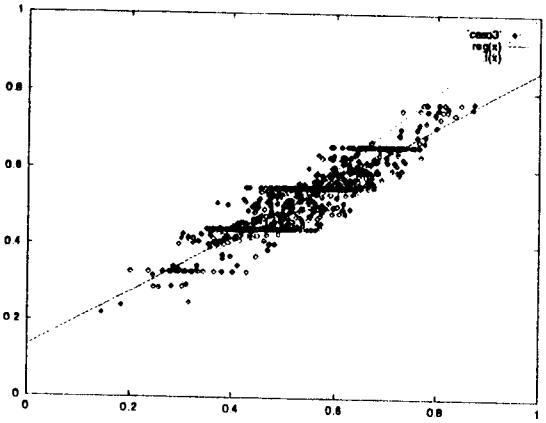


Figura 3: Integral de Sugeno (ordenadas) frente a la de Choquet (abcisas) para g_3

	Media	Desviación típica
S(h)	0.500099	0.087413
C(h)	0.500466	0.109700
C(h)-S(h)	0.000367	0.048079
Coef. de correl. C(h), S(h)		0.905369
Regres.	b=0.721433	a=0.139047

Tabla 1: Resultados para la primera medida.

	Media	Desviación típica
S(h)	0.499102	0.113516
C(h)	0.497244	0.135915
C(h)-S(h)	-0.001858	0.057919
Coef. de correl. C(h), S(h)		0.907545
Regres.	b=0.757980	a=0.122201

Tabla 2: Resultados para la segunda medida.

Con la quinta medida (Tabla 5) el resultado muestra una media para la integral de Sugeno significativamente superior ($p < 0.001$) a la correspondiente a la integral de Choquet. Se mantiene una variabilidad ligeramente mayor en la integral de Choquet.

Para la sexta medida (Tabla 6) se observa una media para la integral de Sugeno ampliamente superior ($p < 0.001$) a la correspondiente a la integral de Choquet, con variabilidad ligeramente mayor de la integral de Choquet.

Por lo que se refiere a las g_λ-medidas, de nuevo la integral de Sugeno aparece significativamente superior para las medidas restrictivas (g_7, g_8) y significativamente inferior para las expansivas (g_9, g_{10}), siempre con $p < 0.001$; llama la atención en estos casos que la variabilidad de la integral de Sugeno es ahora algo mayor y que los coeficientes de correlación son claramente menores, debido a una mayor concentración de los resultados.

Si se observan las rectas de regresión ajustadas en cada caso, aparece claramente reafirmada la menor sensibilidad de la integral de Sugeno y su mayor cercanía al valor medio teórico de las funciones, que es 0.50. En general, puede afirmarse que la integral de Choquet es más sensible a modificaciones en la medida con respecto a una referencia uniforme y también algo más sensible a los valores de la función que se integra; la integral de Sugeno actua así como una me-

	Media	Desviación típica
S(h)	0.523650	0.091162
C(h)	0.540660	0.115091
C(h)-S(h)	0.017010	0.049599
Coef. de correl. C(h), S(h)		0.910052
Regres.	b=0.720841	a=0.133920

Tabla 3: Resultados para la tercera medida.

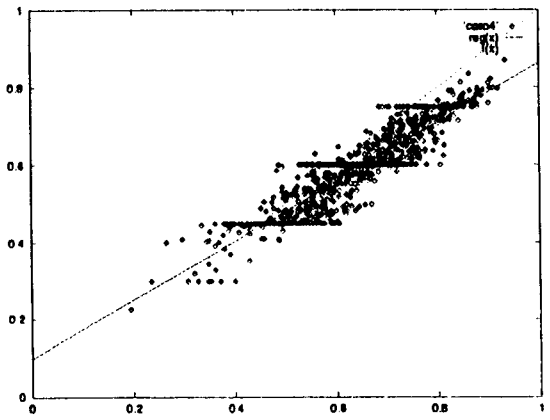


Figura 4: Integral de Sugeno (ordenadas) frente a la de Choquet (abcisas) para g_4

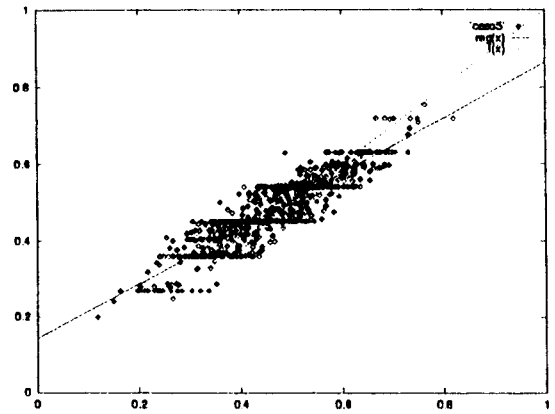


Figura 5: Integral de Sugeno (ordenadas) frente a la de Choquet (abcisas) para g_5

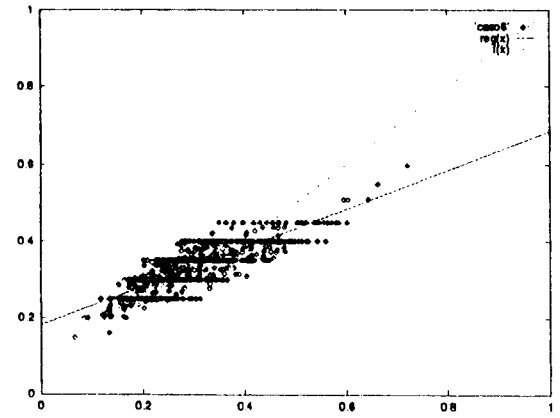


Figura 6: Integral de Sugeno (ordenadas) frente a la de Choquet (abcisas) para g_6

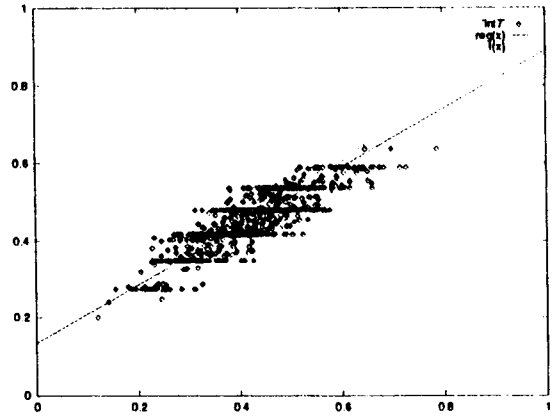


Figura 7: Integral de Sugeno (ordenadas) frente a la de Choquet (abcisas) para g_7

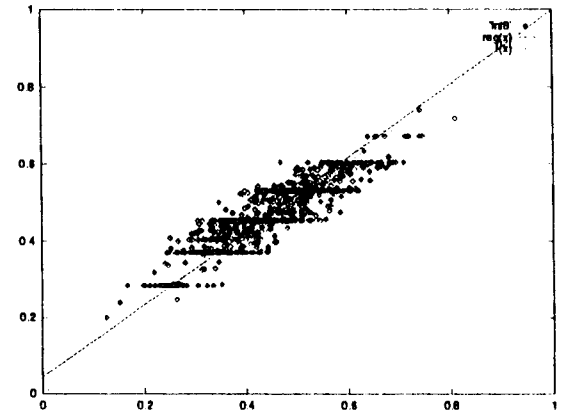


Figura 8: Integral de Sugeno (ordenadas) frente a la de Choquet (abcisas) para g_8

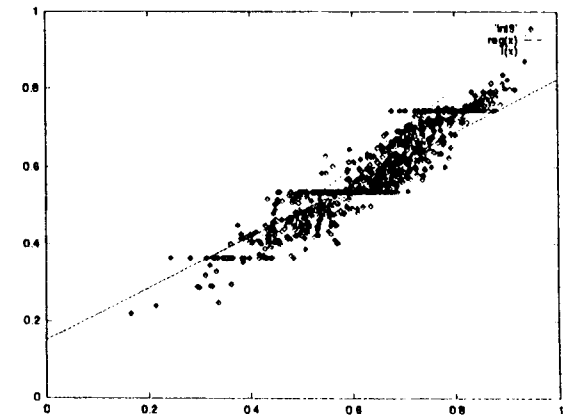


Figura 9: Integral de Sugeno (ordenadas) frente a la de Choquet (abcisas) para g_9

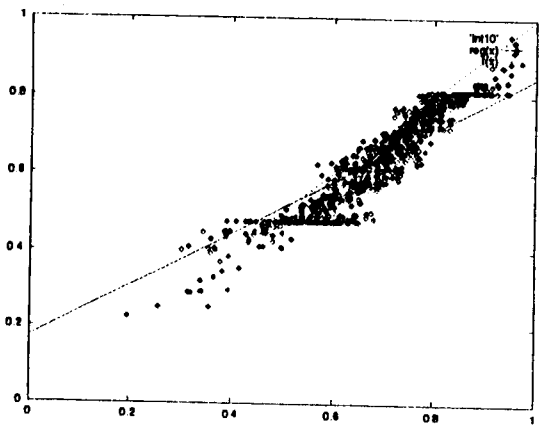


Figura 10: Integral de Sugeno (ordenadas) frente a la de Choquet (abcisas) para g_{10}

	Media	Desviación típica
S(h)	0.593966	0.101550
C(h)	0.646925	0.120577
C(h)-S(h)	0.052959	0.050641
Coef. de correl. C(h), S(h)		0.910064
Regres.	b=0.766457	a=0.098126

Tabla 4: Resultados para la cuarta medida.

	Media	Desviación típica
S(h)	0.475613	0.084324
C(h)	0.460272	0.104752
C(h)-S(h)	-0.015342	0.047521
Coef. de correl. C(h), S(h)		0.895794
Regres.	b=0.721104	a=0.143710

Tabla 5: Resultados para la quinta medida.

	Media	Desviación típica
S(h)	0.333643	0.055272
C(h)	0.299494	0.090752
C(h)-S(h)	-0.034149	0.054687
Coef. de correl. C(h), S(h)		0.827370
Regres.	b=0.503905	a=0.182726

Tabla 6: Resultados para la sexta medida.

	Media	Desviación típica
S(h)	0.450682	0.100068
C(h)	0.417336	0.077592
C(h)-S(h)	-0.033346	0.049804
Coef. de correl. C(h), S(h)		0.586093
Regres.	b=0.755865	a=0.135233

Tabla 7: Resultados para la séptima medida.

	Media	Desviación típica
S(h)	0.474461	0.102634
C(h)	0.451082	0.081674
C(h)-S(h)	-0.023380	0.047924
Coef. de correl. C(h), S(h)		0.758637
Regres.	b=0.953327	a=0.044433

Tabla 8: Resultados para la octava medida.

	Media	Desviación típica
S(h)	0.575307	0.138837
C(h)	0.629780	0.123005
C(h)-S(h)	0.054473	0.047231
Coef. de correl. C(h), S(h)		0.594171
Regres.	b=0.670650	a=0.152945

Tabla 9: Resultados para la novena medida.

	Media	Desviación típica
S(h)	0.638784	0.137876
C(h)	0.690128	0.139022
C(h)-S(h)	0.051344	0.041511
Coef. de correl. C(h), S(h)		0.679747
Regres.	b=0.674144	a=0.173539

Tabla 10: Resultados para la décima medida.

dida basada en el orden, y por tanto más estable mientras que la integral de Choquet es similar a una media aritmética habitual y en consecuencia más sensible y variable (nótese que las diferencias con respecto al valor de 0.50 son muy superiores a las de la integral de Sugeno cuando la medida de probabilidad se "deforma").

5 Conclusiones

De los datos obtenidos se desprende que, reafirmando la forma de cálculo de ambas integrales, la integral de Choquet puede verse como una media ponderada, con la variabilidad y sensibilidad propias de este tipo de medidas, mientras que la integral de Sugeno se asemeja claramente a una generalización del concepto de mediana. Es recomendable, por tanto, la utilización de la integral de Sugeno cuando se pretenda medir el "tamaño" de la función, en el sentido de las coincidencias entre medidas y valores de la función, mientras que la de Choquet es más aconsejable si nos interesamos por la determinación de los valores medios en el sentido aritmético del término.

Por tanto, parece deducirse claramente de los resultados la complementariedad entre ambas definiciones, lo que permitirá la configuración de sistemas teóricos y aplicaciones prácticas bien diferenciadas.

Referencias

[1] Bolaños, M.J, Lamata, M.T., Moral, S. Extensión de medidas difusas usando la esperanza monótona. *Stochastica* XI-2,3, pp.75-92, 1987.

[2] Bolaños, M.J, de Campos, L.M., González, A. Convergence properties of the monotone expectation and its application to the extension of fuzzy measures. *Fuzzy Sets and Systems* 33, pp.201-213. 1989.

[3] Choquet, G. Theory of capacities. *Ann. Ins. Fourier* (Grenoble) 5, pp.131-293. 1953/54.

[4] De Cooman, G., Kerre, E.E. Possibility theory: an integral theoretic approach. *Fuzzy Sets and Systems* 46, pp.287-299. 1992

[5] Friedman, M., Schneider, M., Kandel, A. The use of weighted fuzzy expected value (WFEV) in fuzzy expert systems. *Fuzzy Sets and Systems* 31, pp.37-45. 1989.

[6] Kandel, A. Theory and applications of fuzzy statistics. *In Combining fuzzy imprecision with probabilistic uncertainty in decision making* , 89-112. Lecture notes in Econom. and Math. Systems, 1988.

[7] Murofushi, T., Sugeno, M. An interpretation of fuzzy measures and the Choquet integral w.r.t. a fuzzy measure. *Fuzzy Sets and Systems* 29, pp.201-227. 1989.

[8] Ralescu, D., Adams, G. The fuzzy integral. *J. Math. Anal. Appl.* 75, pp.562-570. 1980.

[9] Squilante, M., Ventre, A. Representations of the fuzzy integral. *Fuzzy Sets and Systems* 29, pp.165-169. 1989.

[10] Suarez, F., Gil, P. Two families of fuzzy integrals. *Fuzzy Sets and Systems* 18, 67-81. 1986.

[11] Weber, S. Decomposable measures and integrals for Archimedian t-conorms. *J. Math. Anal. Appl.* 101, pp.114-138. 1984.

[12] Zhang, W. Fuzzy contactability and fuzzy variables. *Fuzzy Sets and Systems* 8, pp.81-92. 1982.

Las ϵ-Particiones Difusas

Susana Montes, Mª Jesús López-Palomo y Pedro Gil
Facultad de Ciencias, Dpto. de Matemáticas.
Universidad de Oviedo
C/Calvo Sotelo s/n, 33007 Oviedo
Tfno: (98) 510 31 90, Fax: (98) 510 33 56
e-mail: smr@pinon.ccu.uniovi.es

Junio 15, 1995

Resumen

El objetivo de este trabajo es definir un nuevo concepto de partición difusa (ϵ-partición) de un conjunto difuso en un referencial finito. Destacamos para ello las ideas esenciales de las particiones nítidas, en lugar de los aspectos operativos empleados por Ruspini (R-partición). La ϵ-partición resulta más intuitiva que la R-partición desde el punto de vista de los análisis de clasificación y conglomerados. Se caracteriza la ϵ-partición más próxima a una R-partición dada en el sentido de la mínima distancia de Hamming. Se señalan otras propiedades de las ϵ-particiones.

Palabras clave: *Conjuntos difusos, Partición difusa, Distancia de Hamming.*

1 Introducción: R-partición difusa

La definición de Ruspini de partición difusa ([1]) corresponde a una colección de subconjuntos difusos, cuya suma de funciones de pertenencia en cada punto es la función de pertenencia en ese punto del conjunto que queremos particionar. Formalmente,

1.1 Definición

La familia de subconjuntos $\{\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_n\}$ de un referencial discreto X es una **R-partición** de un conjunto difuso $\widetilde{A}$ si y sólo si se cumple que:

$$\sum_{i=1}^{n} \mu_{\widetilde{A}_i}(x) = \mu_{\widetilde{A}}(x) \quad \forall x \in X$$

Esta definición se puede aplicar en particular a cualquier conjunto nítido A, teniendo en cuenta que en este caso la función de pertenencia vale 1 ó 0, dependiendo de si el punto considerado pertenece o no al conjunto A, en todos los puntos del referencial X.

La definición de R-partición anteriormente dada, muy interesante desde el punto de vista operativo, presenta el inconveniente de que conjuntos "muy parecidos" o incluso identicos, verifican las condiciones para ser una partición.

1.2 Ejemplos

1. Sea X un referencial discreto y sea el conjunto difuso $\widetilde{A}$ con función de pertenencia $\mu_{\widetilde{A}}(x) = 1/6 \ \forall x \in X$. Si consideramos $\{\widetilde{A}, \widetilde{A}, \widetilde{A}, \widetilde{A}, \widetilde{A}, \widetilde{A}\}$, esta familia de subconjuntos de X es una R-partición del referencial, pero este ejemplo muestra, cómo conjuntos casi vacíos (podría considerarse un conjunto con función de pertenencia 1/100 en todos los puntos, repetido 100 veces) constituyen una partición del total.

2. Otro ejemplo interesante consiste en considerar un conjunto $\widetilde{A}$ con función de pertenencia $\mu_{\widetilde{A}}(x) = 1/2 \quad \forall x \in X$, junto con su complementario $\widetilde{A}^c$ que cumple evidentemente que $\mu_{\widetilde{A}^c}(x) = 1/2 \quad \forall x \in X$. Ahora bien, es inmediato que $\{\widetilde{A}, \widetilde{A}^c\}$ son una R-partición del referencial X, pues se cumple la condición

$$\mu_{\widetilde{A}}(x) + \mu_{\widetilde{A}^c}(x) = 1 = \mu_X(x) \quad \forall x \in X$$

   Se tiene así un ejemplo, donde un conjunto claramente distinto de X y él mismo, son una R-partición del referencial, lo cual vuelve a parecer totalmente contrario a la idea intuitiva de partición, como clasificación de un conjunto en clases sin elementos comunes.

3. Aunque los dos ejemplos anteriores concluían en particiones formadas por un conjunto repetido un número suficiente de veces, no sólo en estos casos las particiones de Ruspini no respetan la idea de partición, por ejemplo los conjuntos $\widetilde{A}$ y $\widetilde{B}$ con funciones de pertenencia

$$\mu_{\widetilde{A}}(x) = 0.55 \quad \forall x \in X$$

$$\mu_{\widetilde{B}}(x) = 0.45 \quad \forall x \in X$$

   son una R-partición del total, y sin embargo la intersección de ambos conjuntos es un conjunto claramente distinto del vacío.

Surge así la idea de reformar esta definición de partición difusa, extendiendo las particiones nítidas habituales a particiones difusas que sigan recogiendo las ideas de recubrimiento del total por subconjuntos disjuntos.

Pedirle a la partición exactamente esto quizá esté fuera del espíritu fuzzy, pero pueden suavizarse estas restricciones sustituyéndolas por otras que recojan a su vez la borrosidad que existe en esa familia de conjuntos, pidiendo que la unión sea "casi" el total y la intersección de cada dos de ellos fuese "casi" el vacío.

2 Partición difusa

2.1 Definición

Sea $\widetilde{A}$ un conjunto difuso sobre un referencial X. Se dice que la familia $\{\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_n\}$ de subconjuntos de $\widetilde{A}$ es una **ϵ-partición difusa** de $\widetilde{A}$ si verifican que:

$$(\widetilde{A}_i \cap \widetilde{A}_j)_{\overline{\alpha}} = \emptyset \quad \forall i,j = 1,2,\ldots,n \quad \forall \alpha \geq \epsilon$$

$$(\bigcup_{i=1}^{n} \widetilde{A}_i)_\alpha = (\widetilde{A})_\alpha \quad \forall \alpha \leq 1-\epsilon$$

siendo $(\widetilde{D})_\alpha$ el α-corte débil y $(\widetilde{D})_{\overline{\alpha}}$ el α-corte fuerte del conjunto $\widetilde{D}$ según las definiciones tradicionales de estos conceptos ([2]), y siendo las operaciones de unión e intersección las clásicas del máximo y el mínimo de las funciones de pertenencia ([3]).

Por lo tanto, las anteriores condiciones se pueden reescribir como:

$$\{x \in X / \mu_{\widetilde{A}_i \cap \widetilde{A}_j}(x) > \alpha\} = \emptyset \quad \forall i,j = 1,2,\ldots,n \quad \forall \alpha \geq \epsilon$$

$$\{x \in X / \mu_{\bigcup_{i=1}^{n} \widetilde{A}_i}(x) \geq \alpha\} = \{x \in X / \mu_{\widetilde{A}}(x) \geq \alpha\} \quad \forall \alpha \leq 1-\epsilon$$

Es inmediato que tales condiciones equivalen a las siguientes, más ilustrativas de la importancia del valor de ϵ en esta definición de partición difusa:

$$\mu_{\widetilde{A}_i \cap \widetilde{A}_j}(x) \leq \epsilon \quad \forall i,j = 1,2,\ldots,n \quad i \neq j \quad \forall x \in X$$

$$\mu_{\bigcup_{i=1}^{n} \widetilde{A}_i}(x) \geq 1-\epsilon \quad \forall x \in X \;/\; \mu_{\widetilde{A}}(x) \geq 1-\epsilon$$

$$\mu_{\bigcup_{i=1}^{n} \widetilde{A}_i}(x) = \mu_{\widetilde{A}}(x) \quad \forall x \in X \;/\; \mu_{\widetilde{A}}(x) < 1-\epsilon$$

En el caso de un conjunto nítido A dicha definición expresa el cumplimiento de las condiciones siguientes:

$$\mu_{\widetilde{A}_i \cap \widetilde{A}_j}(x) \leq \epsilon \quad \forall i,j = 1,2,\ldots,n \quad i \neq j \quad \forall x \in X$$

$$\mu_{\bigcup_{i=1}^{n} \widetilde{A}_i}(x) \geq 1-\epsilon \quad \forall x \in A$$

que define una partición de un conjunto nítido, formada por subconjuntos difusos. Se puede considerar en particular, el caso en el cual A es el propio referencial X.

2.2 Ejemplo

Una ϵ-partición del referencial X, tomando como ϵ el valor 0.1, sería por ejemplo $\{\widetilde{A}^*, \widetilde{B}^*\}$ donde $\widetilde{A}^*$ y $\widetilde{B}^*$ son subconjuntos difusos del referencial con funciones de pertenencia:

$$\mu_{\widetilde{A}^*}(x) = 0.9 \qquad \forall x \in X$$

$$\mu_{\widetilde{B}^*}(x) = 0.1 \qquad \forall x \in X$$

2.3 Observaciones

- Se considerará $\epsilon < 0.5$ para evitar problemas como los descritos para la R-partición, es decir, que un conjunto repetido n veces sea una partición de n elementos de otro conjunto. En problemas reales, el valor de ϵ deberá ser fijado por el investigador y de este modo es subjetivo, pudiendo incluso llegar a exigir distinta "libertad" para la unión y la intersección, lo que daría lugar a la definición de las δ-ϵ-particiones.

- Cuando ϵ vale cero, estamos exigiendo que los conjuntos que forman la partición sean un verdadero recubrimiento (en el sentido clásico) formado por conjuntos disjuntos dos a dos, del conjunto particionado. Por lo tanto, cuando éste último sea un conjunto nítido, con ϵ igual a cero, la definición dada no es más que la habitual de partición, formada ésta, como es lógico por subconjuntos nítidos. Se trata pues de una generalización de tal concepto.

- Las R-particiones no tienen que descartarse por el uso de las ϵ-particiones; de hecho algunas de estas últimas son verdaderas R-particiones, como por ejemplo las particiones atómicas de un conjunto difuso, formadas por tantos subconjuntos como elementos tenga el conjunto a particionar $\widetilde{A}$, donde la función de pertenencia de cada uno de estos subconjuntos es:

$$\mu_{\widetilde{A}_i}(x) = \begin{cases} 0 & \text{si } x \neq x_i \\ \mu_{\widetilde{A}}(x) & \text{si } x = x_i \end{cases} \qquad \forall i = 1,2,\ldots,n$$

  donde $\widetilde{A} = \{x_1, x_2, \ldots, x_n\}$. En el caso de que $\widetilde{A}$ sea nítido, la partición anterior, no es más que la formada por cada uno de sus puntos, considerados como conjuntos unitarios.

3 Distancia entre particiones

Una vez que hemos dado esta nueva idea de partición difusa, y debido a que hay gran cantidad de trabajos hechos sobre la idea de R-partición, parece conveniente poder encontrar la ϵ-partición que menos dista de una partición de Ruspini dada. Para ello es preciso poder medir la distancia entre dos particiones de un conjunto y la medida de esta proximidad, va a basarse en la definición de la distancia de Hamming ([3]) entre dos conjuntos difusos, cuya expresión es la siguiente,

$$d_H(\widetilde{A}, \widetilde{B}) = \sum_{x \in X} |\mu_{\widetilde{A}}(x) - \mu_{\widetilde{B}}(x)|$$

Se define así la distancia entre dos particiones como sigue,

3.1 Definición

Sean $\Pi_{\widetilde{A}} = \{\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_n\}$ y $\Pi_{\widetilde{B}} = \{\widetilde{B}_1, \widetilde{B}_2, \ldots, \widetilde{B}_n\}$, dos particiones formadas por el mismo número de elementos, dentro de un referencial X. Se define la distancia entre $\Pi_{\widetilde{A}}$ y $\Pi_{\widetilde{B}}$ como:

$$d(\Pi_{\widetilde{A}}, \Pi_{\widetilde{B}}) = \min_{\sigma_{\Pi_{\widetilde{B}}}} \sum_{i=1}^{n} d_H(\widetilde{A}_i, \widetilde{B}_i)$$

$$= \min_{\sigma_{\Pi_{\widetilde{B}}}} \sum_{i=1}^{n} \sum_{x \in X} |\mu_{\widetilde{A}_i}(x) - \mu_{\widetilde{B}_i}(x)|$$

donde hemos denotado por $\sigma_{\Pi_{\widetilde{B}}}$ el conjunto de todas las permutaciones de la partición $\Pi_{\widetilde{B}}$.

4 ϵ-Partición mas próxima a una R-partición dada

Una vez que hemos visto el criterio a seguir, para medir la proximidad entre particiones, vamos a calcular la ϵ-partición más próxima, según lo visto en el apartado 3, a una R-partición dada

4.1 Teorema

Sea $\{\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_n\}$ una R-partición de $\widetilde{A}$, la ϵ-partición de n elementos $\{\widetilde{B}_1, \widetilde{B}_2, \ldots, \widetilde{B}_n\}$ más próxima a ésta es la definida como sigue:

Si $\mu_{\widetilde{A}}(x) \geq 1 - \epsilon$ entonces

$$\begin{cases} \mu_{\widetilde{B}_{i_x}}(x) = \begin{cases} \mu_{\widetilde{A}_{i_x}}(x) & \text{si } \mu_{\cup \widetilde{A}_i}(x) \geq 1 - \epsilon \\ 1 - \epsilon & \text{si } \mu_{\cup \widetilde{A}_i}(x) < 1 - \epsilon \end{cases} \\ \mu_{\widetilde{B}_i}(x) = \begin{cases} \epsilon & \text{si } \mu_{\widetilde{A}_i}(x) > \epsilon \\ \mu_{\widetilde{A}_i}(x) & \text{si } \mu_{\widetilde{A}_i}(x) \leq \epsilon \end{cases} \end{cases}$$

donde i_x es el subíndice que cumple que $\mu_{\cup \widetilde{A}_i}(x) = \mu_{\widetilde{A}_{i_x}}(x)$, es decir, $\widetilde{A}_{i_x}$ es el conjunto donde la función de pertenencia alcanza el valor máximo para x, de entre todas las funciones de pertenencia de los conjuntos en la partición, y donde i es cualquier otro subíndice distinto de i_x.

Si $\mu_{\widetilde{A}}(x) < 1 - \epsilon$

$$\begin{cases} \mu_{\widetilde{B}_{i_x}}(x) = \mu_{\widetilde{A}}(x) \\ \mu_{\widetilde{B}_i}(x) = \begin{cases} \epsilon & \text{si } \mu_{\widetilde{A}_i}(x) > \epsilon \\ \mu_{\widetilde{A}_i}(x) & \text{si } \mu_{\widetilde{A}_i}(x) \leq \epsilon \end{cases} \end{cases}$$

donde i_x e i son los descritos en el caso anterior.

Para probar el resultado propuesto, debemos demostrar cuatro cosas:

1. Que la familia de conjuntos de X así definida es una familia de subconjuntos de $\widetilde{A}$.
2. Que el α-corte débil de la unión de los subconjuntos $\{\widetilde{B}_1, \widetilde{B}_2, \ldots, \widetilde{B}_n\}$ coincide con el α-corte débil de $\widetilde{A}$, en los $\alpha \leq 1 - \epsilon$.
3. Que el α-corte fuerte de la intersección de cada dos subconjuntos es el vacío, en los $\alpha \geq \epsilon$.
4. Que es la más próxima a $\{\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_n\}$, medida esta proximidad con la distancia definida en el apartado 3, es decir, es la ϵ-partición de $\widetilde{A}$ que verifica que

$$d(\Pi_{\widetilde{A}}, \Pi_{\widetilde{B}}) \leq d(\Pi_{\widetilde{A}}, \Pi_{\widetilde{C}}) \quad \forall \Pi_{\widetilde{C}} \ \epsilon\text{-partición de } \widetilde{A}$$

Las tres primeras condiciones, demostrarán que $\{\widetilde{B}_1, \widetilde{B}_2, \ldots, \widetilde{B}_n\}$ es una ϵ-partición de $\widetilde{A}$, y la cuarta que es realmente la más próxima a la R-partición $\{\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_n\}$.

4.2 Ejemplo

Considerando la R-partición $\{\widetilde{A}, \widetilde{B}\}$ dada en el ejemplo 1.2(c), si aplicamos el teorema anterior obtenemos que la ϵ-partición más próxima a ésta, es precisamente la del ejemplo 2.2.

5 Algunas propiedades de las ϵ-particiones

Vamos a ver a continuación un par de propiedades de las ϵ-particiones que destacan por su interés en otros campos.

Antes de enunciar tales propiedades, vamos a definir dos conceptos previos:

5.1 Definición

Sea $\Pi_{\widetilde{A}} = \{\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_n\}$ una ϵ-partición de un conjunto difuso $\widetilde{A}$. Se dice que $\Pi^*_{\widetilde{A}} = \{\widetilde{B}_1, \widetilde{B}_2, \ldots, \widetilde{B}_m\}$ es un **refinamiento** de $\Pi_{\widetilde{A}}$ si $\forall i = 1, 2, \ldots, n$ se verifica que

$$\left.\begin{array}{rl} \mu_{\widetilde{B}_{ik} \cap \widetilde{B}_{il}}(x) \leq \epsilon & \forall x \in X \quad \forall k, l = 1, 2, \ldots, r_i \\ \mu_{\cup_{k=1}^{r_i} \widetilde{B}_{ik}}(x) \geq 1 - \epsilon & \forall x \in X \ / \ \mu_{\widetilde{A}_i}(x) \geq 1 - \epsilon \\ \mu_{\cup_{k=1}^{r_i} \widetilde{B}_{ik}}(x) = \mu_{\widetilde{A}_i}(x) & \forall x \in X \ / \ \mu_{\widetilde{A}_i}(x) < 1 - \epsilon \end{array}\right\}$$

es decir, que cada $\widetilde{A}_i$ se puede "recubrir" por elementos "disjuntos" de $\Pi^*_{\widetilde{A}}$.

5.2 Definición

Sean $\Pi_{\widetilde{A}} = \{\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_n\}$ y $\Pi_{\widetilde{B}} = \{\widetilde{B}_1, \widetilde{B}_2, \ldots, \widetilde{B}_m\}$ ϵ-particiones de $\widetilde{A}$ y $\widetilde{B}$ respectivamente. Se define la **partición producto** $\Pi_{\widetilde{A}} \wedge \Pi_{\widetilde{B}}$ como

$$\Pi_{\widetilde{A}} \wedge \Pi_{\widetilde{B}} = \{\widetilde{A}_i \cap \widetilde{B}_j / \widetilde{A}_i \in \Pi_{\widetilde{A}}, \widetilde{B}_j \in \Pi_{\widetilde{B}}\}$$

Estas dos definiciones no son más que una extensión para ϵ-particiones de los conceptos de refinamiento y producto de particiones habituales en nítidos([4]). Las propiedades que vamos a ver, van a decir cómo ciertamente estas definiciones que acabamos de dar, conservan las características de las particiones, es decir, que un refinamiento de una ϵ-partición de un conjunto es una verdadera ϵ-partición del conjunto y que la partición producto es una ϵ-partición de la intersección de los soportes de ambas particiones, como cabía esperar. Todo esto estará probado en el momento en que hayamos demostrado las dos siguientes propiedades:

1. Si $\Pi_{\widetilde{A}}$ es una ϵ-partición de un conjunto difuso $\widetilde{A}$, cualquier refinamiento $\Pi^*_{\widetilde{A}}$ de dicha partición es a su vez una ϵ-partición de $\widetilde{A}$; es decir, si $\Pi^*_{\widetilde{A}}$ es un refinamiento de una ϵ-partición de $\widetilde{A}$, se verifica que

$$\mu_{\widetilde{B}_{ik}\cap\widetilde{B}_{jl}}(x) \leq \epsilon \quad \forall i,j = 1,2,\ldots,n \quad \forall x \in X$$

$$\mu_{\bigcup_{i=1}^{n}\widetilde{B}_i}(x) \geq 1-\epsilon \quad \forall x \in X \;/\; \mu_{\widetilde{A}}(x) \geq 1-\epsilon$$

$$\mu_{\bigcup_{i=1}^{n}\widetilde{B}_i}(x) = \mu_{\widetilde{A}}(x) \quad \forall x \in X \;/\; \mu_{\widetilde{A}}(x) < 1-\epsilon$$

por lo tanto $\Pi^*_{\widetilde{A}}$ es una verdadera ϵ-partición de $\widetilde{A}$.

2. Si $\Pi_{\widetilde{A}} = \{\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_n\}$ y $\Pi_{\widetilde{B}} = \{\widetilde{B}_1, \widetilde{B}_2, \ldots, \widetilde{B}_m\}$ son ϵ-particiónes de $\widetilde{A}$ y $\widetilde{B}$ respectivamente, la **partición producto** $\Pi_{\widetilde{A}} \wedge \Pi_{\widetilde{B}}$ (definida según 5.2) es una ϵ-partición de $\widetilde{A}\cap\widetilde{B}$, que tiene como soporte la intersección de los soportes de $\Pi_{\widetilde{A}}$ y de $\Pi_{\widetilde{B}}$, con lo cual debemos probar que verifica:

$$\mu_{(\widetilde{A}_i\cap\widetilde{B}_k)\cap(\widetilde{A}_j\cap\widetilde{B}_l)}(x) \leq \epsilon \quad \forall i,j = 1,2,\ldots,n$$
$$\forall k,l = 1,2,\ldots,m \quad \forall x \in X$$

$$\mu_{\bigcup_{i,k}(\widetilde{A}_i\cap\widetilde{B}_k)}(x) \geq 1-\epsilon$$
$$\forall x \in X \;/\; \mu_{\widetilde{A}\cap\widetilde{B}}(x) \geq 1-\epsilon$$

$$\mu_{\bigcup_{i,k}(\widetilde{A}_i\cap\widetilde{B}_k)}(x) = \mu_{\widetilde{A}\cap\widetilde{B}}(x)$$
$$\forall x \in X \;/\; \mu_{\widetilde{A}\cap\widetilde{B}}(x) < 1-\epsilon$$

y además

$$\{x \in X/\mu_{\widetilde{A}_i\cap\widetilde{B}_k}(x) > 0 \text{ para algún } i,k\} =$$
$$\{x \in X/\mu_{\widetilde{A}_i}(x) > 0 \text{ para algún } i\}\cap$$
$$\{x \in X/\mu_{\widetilde{B}_k}(x) > 0 \text{ para algún } k\}$$

6 Conclusiones

En este trabajo hemos obtenido un nuevo concepto de partición difusa, que consideramos puede ser de gran utilidad en muchos campos en los que se aplica la Teoría de los Conjuntos Difusos, pues recoge y extiende las ideas que fundamentan las particiones nítidas. Además estas particiones permiten prefijar el "grado de borrosidad" que permitimos a nuestra partición, en función del valor ϵ, pudiendo incluso llegar a hablar de las δ-ϵ-particiones, como ya hemos apuntado, si esto fuese más conveniente para el investigador. Así mismo, si éste lo considera más oportuno, puede sustituir los operadores de unión e intersección considerados, por otros cualesquiera.

Referencias

[1] E.H. RUSPINI (1969), *A new approach to clustering*, Inf. Control 15, 22-32.

[2] DUBOIS AND PRADE (1980), *Fuzzy Sets and Systems. Theory and Applications*. Academic Press, Inc.

[3] KLIR AND FOLGER (1988), *Fuzzy Sets, Uncertainty, and Information* - Prentice-Hall International (UK) Limited.

[4] B. FORTE (1969), *Measures of information: the general axiomatic theory* R.I.R.O. 3-me année R-2.

[5] M.T. LOPEZ and P.GIL (1986), *Una medida de incertidumbre probabilística para sucesos difusos*. Trabajos de Estadística - 1,1, 60-69.

Indistinguibilidad y funciones de analogia *

Llorenç Valverde
Dept. de Matemàtiques i Informàtica
Universitat de les Illes Balears
e-mail: dmilvg0@ps.uib.es

Resumen

La Regla Composicional de Inferencia muestra la inferencia en lógica borrosa como un procedimiento que permite trasladar un cierto tipo de analogia entre la hipótesis y el antecedente de una regla, a la correspondiente analogia -medida ahora en el espacio de las tesis- que ha de permitir determinar la tesis a partir del consecuente de la regla. Resulta, pues, que diferentes formas de medir la analogia deben conducir a diferentes modelos de procesos de razonamiento lo que -a su vez- lleva a la introducción de las llamadas funciones de razonamiento por analogia, que son simplemente funciones que conservan estos tipos de relaciones de analogia. Se demuestra que, en el caso particular en que estas vengan dadas por relaciones difusas transitivas respecto de una t-norma, ciertos conjuntos de posibles imágenes por analogia superiores (resp. inferiores) de un conjunto dado son v-partes (resp. f-partes) respecto del preorden definido por la relación transitiva. Se analizan otras propiedades asociadas al uso de estas funciones para representar procesos de razonamiento por analogia.

Palabras clave: *Relaciones borrosas T-transitivas, lógica borrosa, razonamiento por analogia*

1 Introducción

El uso de técnicas emanadas de la lógica borrosa para representar procesos de razonamiento por analogia se halla latente en la propia definición de su herramienta básica, es decir, la regla composicional de inferencia, introducida por Zadeh ([15]) y posteriormente modificada por otros autores. En realidad, la formulación de Baldwin ([1]) de la citada regla, basada en los llamados valores de verdad fuzzy, permite ver el proceso básico de inferencia de la lógica borrosa compuesto por dos pasos, en el primero se compara el grado de adecuación entre la hipótesis y el antecedente de la regla para, a continuación, calcular como se transforma dicho grado en el grado en que debe modificarse el consecuente de la regla para obtener la conclusión ([8]). Tal y como han puesto de manifiesto Bouchon-Meunier y Valverde ([5]), dichos grados definen una pseudo-métrica generalizada en cada uno de los espacios de hipótesis y de tesis, respectivamente.

Desde este punto, con sólo modificar los criterios usados para comparar grados de adecuación entre antecedente e hipótesis, así como los correspondientes para tesis y consecuente, se obtiene de forma natural una modelación del llamado razonamiento por analogia, es decir, una vez establecida la relación inferencial entre A y B, el grado —medido de acuerdo con el criterio fijado— en que B' puede deducirse por analogia de A', depende exclusivamente del grado de adecuación entre A y A'. Este, es, esencialmente, el punto de vista adoptado en los trabajos como los de Turksen y Zhao ([11]), Mukaidono y Ding ([9]), Dubois y Prade ([7]) y el ya mencionado de Bouchon-Meunier y Valverde. ([5]). En este trabajo se profundiza en el estudio de las propiedades de las funciones de razonamiento por analogia introducidas eneste último trabajo. Para ello se parte del supuesto que los grados de analogia están dados por una relación borrosa T-transitiva.

2 Funciones de analogia

En los trabajo de Bouchon-Meunier y Valverde ([4, 5]) se presenta una formulación para el razonamiento por analogia basada en relaciones de similaridad fuzzy (es decir, relaciones fuzzy, reflexivas y transitivas respecto de una t-norma) mediante las llamadas funciones de razonamiento analògico. Dichas funciones son introducidas simplemente como aquellas funciones que preservan las similaridades definidas en los espacios de hipótesis y tesis respectivamente, es decir, si R (resp. R^*) es una relación de similaridad en el conjunto $[0,1]^X$ (resp. $[0,1]^Y$) de subconjuntos borrosos de un universo X (resp. Y), para cada nexo causal entre A y B, $A\delta B$, entonces

Definición 1 *([5]) Una $R - R^*$-función de razonamiento analógico será una función $\mathcal{A}$, tal que*

1. $\mathcal{A}(A, A\delta B) = B$,
2. *Para cada $s*$ existe un s tal que si $R(A, A') \geq s$, entonces $R^*(B, \mathcal{A}(A', A\delta B)) \geq s*$.*

es decir, una $R - R^*$-función de razonamiento por analogia respecto de A y B es una función $R - R^*$ contínua que transforma A en B.

La función definida mediante la regla composicional de inferencia proporciona un primer ejemplo de función de razonamiento por analogia como han demostrado recientemente, Boixader y Jacas ([2]) En este caso tanto R como R^*, vienen dadas por las relaciones de indistinguibilidad —que ellos llaman naturales— inducidas por una t-norma en $[0,1]^X$ y $[0,1]^Y$, i.e.

*Trabajo subvencionado, en parte, por la DGICYT proyecto n. PB91-0334

$$E_X(A, A') = Inf_{x \in X} (E_T(A(x); A'(x)) \wedge E_T(A'(x); A(x)))$$

siendo T una t-norma y

$$\begin{aligned} E_T(a; b) &= T^{\wedge}(a|b) \\ &= Sup\{\alpha \in [0,1]; T(a, \alpha) \leq b\}. \end{aligned}$$

Dicho de otra forma, si $R(A, B)$ indica la relación condicional entre A y B, entonces

$$CRI_{AB}(A')(y) = Sup_{x \in X} T(A'(x), R(A(x), B(y)))$$

es tal que

$$E_Y(CRI_{AB}(A'), CRI_{AB}(A")) \geq E_X(A', A")$$

Con todo, dado que siempre se verifica que $CRI_{AB}(A') \geq B$, resulta que dicha clase de funciones de razonamiento por analogia es muy restrictiva y es por lo que se propone en este trabajo, iniciar el estudio de las propiedades de la funciones más generales de razonamiento por analogia, definidas más arriba, estudio que tambien proporciona métodos para construir tales funciones.

3 Analogia superior e inferior

Se estudian en este apartado las propiedades de los conjuntos de analogias posibles asociadas a un nexo causal entre las dos distribuciones de posibilidad que definen los subconjuntos borrosos $A \in [0,1]^X$ y $B \in [0,1]^Y$. Para ello, y sin pérdida de generalidad, una vez fijadas, A, A', B y las relaciones de similaridad, R y R^*, en el espacio de las hipótesis $[0,1]^X$ y en el de las tesis $[0,1]^Y$ respectivamente,se procede a caracterizar los conjuntos B' para los que se verifica la relación

$$R^*(B; B') \geq R(A; A').$$

A este respecto cabe mencionar que los teoremas de representación para relaciones borrosas transitivas ([12]) permiten reducir dicho estudio al caso en que la relación R^* viene generada por una función h de $[0,1]^Y$ en $[0,1]$, es decir para aquellas relaciones de la forma

$$R^*(B; B') = E_T(h(B); h(B')) = E_h(B; B'),$$

puesto que dichos resultados establecen que una relación borrosa T-transitiva es, en general, de la forma

$$R^*(B; B') = Inf_{j \in J} E_{h_j}(B; B'),$$

para una cierta familia $\{h_j\}_{j \in J}$ de funciones de $[0,1]^Y$ en $[0,1]$. Por consiguiente, la desigualdad $R^*(B; B') \geq R(A; A')$. valdrá si, y sólo si, vale cada una de las correspondientes desigualdades asociadas a cada E_{h_j}, es decir si, para cada $j \in J$, es

$$E_{h_j}(B; B') \geq R(A; A')$$

Lo cual, a su vez, lleva a considerar las siguientes definiciones:

Definición 2 *(Analogia superior) Si T es una t-norma, B un subconjunto borroso de Y y h una función de $[0,1]^Y$ en $[0,1]$, se llamará conjunto de analogias superiores de B al nivel α, al conjunto*

$$\begin{aligned} U_{h,\alpha}(B) &= h^{[-1]}([T(h(B), \alpha), 1]) \\ &= \{C \in [0,1]^Y; h(C) \geq T(h(B), \alpha)\} \end{aligned}$$

Analogamente,

Definición 3 *(Analogia inferior) Si T es una t-norma, B un subconjunto borroso de Y y h una función de $[0,1]^Y$ en $[0,1]$, se llamará conjunto de analogias inferiores de B al nivel α, al conjunto*

$$\begin{aligned} L_{h,\alpha}(B) &= h^{[-1]}([0, T^{\wedge}(\alpha|h(B))]) \\ &= \{C \in [0,1]^Y; T(\alpha, h(C)) \leq h(B)\} \end{aligned}$$

Como quiera que, para cualquier t-norma T y para cualquier α es

$$T(h(B), \alpha) \leq T^{\wedge}(\alpha|h(B))$$

y dado que, además $h(B)$ está tanto en $L_{h,\alpha}(B)$ como en $U_{h,\alpha}(B)$ resulta que tiene sentido la siguiente definición.

Definición 4 *Para cada función h de $[0,1]^Y$ en $[0,1]$ y para cada $\alpha \in [0,1]$*

$$A_{h,\alpha}(B) = L_{h,\alpha}(B) \cap U_{h,\alpha}(B)$$

$$= \{C \in [0,1]^Y; T(h(B), \alpha) \leq h(C) \leq T^{\wedge}(\alpha|h(B))\}$$

indicará el conjunto de las $h - \alpha$ -analogias de B. Dependiendo de que sea $\alpha = R(A; A')$ o $\alpha = R(A'; A)$ se tratará de analogias directas o inversas.

En el caso de relaciones simétricas ambas analogias, las directas y las inversas, coinciden. Por su parte, la regla composicional de inferencia corresponderia al caso particular en el que las posibles analogias asociadas a B respecto de A y A' están todas en el conjunto $U_{h,\alpha}(B)$.

En definitiva, cuando se trata con relaciones T-transitivas, para poder hablar de analogia es suficiente disponer de una función h de $[0,1]^Y$ en $[0,1]$, para cada nivel $\alpha \in [0,1]$ el conjunto de analogias queda perfectamente determinado. Obviamente, las anteriores definiciones pueden ser extendidas a una relación T-transitiva R general, de forma que

$$\begin{aligned} U_{R,\alpha}(B) &= \{C \in [0,1]^Y; R(B'; B) \geq \alpha)\}, \text{ y} \\ L_{R,\alpha}(B) &= \{C \in [0,1]^Y; R(B; B') \geq \alpha)\}, \end{aligned}$$

en realidad, en este caso, tanto las analogias superiores como las inferiores son simplemente la intersección de las analogias superiores o inferiores asociada a cada una de las funciones h_j generadoras de la relación, es decir

$$\begin{aligned} U_{R,\alpha}(B) &= \bigcap_{j \in J} U_{h_j,\alpha}(B), \text{ y} \\ L_{R,\alpha}(B) &= \bigcap_{j \in J} L_{h_j,\alpha}(B), \end{aligned}$$

ahora, a partir de estas dos definiciones se obtiene el conjunto de analogias

$$A_{R,\alpha}(B) = L_{R,\alpha}(B) \cap U_{R,\alpha}(B).$$

Es inmediato comprobar, a partir de la propia definición, que para cada α , $U_{h,\alpha}(B)$ es una v-parte ([10]) respecto del preorden inducido en $[0,1]^Y$ por la aplicación h; en tanto que $L_{h,\alpha}(B)$ es una f-parte para el mismo preorden. Dicho de otra forma, las analogias superiores satisfacen el Modus Ponens respecto del preorden inducido por h, en tanto que las inferiores satisfacen el Modus Tollens.

Proposición 1 *1. Si C está en$U_{h,\alpha}(B)$ y $h(C) \leq h(C')$ entonces C' está en$U_{h,\alpha}(B)$*

2. Si C está en$L_{h,\alpha}(B)$ y $h(C) \geq h(C')$ entonces C' está en$L_{h,\alpha}(B)$

Dado que tanto la intersección de v-partes como la de f-partes es, a su vez, v-parte o f-parte, resulta que $U_{R,\alpha}(B)$ es una v-parte y que $L_{R,\alpha}(B)$ es una f-parte. Esta adecuación a conceptos característicos de las lógicas multivaluadas proporciona, a su vez, ciertas garantias para la consideración efectiva de funciones de razonamiento que se pretende dar a las funciones de razonamiento por analogia.

Por otra parte, resulta que tanto $L_{h,\alpha}(B)$ como $U_{h,\alpha}(B)$, forman cadenas descendentes respecto de α. Cadenas que empiezan ambas en $[0,1]^Y = U_{h,0}(B) = L_{h,0}(B)$, terminando las superiores en

$$U_{h,1}(B) = \{B'; h(B') \geq h(B)\},$$

en tanto que las inferiores lo hacen en

$$L_{h,1}(B) = \{B'; h(B') \leq h(B)\}.$$

Es decir, valen los siguientes enunciados:

Proposición 2 *Si $\alpha \leq \alpha'$ entonces*

1. $U_{h,\alpha'}(B) \subseteq U_{h,\alpha}(B)$

2. $L_{h,\alpha'}(B) \subseteq L_{h,\alpha}(B)$

Una propiedad interesante —especialmente desde el punto de vista de la inferencia aproximada— donde no conviene perder la relación entre la tesis, lo inferido, lo inferido de lo inferido, ..., viene expresada en la siguiente proposición que pone de manifiesto la relación entre analogias inferiores y superiores.

Proposición 3 *B' está en $L_{h,\alpha}(B)$ si, y sólo si, B está en $U_{h,\alpha}(B')$*

Una consecuencia inmediata de esta proposición nos muestra que en la equivalencia inducida por h, la clase representada por B es la intersección de todas las analogias inferiores de sus analogias superiores, y reciprocamente.

Proposición 4 *Valen las siguientes igualdades:*

1. $\bigcap_{B' \in L_{h,\alpha}(B)} U_{h,\alpha}(B') = \{C; h(C) = h(B)\}$.

2. $\bigcap_{B' \in U_{h,\alpha}(B)} L_{h,\alpha}(B') = \{C; h(C) = h(B)\}$.

4 Reglas con hipótesis compuestas

Finalmente, la consideración de los conjuntos $L_{h,\alpha}(B)$ y $U_{h,\alpha}(B)$, permite plantear, de forma natural el caso de hipótesis compuestas, es decir, el caso en el que la hipótesis es de la forma $(A_1, \cdots, A_n)$, en este supuesto, los conjuntos de analogia superior e inferior de B asociados a $(A_1, \cdots, A_n)$ y $(A'_1, ..., A'_n)$ vendrian determinados por los valores $\alpha_i = R(A_i; A'_i)$, en el caso de las analogias directas, o por $\alpha_i = R(A'_i; A_i)$, para las inversas. Es decir, si $\overline{\alpha} = (\alpha_1, \cdots, \alpha_n)$, entonces

$$U_{h,\overline{\alpha}}(B) = \bigcap_i U_{h,\alpha_i}(B) \text{ y}$$

$$L_{h,\overline{\alpha}}(B) = \bigcap_i L_{h,\alpha_i}(B)$$

que, como ya se ha mencionado, continuarian siendo v-parte y f-parte respectivamente.

Finalmente, el conjunto de analogias vendria, en este caso, dado por

$$A_{R,\overline{\alpha}}(B) = \left(\bigcap_i L_{h,\alpha_i}(B')\right) \bigcap \left(\bigcap_i U_{h,\alpha_i}(B')\right)$$

5 Conclusiones y perspectivas

Para concluir, las propiedades de los conjuntos de analogias inferiores y superiores de un subconjunto borroso B respecto de otros dos dados A y A', permiten una primera aproximación a la caracterización como funciones de inferencia de las funciones de razonamiento por analogia, en tanto que, como se ha probado, cualquiera de dichas analogias debe estar bien en la intersección de v-partes asociadas a una familia de generadores de la relación de indistinguibilidad asociada, o bien en la intersección de f-partes asociadas a la misma familia de funciones. En próximos trabajos se pretende introducir ejemplos naturales de analogias inferiores, en el mismo sentido que la regla composicional de inferencia proporciona un ejemplo natural de analogia superior.

6 Agradecimientos

El autor está en deuda con la profesora B. Bouchon-Meunier, ya que esta comunicación es parte de una serie de trabajos colaboración que, sobre este tema, vienen realizando conjuntamente.

Referencias

[1] Baldwin, J.F.: A New Approach to Approximate Reasoning Using Fuzzy Logic. Fuzzy Sets and Systems, 2, 1979. pp. 309-325

[2] Boixader, D. y Jacas, J.: Generators and duals similarities. Proc. Fifth IPMU. Paris, 1994. pp. 993-998

[3] Bouchon-Meunier, B.: Logique floue et analyse des similitudes. Actes des Journées Pôle-A-Pôle E du PRC Intelligence Artificialle, CNRS, Plestin-les-Grèves (1991). Rapport LAFORIA 92/10, 1992

[4] Bouchon-Meunier, B. y Valverde, L.: Analogy Relations and Inference. Proceedings Second IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'93), San Francisco, 1993. pp. 1140-1144.

[5] Bouchon-Meunier, B. y Valverde, L.: Analogical Reasoning and Fuzzy Resemblance. En B. Bouchon-Meunier et al. (Eds.): "Uncertainty in Intelligent Systems". North-Holland, 1993. pp 247-255.

[6] Bouchon-Meunier, B; Ramdani, M. y Valverde, L.: Fuzzy Logic, inductive and analogical reasoning. En A. Ralescu (Ed.): "Fuzzy Logic in Artificial Intelligence". Springer-Verlag, 1994.pp 38-50.

[7] Dubois, D. and Prade, H.: Similarity-based approximate reasoning. Rapport IRIT/94-10-R, 1994.

[8] Godo, L.; Jacas, J. y Valverde, L.: Fuzzy Values in Fuzzy Logic. Int. J. of Intelligent Systems, vol. 6.1991.pp. 199-212

[9] Mukaidono, M.; Ding, L. y Shen, Z.: Approximate reasoning based on revision principle. Proc. NAFIPS'90. Toronto, 1990.

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

[11] Turksen, I.B. y Zhao Zhong: An approximate analogical reasoning approach based on similarity measures. IEEE Trans. on Systems, Man and Cybernetics 18(1988), 1049-1056

[12] Valverde, L.: On the structure of F-indistinguishability operators. Fuzzy Sets and Systems, 17(1985). 313-328

[13] Zadeh, L.A.: The concept of linguistic variable and its application to approximate reasoning I, II, III Information Sciences (1975) 8 pp. 199-249; pp. 301-357; 9 pp 43-80

[14] Zadeh, L.A.: Similarity relations and fuzzy orderings. Information Sciences 3(1971). 117-200

[15] Zadeh, L.A.: A theory of approximate reasoning. In J. Hayes, D. Michie and L.I. Mikulich (Eds.): Machine Intelligence, vol 9. Halstead Press, New York, 1979. 149-194.

[16] Zwick, R.; Carlstein, E. y Budescu, D.V.: Measures of similarity among fuzzy concepts: a comparative analysis. Int. J. Appr. Reasoning 1 (1987), 221-242.

Clases de Subconjuntos Borrosos con el mismo Condicional Material

Enric Trillas
Dept. Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de
Madrid.

Adolfo R. De Soto
Dept. Dirección y Economía
de la Empresa
Edificio de Empresariales
Universidad de León
240761 León.
FAX: 987-29 14 54

Susana Cubillo
Dept. Matemática Aplicada
Facultad de Informática
Universidad Politécnica de
Madrid.

Resumen

Por el teorema de representación de preórdenes es conocido que una relación borrosa R es un preorden si y sólo si $R = \inf_{\mu \in F} I^T_\mu$, donde F es el conjunto de estados lógicos de R. En el presente trabajo se muestra como, en casi todos los casos, para las t-normas Min y arquimedianas, se pueden considerar que los subconjuntos borrosos μ están normalizados. En particular, cuando el universo es finito, siempre es posible tomar representantes normalizados. Además se estudia un orden parcial entre estados lógicos definido a partir de los preórdenes elementales I^T_μ que generan.

Palabras Clave: *Relaciones borrosas, condicional material borroso, preórdenes elementales, estados lógicos borrosos.*

1 Introducción

Dada una relación borrosa R sobre cierto universo E, se define el conjunto de estados lógicos [4] de R respecto a una t- norma T [1], que supondremos siempre continua, como

$$L^T_R = \{\mu \in [0,1]^E : T(\mu x, R(y/x)) \leq \mu y, \forall\, x, y \in E\}.$$

El conjunto de T-estados lógicos de un preorden R es la familia más grande que verifica el teorema de representación para preórdenes:

$$R(y/x) = \inf_{\mu \in L^T_R} I^T_\mu(y/x),$$

siendo $I^T_\mu(y/x) = \sup\{z \in [0,1] : T(\mu x, z) \leq \mu y\}$ el preorden elemental asociado a T y al subconjunto borroso μ. Sobre los elementos de L^T_R puede definirse la siguiente relación de equivalencia:

$$\mu_1 \sim \mu_2 \text{ si y sólo si } I^T_{\mu_1} = I^T_{\mu_2}$$

En [3] se mostraron las condiciones necesarias y suficientes para que $\mu_1 \sim \mu_2$, considerando tanto la t-norma $T = Min$ como las t-normas arquimedianas. Los resultados obtenidos fueron:

Teorema 1 *Se cumplirá que $I^{Min}_{\mu_1} = I^{Min}_{\mu_2}$ si y sólo si se verifica la siguiente condición para cualquier $x \in E$:*

Si $\mu_1 x \neq \mu_2 x$ entonces $\mu_1 x \geq \mu_1 y$ y $\mu_2 x \geq \mu_2 y$ para todo $y \in E$.

Teorema 2 *Si T es una t-norma arquimediana, entonces $I^T_{\mu_1} = I^T_{\mu_2}$ si y sólo si existe un número real $k \in \mathbb{R}$ tal que para todo $x \in E$ se verifica que*

$$h(\mu_1 x) = k + h(\mu_2 x),$$

siendo h el generador aditivo de la t-norma T. Además si T es arquimediana no estricta, se ha de cumplir $k \leq h(0) - \sup_{x \in E} h(\mu_2 x)$.

La relación de equivalencia $\sim$ permite definir el conjunto cociente $L^T_R/\sim$. Estudiaremos cómo puede ser elegido, para cada clase $\| \mu \| \in L^T_R/\sim$, un representante canónico que esté normalizado.

Recordemos que $[\mu]_\alpha = \{x \in E : \mu x \geq \alpha\}$ son los α-cortes de un subconjunto borroso μ. Definimos α_μ como el supremo de los valores que toma μ,

$$\alpha_\mu = \sup_{x \in E} \mu x.$$

Es evidente que α_μ siempre existe ya que $\mu x \leq 1$. Es fácil comprobar que

$$\alpha_\mu = \sup\{\alpha \in [0,1] : [\mu]_\alpha \neq \emptyset\}.$$

El conjunto de puntos $[\mu]_{\alpha_\mu}$ puede ser vacío cuando μ tenga un comportamiento asintótico. Por ejemplo, si tomamos $E = I\!R$, el subconjunto borroso

$$\mu x = 1 - e^{-x^2}$$

verifica que $\alpha_\mu = 1$, pero no existe un elemento $x \in I\!R$ con $\mu x = 1$, luego $[\mu]_1 = \emptyset$.

También es evidente que se cumple la igualdad

$$[\mu]_{\alpha_\mu} = \{x \in E : \mu x \geq \mu y \ \forall y \in E\}.$$

El conjunto $[\mu]_{\alpha_\mu}$ se conoce habitualmente como el núcleo de un subconjunto borroso, y se dice que un subconjunto borroso está normalizado o que "tiene puntos" si $\alpha_\mu = 1$ y $[\mu]_1 \neq \emptyset$.

2 Obtención de representantes normalizados

Dividiremos el estudio en dos casos: cuando la t-norma es el mínimo y cuando es arquimediana.

2.1 Caso T = Min

Consideremos dos posibilidades:

a) $[\mu]_{\alpha_\mu} = \emptyset$

Entonces se verifica que $\| \mu \| = \{\mu\}$. Si existiera η con $\eta \sim \mu$, es decir con $I_\mu^T = I_\eta^T$, tendríamos por el teorema 1 que μ y η sólo se diferenciarían en sus valores máximos, conclusión que se obtiene de la condición de dicho teorema: para cualquier $x \in E$, si $\mu x \neq \eta x$ entonces $\mu x \geq \mu y$ y $\eta x \geq \eta y$para todo $y \in E$. Pero eso es imposible ya que implicaría la existencia de un máximo valor para μ.

b) $[\mu]_{\alpha_\mu} \neq \emptyset$.

Por hipótesis consideraremos que μ toma algún valor máximo. Podemos tomar como representante

$$\tilde{\mu}a = \begin{cases} \mu a & \text{si } a \notin [\mu]_{\alpha_\mu} \\ 1 & \text{si } a \in [\mu]_{\alpha_\mu} \end{cases}$$

El siguiente teorema es de demostración evidente.

Teorema 3 *$\tilde{\mu} \in \| \mu \|$ y está normalizado.*

Evidentemente $\tilde{\mu}$ no tiene por qué ser continuo.

2.2 Caso T arquimediana

Por el teorema 2 sabemos que $\mu \sim \eta$ si y sólo si existe un $k \in I\!R$ verificando $h(\mu x) = k + h(\eta x)$ para todo x, y con k acotado si T es no estricta. Dada la clase $\| \mu \|$ definimos el subconjunto borroso

$$\tilde{\mu}x = h^{(-1)}(h(\mu x) - h(\alpha_\mu)),$$

que debido a que $h(\mu x) - h(\alpha_\mu) \in [0, h(0)]$, será igual a

$$\tilde{\mu}x = h^{-1}(h(\mu x) - h(\alpha_\mu)).$$

La comprobación de que $h(\mu x) - h(\alpha_\mu) \in [0, h(0)]$ es inmediata ya que al ser $\mu x \leq \alpha_\mu$tendremos que $h(\mu x) \geq h(\alpha_\mu)$ por ser h decreciente, y así $h(\mu x) - h(\alpha_\mu) \geq 0$. Por otra parte como $h(\mu x) \leq h(0)$, tendremos $h(\mu x) - h(\alpha_\mu) \leq h(0) - h(\alpha_\mu) \leq h(0)$.

Teorema 4 *$\tilde{\mu}$ es un representante canónico de la clase $\| \mu \|$.*

Demostración: En primer lugar $\tilde{\mu}$ está bien definido ya que $h(\mu x) - h(\alpha_\mu) \in [0, h(0)]$ y por tanto $h^{-1}(h(\mu x) - h(\alpha_\mu)) \in [0,1]$. Comprobemos que pertenece a la clase de μ. Si T es arquimediana estricta tendremos $h(\tilde{\mu}x) = h(\mu x) - h(\alpha_\mu)$ y será cierto con $k = -h(\alpha_\mu)$. Cuando T sea no estricta habrá que comprobar que la constante $k = -h(\alpha_\mu)$ verifica la acotación superior dada por el teorema 2, lo cual es cierto ya que

$$\begin{aligned} -h(\alpha_\mu) &\leq h(0) - h(\alpha_\mu) = h(0) - h(\sup_x \mu x) = \\ &= h(0) - \sup_x h(\mu x). \end{aligned}$$

Por último comprobemos que $\tilde{\mu}$ no depende del representante elegido para su obtención. Sea otro subconjunto borroso $\mu' \in \| \mu \|$. Existirá por tanto una constante k con $h(\mu' x) = k + h(\mu x)$. El representante asociado a μ' sería $\tilde{\mu}'x = h^{-1}(h(\mu' x) - h(\alpha_{\mu'}))$. Se deberá cumplir que $\tilde{\mu} = \tilde{\mu}'$. En primer lugar se verifica que $h(\alpha_{\mu'}) = k + h(\alpha_\mu)$ por continuidad de h,

$$\begin{aligned} h(\alpha_{\mu'}) &= h(\sup_x \mu' x) = \sup_x h(\mu' x) = \\ &= \sup_x (h(\mu x) + k) = \sup_x h(\mu x) + k \\ &= h(\alpha_\mu) + k. \end{aligned}$$

Aplicando esta igualdad tendremos que

$$\begin{aligned} \tilde{\mu}'(x) &= h^{-1}(h(\mu' x) - h(\alpha_{\mu'})) = h^{-1}(k + h(\mu x) - \\ &\quad - (k + h(\alpha_\mu)) = h^{-1}(h(\mu x) - h(\alpha_\mu)) = \\ &= \tilde{\mu}(x). \end{aligned}$$

□

El representante $\tilde{\mu}$ tiene la siguiente propiedad,

Teorema 5 $\alpha_{\tilde{\mu}} = \sup_x \tilde{\mu}x = 1$.

Demostración:

$$\begin{aligned}\alpha_{\tilde{\mu}} &= \sup_x \tilde{\mu}x = \sup_x h^{-1}(h(\mu x) - h(\alpha_\mu)) = \\ &= h^{-1}(h(\sup_x \mu x) - h(\alpha_\mu)) = h^{-1}(h(\alpha_\mu) - h(\alpha_\mu)) \\ &= h^{-1}(0) = 1.\end{aligned}$$

□

Observación. De nuevo que $\alpha_{\tilde{\mu}} = 1$ no implica necesariamente que $\tilde{\mu}$ esté normalizado debido a que puede presentar un comportamiento asintótico.

3 Un orden entre estados lógicos

El condicional material borroso, o preorden elemental, asociado a un subconjunto borroso μ, permite definir el siguiente preorden entre estados lógicos:

$$\eta \preceq \mu \text{ si y sólo si } I_\eta^T \geq I_\mu^T.$$

El preorden $\preceq$ dará lugar a un orden parcial en el conjunto cociente $L_R^T/\sim$.

Teorema 6

$$\mu \in L_R^T \text{ si y sólo si } R \leq I_\mu^T.$$

Demostración: Si $\mu \in L_R^T$ entonces $T(\mu x, R(y/x)) \leq \mu y$ para todo $x, y \in E$ y por tanto $R(y/x) \leq I_\mu^T(y/x)$ por definición de I_μ^T.

Por otra parte si $R \leq I_\mu^T$ entonces

$$T(\mu x, R(y/x)) \leq T(\mu x, I_\mu^T(y/x)) \leq \mu y$$

□

Teniendo en cuenta el resultado anterior la definición del orden $\preceq$ es equivalente a la condición

$$\eta \preceq \mu \text{ si y sólo si } \eta \in L_{I_\mu^T}^T.$$

Dicho orden presenta un elemento mínimo formado por los subconjuntos borrosos constantes $\mu = \alpha, \alpha \in [0,1]$, puesto que éstos dan lugar al preorden constante $I_\alpha^T = 1$ y son siempre estados lógicos; y no posee, en general, elemento máximo. Para verlo consideremos el siguiente teorema,

Teorema 7 *El orden parcial $\preceq$ sobre $L_R^T/\sim$ tiene elemento máximo si y sólo si R es un preorden elemental.*

Demostración: Si existe $\mu \in L_R^T/\sim$ con $\eta \preceq \mu$ para todo $\eta \in L_R^T/\sim$, entonces es que $I_\eta^T \geq I_\mu^T$ para todo $\eta \in L_R^T/\sim$, y por el teorema de representación

$$R = \inf_{\eta \in L_R^T} I_\eta^T = I_\mu^T.$$

El recíproco es también sencillo, ya que si $R = I_\mu^T$ entonces para todo $\eta \in L_R^T/\sim$ se verificará que $R = I_\mu^T \leq I_\eta^T$, por ser η estado lógico.

□

Basta por tanto tomar un preorden no elemental R para que el orden $\preceq$ no tenga elemento máximo. Es evidente que si $\eta \preceq \mu$ entonces η es irrelevante en el cálculo del preorden R a través del teorema de representación. La equivalencia entre $\eta \preceq \mu$ y $\eta \in L_{I_\mu^T}^T$ conduce a estudiar la pertenencia al conjunto $L_{I_\mu^T}^T$. En el caso de la t-norma Mínimo tenemos el siguiente resultado de caracterización.

Teorema 8 *Se cumplirá que un subconjunto η pertenece a $L_{I_\mu^{Min}}^{Min}$ si y sólo si se verifican las dos condiciones siguientes:*

1. *Si $\mu x \leq \mu y$ entonces $\eta x \leq \eta y$;*
2. *Si $\mu x > \mu y$ y $\eta x > \eta y$ entonces $\mu y \leq \eta y$*

Demostración: Las condiciones son necesarias. En efecto,

1. Si $\mu x \leq \mu y$ entonces $I_\mu^{Min}(y/x) = 1 \leq I_\eta^{Min}(y/x)$ si y sólo si $\eta x \leq \eta y$.
2. Si $\mu x > \mu y$ y $\eta x > \eta y$ tendremos que $I_\mu^{Min}(y/x) = \mu y \leq I_\eta^{Min}(y/x) = \eta y$

Para probar que las condiciones también son suficientes basta considerar que:

- Si $\mu x \leq \mu y$ entonces $Min(\eta x, I_\mu^{Min}(y/x)) = \eta x \leq \eta y$ por 1,
- Si $\mu x > \mu y$ entonces $Min(\eta x, I_\mu^{M}in(y/x)) = Min(\eta x, \mu y)$. Al ser $\mu x > \mu y$ entonces por 1 $\eta x \geq \eta y$. Si $\eta x = \eta y$, tendremos $Min(\eta x, \mu y) = Min(\eta y, \mu y) \leq \eta y$; y si $\eta x > \eta y$ entonces por 2 $\mu y \leq \eta y$ y por tanto $Min(\eta x, \mu y) \leq \mu y \leq \eta y$.

En cualquiera de los dos casos $\eta \in L_{I_\mu^{Min}}^{Min}$.

□

La condición 1 del teorema afirma que los estados lógicos tienen el mismo carácter de monotonía que μ. De hecho se verificarán las siguientes condiciones de monotonía:

- Si μ es constante entonces η es constante, ya que si $\mu x = \mu y$ se darán a la vez $\mu x \leq \mu y$ y $\mu x \geq \mu y$, con lo que $\eta x \leq \eta y$ y $\eta x \geq \eta y$ y así $\eta x = \eta y$.
- Si μ es creciente entonces η es no decreciente.
- Si μ es decreciente entonces η es no creciente.

Por otra parte la condición 2 afirma que $\mu x \leq \eta x$, siempre que exista un elemento $y \in E$ tal que $\mu y > \mu x$ y $\eta y > \eta x$. El siguiente teorema establece la relación existente entre el núcleo y el soporte de un estado lógico η de I_μ^{Min}. Recordemos que el soporte de un subconjunto borroso μ es el subconjunto clásico de E definido como

$$(\mu)_0 = \{x \in E : \mu x > 0\}.$$

Teorema 9 *Si $\eta \in L^{Min}_{I^{Min}_{\mu}}$ y $\eta \neq 0$ entonces se verifican las siguientes condiciones:*

1. $[\mu]_{\alpha_\mu} \subseteq [\eta]_{\alpha_\eta}$
2. $(\mu)_0 \subseteq (\eta)_0$

Demostración: 1. Si $[\mu]_{\alpha_\mu} = \emptyset$ entonces es trivial. Sea $x \in [\mu]_{\alpha_\mu}$ tendremos que

$$\sup_y Min(\eta y, I^M_\mu in(x/y) = \sup_y Min(\eta y, 1) = \sup_y \eta y \leq \eta x$$

por ser η estado lógico, luego $x \in [\eta]_{\alpha_\eta}$.

2. Sea $x \in (\mu)_0$, ya que de nuevo si $(\mu)_0 = \emptyset$ sería trivial. Si $x \in [\eta]_{\alpha_\eta}$ entonces $x \in (\eta)_0$. Supongamos que $x \notin [\eta]_{\alpha_\eta}$ lo que implica por 1 que $x \notin [\mu]_{\alpha_\mu}$. Pueden presentarse dos posibilidades: que $[\mu]_{\alpha_\mu} \neq \emptyset$ y entonces exista un $y \in [\mu]_{\alpha_\mu} \subseteq [\eta]_{\alpha_\eta}$ verificando $\mu y > \mu x$ y $\eta y > \eta x$ para cualquier x, y por tanto $0 < \mu x \leq \eta x$ por el teorema 8, de lo que se deduciría que $x \in (\eta)_0$. O bien que $[\mu]_{\alpha_\mu} = \emptyset$, en cuyo caso definimos los siguientes conjuntos:

$$(\mu)_{\mu x} = \{y \in E : \mu y > \mu x\}.$$

Se verifica que $(\eta)_{\eta x} \subseteq (\mu)_{\mu x}$ para cualquier $x \in E$, ya que si $y \notin (\mu)_{\mu x}$ entonces $\mu y \leq \mu x$ y por el teorema 8 tendremos $\eta y \leq \eta x$ con lo que $y \notin (\eta)_{\eta x}$. Al tener que $x \notin [\eta]_{\alpha_\eta}$, entonces $(\eta)_{\eta x} \neq \emptyset$ y por tanto existirá un $y \in (\eta)_{\eta x} \subseteq (\mu)_{\mu x}$ con $\mu y > \mu x$ y $\eta y > \eta x$, y aplicando la condición 1 del teorema 8 se cumplirá que $0 < \mu x \leq \eta x$, es decir $x \in (\eta)_0$.

□

4 Conclusiones

Mediante la relación de equivalencia $\sim$ entre estados lógicos definida como $\mu \sim \eta$ si y sólo si $I^T_\mu = I^T_\eta$, hemos comprobado que existe la posibilidad de elegir un representante normalizado para la clase de estados lógicos que definen un mismo preorden elemental. La elección del representante normalizado se puede hacer, para las t-normas Min y arquimedianas, siempre que el subconjunto borroso no tenga un comportamiento asintótico. Si el conjunto universo es finito, éste comportamiento no se puede dar y por tanto siempre es posible elegir tal representante. Por otra parte se ha iniciado el estudio de un orden parcial entre estados lógicos definido mediante los preórdenes elementales que definen. Este orden presenta un elemento mínimo y no posee en el caso más general un elemento máximo. Además se ha caracterizado, para la t-norma Min cuándo se da $\mu \preceq \eta$. Esto abre una línea de investigación futura sobre la estructura, reticular o no, del orden parcial $\preceq$ y sobre la existencia de elementos maximales.

References

[1] SCHWEIZER, B.; SKLAR, A. (1983) *Probabilistic Metric Spaces.* Elsevier North-Holland, New York.

[2] TRILLAS, E.; ALSINA, C. (1993) *Logic: going farther from Tarski?.* Fuzzy Sets and Systems 53, 1-13.

[3] TRILLAS, E., CUBILLO, S., RODRÍGUEZ, A. (1994) *Sobre la identidad de condicionales materiales.* Actas IV Congreso de la Asociación Española sobre Tecnologías y Lógica Fuzzy, 75-77, Blanes, Girona.

[4] TRILLAS, E. (1993) *On Logic and Fuzzy Logic.* Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 1, No. 2 107-137.

Funciones de Agregación Extendidas

G. Mayor
Dep. de Matemáticas e Informática
Universitat de les Illes Balears
e-mail: dmigmf0@ps.uib.es

Resumen

Introducimos en este trabajo el concepto de Función de Agregación Extendida como una aplicación entre la colección L de listas ordenadas (de cualquier dimension) de elementos de un retículo dado V (espacio de valores que tienen que ser agregados) y el mismo V.

Definimos una estructura de retículo sobre L que nos permite presentar la condición de monotonía extendida para una Función de Agregación. Consideramos, aparte de la monotonía, otras condiciones que parecen razonables exigir a un proceso de agregación. En el caso V = [a,b] se muestran algunos ejemplos.

Palabras clave: agregación, retículo, monotonía.

1. El retículo de las listas ordenadas de elementos de un retículo

En este apartado introducimos y estudiamos dos órdenes en el conjunto de las listas ordenadas y finitas (de cualquier dimensión) formadas a partir de un retículo de valores V.

Sea V un retículo (orden indicado por $\leq$) con mínimo 0 y máximo 1. Consideremos, para cada $n \geq 1$, el conjunto V^n de n-plas ordenadas $(x1,...,xn)$ con $xi \in V$ para todo $i:1,..,n$. Sea $L = \cup V^n$.

Definición 1. Sean $x = (x1,...,xn)$ $y = (y1,...,ym)$ elementos de L.

$x \leq_1 y$ si y sólo si $n \leq m$, $x1 \leq y1,....,xn \leq yn$, y si $n < m$ entonces $\sup(x1,..,xn) \leq \inf(y(n+1),...,ym)$.

$x \leq_2 y$ if si y sólo si $n \geq m$, $x1 \leq y1,....,xm \leq ym$, y si $n > m$ entonces $\sup(x(m+1),...,xn) \leq \inf(y1,...,ym)$

Por ejemplo si $L = [0,10]$ (orden usual) tendríamos: $(2,1,7,1) <_2 (2,1,7) <_1 (2,2,7,7) <_2 (8,7) <_1 (8,7,8)$

Teorema 1. Las relaciones $\leq_1$ y $\leq_2$ son de orden sobre L, con (0) y (1) elementos mínimo ($\leq_1$) y máximo ($\leq_2$) respectivamente. L es un retículo respecto de cada uno de estos órdenes.

Nota 1

1) Para cada $(x1,...,xn) \in L$ tenemos: $(x1,...,xn,\inf(x1,...,xn)) <_2 (x1,...,xn) <_1 (x1,...,xn,\sup(x1,...,xn))$.

Además, $(x1,...,xn)$ es siguiente de $(x1,...,xn,\inf(x1,...,xn))$ y $(x1,...,xn)$ es antecedente de $(x1,...,xn,\sup(x1,...,xn))$.

Como consecuencia no existen elementos maximales para $\leq_1$ (ni minimales para $\leq_2$).

2) L no es modular (aunque V sea distributivo).

3) $x \leq_1 y$ y $x \leq_2 y$ si y sólo si $x \leq_p y$ (orden producto)

4) $x \leq_1 y$ implica $x \leq_L y$ (orden lexicográfico)

5) $(V^n ; \leq_p) \rightarrow (L ; \leq_1 , \leq_2)$ (inclusión) es un monomorfismo de retículos.

Es interesante observar que podríamos considerar también listas no ordenadas de elementos de V. Basta definir sobre L la equivalencia $(x,y \in L)$:

$x \, R \, y$ si y sólo si $n = m$ y existe una permutación p de $\{1,2,...,n\}$ tal que $xp(1) = y1,...,xp(n) = yn$.

Al considerar $L' = L/R$ (conjunto de listas no ordenadas) podemos definir :

$[x] \leq_1 [y]$ si y sólo si $n \leq m$ y existe $x'Rx$, $y'Ry$ tal que $x' \leq_1 y'$. Lo mismo para $\leq_2$. A partir de aquí se podría estudiar la estructura bi-ordenada $(L' ; \leq_1, \leq_2)$.

2. La condición de monotonía. Dualidad.

Definición 2. Sea A una aplicación entre L y V. Diremos que A es monótona (creciente) si se verifica: $x \leq_i y$ implica $A(x) \leq A(y)$ con i: 1, 2.

Obviamente podemos considerar funciones monótonas con respecto sólo a uno de los órdenes.

Ejemplos 1
1. $A(x1,...,xn) = \inf(x1,...,xn)$ $(\sup(x1,...,xn))$ son monótonas.
2. En el caso V = [0,1], sea S una t-conorma y consideremos $A(x1,...,xn) = S(x1,...,xn)$. Es trivial demostrar que A es mónotona con respecto a $\leq_1$. Igualmente $A(x1,...,xn) = T(x1,...,xn)$, donde T es una t-norma , es monótona con respecto a $\leq_2$.

Teorema 2. Sea T una t-norma contínua y $A(x1,...,xn) = T(x1,...,xn)$. A es monótona con respecto a $\leq_1$ si, y sólo si, T = Min.

Es interesante observar que la t-norma no contínua: $Z(x,y) = 0$ si $\max(x,y) < 1$, $Z(x,y) = \min(x,y)$ si $\max(x,y)=1$, define una aplicación que es monótona con respecto a $\leq_1$.

Por otra parte, consideremos una negación (fuerte), N, sobre V (N: V → V involutiva y decreciente). Dada A: L → V podemos definir A^*: L → V mediante $A^*(x1,...,xn) = N(A(N(x1),...,N(xn)))$ (A^* es la N-dual de A). Podemos dar el siguiente

Lema 1. Sea N una negación sobre V, y A: L → V. Si A es monótona respecto a $\leq_1$ entonces A^* es monótona respecto $\leq_2$ y si A es monótona respecto $\leq_2$ entonces A^* es monótona respecto $\leq_1$.

Como consecuencia de este Lema, podemos enunciar:

Teorema 3. Sea S una t-conorma contínua y $A(x1,...,xn) = S(x1,...,xn)$. A es monótona respecto ≤ 2 si, y sólo si, , S = Max.

Corolario 1. Dada A: L → [0,1], donde $A(x1,...,xn) = F(x1,...,xn)$ con F una t-norma o una t-conorma contínua. Entonces A es monótona (ambos órdenes) si, y sólo si, F= Min o F = Max.

Este último resultado parece indicar que la condición de monotonía que hemos introducido (para ser incluída en la definición de Función de Agregación Extendida) es bastante razonable, ya que aparte de Min y Max no hay t-normas ni t-conormas contínuas (utilizadas para modelizar "y" y "o") que sean monótonas.

Nota 2. Al considerar el conjunto de listas no ordenadas sobre V, L', podemos identificar la clase de aplicaciones simétricas A: L→ V con la clase de aplicaciones A': L' → V mediante la biyección $A \rightarrow A'$: $A'(|(x1,...,xn)|) = A(x1,...,xn)$.

Teorema 4. Dada A: L→ V monótona con respecto a $\leq_p$, entonces:
a) A es monótona respecto $\leq_1$ si y sólo si $A(x1,...,xn) \leq A(x1,...,xn,\sup(x1,...,xn))$
b) A es monótona respecto $\leq_2$ si y sólo si $A(x1,...,xn,\inf(x1,...,xn)) \leq A(x1,...,xn)$ para todo (x1,...,xn) en L.

3. Hacia una definición de Funciónde Agregación Extendida.

La definición que se propone a continuación presenta la novedad respecto de otras que se han dado recientemente ([4]) de introducir una condición de monotonía "global" y no únicamente para listas de una determinada dimensión.

Definición 3. A: L → V es una Función de Agregación Extendida si verifica las siguientes dos condiciones:
1) Monotonía
2) Idempotencia: $A(x) = a$ para todo x = (a,...,a) de L

Otras condiciones razonables a considerar serían:
3) Simetría
4) Auto dualidad: $A(N(x1),...,N(xn)) = N(A(x1,...,xn))$
5) Auto identidad: $A(x1,...,xn,x(n+1)) = A(x1,...,xn)$ siempre que $x(n+1) = A(x1,...,xn)$
6) Continuidad

Nota 3. Supongamos que A: E → L es monótona respecto $\leq_p$, entonces A es idempotente si, y sólo si, verifica $\inf(x1,...,xn) \leq A(x1,...,xn) \leq \sup(x1,...,xn)$ para todo (x1,...,xn) en L.

Ejemplo 2. Consideremos L = [a,b] (orden usual). Dada w1,w2,.... una sucesión de números reales positivos y f: [a,b] → [-∞,+∞] una función contínua y estrictamente monótona, podemos definir A: ∪ $[a,b]^n$ → [a,b] (media casi lineal extendida)

$$A(x_1,\ldots,x_n) = f^{-1}\left(\frac{\sum_{i=1}^{n} w_i f(x_i)}{\sum_{i=1}^{n} w_i}\right)$$

A es una Función de Agregación Extendida (FAE). A es simétrica si y sólo si w1=w2=.... Por otra parte, A(x1,...,xn,x(n+1)) = A(x1,...,xn) si y sólo si x(n+1) = A(x1,...,xn) (auto identidad).

Con respecto a la condición de auto dualidad, puede demostrarse que A (en caso de ser f acotada) es auto dual (N negación sobre [a,b]) si y sólo si $N(x) = f^{-1}(r+s + (s-r)/(r-s)f(x))$ para todo x en [a,b], donde r = inf f y s = sup f. En particular si L = [0,1] y N(x) = 1-x, entonces la anterior condición se convierte en f(x) + f(1-x) = r+s para todo x en [0,1].

4. Una distancia sobre L

Con el objetivo de considerar la condición de continuidad para una función A: L→ V, vamos a definir una distancia sobre L. Supongamos que tenemos una distancia, d, sobre V y sean x=(x1,...,xn) , y=(y1,...,ym) en L. Definimos:

$$d(x,y) = |n-m| + \sum_{i=1}^{\min(n,m)} d(x_i, y_i)$$

Efectivamente d es una distancia en L y obviamente la restricción de d a V es la distancia de partida sobre V.

Referencias

[1]. Aczél, J., Saaty, T.L. Procedures for Synthesizing Ratio Judgements. *Journal of Mathematical Psychology* 27, 93-102 (1983).

[2]. Aczél, J. *Lectures on functional equations and their applications*. New York/London: Academic Press. 1966.

[3]. Schweizer, B., Sklar, A. *Probabilistic Metric Spaces*. North-Holland. 1983.

[4]. Yager, R.R. On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics* 18 (1) 183-190 (1988).

Retículos de conceptos asociados a W-contextos L-Fuzzy.

Ana Burusco Juandeaburre
Dpto. de Matemática e Informática
Universidad Pública de Navarra
e-mail: burusco@si.upna.es

Ramón Fuentes González
Dpto. de Matemática e Informática
Universidad Pública de Navarra
e-mail: rfuentes@si.upna.es

Resumen

En este artículo, utilizando la definición de composición de relaciones $L-Fuzzy$, generalizamos la teoría de conceptos $L-Fuzzy$ que hemos desarrollado en un trabajo anterior ([1]) y que trata de modelizar la adquisición y clasificación del conocimiento. Dicha teoría toma como punto de partida la idea de concepto definida por Wille ([5]).

Comenzamos el trabajo definiendo los W-contextos $L-Fuzzy$ como aquellas tuplas $(L,W,X,Y,\underset{\sim}{R})$ donde W, X e Y son el conjunto de etiquetas, objetos y atributos respectivamente, y $\underset{\sim}{R} \in L^{X\times Y}$ es una relación $L-Fuzzy$.

A partir de estos contextos definiremos varios operadores que nos servirán para construir los W-conceptos $L-Fuzzy$, que serán pares de relaciones de la forma $(\underset{\sim}{P},\underset{\sim}{Q})$ con $\underset{\sim}{P} \in L^{W\times X}$, $\underset{\sim}{Q} \in L^{W\times Y}$ cumpliendo $\underset{\sim}{P}_1 = \underset{\sim}{Q}$ y $\underset{\sim}{Q}_2 = \underset{\sim}{P}$ con las definiciones dadas de los operadores de subíndices 1 y 2.

Una vez estudiada la estructura reticular del conjunto de los W-conceptos $L-Fuzzy$ analizamos un ejemplo práctico donde se interpreta esta nueva definición de concepto.

Palabras clave: *Conceptos $L-Fuzzy$, W-conceptos $L-Fuzzy$, W-contextos $L-Fuzzy$, conocimiento conceptual.*

1 Introducción

A través de la teoría de conceptos $L-Fuzzy$ ([1]) se analizan contextos (L,X,Y,$\underset{\sim}{R}$) donde X e Y son el conjunto de objetos y el de atributos respectivamente, y $\underset{\sim}{R} \in L^{X\times Y}$. Este análisis se realiza por medio de los conceptos $L-Fuzzy$, que son pares $(\underset{\sim}{A},\underset{\sim}{B})$ donde $\underset{\sim}{A} \in L^X$, $\underset{\sim}{B} \in L^Y$ y se verifica que $\underset{\sim}{A}_1 = \underset{\sim}{B}$ y $\underset{\sim}{B}_2 = \underset{\sim}{A}$.

Hemos considerado que puede ser interesante definir un nuevo tipo de contexto que nos permita manejar conceptos cuyos elementos sean relaciones $L-Fuzzy$. El ejemplo 1 que aparece más adelante es una muestra de ello. Con este fin, variaremos la definición de contexto $L-Fuzzy$ dada en [1] para que se ajuste a esta nueva situación y definiremos los W-contextos $L-Fuzzy$ a partir de los cuales se construirán sus correspondientes conceptos $L-Fuzzy$.

2 Definición de W-contexto L-Fuzzy

Sea $(L,\leq)$ un retículo completo, $'$ una complementación o negación en L y $\intercal$ una t-conorma en L. Sean también W, X, Y conjuntos no vacíos, y $\underset{\sim}{R} \in L^{X\times Y}$ una relación $L-Fuzzy$.

Definición 1. Llamaremos W-contexto L-Fuzzy a la tupla (L,W,X,Y,$\underset{\sim}{R}$) donde W, X e Y serán los conjuntos de etiquetas, objetos y atributos respectivamente.

Ejemplo 1. Supongamos que se imparte una serie de cursos de formación para personas en paro que optarán posteriormente a varios empleos.

Para trabajar en esta situación tomaremos el W-contexto $(L,W,X,Y,\underset{\sim}{R})$ donde L={0,0.1,0.2,0.3, ... 0.9,1}, W={interesante, muy específico, de nivel adecuado a sus conocimientos } contiene opiniones personales sobre los cursos, X={informática, contabilidad, mecánica } es el conjunto de los cursos de formación, Y={empleado del hogar, camarero, secretario, vendedor de coches} está formado por los distintos empleos, y la siguiente Tabla 1 representa la relación existente entre los cursos de formación y los trabajos ofertados ($\underset{\sim}{R}(x,y)$=1 indica que el curso de formación x está estrechamente relacionado con el empleo y, mientras que $\underset{\sim}{R}(x,y) = 0$ indicaría todo lo contrario.)

$\underset{\sim}{R}$	empl.hogar	camarero	secretario	vend.coches
informática	0	0.2	0.9	0.5
contabilidad	0	0.4	1	0.7
mecánica	0.2	0	0	0.5

Tabla 1

3 Operadores derivación y operadores constructores

Definición 2. Dada la relación $\underset{\sim}{P} \in L^{W\times X}$, le asociamos un elemento $\underset{\sim}{P}_1$ de $L^{W\times Y}$ tal que

$$\underset{\sim}{P}_1(w,y) = \inf_{x\in X}(\underset{\sim}{P}'(w,x) \top \underset{\sim}{R}(x,y)) \qquad (1)$$

Análogamente, dado $\underset{\sim}{Q} \in L^{W\times Y}$ le asociamos $\underset{\sim}{Q}_2 \in L^{W\times X}$ tal que

$$\underset{\sim}{Q}_2(w,x) = \inf_{y\in Y}(\underset{\sim}{Q}'(w,y) \top \underset{\sim}{R}(x,y)) \qquad (2)$$

Se puede demostrar de forma sencilla la relación existente entre las definiciones dadas de operadores derivación y la composición de relaciones de la teoría fuzzy.

Así, si tomamos la definión $sup - \star$ dada por Dubois y Prade ([3]) y la generalizamos de forma natural al caso de un retículo completo L cualquiera, tenemos la siguiente expresión de dicha composición

$$\forall \underset{\sim}{P} \in L^{W\times X}, \forall \underset{\sim}{R} \in L^{X\times Y}$$

$$(\underset{\sim}{P} \star \underset{\sim}{R})(w,y) = \sup_{x\in X}(\underset{\sim}{P}(w,x) \star \underset{\sim}{R}(x,y))$$

Por tanto, se comprueba de forma inmediata la siguiente

Proposición 1. Para toda t-conorma triangular $\top$ en L y para toda negación $'$, son ciertas las siguientes igualdades:

$$\underset{\sim}{P}_1 = (\underset{\sim}{P} \star \underset{\sim}{R}')', \ \forall \underset{\sim}{P} \in L^{W\times X}$$
$$\underset{\sim}{Q}_2 = (\underset{\sim}{Q} \star (\underset{\sim}{R}^{op})')', \ \forall \underset{\sim}{Q} \in L^{W\times Y}$$

donde $\star$ es la t-norma triangular en L asociada a $(\top,')$ y R^{op} es la relación opuesta de R definida por $R^{op}(y,x) = R(x,y), \forall x \in X, \forall y \in Y$.

También en esta ocasión podremos definir los operadores constructores([1]) φ y ψ , componiendo los anteriores, que se representarán de la siguiente manera:

$$\varphi : L^{W\times X} \to L^{W\times X} \ t.q. \ \varphi(\underset{\sim}{P}) = \underset{\sim}{P}_{12} \qquad (3)$$

$$\psi : L^{W\times Y} \to L^{W\times Y} \ t.q. \ \psi(\underset{\sim}{Q}) = \underset{\sim}{Q}_{21} \qquad (4)$$

Se comprueba que para la construcción de $\varphi(\underset{\sim}{P})$ y $\psi(\underset{\sim}{Q})$ basta aplicar a cada una de las filas de $\underset{\sim}{P}$ y $\underset{\sim}{Q}$ los operadores constructores definidos en la teoría $L - Fuzzy$.

Todos estos operadores cumplen las propiedades demostradas en [1] y en particular, como φ y ψ preservan el orden usual en relaciones $L-Fuzzy$, $\Omega = fix(\varphi)$ y $\Sigma = fix(\psi)$ son retículos completos (Tarski [4]), lo que nos permitirá dar nuestra definición de W-concepto $L - Fuzzy$([1]).

Por otra parte, debido a la monotonía de φ y ψ, podemos utilizar la teoría de P. y R. Cousot ([2]), que nos proporciona una versión constructiva del teorema de Tarski ([4]), para calcular los elementos máximo y mínimo de los retículos $fix(\varphi)$ y $fix(\psi)$, así como el supremo y el ínfimo de una familia:

$$\underset{\sim}{0}_\Omega = luis(\varphi)(\underset{\sim}{0}) \qquad \underset{\sim}{0}_\Sigma = luis(\psi)(\underset{\sim}{0}) \qquad (5)$$
$$\underset{\sim}{1}_\Omega = llis(\varphi)(\underset{\sim}{1}) \qquad \underset{\sim}{1}_\Sigma = llis(\psi)(\underset{\sim}{1}) \qquad (6)$$

$$\forall \{\underset{\sim}{P}_i, i \in I\} \in \Omega, \forall \{\underset{\sim}{Q}_i, i \in I\} \in \Sigma$$

$$\bigvee_\Omega \underset{\sim}{P}_i = luis(\varphi)\left(\bigvee \underset{\sim}{P}_i\right) \qquad (7)$$

$$\bigvee_\Sigma \underset{\sim}{Q}_i = luis(\psi)\left(\bigvee \underset{\sim}{Q}_i\right) \qquad (8)$$

$$\bigwedge_\Omega \underset{\sim}{P}_i = llis(\varphi)\left(\bigwedge \underset{\sim}{P}_i\right) \qquad (9)$$

$$\bigwedge_\Sigma \underset{\sim}{Q}_i = llis(\psi)\left(\bigwedge \underset{\sim}{Q}_i\right) \qquad (10)$$

donde en general, $luis(f)(\underset{\sim}{P})$, con f una función que preserva el orden, es el límite de la secuencia iterativa superior estacionaria de f que comienza en $\underset{\sim}{P}$ y $llis(f)(\underset{\sim}{Q})$ el límite de la secuencia iterativa inferior estacionaria de f que comienza en $\underset{\sim}{Q}$.

4 Retículo de W-conceptos L-Fuzzy

Sea (L,W,X,Y,$\underset{\sim}{R}$) un W-contexto $L-Fuzzy$ y sean $fix(\varphi)$ y $fix(\psi)$ los conjuntos de puntos fijos de los operadores constructores que hemos definido.

Definición 3. Si $\underset{\sim}{P} \in fix(\varphi)$, al par $(\underset{\sim}{P},\underset{\sim}{P_1})$ lo llamaremos W-concepto L-Fuzzy del W-contexto $L-Fuzzy$ $(L,W,X,Y,\underset{\sim}{R})$.

Esta definición puede darse de la misma forma tomando $\underset{\sim}{Q} \in fix(\psi)$. En este caso, $(\underset{\sim}{Q_2},\underset{\sim}{Q})$ será el W-concepto $L-Fuzzy$.

Además, de la misma forma que vimos en [1], se demuestra el siguiente:

Teorema 1. El conjunto de W-conceptos $L-Fuzzy$:

$$\underset{\sim}{\mathcal{L}_W} = \{(\underset{\sim}{P},\underset{\sim}{P_1})\, t.q.\, \underset{\sim}{P} \in fix(\varphi)\} = \{(\underset{\sim}{Q_2},\underset{\sim}{Q})\, t.q.\, \underset{\sim}{Q} \in fix(\psi)\}$$

con el orden $\preceq$ definido de la siguiente forma:

$$\forall(\underset{\sim}{P},\underset{\sim}{P_1}),(\underset{\sim}{T},\underset{\sim}{T_1}) \in \underset{\sim}{\mathcal{L}_W}$$
$$(\underset{\sim}{P},\underset{\sim}{P_1}) \preceq (\underset{\sim}{T},\underset{\sim}{T_1}) \Longleftrightarrow \underset{\sim}{P} \leq \underset{\sim}{T}$$

tiene estructura de retículo completo.

Llamaremos a $(\underset{\sim}{\mathcal{L}_W},\preceq)$ retículo de W-conceptos $L-Fuzzy$.

Se puede demostrar, como en [1], que los elementos mínimo y máximo del retículo $(\underset{\sim}{\mathcal{L}_W},\preceq)$ son respectivamente:

$$\underset{\sim}{0}_{\underset{\sim}{\mathcal{L}_W}} = (\underset{\sim}{0}_\Omega,\underset{\sim}{1}_\Sigma) \text{ y } \underset{\sim}{1}_{\underset{\sim}{\mathcal{L}_W}} = (\underset{\sim}{1}_\Omega,\underset{\sim}{0}_\Sigma)$$

Además, si utilizamos una t-conorma triangular $\intercal$ semicontinua superiormente para las definiciones de las funciones φ y ψ, tenemos la siguiente:

Proposición 2. Para cada familia

$$\mathcal{F} = \{(\underset{\sim}{P_i},(\underset{\sim}{P_i})_1),\ \underset{\sim}{P_i} \in \Omega\} = \{((\underset{\sim}{Q_i})_2,\underset{\sim}{Q_i}),\ \underset{\sim}{Q_i} \in \Sigma\} \subseteq \underset{\sim}{\mathcal{L}_W}$$

se puede expresar el supremo y el ínfimo de $\mathcal{F}$ de la siguiente forma:

$$\bigvee_{\underset{\sim}{\mathcal{L}_W}} \left(\underset{\sim}{P_i},(\underset{\sim}{P_i})_1\right) = \left(\bigvee_{\Omega} \underset{\sim}{P_i}, \bigwedge_{\Sigma} (\underset{\sim}{P_i})_1\right)$$

$$\bigwedge_{\underset{\sim}{\mathcal{L}_W}} \left((\underset{\sim}{Q_i})_2,\underset{\sim}{Q_i}\right) = \left(\bigwedge_{\Omega} (\underset{\sim}{Q_i})_2, \bigvee_{\Sigma} \underset{\sim}{Q_i}\right)$$

donde los supremos e ínfimos en Σ y Ω se calcularán de acuerdo con las expresiones (7),(8),(9) y (10).

5 Relación entre $\underset{\sim}{\mathcal{L}}$ y $\underset{\sim}{\mathcal{L}_W}$

Si tomamos un elemento $(\underset{\sim}{P},\underset{\sim}{P_1}) \in \underset{\sim}{\mathcal{L}_W}$, y tenemos en cuenta las definiciones dadas en [1], se observa que dicho elemento está formado por un par de relaciones donde cada una de las filas de $\underset{\sim}{P}$, etiquetada por un elemento de W, es la $Fuzzy$-extensión de un concepto $L-Fuzzy$ del contexto $(L,X,Y,\underset{\sim}{R})$, y cada una de las filas de $\underset{\sim}{P_1}$ su correspondiente $Fuzzy$-comprensión. Esto nos permitirá construir el retículo de W-conceptos $L-Fuzzy$ en función del retículo de conceptos $L-Fuzzy$.

En particular, su cardinal se calculará de la siguiente forma:

Proposición 3. Sea $(L,W,X,Y,\underset{\sim}{R})$ un W-contexto $L-Fuzzy$ con $Card(W)$=k. Si llamamos $\underset{\sim}{\mathcal{L}}$ al retículo de conceptos $L-Fuzzy$ del contexto $(L,X,Y,\underset{\sim}{R})$ y si $Card(\underset{\sim}{\mathcal{L}})$=n finito, entonces $Card(\underset{\sim}{\mathcal{L}_W}) = n^k$.

Dem: Basta tener en cuenta que dado $\underset{\sim}{P} \in L^{W\times X}$, para calcular $\varphi(\underset{\sim}{P}) = \underset{\sim}{P_{12}}$ se aplica el operador φ definido en la teoría de conceptos $L-Fuzzy$ a cada una de las filas de la relación $\underset{\sim}{P}$. Por tanto, $Card(\underset{\sim}{\mathcal{L}_W}) = VR(n,k) = n^k$.■

De alguna forma, el W-concepto $(\underset{\sim}{P},\underset{\sim}{P_1})$ recoge los conceptos $L-Fuzzy$ que se derivan de cada uno de los conjuntos $L-Fuzzy$ que forman las filas de la relación original.

Por otra parte, si tenemos una única etiqueta, los W-conceptos que se derivan de $(L, W, X, Y, \underset{\sim}{R})$ estarán estrechamente ligados con los que se obtienen a partir de $(L, X, Y, \underset{\sim}{R})$, es decir, serán pares de conjuntos $L - Fuzzy$ como se ve en la siguiente:

Proposición 4. Si $Card(W) = 1$, entonces $\underset{\sim}{\mathcal{L}}_W(L, W, X, Y, \underset{\sim}{R}) \backsimeq \underset{\sim}{\mathcal{L}}(L, X, Y, \underset{\sim}{R})$.

Dem: Si $Card(W) = 1$, entonces $L^{W\times X} \backsimeq L^X$ y $L^{W\times Y} \backsimeq L^Y$, y teniendo en cuenta estos isomorfismos, las definiciones (1) y (2) y sus correspondientes en el caso $L - Fuzzy([1])$, es inmediato que $\underset{\sim}{\mathcal{L}}_W(L, W, X, Y, \underset{\sim}{R}) \backsimeq \underset{\sim}{\mathcal{L}}(L, X, Y, \underset{\sim}{R})$.■

6 Interpretación de los W-conceptos L-Fuzzy a través de un ejemplo

Volviendo al Ejemplo 1 (pg. 1), vamos a suponer que la Tabla 2 (ver final de página) representa la opinión de un individuo sobre los distintos cursos (objetos), atendiendo a una serie de aspectos concretos (etiquetas) sobre los que se le pregunta.

Aplicando la teoría de P. y R. Cousot([2]) a nuestro trabajo y teniendo en cuenta que φ preserva el orden, se puede calcular el punto fijo de φ que se deriva de $\underset{\sim}{Q}$ al que llamaremos $\underset{\sim}{Q^*}$ y el W-concepto $L - Fuzzy$ $(\underset{\sim}{Q^*}, \underset{\sim}{Q_1^*})$ (ver Tablas 3 y 4). De la observación de este concepto se pueden extraer las siguientes conclusiones:

Si lo que prima es la adecuación de los cursos y los trabajos a los conocimientos del alumno, que corresponde a la última etiqueta, éste deberá hacer un curso de contabilidad y también de informática, siendo el puesto que más se ajusta a sus conocimientos el de secretario.

Si de lo que se trata es de tener en cuenta la preparación del alumno sobre temas específicos, no hay mucha diferencia entre unas opciones y otras, pero si tuviéramos que decantarnos por un curso elegiríamos el de contabilidad, y el puesto más adecuado sería el de secretario o vendedor.

Finalmente, si se pretende que el alumno estudie y posteriormente trabaje en aquello que le resulte interesante, deberá realizar un curso de informática y otro de contabilidad, para trabajar posteriormente como secretario.

También podemos ver que, teniendo en cuenta la opinión de éste individuo sobre distintos aspectos de los diversos cursos y observando la relación existente entre estos cursos y los tipos de trabajo, los oficios de empleado del hogar y camarero no son adecuados para él. Además hay una estrecha relación entre lo que es interesante y de nivel adecuado para esta persona.

$\underset{\sim}{Q}$	informática	contabilidad	mecánica
interesante	0.5	0.5	0.2
muy específico	0.7	0.8	0.1
de nivel adecuado a conoc.	0.8	0.3	0

Tabla 2

$\underset{\sim}{Q^*}$	informática	contabilidad	mecánica
interesante	0.5	0.5	0.2
muy específico	0.5	0.6	0.5
de nivel adecuado a conoc.	0.5	0.6	0.1

Tabla 3

$\underset{\sim}{Q_1^*}$	empl. del hogar	camarero	secretario	vend. de coches
interesante	0.5	0.5	0.8	0.5
muy específico	0.4	0.4	0.5	0.5
de nivel adec. a conoc.	0.4	0.4	0.9	0.5

Tabla 4

Referencias

[1] BURUSCO, A., FUENTES, R.; Estudio del retículo de conceptos L-Fuzzy, *IV Congreso Español sobre Tecnologías y Lógica Fuzzy*, Blanes, (1994)

[2] COUSOT, P., COUSOT, R.; Constructive versions of Tarski's fixed point theorems, *Pacific Journal of Mathematics*, **82**, (1979), 43-57.

[3] DUBOIS, D., PRADE, H.; *Fuzzy sets and systems. Theory and Applications*, Academic Press, (1980).

[4] TARSKI, A.; A Lattice Theoretical Fixpoint Theorem and its Applications, *Pathific J.Math.*, 5,(1955),285-310.

[5] WILLE, R.; Restructuring lattice theory: an approach based on hierarchies of concepts, *in Rival I.(ed.),Ordered Sets.* Reidel, Dordrecht-Boston (1982), 445-470.

Relaciones extendidas por agregación y principio de extensión

León González Angel Marín

Universidad Pública de Navarra.
Departamento de Matemática e Informática.
Campus Arrosadía. 31006 Pamplona
E-mail: amarin@si.upna.es

Resumen

En este trabajo se estudia la conexión entre las relaciones borrosas sobre subconjuntos borrosos obtenidas por agregación a partir de relaciones elementales y el resultado de aplicar el principio de extensión de Zadeh a la función de pertenencia de dichas relaciones.

Palabras Clave: *Relaciones borrosas, principio de extensión, agregación.*

1 Introducción

El estudio de relaciones extendidas o inducidas en $\mathcal{F}(X)$ a partir de relaciones borrosas elementales en X permite orientar desde una nueva perspectiva problemas clásicos en la teoría de subconjuntos borrosos como el de la indistinguibilidad y el de la clasificación borrosa.

Uno de los métodos de construcción de relaciones, que denominamos método de agregación, da un grado de relación entre subconjuntos borrosos de X mediante la agregación de los valores $R(x,y), A(x), B(y)$ con la conorma y norma triangulares S y T [1]. El principio de extensión de Zadeh [8] propone una extensión de las correspondencias de tipo funcional con ayuda de la conorma *max* y norma *min*. Dicha extensión fue posteriormente generalizada por Yager [6] a conormas y normas cualesquiera. También algunos otros autores han propuesto una versión para correspondencias dadas por una relación borrosa cualquiera [2].

La aplicación de la identidad de resolución [7] a la relación inicial en X permite, tras la segmentación, el uso de la representación formada por la familia de cortes de nivel. Esta cuestión se trata en el apartado segundo, donde además se establecen diversos resultados en torno a las operaciones de segmentación y agregación planteadas. El tercer apartado presenta las consecuencias de la aplicación de la identidad de resolución a cada uno de los componentes de la relación extendida por agregación. La reducción a relaciones y subconjuntos nítidos da lugar a un algoritmo de cálculo de $R^*(A,B)$ cuando el referencial X es finito. Finalmente en el último apartado se establece la conexión entre la relación extendida por agregación con $max - min$ y la función obtenida a partir del principio de extensión.

2 Agregación y segmentación

A partir de relaciones borrosas elementales R sobre un referencial X se pueden construir relaciones extendidas por agregación según el par $[max, min]$ mediante la expresión

$$R^*(A,B) = \sup_{x,y\in X} \min(R(x,y), A(x), B(y)) \qquad (1)$$

Como todos los subconjuntos borrosos, R^* queda determinado por la familia de cortes de nivel $\{(R^*)_\alpha\}$ de acuerdo con el principo de resolución de Zadeh [7], según el cual

$$R^* = \bigcup_{\alpha\in[0,1]} \alpha \cdot R^*_\alpha$$

que para $A, B \in \mathcal{F}(X)$ cualesquiera se interpretaría como

$$R^*(A,B) = \sup_{\alpha\in[0,1]} \min(\alpha, R^*_\alpha(A,B))$$

donde los cortes de nivel R^*_α se definen en la forma habitual, de modo que $R^*_\alpha(A,B) = 1$ cuando $R^*(A,B) \geq \alpha$ y 0 en caso contrario.

No obstante, existen otras representaciones de R^*.

Proposición 1 *La familia $\{(R_\alpha)^*\}_{\alpha\in[0,1]}$ es una representación de R^*.*

Demostración : Si tomamos A y B cualesquiera, por la definición dada en (1)

$$R^*(A,B) = \sup_{x,y} \min(R(x,y), A(x), B(y))$$
$$= \sup_{x,y} \min(\sup_{\alpha} \min(\alpha, R_\alpha(x,y)), A(x), B(y))$$

Por ser *min* norma semicontinua inferior se verificará

$$R^*(A,B) = \sup_{x,y} \sup_{\alpha} \min(\alpha, R_\alpha(x,y), A(x), B(y))$$
$$= \sup_{\alpha} \min(\alpha, \sup_{x,y} \min(R_\alpha(x,y), A(x), B(y)))$$
$$= \sup_{\alpha} \min(\alpha, (R_\alpha)^*(A,B))$$

y por tanto constituye una representación de R^*. ∎

Podemos señalar que las dos familias de relaciones intermedias $\{(R^*)_\alpha\}_{\alpha\in[0,1]}$ y $\{(R_\alpha)^*\}_{\alpha\in[0,1]}$ componen R^* y constituyen sendas cadenas respecto a la inclusión de relaciones, pero mientras la primera está formada por conjuntos ordinarios, la segunda la forman subconjuntos borrosos.

Proposición 2 *Si α_1 y α_2 son dos números reales pertenecientes a $[0,1]$ y tales que $\alpha_1 \geq \alpha_2$, entonces*

$$(R_{\alpha_1})^* \subseteq (R_{\alpha_2})^*$$

es decir para cada par de subconjuntos borrosos A, B se tiene que $(R_{\alpha_1})^*(A,B) \leq (R_{\alpha_2})^*(A,B)$

Demostración : En efecto, de cumplirse $R(x,y) \geq \alpha_1 \geq \alpha_2$ resulta que cuando $R_{\alpha_1}(x,y) = 1$ forzosamente $R_{\alpha_2}(x,y) = 1$. Por tanto

$$R_{\alpha_1} \subseteq R_{\alpha_2}$$

y por la isotonía de la extensión por agregación resulta $(R_{\alpha_1})^* \subseteq (R_{\alpha_2})^*$. ∎

Tratándose en cada caso de familias de subconjuntos ordinarios y borrosos, cabe la posibilidad de que para ciertos subconjuntos A y B, y algún real α se tenga

$$(R^*)_\alpha(A,B) \neq (R_\alpha)^*(A,B)$$

Para comprobarlo basta considerar los átomos borrosos $A = m_a$ y $B = n_b$ definidos como

$$m_a(x) = \begin{cases} m & si \quad x = a \\ 0 & si \quad x \neq a \end{cases} \qquad n_b(x) = \begin{cases} n & si \quad x = b \\ 0 & si \quad x \neq b \end{cases} \tag{2}$$

donde los valores m, n son tales que $m, n \in (0,1)$, mientras que a y b verifican $R(a,b) \neq 0$. La condición no es restrictiva ya que tales a y b existen en cualquier $R \neq \emptyset$. Al hacer $\alpha = R(a,b)$ se tendrá $R_\alpha(a,b) = 1$. En estas condiciones los valores de $(R_\alpha)^*(m_a, n_b)$ y $(R^*)_\alpha(m_a, n_b)$ son distintos. El primero de ellos resulta

$$(R_\alpha)^*(m_a, n_b) = \min(m,n)$$

y por tanto es distinto de 0 y 1, mientras que el segundo da o bien 0 o bien 1 ya que

$$(R^*)_\alpha(m_a, n_b) = \begin{cases} 1 & si \quad \min(m,n) \geq \alpha \\ 0 & si \quad no \end{cases}$$

Podemos ahora, por el contrario, preguntarnos si existe alguna familia restringida de pares (A,B) para los que se cumpla la igualdad anterior cualquiera que sea α.

Teorema 1 *Dada una relación R binaria borrosa no nula, se cumplirá la igualdad $(R_\alpha)^*(A,B) = (R^*)_\alpha(A,B)$ si y solo si los subconjuntos A, B verifican una de las siguientes condiciones:*

i) $\min(A(x), B(y)) = 0$ *para todo* x, y

ii) existe $r \in (0,1]$ *tal que*

$$\sup_{x,y} \{\min(A(x), B(y)) \mid R(x,y) = r\} = 1$$

$$\min(A(x), B(y)) = 0 \text{ si } R(x,y) > r$$

Demostración : Veamos en primer lugar que para las condiciones establecidas se verifica la igualdad. En el caso (i) la igualdad de ambos elementos a 0 resulta inmediata. Supongamos pues A y B elegidos a partir de un valor r según las condiciones establecidas. Vamos a calcular los valores de $(R^*)_\alpha(A,B)$ y de $(R_\alpha)^*(A,B)$. Para el primero partiremos de $R^*(A,B)$. Si denominamos $M(x,y)$ a $\min(A(x), B(y))$

$$R^*(A,B) = \sup_{x,y} \min(M(x,y), R(x,y))$$

Para el cálculo podemos distinguir tres tipos de pares según sea $R(x,y)$ mayor, igual o menor que r. El mayor de los tres supremos dará el valor buscado. Por las condiciones de A y B

$$\sup_{\substack{x,y \\ R(x,y)>r}} \min(M(x,y), R(x,y)) = 0$$

$$\sup_{\substack{x,y \\ R(x,y)=r}} \min(M(x,y), R(x,y)) = r$$

En el tercer caso se tiene

$$\sup_{\substack{x,y \\ R(x,y)<r}} \min(M(x,y), R(x,y)) \leq \sup_{\substack{x,y \\ R(x,y)<r}} \min(M(x,y), r)$$

Aplicando la distributividad, por ser *min* norma semicontinua inferiormente, el segundo término resultaría

$$\min\Big(\sup_{\substack{x,y \\ R(x,y)<r}} M(x,y), r\Big) = n \leq r$$

Como consecuencia de los tres resultados obtenidos, se deduce que

$$R^*(A,B) = \max(0, r, n) = r \tag{3}$$

Una vez obtenido $R^*(A,B)$, analicemos los posibles valores de $(R^*)_\alpha(A,B)$. En el caso de $\alpha > r$, según (3) $R^*(A,B) < \alpha$ y se tendrá $(R^*)_\alpha(A,B) = 0$. A su vez en dicho caso para $(R_\alpha)^*(A,B)$ tenemos

$$(R_\alpha)^*(A,B) = \sup_{\substack{x,y \\ R(x,y)\geq\alpha}} \min(A(x), B(y)) = 0$$

ya que los pares en que $R(x,y) \geq \alpha$, por la elección inicial de α cumplen $R(x,y) > r$ y entonces por la segunda de las condiciones de (ii) todos los términos son nulos. Al ser en este caso $(R^*)_\alpha(A,B)$ y $(R_\alpha)^*(A,B)$ nulos se cumple la igualdad del enunciado.

En el caso de que sea $\alpha \leq r$, los pares (x,y) tales que $R(x,y) = r$ cumplen $R(x,y) \geq \alpha$. Por tanto al calcular $(R_\alpha)^*(A,B)$ tenemos

$$(R_\alpha)^*(A,B) = \sup_{\substack{x,y \\ R(x,y)\geq\alpha}} \min(A(x), B(y)) \geq \sup_{\substack{x,y \\ R(x,y)=r}} \min(A(x), B(y)) = 1$$

por la primera de las condiciones de la hipótesis (ii). Luego $(R_\alpha)^*(A,B) = 1$. Como además $R^*(A,B) = r \geq \alpha$, queda $(R^*)_\alpha(A,B) = 1$ y se cumple la igualdad también en este caso.

Para demostrar el recíproco, supongamos que para cualquier valor de $\alpha \in [0,1]$ se cumple la igualdad $(R_\alpha)^*(A,B) = (R^*)_\alpha(A,B)$. Como la relación del segundo término es nítida solo toma los valores 0, 1. Así que para cualquier α se tiene $(R_\alpha)^*(A,B) \in \{0,1\}$. Además, según la proposición anterior, el conjunto de valores $\{(R_\alpha)^*(A,B)\}_{\alpha\in[0,1]}$ forma una cadena.

Consideremos ahora el conjunto de reales $\{\alpha|(R_\alpha)^*(A,B)=1\}$. Si fuera vacío, se tendría para todo α

$$(R^*)_\alpha(A,B) = (R_\alpha)^*(A,B) = 0$$

Como consecuencia $R^*(A,B) = 0$, en cuyo caso o bien $\min(A(x),B(y)) = 0$ para todo x,y con lo que se cumpliría la condición (i), o bien R es nula, lo que es imposible por hipótesis.

Cuando dicho conjunto sea no vacío, tomamos $\delta = \sup\{\alpha|(R_\alpha)^*(A,B)=1\}$. En tal caso se tendrá

$$(R_\alpha)^*(A,B) = 0 \quad \text{para } \alpha > \delta$$
$$(R_\alpha)^*(A,B) = 1 \quad \text{para } \alpha \leq \delta$$

La primera expresión equivaldría

$$\sup_{\substack{x,y \\ R(x,y)\geq\alpha}} \min(A(x),B(y)) = 0 \quad \text{para } \alpha > \delta$$

de donde se deduce la segunda condición en (ii) para A y B ya que

$$\min(A(x),B(y)) = 0 \quad \text{para } R(x,y) > \delta$$

Por otro lado particularizando la segunda expresión para $\alpha = \delta$ se tiene

$$\sup_{\substack{x,y \\ R(x,y)\geq\delta}} \min(A(x),B(y)) = 1$$

Como para $R(x,y) > \delta$, según se ha visto, $\min(A(x),B(y))$ se anula, forzosamente se tiene

$$\sup_{\substack{x,y \\ R(x,y)=\delta}} \min(A(x),B(y)) = 1$$

Por tanto al hacer $r = \delta$, se cumplen las dos condiciones de A y B. ∎

Los siguientes corolarios proporcionan ejemplos de subconjuntos en los que se cumplen estas condiciones.

Corolario 1 *Sea R una relación binaria borrosa y A,B dos subconjuntos ordinarios, entonces*

$$(R^*)_\alpha(A,B) = (R_\alpha)^*(A,B)$$

Demostración : Si A o B es vacío se cumple para todo x,y que $\min(A(x),B(y)) = 0$ y por la condición (i) del teorema se verifica la igualdad.

Si A y B son no vacíos podemos tomar el valor

$$r = \sup\{R(x,y)|x \in A, \quad y \in B\}$$

Para dicho valor tendremos que si $R(x,y) > r$ no se cumple $x \in A$, $y \in B$, es decir $\min(A(x),B(y)) = 0$.

Por otro lado, si $R(x,y) = r$ entonces $A(x) = B(y) = 1$, y $\min(A(x),B(y)) = 1$. Luego

$$\sup_{x,y}\{\min(A(x),B(y))|R(x,y) = r\} = 1$$

Como se cumplen las dos condiciones de (ii) se verifica la igualdad. ∎

Para cualquier relación R podemos encontrar también subconjuntos borrosos que verifiquen la igualdad.

Corolario 2 *Sea R una relación binaria borrosa y A,B dos subconjuntos borrosos normales tales que existe x_0,y_0 con $A(x_0) = B(y_0) = 1$ y $R(x_0,y_0) = hgt(R)$, entonces*

$$(R^*)_\alpha(A,B) = (R_\alpha)^*(A,B)$$

Demostración : Como A,B son normales no serán vacíos. Entonces si se considera $r = \sup\{R(x,y)|A(x) = B(y) = 1\}$ por la condición establecida $r = hgt(R)$. En tal caso no puede darse $R(x,y) > r$ y cuando $R(x,y) = r$ por la construcción se tiene $A(x) = B(y) = 1$, con lo que

$$\sup_{x,y}\{\min(A(x),B(y))|R(x,y) = r\} = 1$$

Luego se cumplen las condiciones de la proposición. ∎

3 Doble resolución y cálculo de valores

Aplicando el principio de resolución a los subconjuntos borrosos A y B y teniendo en cuenta las propiedades de R^* se obtenía en [1] la expresión

$$R^*(A,B) = \sup_{\alpha\in[0,1]} \min(\alpha, R^*(A_\alpha,B_\alpha)) \qquad (4)$$

Podemos ahora utilizar la resolución en cortes de R^* para obtener a partir de ella una nueva expresión que reduzca el valor $R^*(A,B)$ a un cálculo con subconjuntos y relaciones ordinarias.

Proposición 3 *Dada una relación binaria borrosa R y dos subconjuntos borrosos A y B de X, la relación extendida R^* obtenida con el par canónico $max-min$ cumple*

$$R^*(A,B) = \sup_{\alpha\in[0,1]} \min(\alpha, (R_\alpha)^*(A_\alpha,B_\alpha))$$

Demostración : Sustituyendo en (4) la resolución de R^* en cortes tendremos, siendo $\alpha,\beta \in [0,1]$

$$R^*(A,B) = \sup_\alpha \min(\alpha, \sup_\beta \min(\beta, (R^*)_\beta(A_\alpha,B_\alpha)))$$

Como A_α, B_α son subconjuntos ordinarios, se verifica para ellos según el Cor.1 del Teorema 1

$$(R^*)_\beta(A_\alpha,B_\alpha) = (R_\beta)^*(A_\alpha,B_\alpha)$$

lo que supone

$$\begin{aligned} R^*(A,B) &= \sup_\alpha \min(\alpha, \sup_\beta \min(\beta, (R_\beta)^*(A_\alpha,B_\alpha))) \\ &= \sup_{\alpha,\beta} \min(\min(\alpha,\beta), (R_\beta)^*(A_\alpha,B_\alpha))) \end{aligned}$$

Restringiendo el supremo a los pares tales que $\alpha = \beta$ quedaría un valor menor

$$R^*(A,B) \geq \sup_{\alpha} \min(\alpha, (R_\alpha)^*(A_\alpha, B_\alpha))$$

Por otro lado, si $\alpha \leq \beta$ se tendrá

$$(R_\beta)^*(A_\alpha, B_\alpha)) \leq (R_\alpha)^*(A_\alpha, B_\alpha))$$

de donde

$$\sup_{\substack{\alpha,\beta \\ \alpha \leq \beta}} \min(\min(\alpha,\beta), (R_\beta)^*(A_\alpha, B_\alpha))) \leq \sup_{\alpha} \min(\alpha, (R_\alpha)^*(A_\alpha, B_\alpha))$$

y si $\alpha > \beta$ quedará

$$\sup_{\substack{\alpha,\beta \\ \alpha > \beta}} \min(\min(\alpha,\beta), (R_\beta)^*(A_\alpha, B_\alpha))) \leq \sup_{\beta} \min(\beta, (R_\beta)^*(A_\beta, B_\beta))$$

Como los dos últimos términos de ambas desigualdades son idénticos, al obtener el supremo global resulta

$$R^*(A,B) \leq \sup_{\alpha} \min(\alpha, (R_\alpha)^*(A_\alpha, B_\alpha))$$

Este resultado junto con el anterior da lugar a la igualdad del enunciado. ∎

El resultado obtenido permite ofrecer una nueva formulación de $R^*(A,B)$. Basta observar que $(R_\alpha)^*(A_\alpha, B_\alpha)$ solo puede ser 0 o 1 y que para los distintos α la familia resultante forma cadena, y como consecuencia tendríamos que la igualdad de la proposición se convierte en

$$R^*(A,B) = \sup\{\alpha | (R_\alpha)^*(A_\alpha, B_\alpha) = 1\}$$

Esta última condición para α equivale a

$$\sup_{x,y} \min(R_\alpha(x,y), A_\alpha(x), B_\alpha(y)) = 1$$

entonces si llamamos $T_\alpha = \{(x,y) \mid A(x) \geq \alpha,\ B(y) \geq \alpha\}$ podríamos reformular la igualdad como

$$R^*(A,B) = \sup\{\alpha \mid T_\alpha \neq \emptyset,\ \sup_{(x,y)\in T_\alpha} R(x,y) \geq \alpha\} \tag{5}$$

En esta formulación el valor de $R^*(A,B)$ se obtiene variando progresivamente el nivel de corte aplicado hasta que una condición booleana deje de cumplirse. Gracias a ella se puede describir con facilidad un algoritmo para el cálculo de $R^*(A,B)$ en el caso en que X es finito. En este caso la relación R se puede representar mediante una matriz cuadrada M_R que contenga los valores $R(x,y)$. Partiendo de la ecuación (5) obtendríamos el siguiente algoritmo.

Algoritmo para el cálculo de $R^*(A,B)$

1. Formar una lista con los grados existentes en A y B tales que sean menores que $m = \min(hgt(R), hgt(A), hgt(B))$, y ponerla en orden creciente $\{\alpha_1, \ldots, \alpha_r\}$.
2. $i \leftarrow 0$.
3. Repetir
   (a) $i \leftarrow i + 1$
   (b) Determinar los conjuntos de valores $\{f_i\}$ tales que $A(f_i) < \alpha_i$ y $\{c_i\}$ tales que $B(c_i) < \alpha_i$.
   (c) A partir de M_R formar una submatriz M_{α_i}, eliminando las filas $\{f_i\}$ y las columnas $\{c_i\}$.
   (d) $s \leftarrow \max M_{\alpha_i}$.

   hasta que $i = r$ ó $s < \alpha_i$
4. Si $i = r$ entonces $R^*(A,B) = \alpha_r$ si no $R^*(A,B) = \alpha_{i-1}$.

El algoritmo es correcto. No puede darse $s < \alpha_1$ y por tanto un resultado que apunte a un inexistente α_0 ya que para α_1 se tiene $M_{\alpha_1} = M_R$ de donde resulta que $s = \sup_{x,y} R(x,y) = hgt(R)$. Por la elección de los elementos se tendría que $\alpha_1 \leq m \leq s$, lo que contradice la hipótesis inicial.

Además la submatriz M_{α_i} siempre existe y por tanto su máximo, ya que en el caso final de α_r, que coincide con $hgt(A)$ o $hgt(B)$, todavía quedarán filas o columnas sin eliminar.

Ejemplo : Si tomamos en $X = \{a,b,c\}$ los subconjuntos $A = \{0.7/a, 0.8/b, 1/c\}$ y $B = \{0.2/a, 1/b, 0.5/c\}$, y consideramos R tal que

$$M_R = \begin{bmatrix} 0.7 & 0.3 & 0.4 \\ 1 & 0 & 0.8 \\ 0.6 & 0.5 & 0.3 \end{bmatrix}$$

por el algoritmo convencional del producto matricial, expuesto en [1], obtenemos $R^*(A,B) = 0.5$. Al aplicar el algoritmo anterior resulta en el paso 1 la lista de valores $\{0.2, 0.5, 0.7, 0.8, 1\}$. Para $\alpha_1 = 0.2$ la submatriz, que es M_R, tiene por supremo 1 y las condiciones de parada del ciclo no se cumplen. Para $\alpha_2 = 0.5$ la submatriz, formada por la segunda y tercera columnas, tiene supremo 0.8, por lo que tampoco se cumplen. En el caso de $\alpha_3 = 0.7$ el supremo es 0.5 con lo que se cumple la condición de parada. Al aplicar el punto 4 del algoritmo resulta que $R^*(A,B) = \alpha_2 = 0.5$.

4 Agregación y principio de extensión

La extensión de operaciones numéricas al marco borroso puede basarse en el llamado principio de extensión de Zadeh [8], según el cual dada una función $f : X \rightarrow Y$ podemos construir para un subconjunto borroso A de X un subconjunto borroso en Y mediante la función

$$B(y) = \begin{cases} \sup_{x \in f^{-1}(y)} A(x) & si \quad f^{-1}(y) \neq \emptyset \\ 0 & si \quad f^{-1}(y) = \emptyset \end{cases}$$

Con dicho método de una función ordinaria se obtiene una función con argumentos y valores borrosos, mediante la aplicación de una agregación selectiva. La

posibilidad de diversificar la agregación según el tipo de normas y conormas escogidas [6] ha dado lugar a una familia de métodos o principios de extensión semejantes.

Con este tipo de métodos se puede construir también una función g^p de $\mathcal{F}(X) \times \mathcal{F}(Y)$ en $\mathcal{F}(Z)$ a partir de una función dada $g : X \times Y \to Z$. Y de igual modo podemos aplicarlos, dada una relación R borrosa en X, a su función de pertenencia $R : X \times X \to [0,1]$. Al aplicar el principio de extensión a la función R, según la conorma extendida S y la norma T, se obtendrá la función $R^p_{ST} : \mathcal{F}(X) \times \mathcal{F}(X) \to \mathcal{F}([0,1])$. Esta función extendida daría lugar, para $A, B \in \mathcal{F}(X)$, al conjunto borroso $C = R^p_{ST}(A,B) = \mathcal{F}([0,1])$, que en virtud del principio de extensión quedaría definido por

$$R^p_{ST}(A,B)(z) = \begin{cases} \underset{(x,y)\in R^{-1}(z)}{S} T(A(x),B(y)) & si \quad R^{-1}(z) \neq \emptyset \\ 0 & si \quad R^{-1}(z) = \emptyset \end{cases}$$

y en la versión canónica del principio de extensión, para cada uno de los dos casos

$$R^p(A,B)(z) = \begin{cases} \sup_{(x,y)\in R^{-1}(z)} \min(A(x),B(y)) \\ 0 \end{cases}$$

Una vez fijado el par $[S,T]$ que determina el método de extensión empleado, el subconjunto $R^p_{ST}(A,B)$ borroso de $[0,1]$ resultante ofrece un grado borroso de la relación a través de R entre los subconjuntos A y B. De dicho subconjunto es posible obtener por agregación valores en $[0,1]$. A la vista de las operaciones con las que se define el subconjunto, cabe preguntarse si el valor que pueda obtenerse tiene que ver con $R^*(A,B)$, donde R^* es la relación extendida por agregación según el par $[S,T]$.

Para la obtención de dicho valor proponemos inicialmente una agregación sobre los valores $z \in [0,1]$, tomando como conorma iterada la misma conorma S utilizada en el principio de extensión.

Proposición 4 *Dada una una relación R binaria borrosa no vacía, A, B subconjuntos borrosos de X, sean $R^p_{ST}(A,B)$ el subconjunto borroso obtenido mediante el principio de extensión y $\mathbf{U}^*$ la relación extendida de la universal $\mathbf{U} = X \times X$ (ambas extensiones según el par $[S,T]$), entonces*

$$\underset{z\in[0,1]}{S} R^p_{ST}(A,B)(z) = \mathbf{U}^*(A,B)$$

Demostración : Al aplicar la conorma S a $C = R^p_{ST}(A,B)$ tendremos

$$\begin{aligned} \underset{z\in[0,1]}{S} C(z) &= \underset{z\in[0,1]}{S} R^p_{ST}(A,B)(z) \\ &= \underset{z\in[0,1]}{S} \underset{z=R(x,y)}{\underset{x,y}{S}} T(A(x),B(y)) \\ &= \underset{x,y}{S} T(A(x),B(y)) \end{aligned}$$

ya que al tener cada par x, y asociado un valor $z = R(x,y)$, todos resultarán agregados.

Dado que $\mathbf{U}(x,y) = 1$ para todo x, y, este último término puede reescribirse como $\mathbf{U}^*(A,B)$, resultando la igualdad del enunciado. ∎

Dicho resultado se obtiene con independencia de la conorma y norma escogidas. En particular, para el par $[max, min]$ se cumple

$$\sup_{z\in[0,1]} R^p(A,B)(z) = \mathbf{U}^*(A,B) \qquad (6)$$

En este último caso el resultado de la agregación en la relación R resulta

$$\sup_{x,y} \min(A(x),B(y))$$

que es un valor independiente de R, lo que hace desaconsejable esta expresión como una nueva extensión de R. En realidad el principio de extensión propicia una agregación segmentada a través de las imágenes de la función R, cuyo papel en la agregación global desaparece.

La inoperancia de la agregación global sobre el subconjunto $R^p(A,B)$ nos induce al estudio de agregaciones parciales, de las que vamos a ver que se pueden emparentar a través de los cortes de nivel con la relación extendida R^*.

Proposición 5 *Sea R una relación binaria borrosa no vacía, A y B dos subconjuntos borrosos en X y $R^p(A,B)$ el subconjunto borroso en $[0,1]$ obtenido por el principio de extensión canónico, entonces*

$$(R_\alpha)^*(A,B) = \sup_{z\geq\alpha} R^p(A,B)(z)$$

Demostración : Dado que, por definición de $(R_\alpha)^*$

$$(R_\alpha)^*(A,B) = \sup_{x,y} \min(A(x),B(y),R_\alpha(x,y))$$

podemos expresarlo

$$(R_\alpha)^*(A,B) = \sup_{\substack{x,y \\ R(x,y)\geq\alpha}} \min(A(x),B(y))$$

Por otro lado se tiene que

$$R^p(A,B)(z) = \sup_{\substack{x,y \\ R(x,y)=z}} \min(A(x),B(y))$$

o bien 0 cuando $R^{-1}(z) = \emptyset$.

Teniendo en cuenta que para cualquier función $M(x,y)$ se tendría la igualdad

$$\sup_{\substack{x,y \\ R(x,y)\geq\alpha}} M(x,y) = \sup_{z\geq\alpha} \sup_{\substack{x,y \\ R(x,y)=z}} M(x,y)$$

sustituyendo $M(x,y)$ por $\min(A(x),B(y))$ se deduce el resultado. ∎

De esta proposición resulta como caso particular, cuando $\alpha = 0$, el resultado obtenido en (6). Se pueden,

de igual forma, obtener algunos otros casos particulares de interés.

Corolario 1 *Sea R una relación binaria borrosa, A y B dos subconjuntos borrosos en X y $R^p(A,B)$ el subconjunto borroso en $[0,1]$ obtenido por el principio de extensión, entonces*

i) $(R_1)^*(A,B) = R^p(A,B)(1)$.

ii) $R^*(A,B) = R^p(A,B)(1)$, *si R es ordinaria.*

Demostración : i) Basta ver que al hacer $\alpha = 1$ queda

$$(R_1)^*(A,B) = \sup_{z \geq 1} R^p(A,B)(z) = R^p(A,B)(1)$$

ii) Como $R_1 = R$ si R es ordinaria, basta aplicar el apartado anterior. ∎

El resultado de la proposición permite también relacionar R^* y R^p en el caso general.

Teorema 2 *Sea R una relación binaria borrosa, A y B dos subconjuntos borrosos en X y $R^p(A,B)$ el subconjunto borroso en $[0,1]$ obtenido por el principio de extensión, entonces*

$$R^*(A,B) = \sup_{\alpha} \min(\alpha, \sup_{z \geq \alpha} R^p(A,B)(z))$$

Demostración : En la Prop.1 se demostraba que $\{(R_\alpha)^*\}_{\alpha \in [0,1]}$ era una representación de R^*, es decir

$$R^*(A,B) = \sup_{\alpha} \min(\alpha, (R_\alpha)^*(A,B))$$

sustituyendo el valor obtenido en la proposición anterior se obtiene el resultado. ∎

5 Conclusiones

Se ha puesto de manifiesto que las relaciones extendidas R^* pueden ser obtenidas mediante el empleo del principio de extensión, por lo que podemos declarar que la extensión por agregación es formulable en términos de dicho principio. La cuestión recíproca puede plantearse formulando el principio de extensión mediante el empleo de relaciones extendidas por agregación, dando lugar así a nuevos enfoques sobre el mismo. Así pues, a través del último resultado puede desarrollarse un estudio de las operaciones entre relaciones extendidas que encuentre su apoyo en los resultados de la aritmética borrosa.

Referencias

[1] González L., Marín A., Extending fuzzy relations, *Fuzzy Sets and Systems* **69** (1995) 157-169.

[2] Keyun Q., Yang X., An extended form of the extension principle for fuzzy sets, *Busifal* 66-71

[3] Marín A., Sobre redundancia y similaridad, Tesis doctoral, Univ. Pública de Navarra, 1994.

[4] Ovchinnikov S., The duality principle in fuzzy set theory, *Fuzzy Sets and Systems* **42** (1991) 133-144.

[5] Nguyen H.T. A note on the extension principle for fuzzy sets, *Journal of Math. Anal. and Appl.* **64** (1978) 369-380.

[6] Yager R.R., A characterization of the extension principle, *Fuzzy Sets and Systems* **18** (1986) 205-217.

[7] Zadeh L.A., Similarity relations and fuzzy orderings, *Information Sciences* **3** (1971) 177-200.

[8] Zadeh L.A., The concept of linguistic variable and its application to approximate reasoning, *Information Sciences* **8** (1975) 199-249.

Construcción de correspondencias borrosas entre subconjuntos

León González Angel Marín

Universidad Pública de Navarra.
Departamento de Matemática e Informática.
Campus Arrosadía. 31006 Pamplona
E-mail: amarin@si.upna.es

Resumen

En este trabajo se presenta una forma de construcción sistemática de relaciones entre subconjuntos borrosos a partir de un análisis del concepto de correspondencia entre subconjuntos inducida por una relación binaria entre elementos.

Palabras Clave: *Correspondencia, relaciones borrosas, extensión, agregación, extensión a nivel óptimo.*

1 Introducción

El desarrollo de aplicaciones de la lógica borrosa a la teoría de control, los sistema de de clasificación o las bases de datos relacionales ha obligado a plantearse el problema de la extensión de relaciones borrosas en X a relaciones en $\mathcal{F}(X)$. El problema admite diversos enfoques. Desde un punto de vista conjuntista algunas soluciones han sido tratadas por Badard [2], Brink et al. [5] y los autores [6][7]. Sin embargo no se han llegado a estudiar ni las relaciones entre los diversos modos de extensión ni un método que permita generar de forma sistemática construcciones o formas de correspondencia borrosa entre subconjuntos borrosos.

En trabajos anteriores hemos considerado las características y aplicaciones de las llamadas extensiones por agregación [6] y extensiones a nivel óptimo [7]. Aquí pretendemos encuadrar de modo general el método, desarrollando para subconjuntos borrosos construcciones semejantes a las formas de correspondencia entre subconjuntos ordinarios. La amplitud y diversidad de las posibilidades aconsejan empezar restringiendo el ámbito de estudio, aplazando cuestiones como la persistencia de las propiedades de la relación original y las posibles aplicaciones que de ahí se derivan. En este trabajo nuestro propósito es fundamentalmente analítico y orientado a la formulación sistemática de construcciones que revelen potenciales grados de relación entre subconjuntos borrosos a partir de una relación borrosa inicial.

Los temas presentados están organizados de la forma siguiente: En el segundo apartado se formulan a partir de una relación binaria diversos tipos de correspondencia entre subconjuntos. Cada tipo se caracteriza mediante una expresión en el cálculo de predicados y mediante condiciones booleanas de naturaleza conjuntista. A continuación, en el tercer apartado, se propone la construcción por agregación, mediante normas y conormas iteradas, de un grado de relación entre subconjuntos para las distintas formas de correspondencia. En el cuarto apartado se construye para cada correspondencia un nuevo grado mediante la familia de relaciones ordinarias que se obtiene de la inicial mediante cortes de nivel. Finalmente, consideramos las construcciones que derivan de la complementación mediante la negación ordinaria, y la conexión con la construcción obtenida para la relación ordinaria.

2 Correspondencias entre conjuntos

En esta sección se estudian las correspondencias entre subconjuntos ordinarios que resultan inducidas por las correspondencias puntuales, lo que constituye un punto de partida para lograr un estudio global y homogéneo de la formación de correspondencias no puntuales en el caso de subconjuntos borrosos.

Para ello partimos de dos conjuntos X e Y ordinarios, no necesariamente finitos. Una relación binaria ordinaria R en el conjunto $X \times Y$ permite establecer correspondencias entre los elementos de X e Y. Dos elementos $x \in X$ e $y \in Y$ están en correspondencia si se cumple la condición xRy, o bien si $(x, y) \in R$. La correspondencia elemental puede ser también representada por un predicado binario, y como tal admite las interpretaciones que le sean transferidas a través de la definición de R.

Cuando pretendemos generalizar la correspondencia elemental establecida a través de R a subconjuntos de $A \subseteq X$ y de $B \subseteq Y$ y dar significado a la expresión ARB, obtenemos una serie de casos con independencia de las características de R. Si asociamos la pertenencia a los subconjuntos A, B con predicados unarios $A(x), B(y)$, aparecerá una expresión del cálculo de predicados cuya cuantificación puede servir para enumerar distintos casos.

Tipos de correspondencia

Para dos subconjuntos $A \subseteq X$ y $B \subseteq Y$ cualesquiera se puede establecer a través de R una correspondencia *mínima* si

$$\exists x \in A \quad \exists y \in B \text{ tal que } xRy$$

Dicha expresión puede reescribirse

$$\exists x \exists y [R(x, y) \wedge A(x) \wedge B(y)] \qquad (1)$$

si representamos la pertenencia mediante predicados unarios y la conjunción de predicados con $\wedge$. La expresión, que es una fórmula del cálculo de predicados, será verdadera o falsa en virtud de las características de R, A, B, y este resultado booleano define la correspondencia mínima entre A y B a través de R, cuya existencia denotaremos con $AR^{\vee}B$.

La correspondencia será *dirigida a derecha* cuando se cumpla

$$\forall x \exists y[R(x,y) \wedge A(x) \wedge B(y)] \qquad (2)$$

ya que en este caso cualquier elemento de A tiene correspondiente en B a través de R. Denotaremos este hecho mediante $AR^{>}B$.

De igual modo se tendría la correspondencia *dirigida a izquierda*. La expresión sería entonces

$$\exists x \forall y[R(x,y) \wedge A(x) \wedge B(y)] \qquad (3)$$

lo que viene a indicar que cualquier elemento de B tiene su correspondiente en A. Lo que se denotará como $AR^{<}B$. En el caso en que la correspondencia sea dirigida a izquierda y a derecha hablaremos de correspondencia mutua, que denotaremos $AR^{\bowtie}B$.

El último caso representa la correspondencia en su más alto grado, o correspondencia *máxima* ya que

$$\forall x \forall y[R(x,y) \wedge A(x) \wedge B(y)] \qquad (4)$$

señala que entre todos los elementos de A y B hay correspondencia, y lo denotaremos con $AR^{\vee}B$.

Expresión conjuntista de las correspondencias

Las condiciones, que anteriormente se han expresado como fórmulas del cálculo de predicados, pueden ser descritas mediante operaciones conjuntistas. Para ello se hace necesaria una serie de definiciones que describan los conjuntos y operaciones asociados a una correspondencia.

Dada una relación binaria R introduzcamos, para describir el nivel elemental, las nociones de posconjunto xR de $x \in X$ y de preconjunto Ry de $y \in Y$ de modo que

$$xR = \{y \in Y \mid xRy\} \qquad Ry = \{x \in X \mid xRy\}$$

En el caso de dos relaciones $R \subseteq X \times Y$ y $S \subseteq Y \times Z$ podemos formar correspondencias compuestas para $x \in X, z \in Z$ cuando existe $y \in Y$ a través del cual se logra la correspondencia, resultando

$$x(R \circ S)z \quad \equiv \quad \exists y[xRy \wedge ySz]$$

Estas correspondencias permiten definir la relación compuesta $R \circ S$, que en términos de preconjuntos y posconjuntos podría expresarse

$$R \circ S = \{(x,z) \mid xR \cap Sz \neq \emptyset\} \qquad (5)$$

Podemos destacar que no es ésta la única definición posible de composición, tal como muestran las definiciones alternativas de Bandler-Kohout [4] y De Baets-Kerre [3], que dan las relaciones siguientes

$$\begin{aligned} R \triangleleft S &= \{(x,z) \mid \emptyset \subset xR \subseteq Sz\} \\ R \triangleright S &= \{(x,z) \mid \emptyset \subset Sz \subseteq xR\} \\ R \circ S &= \{(x,z) \mid \emptyset \subset xR = Sz\} \end{aligned}$$

Por extensión, cabría aplicar estas definiciones de composición a un subconjunto $A \subseteq X$, representado por una relación unaria, y una relación binaria $R \subseteq X \times Y$. En tal caso tendríamos

$$A \circ R = \{y \in Y \mid A \cap Ry \neq \emptyset\} \qquad (6)$$

que viene a reflejar el posconjunto asociado al subconjunto A de X. De idéntica forma, el preconjunto asociado al subconjunto B de Y sería

$$R \circ B = \{x \in X \mid xR \cap B \neq \emptyset\} \qquad (7)$$

Y con el mismo criterio se podrían obtener las definiciones para $A \triangleleft R$, $A \triangleright R$ y $A \diamond R$.

Las definiciones introducidas representan una extensión de las nociones de preconjunto y posconjunto, puesto que $xR = \{x\} \circ R$ y $Ry = R \circ \{y\}$. Por otro lado, con estas definiciones estamos en condiciones de caracterizar mediante conjuntos las correspondencias mínima, dirigida a derecha, dirigida a izquierda y máxima entre subconjuntos establecidos en el apartado anterior.

Proposición 1 *Dada una relación R en $X \times Y$, y dos subconjuntos $A \subseteq X$ y $B \subseteq Y$, entonces*

i) $AR^{\vee}B$ si y solo si se cumple cualquiera de las condiciones equivalentes

$$A \cap (R \circ B) \neq \emptyset \qquad (A \circ R) \cap B \neq \emptyset \qquad R \cap (A \times B) \neq \emptyset$$

ii) $AR^{\bowtie}B$ si y solo si $A \subseteq R \circ B$ y $A \circ R \supseteq B$

iii) $AR^{\wedge}B$ si y solo si se cumple cualquiera de las condiciones equivalentes

$$A \subseteq R \triangleright B \qquad B \subseteq A \triangleleft R$$

La complejidad de las caracterizaciones concuerda con la progresiva globalidad de las correspondencias, ya que se tiene

mínima	$AR^{\vee}B$	$A \cap (R \circ B) \neq \emptyset$
derecha	$AR^{>}B$	$A \subseteq R \circ B$
máxima	$AR^{\wedge}B$	$A \subseteq R \triangleright B$

3 Formación por agregación de correspondencias borrosas

En la sección anterior se muestra cómo la expresión de las correspondencias entre subconjuntos inducidas por una relación en $X \times Y$ puede ser formulada en el cálculo de predicados ordinario. Una forma de incorporar la borrosidad a los distintos tipos de correspondencia consiste en considerar e interpretar dichas fórmulas en un cálculo de predicados con borrosidad.

Para establecer una valoración de estas fórmulas podemos utilizar, siguiendo a Gottwald [8] y otros autores, el siguiente conjunto de reglas: Si H_i representa una fórmula del cálculo de predicados borroso y $[[H_i]]$ constituye su valoración en el intervalo $[0,1]$

$$[[H_1 \wedge H_2]] = T([[H_1]], [[H_2]]) \qquad (8)$$

$$[[H_1 \vee H_2]] = S([[H_1]], [[H_2]]) \qquad (9)$$

$$[[\neg H]] = n([[H]]) \qquad (10)$$

donde T, S, n representan una norma, una conorma y una función de negación (cf.[1], [11]). Para los cuantificadores existencial y universal una primera interpretación consistiría en aplicar las reglas

$$[[\forall x H(x)]] = \inf_{a \in X} [[H(x/a)]] \qquad (11)$$

$$[[\exists x H(x)]] = \sup_{a \in X} [[H(x/a)]] \qquad (12)$$

donde $H(x/a)$ indica que la variable libre x se sustituye por un valor a del referencial. Se han formulado propuestas para interpretar estas fórmulas mediante normas y conormas extendidas por iteración. Estos operadores, introducidos por KLEMENT [9], representarían una generalización del *supremo* e *ínfimo*. Por ejemplo, en el caso de una aplicación iterada de la conorma S a los valores del predicado $H(x)$ sobre un conjunto $M = \{m_1, \ldots, m_r\}$ finito se tendría

$$\underset{m_i \in M}{S} H(x) =$$
$$S\,(H(m_1),\ S\,(\ldots,\ S\,(H(m_{r-1}), H(m_r))\ldots))$$

El empleo de normas y conormas, extendidas en este sentido, daría lugar a otras dos reglas semejantes

$$[[\forall x H(x)]] = \underset{a \in X}{T} [[H(x/a)]] \qquad (13)$$

$$[[\exists x H(x)]] = \underset{a \in X}{S} [[H(x/a)]] \qquad (14)$$

La aplicación recursiva de este conjunto de reglas permite dar interpretaciones multivaluadas, es decir valores en el intervalo $[0, 1]$, de las fórmulas del cálculo de predicados. En particular podemos establecer valoraciones de las fórmulas que expresaban cada uno de los tipos de correspondencia.

En la sección anterior establecimos para una relación $R \subseteq X \times Y$ una correspondencia booleana ARB, que llamábamos máxima, entre $A \subseteq X$ y $B \subseteq Y$ cuando se cumplía

$$\exists x \exists y [R(x, y) \wedge A(x) \wedge B(y)]$$

Si consideramos relaciones y subconjuntos borrosos sobre X, de las fórmulas que expresaban cada uno de los tipos de correspondencia se puede hacer una valoración en la lógica multivaluada con arreglo a las reglas anteriormente presentadas. Entendiendo que $R(x, y), A(x), B(y)$ representan ahora el grado de pertenencia a los respectivos subconjuntos, podemos expresar el grado de correspondencia mínima entre A y B como una valoración y por tanto

$$\begin{aligned} R^*_{\vee}(A, B) &= [[\ \exists x \exists y [R(x, y) \wedge A(x) \wedge B(y)]\ ]] \\ &= \sup_{x \in X} \sup_{y \in X} T(R(x, y), T(A(x), B(y))) \end{aligned}$$

si aplicamos las reglas (8) y (12). Al aplicar una conorma S, según la regla (14) se obtendrá

$$R^*_{\vee}(A, B) = \underset{x \in X}{S} \underset{y \in X}{S} T(R(x, y), T(A(x), B(y))) \qquad (15)$$

En una primera interpretación el valor obtenido refleja la agregación mediante el supremo de los grados de pertenencia de la relación $R \cap (A \times B)$. La actuación se realiza a través de dos operadores: la norma T que sería una interpretación de la intersección y del producto cartesiano, y la conorma S que permite la agregación.

En la interpretación clásica o canónica la norma es $T = \min$ y $S = \max$ y se obtiene por agregación el valor

$$R^*_{\vee}(A, B) = \sup_{x, y \in X} \min(R(x, y), A(x), B(y)) \qquad (16)$$

pero en general el valor de la correspondencia mínima depende del par de operadores $[S, T]$ escogidos.

Este tipo de construcción ha sido estudiado en [6] como relación extendida según el triple $t = [S, T, T']$, en la que se desdobla la norma asociada a la intersección y al producto cartesiano haciendo

$$R^*_t(A, B) = \underset{x, y \in X}{S} T(R(x, y), T'(A(x), B(y)))$$

Aplicando este mismo método a las fórmulas (2)-(4) resultan otras tres correspondencias borrosas

$$\begin{aligned} R^*_{>}(A, B) &= \underset{x \in X}{T'} \underset{y \in X}{S} T(R(x, y), T(A(x), B(y))) \\ R^*_{<}(A, B) &= \underset{x \in X}{S} \underset{y \in X}{T''} T(R(x, y), T(A(x), B(y))) \\ R^*_{\wedge}(A, B) &= \underset{x, y \in X}{T''} T(R(x, y), T(A(x), B(y))) \end{aligned}$$

definidas, al interpretar $\wedge$ según la norma T y los cuantificadores según la conorma S y la norma T', por las agregaciones

4 Correspondencias obtenidas por segmentación

Junto a la expresión en el cálculo de predicados de las correspondencias entre subconjuntos inducidas por una relación en $X \times Y$, la Prop.1 muestra condiciones conjuntistas booleanas equivalentes. En estas expresiones podemos hacer que una de las variables cuantificadas quede implícita en las operaciones conjuntistas

$$\begin{aligned} \exists x C_{\vee}(R, A, B) &\equiv \exists x \in X (x \in A \cap (R \circ B)) \\ \forall x C_{>}(R, A, B) &\equiv \forall x \in X (x \in A \Rightarrow x \in R \circ B) \\ \forall x C_{\wedge}(R, A, B) &\equiv \forall x \in X (x \in A \Rightarrow x \in R \triangleright B) \end{aligned}$$

Podemos obtener nuevas construcciones del grado de correspondencia usando conjuntos borrosos en vez de ordinarios y obteniendo las correspondientes condiciones booleanas $\tilde{C}(R, A, B)$.

$$\begin{aligned} \tilde{C}_{\vee}(R, A, B) &\equiv (A \cap (R \circ B))(x) \neq 0 \\ \tilde{C}_{\bowtie}(R, A, B) &\equiv A(x) \leq (R \circ B)(x), \\ &\quad\ B(y) \leq (A \circ R)(y) \\ \tilde{C}_{\wedge}(R, A, B) &\equiv A(x) \leq (R \triangleright B)(x) \end{aligned}$$

Cuando R, A, B satisfagan dichas condiciones hablaremos de interrelación entre los subconjuntos según correspondencia mínima, mutua o máxima. Las

interrelaciones, que son relaciones ordinarias en $\mathcal{P}(X)$ quedarán definidas del modo siguiente

$$\begin{aligned} A R^{+}_{\vee} B &\equiv \exists x \in X \quad (A \cap (R \circ B)(x) \neq 0 \\ A R^{+}_{\bowtie} B &\equiv A \subseteq R \circ B, \quad B \subseteq A \circ R \\ A R^{+}_{\wedge} B &\equiv A \subseteq R \triangleright B \end{aligned}$$

Todavía es posible diversificar las construcciones según sean las normas y conormas asociadas a las operaciones $\cap, \circ, \triangleright$. La interpretación canónica será *min* para la intersección, $sup - min$ para la composición y, siguiendo a Baets-Kerre, en el caso de $\triangleright$

$$\begin{aligned} (R \triangleright S)(x,z) &= \min(\inf_{y}(S(y,z) \rightarrow R(x,y)), \\ &\qquad \sup_{y} R(x,y), \sup_{y} S(y,z)) \end{aligned}$$

donde $\rightarrow$ representa el operador implicación gödeliana, para el que

$$a \rightarrow b = \begin{cases} 1 & si \quad a \leq b \\ 0 & si \quad a > b \end{cases}$$

Si consideramos la relación R en X y la familia de cortes de nivel $\{R_\alpha\}$ donde $\alpha \in [0,1]$, podemos aplicar a cada una de las relaciones ordinarias R_α las condiciones que definen cada una de las correspondencias. En el caso de una relación ordinaria las distintas interrelaciones admiten las caracterizaciones que muestra la siguiente proposición.

Proposición 2 *Sea R una relación binaria ordinaria en X y A, B dos subconjuntos borrosos entonces*

i) $A R^{+}_{\vee} B \Leftrightarrow \sup_{x,y}\{\min(A(x), B(y)) \mid xRy\} \neq 0$

ii) $A R^{+}_{\bowtie} B \Leftrightarrow$
$\forall x \quad A(x) \neq 0 \Rightarrow A(x) \leq \sup_{y \in xR} B(y)$
$\forall y \quad B(y) \neq 0 \Rightarrow B(y) \leq \sup_{x \in Ry} A(x)$

iii) $A R^{+}_{\wedge} B \Leftrightarrow$
$\forall x, y \quad A(x) \neq 0, B(y) \neq 0 \Rightarrow xRy$
$hgt(A) = hgt(B)$ *siendo* $hgt(M) = \sup_x M(x)$

Por otro lado, el resultado anterior permite deducir que para cualquiera de las tres correspondencias consideradas la familia de interrelaciones $\{R^{+}_{\alpha}\}_{\alpha \in [0,1]}$ esta ordenada linealmente según un orden inducido por el de $[0,1]$.

Proposición 3 *Sea R una relación binaria borrosa, entonces dados $\alpha, \beta \in [0,1]$ tales que $\alpha \leq \beta$ resulta, cualquiera que sea el tipo de interrelación considerada, que $R^{+}_{\beta} \subseteq R^{+}_{\alpha}$.*

Para A, B dados, esta proposición determina la existencia de una partición de $[0,1]$ en dos subintervalos con los valores para los que hay interrelación y los valores para los que no hay. Al mayor valor para el que se cumple la interrelación lo llamaremos nivel óptimo y determinará un grado de correspondencia entre los subconjuntos.

En el caso de la correspondencia mínima se obtiene el nivel óptimo de interrelación mínima

$$\begin{aligned} R^{o}_{\vee}(A,B) &= \sup\{\alpha \in [0,1] \mid A(R_\alpha)^{+}_{\vee} B\} \\ &= \sup\{\alpha \in [0,1] \mid \\ &\qquad \exists x \in X \quad (A \cap (R_\alpha \circ B)(x) \neq 0\} \end{aligned}$$

De igual modo resulta para la interrelación mutua y máxima

$$\begin{aligned} R^{o}_{\bowtie}(A,B) &= \sup\{\alpha \in [0,1] \mid \\ &\qquad A \subseteq R_\alpha \circ B, \quad B \subseteq A \circ R_\alpha\} \\ R^{o}_{\wedge}(A,B) &= \sup\{\alpha \in [0,1] \mid A \subseteq R \triangleright B\} \end{aligned}$$

Estos grados se obtienen en la hipótesis de que sean no vacíos los conjuntos

$$\begin{aligned} C_{\vee} &= \{\alpha \in [0,1] \mid A(R_\alpha)^{+}_{\vee} B\} \\ C_{\bowtie} &= \{\alpha \in [0,1] \mid A(R_\alpha)^{+}_{\bowtie} B\} \\ C_{\wedge} &= \{\alpha \in [0,1] \mid A(R_\alpha)^{+}_{\wedge} B\} \end{aligned}$$

como consecuencia se impone una restricción sobre los pares de subconjuntos a los que es aplicable cada una de las relaciones obtenidas.

Proposición 4 *Sea R una relación binaria borrosa y A, B un par de subconjuntos borrosos, entonces*

i) $C_{\vee} \neq \emptyset \Leftrightarrow A \neq \emptyset, \ B \neq \emptyset$

ii) $C_{\bowtie} \neq \emptyset \leftrightarrow hgt(A) = hgt(B)$

iii) $C_{\wedge} \neq \emptyset \leftrightarrow hgt(A) = hgt(B)$

Un marco común en el que se pueden obtener todas las relaciones es la clase de subconjuntos borrosos normales $\mathcal{N}(X)$. En este marco de estudio se han obtenido resultados (cf.[10],[7]) que establecen una comparación entre las distintas relaciones obtenidas por agregación $sup - min$ y por segmentación, a partir de una elemental R.

Proposición 5 *Dada una relación binaria borrosa R y dos subconjuntos normales A, B cualesquiera se cumple que*

$$R^{o}_{\wedge}(A,B) \leq R^{o}_{\bowtie}(A,B) \leq R^{*}_{\vee}(A,B) \leq R^{o}_{\vee}(A,B)$$

Correspondencias duales

La introducción de la complementación a través de una función de negación [11] como $n(x) = 1 - x$ permite obtener un conjunto de construcciones duales de las anteriores. En la segmentación podemos partir de dos familias, $\{(R_\alpha)^c\}$ o $\{(R^c)_\alpha\}$.

Proposición 6 *Dada una relación binaria borrosa, para todo $\alpha \in [0,1]$ se cumple*

i) $(R_\alpha)^c \subseteq (R^c)_{1-\alpha}$ *ii)* $(\bar{R}_\alpha)^c = (R^c)_{1-\alpha}$

donde $\bar{R}_\alpha = \{(x,y) \mid R(x,y) > \alpha\}$ son cortes de nivel estrictos sobre la relación R.

Este resultado da pie a la formación con la familia $\{\bar{R}_\alpha\} = \{Q_\alpha\}$ de nuevas construcciones, que podríamos denominar niveles óptimos complementarios de interrelación y que serían

$$R^{\bullet}_{\vee}(A,B) = \inf\{\alpha \mid A(Q_\alpha)^{+}_{\vee} B\} \qquad (17)$$

$$R^{\bullet}_{\bowtie}(A,B) = \inf\{\alpha \mid A(Q_\alpha)^{+}_{\bowtie} B\} \qquad (18)$$

$$R^{\bullet}_{\wedge}(A,B) = \inf\{\alpha \mid A(Q_\alpha)^{+}_{\wedge} B\} \qquad (19)$$

La siguiente proposición establece la conexión entre las relaciones complementarias y las originales.

Proposición 7 *Sea R una relación binaria borrosa, para cada una de las interrelaciones mínima, mutua y máxima se cumple $R^{\bullet} = 1 - (R^c)^o$*

No obstante, existen contraejemplos en el caso de correspondencia mutua que muestran que, en general, no se cumple $R^{\bullet} = R^o$.

La correspondencia complementaria en el caso de mutua interrelación ha sido apuntada por Brink et al. como método de extensión de relaciones [5].

5 Conclusión

Hemos presentado una sistemática que abarca una amplia variedad de casos, algunos de los cuales ya han sido objeto de estudios particulares. Siguiendo el enfoque global de este trabajo cabe continuar el estudio de las construcciones que resulten coherentes al extender la relación inicial. Para ello se hace necesario fijar axiomáticamente las condiciones que expresan la extensión de relaciones y la persistencia de propiedades, particularmente en el caso de similaridades. Asimismo quedan por fijar las relaciones entre la construcción por agregación y el principio de extensión de Zadeh.

Otro de los aspectos estudiados ha sido la aplicación a casos finitos de las relaciones extendidas. esta restricción da paso a algoritmos de cálculo del grado de relación y en el caso de similaridades a algoritmos de clasificación. También es de particular interés, el resultado de la aplicación de estos métodos de extensión a la igualdad ordinaria.

Quedan abierto el estudio de las extensiones que se derivan al considerar en las condiciones conjuntistas las inclusiones como implicaciones borrosas. Por otro lado, existen otras formas de agregación parcial y de segmentación mixta que se revelan de indudable interés para las aplicaciones.

Referencias

[1] Alsina C., Trillas E., Valverde L., On some logical connectives for fuzzy sets theory, *Journ. Math. An. and Appl.* **93** (1983) 15-26.

[2] Badard R., Extending lattice structures, *Journal of Math. An. and App.* **136** (1988) 314-324.

[3] De Baets B., Kerre E., Fuzzy relational compositions, *Fuzzy Sets and Systems* **60** (1993) 109-120.

[4] Bandler W., Kohout L., Fuzzy power sets and fuzzy implication operators, *Fuzzy Sets and Systems* **4** (1980) 13-30.

[5] Brink C., Britz C., Melton A., A note on fuzzy power relations, *Fuzzy Sets and Systems* **54** (1993) 115-117.

[6] González L., Marín A., Extending fuzzy relations, *Fuzzy Sets and Systems* **69** (1995) 157-169.

[7] González L., Marín A., Optimum level extension of relations, en Procs of FUBEST94, Sofia, Bulgaria, 89-91.

[8] Gottwald S., Fuzzy set theory with t-norms and phi-operators, en *The Mathematics of Fuzzy Systems* (Di Nola et al. eds., Verlag Tüv Rheinland, Köln 1986) 143-196.

[9] Klement E.P., Construction of fuzz σ-algebra using triangular norms, *Journ. Math. An. and Appl.* **85** (1982) 543-565.

[10] Marín A., Sobre redundancia y similaridad, Tesis doctoral, Univ. Pública de Navarra, 1994.

[11] Trillas E., Sobre funciones de negación en la teoría de conjuntos difusos, *Stochastica*, **3** (1979), 47-59.

GRAMATICAS DE ESTRUCTURAS DE FRASE Y EXPRESIVIDAD DE LOS CONCEPTOS VAGOS

A. Sobrino* y J. A. Olivas
*Area de Lógica y Filosofía de la Ciencia
Universidad de Santiago de Compostela
e-mail: lflgalex@usc.es

"El problema para la lingüística es proveer enunciados (reglas) que separen las cadenas correspondientes a oraciones o discursos del español de otras cadenas juzgadas agramaticales por hispanohablantes nativos. A fin de llegar a tales reglas, debemos estudiar las cadenas particulares que aparecen en los lenguajes naturales.

Examinemos, por ejemplo, las 'coacciones' que se ejercen entre sujetos y verbos en el español. La secuencia de dos palabras *Juan* (sujeto) y *bebe* (verbo)

Juan bebe

es una oración española, pero la secuencia

bebe Juan

no lo es.†

---

† Esto deberá entenderse desde luego en un sentido de *orden* gramatical muy estricto. (N. del T.)."

(M. Gross, *Modelos matemáticos en lingüística*)

INDICE

1. Introducción
2. La estructura canónica de la oración vaga
3. La gramática base G_0
4. Objetivo
5. Gramaticalidad y aceptabilidad
6. Alteraciones de G_0 que preservan gramaticalidad o derivan aceptabilidad
7. Conclusión
8. Referencias

Palabras clave: *Gramáticas formales, semioraciones, expresividad de los conceptos vagos*

1. INTRODUCCION

Ya desde Aristóteles se ha observado que la estructura lingüística sujeto-predicado era muy adecuada para desarrollar el análisis lógico del lenguaje ordinario. La silogística, p. ej., participa de esta convicción. La lógica borrosa, una de las herramientas formales encargadas del análisis del lenguaje de significado vago, fija también como unidad de análisis formal de la lengua a la constituída por sujeto y predicado, si bien en este caso, sujeto y predicado son entidades complejas que contienen a su vez otros elementos. El sujeto, en el caso de máxima transmisión de vaguedad, está conformado por un adverbio de cantidad y un nombre y el predicado, por un adverbio cualificador y un adjetivo, el primero de ellos con el papel de especificar la vaguedad del adjetivo al que acompaña. Un ejemplo-tipo de oración que se puede atener a esta descomposición es,

Pocos suecos son muy bajos

2. LA ESTRUCTURA CANONICA DE LA ORACION VAGA.

La estructura más completa y usual de una oración vaga aparece señalada en los primeros trabajos de Zadeh sobre semántica borrosa [3] y puede resumirse en la siguiente sucesión:

$$\text{Adverbio}_{\text{cantidad}} + \text{Nombre} + \text{Verbo} + \text{Adverbio}_{\text{modificador}} + \text{Adjetivo}$$

abreviadamente, CNVMA.

En efecto, una de las características definitorias de la lógica borrosa es su disposición para tratar el lenguaje natural tal y como de inmediato se presenta, esto es, sin regimentarlo previamente, sin convertirlo a otro lenguaje alternativo diferente del dado. Esto exige que atienda a frases como la señalada más arriba:

'Pocos suecos son muy bajos'

Desde el punto de vista del análisis de la vaguedad, en esta frase son importantes las siguientes unidades lingüísticas:

- Adverbios de cantidad, como 'pocos'.
- Nombres, como 'suecos'.
- Adverbios cualificadores, como 'muy'.
- Adjetivos, como 'bajos'.

Pero en la oración-tipo considerada se puede distinguir otras unidades:

- El verbo ('son'), que ejerce el mismo papel que en el análisis clásico: atribuir propiedades a las entidades que se sitúan en posición de sujeto.
- Artículos y preposiciones, como 'la', 'un' o 'de'. Tal ocurre, p. ej., con 'los' en 'la mayoría de los...'

No obstante, verbos, artículos y preposiciones no merecerán un examen especial en este trabajo. Desde el punto de vista de la vaguedad, no parecen ofrecer un

interés añadido y se obvian, en adelante, las consideraciones que puedan estar ligadas a los mismos.

3. LA GRAMÁTICA BASE G_0

En una acepción común en el análisis formal del lenguaje, analizar con una gramática de estructura frase una oración como la ya señalada es hacer un doble ejercicio:

- desde un punto de vista generador, encontrar una derivación que vaya desde el símbolo inicial o axioma (O) a la oración.
- desde un punto de vista reconocedor, realizar el camino inverso: partir de la oración y reconstruir su gramaticalidad hasta llegar al axioma.

Una gramática canónica o gramática base para frases como 'Pocos suecos son muy bajos' podría ser la siguiente,

Gramática canónica (G_0)

(G_0)

$O_0 \rightarrow SN_0 + SV_0$
$SN_0 \rightarrow C^1 + [] + N$
$SV_0 \rightarrow V + M + [] + A$
$C \rightarrow$ Pocos | La mayoría de
$[] \rightarrow$ los | λ
$N \rightarrow$ suecos
$V \rightarrow$ son
$M \rightarrow$ muy
$A \rightarrow$ bajos | altos

donde O=oración, SN=sintagma nominal, SV=sintagma verbal, C=cuantificador, N=nombre, V=verbo, M=modificador, A=adjetivo y [] está en lugar de posibles preposiciones o artículos , como ocurre en 'la mayor parte de *los*' en posición SN o en 'es muy *de* derechas' en posición SV. Eventualmente, [] también puede ser sustituído por una cadena de símbolos vacía, λ. | indica, como es usual, alternatividad y las letras de no terminales en negrita señalan un elemento añadido a la regla. Esta gramática generaría frases las siguientes,

Ejemplos: Pocos suecos son muy bajos
La mayoría de los suecos son muy altos

[1] Obsérvese que algunos cuantificadores son expresiones adverbiales más bien que adverbios, p. ej., 'la mayoría de' o 'casi todos'. Lo mismo ocurre con los modificadores, que a veces son adverbios sencillos y en otras ocasiones son expresiones adverbiales complejas, como sucede en 'una especie de', 'un tipo de', etc. En este estudio, la letra C y M representan tanto a adverbios sencillos como a expresiones adverbiales.

4. OBJETIVO

El objetivo de este trabajo es indagar en algunas alteraciones estructurales de G_0 cuando se

- invierte,
- suprime o
- añade

algún o algunos de los elementos lingüísticamente distinguidos en el análisis gramatical de una oración vaga, esto es, C, N, M o A.

El beneficio que se espera es extender la estructura de la que se ha considerado expresión canónica de una oración vaga a otros formatos admisibles desde una perspectiva lingüística que se explicitan el apartado que a continuación sigue.

5.GRAMATICALIDAD Y ACEPTABILIDAD

En el ámbito lingüístico de la gramática generativa ha sido usual distinguir [1],

- oraciones gramaticales: aquellas que se atienen a las reglas de la gramática de una lengua. Por ejemplo: pocos suecos son muy bajos.
- oraciones semigramaticales: aquellas que, aún no siendo gramaticalmente correctas, son semánticamente comprensibles por los hablantes de una lengua. Por ejemplo: una idea surgió a mi.

Por transmitir un significado tan fiel como el de su correlato gramaticalmente correcto (Me surgió una idea), estas oraciones se dice que son aceptables, aunque no gramaticales.

- oraciones agramaticales: aquellas que no son gramaticalmente correctas ni aceptables. Por ejemplo: verde la de sol muy.

Si bien las oraciones gramaticales son también aceptables, las semioraciones no son oraciones estrictamente gramaticales.

La oración-ejemplo 'Pocos suecos son muy bajos' puede ser considerada como un paradigma de frase gramaticalmente correcta. Modificaciones estructurales en ella por medio de la adición, supresión o inversión de alguno de sus términos originarán gramaticalidad, semigramaticalidad o antigramaticalidad. Las alteraciones que preserven gramaticalidad incrementarán de modo no problemático la expresividad del significado vago, las que conduzcan a semigramaticalidad aumentarán la expresividad vía aceptabilidad (no gramaticalidad), la agramaticalidad indicará, por fin, qué transgresiones convierten la gramaticalidad o la aceptabilidad en oraciones incomprensibles y, por lo tanto, inaceptables desde el punto de vista de estructuras gramaticales que pueden servir para generar o reconocer frases de significado vago.

Asi pues, la adición, supresión o inversión de alguno de los no terminales de O en G_0 originará gramáticas modificadas como las que en el apartado siguiente se exponen. Para cada caso se indican las alteraciones efectuadas en los elementos no terminales y, también para cada ocasión, de acuerdo con los ejemplos, se especifican los elementos terminales, que se repiten para algunas gramáticas y que deben ser considerados como algunos de los posibles.

6. ALTERACIONES DE G_0 QUE PRESERVAN GRAMATICALIDAD O DERIVAN ACEPTABILIDAD

Como ya se indicó, puede considerarse de modo razonable que el ordenamiento gramaticalmente canónico de una oración vaga es aquel que responde a la ordenación CNVMA. La inversión, supresión y adición de alguno de los elementos de esta serie da lugar a estructuras lingüísticas diferentes de la canónica que, en consecuencia, tienen que ser reconocidas por gramáticas también diferentes de G_0, esto es, por gramáticas modificadas de G_0.

A continuación se indican algunas de las variantes posibles.

1. Variantes de (G_0) que preservan gramaticalidad

En cada caso se recoge: Número de gramática, descripción de la transformación, reglas de la gramática y ejemplo. Las reglas de la gramática son numeradas con subíndices, de acuerdo con las transformaciones hechas sobre los elementos no terminales. Algunas variantes usan transformaciones previas y así son indicadas por la numeración. Las reglas que especifican elementos terminales tienen sólo el papel de permitir generar los ejemplos concretos.

(G_1)

Por inversión de SV y SN en O_0, resultando la nueva regla O_1. SN y SV permanecen con estructura que en (G_0).

$O_1 \rightarrow SV_0 + SN_0$
$SN_0 \rightarrow C + [] + N$
$SV_0 \rightarrow V + M + A$
C → pocos | la mayoría de
[] → los | λ
N → suecos
V → son
M → muy
A → bajos | altos

Ejemplos: Son muy bajos pocos suecos
Son muy altos la mayoría de los suecos

Obsérvese como 'la mayoría de' acepta mejor la inversión que 'pocos'.

(G_2)

Por inversión de C y N en SN_0, resultando SN_1. O y SV permanecen con igual estructura que en (G_0).

$O_0 \rightarrow SN_1 + SV_0$
$SN_1 \rightarrow N + P + C + P$
$SV_0 \rightarrow V + M + A$
C → pocos | la mayoría
P → ,
N → suecos
V → son
M → muy
A → bajos | altos

donde P=signo de puntuación.

Ejemplos: Los suecos, pocos, son muy bajos
Los suecos, la mayoría, son muy altos

(G_3)

Por adición de N o C en SN_0, resultando SN_2. O y SV permanecen con igual estructura que en G_0.

$O_0 \rightarrow SN_2 + SV_0$
$SN_2 \rightarrow C + N + \mathbf{N}_{Adv} \mid C + [] + \mathbf{C} + N + A$
$SV_0 \rightarrow V + M + A$
C → pocos | la mayoría de
[] → los | λ
N → finlandeses | lapones | kurdos
V → son
M → muy
A → morenos | perseguidos

Ejemplos: Pocos finlandeses lapones son muy morenos
La mayoría de los pocos kurdos son muy perseguidos

(G_4)

Por adición de M o A en SV_0, resultando SV_1. O y SN permanecen con igual estructura que en G_0.

$O_0 \rightarrow SN_0 + SV_1$
$SN_0 \rightarrow C + N$
$SV_1 \rightarrow V + \mathbf{M} + M + A \mid V + M + A + \mathbf{A}$
C → la mayoría de
[] → los
N → suecos | españoles
V → son
M → casi | seguramente
A → moreno | morenos | oscuros

Ejemplos:
La mayoría de los suecos son casi seguramente muy morenos
La mayoría de los españoles son morero oscuros

(G_5)

Por adición con repetición de C y N es SN_0, resultando SN_3. O y SV permanecen con igual estructura que en G_0.

$O_0 \rightarrow SN_3 + SV_0$
$SN_3 \rightarrow C + \mathbf{C} + N \mid C + N + \mathbf{N}$
$SV_0 \rightarrow V + M + A$

C → pocos
N → suecos
V → son
M → muy
A → bajos

Ejemplos: Pocos pocos suecos son muy bajos.
Pocos suecos suecos son muy bajos.

queriendo enfatizar, esta última oración, una característica racial (p. ej., el hecho de que esta propiedad no se observe en los suecos hijos de emigrantes).

(G_6)

Por adición con repetición de M y A en SV_0, resultando SV_2. O y SN permanecen igual que en G_0.

$O_0 \rightarrow SN_0 + SV_2$
$SN_0 \rightarrow C + N$
$SV_2 \rightarrow V + M + M + A \mid V + M + A + A$
C → pocos
N → suecos
V → son
M → muy
A → bajos

Ejemplos:
Pocos suecos son muy muy bajos.
Pocos suecos son bajos bajos.

Variantes de (G_0) que preservan gramaticalidad, pero de las que resultan,

a) oraciones ininteligibles (gramaticales, pero no comprensibles; por tanto, no aceptables)

(G_7)

Por supresión de N en SN_0, resultando SN_4. O y SV permanecen con igual estructura que en G_0.

$O_0 \rightarrow SN_4 + SV_0$
$SN_4 \rightarrow C$
$SV_0 \rightarrow V + M + A$
C → pocos | la mayoría
V → son
M → muy
A → bajos | altos

Ejemplos: Pocos son muy bajos
La mayoría son muy altos

b) oraciones de significado genérico.

(G_8)

Por supresión de M en SV_0, resultando SV_3. O y SN permanecen igual que en G_0.

$O_0 \rightarrow SN_0 + SV_3$
$SN_0 \rightarrow C + [] + N$
$SV_3 \rightarrow V + A$
C → pocos | la mayoría de
[] → los | λ
N → suecos
V → son
A → bajos | altos

Ejemplos: Pocos suecos son bajos
La mayoría de los suecos son altos

(G_9)

Por supresión de C en SN_0, resultando SN_5. O y SV permanecen con igual estructura que en G_0.

$O_0 \rightarrow SN_5 + SV_0$
$SN_5 \rightarrow [] + N$
$SV_0 \rightarrow V + M + A$
[] → los
N → suecos
V → son
M → muy
A → bajos | altos

Ejemplos: Los suecos son muy bajos
Los suecos son muy altos[2]

c) oraciones precisas

(G_{10})

Por supresión de C y M en SN_0 y SV_0, respectivamente, resultando SN_5 y SV_4. O permanece con igual estructura que en G_0. Obsérvese que se suprime más de un elemento de los que señalan la vaguedad en la oración; esto es, se elimina C y M conjuntamente.

$O_0 \rightarrow SN_5 + SV_3$
$SN_5 \rightarrow [] + N$

[2]Esto revela que la supresión del cuantificador es más importante que la del modificador para especificar la vaguedad. La supresión del cuantificador deja un nombre, que usualmente es tomado en la frase como de significado preciso. La supresión del modificador, deja un adjetivo, normalmente de significado vago. Esto puede explicar, quizás, porque la supresión del modificador es menos fundamental en el cometido indicado antes.

$SV_3 \rightarrow V + A$
[] → los
N → suecos
V → son
A → altos

Ejemplo: Los suecos son altos

II. Variantes de (G_0) que conducen a semigramaticalidad

(G_{11})

Por inversión de V con M y A en SV_0, resultando SV_4. O y SN permanecen con igual estructura que en G_0.

$O_0 \rightarrow SN_0 + SV_4$
$SN_0 \rightarrow C + N + []$
$SV_4 \rightarrow M + A + V$
C → pocos | la mayoría de
[] → los | λ
N → suecos
V → son
M → muy
A → bajos | altos

Ejemplos: Pocos suecos muy bajos son
La mayoría de los suecos muy altos son

(G_{12})

Por inversión de M y A en SV_0 (sólo para algunos modificadores) e inversión del resultado con V, resultado SV_5. O y SN permanecen con igual estructura que en G_0.

$O_0 \rightarrow SN_0 + SV_5$
$SN_0 \rightarrow C + N + P$
$SV_5 \rightarrow A + M + V$
C → pocos | la mayoría de
[] → los | λ
P → .
N → suecos
V → son
M → bastante | más bien
A → bajos | altos

Ejemplo:
La mayoría de los suecos, altos bastante son
(Frente a la frase gramatical, 'la mayoría de los suecos son bastante altos').
Pocos suecos, bajos más bien son
(Frente a la frase gramatical 'pocos suecos son más bien bajos').

Obsérvese como la colocación de la coma es desambiguadora. De lo contrario, estos ejemplos podrían ser interpretados como oraciones agramaticales, como resulta en el ejemplo: Pocos suecos bajos, más bien son.

III. Variantes de (G_0) que conducen a antigramaticalidad

(G_{13})

Por inversión de M y A (sólo para algunos modificadores) en SV_0, resultando SV_6. O y SN permanecen con igual estructura que en G_0.

$O_0 \rightarrow SN_0 + SV_6$
$SN_0 \rightarrow C + N + []$
$SV_6 \rightarrow V + A + M$
C → pocos | la mayoría de
N → suecos
[] → los | λ
V → son
M → muy
A → altos | bajos

Ejemplos: Pocos suecos son bajos muy
La mayoría de los suecos son altos muy

(G_{14})

Idéntica a G_{12}. Por inversión de M (sólo para algunos modificadores) y A en SV_0, y por inversión de V con la inversión de M y A ya realizada, resultando SV_7. O y SN permanecen con igual estructura que en G_0.

$O_0 \rightarrow SN_0 + SV_7$ $\quad SN_0 \rightarrow C + N$
$SV_7 \rightarrow A + M + V$
C → pocos
N → suecos
V → son
M → muy
A → bajos

Ejemplo: Pocos suecos bajos muy son

(G_{15})

Por supresión de A en SV_0, resultando SV_8. O y SN permanecen igual que en G_0.

$O_0 \rightarrow SN_0 + SV_8$ $\quad SN_0 \rightarrow C + N$
$SV_8 \rightarrow V + M$
C → pocos
N → suecos
V → son
M → muy

Ejemplo: Pocos suecos son muy

Este apartado y sus ejemplos han querido mostrar cómo incrementar la expresividad de oraciones de significado vago, añadiendo, suprimiendo o invirtiendo

los elementos no terminales C, N, M y A en la gramática canónica (G_0); alteraciones de las que resulta:

- preservación de la gramaticalidad respecto a la estructura canónica o
- derivación de semigramaticalidad; esto es, de oraciones no totalmente gramaticales pero semánticamente comprensibles por los hablantes de una lengua.

Como resultado colateral también se ha mostrado qué transgresiones serían inútiles en este propósito (las que conducen a oraciones agramaticales).

7. CONCLUSIONES

En este trabajo se ha examinado cómo aumentar la expresividad de una oración vaga introduciendo la categoría de semigramaticalidad y se ha mostrado qué gramáticas generan este tipo de oraciones. El resultado es el logro de varias gramáticas para expresar conceptos vagos de formas aceptables.

Esto debería ser útil, p. ej., en el campo de los interfaces persona-máquina, ya que discriminaría las oraciones aceptables de aquellas que no lo son, admitiendo frases que, de no contemplarse la categoría 'semigramatical', no serían usualmente reconocidas como aceptables, por más que sean empleadas de modo habitual por los hablantes de una lengua.

Otros beneficio de este trabajo es el ofrecer una metodología que permita agrupar a los cuantificadores y modificadores en clases según el papel que desempeñan en las transformaciones que preservan o no gramaticalidad y semigramaticalidad (aceptabilidad). De ello puede resultar una especificación de las reglas de producción de la gramática diferentes, de modo que en la expresión canónica queden incluídas algunas de las gramáticas que en este estudio se consideran variantes.

REFERENCIAS

[1] Katz, J. J. (1964), Semi-sentences, en Fodor, J. A. & Katz, J. J. (eds.), *The Structure of Language. Readings in the Philosophy of Language*. Prentice-Hall, Inc. Englewood Cliffs, pp. 400-16.

[2] Sobrino, A & Olivas, J. A. (1994), Semioraciones y semicadenas (Estudio de algunos aspectos), en Esteva, F. & García, P (eds.) (1994), *Actas del IV Congreso de la Asociación Española de Tecnologías y Lógica Fuzzy*, Blanes, pp. 139-144.

[3] Zadeh, L. A. (1971), Quantitative fuzzy semantics, *Information Sciences*, 3, pp. 159-176.

AGRADECIMIENTOS

Este trabajo ha sido realizado en el marco del proyecto PB92-0394 de la D.G.I.C.Y.T.

El lenguaje de programación borrosa XL

Rafael Morales-Bueno
Dpto. Lenguajes y Ciencias de la Computación
Facultad de Informática
Universidad de Málaga
e-mail: morales@tecma1.ctima.uma.es

José-Luis Pérez-de-la-Cruz
Dpto. Lenguajes y Ciencias de la Computación
Facultad de Informática
Universidad de Málaga
e-mail: cruz@tecma1.ctima.uma.es

Ricardo Conejo
Dpto. Lenguajes y Ciencias de la Computación
Facultad de Informática
Universidad de Málaga
e-mail: conejo@tecma1.ctima.uma.es

Buenaventura Clares
Dpto. Lenguajes y Sistemas Informáticos
Facultad de Ciencias
Universidad de Granada

Resumen

En esta comunicación definimos un nuevo lenguaje de programación borrosa llamado XL. XL se obtiene del lenguaje de programación borrosa L extendiendo su sintaxis y su semántica con una sentencia de bucle indefinido (while). Desarrollamos algunos ejemplos y en particular describimos en este lenguaje una regla de control borroso.

Palabras Clave: *lenguajes de programación, control borroso, conjuntos borrosos.*

1 Introducción

En [7] definimos el lenguaje de programación L. Este lenguaje tiene un reducido conjunto de instrucciones y una sola instrucción de bucle definido de la forma `repeat Xi times ... semit`. En 2.1 hacemos una breve descripción de L.

El objetivo central de esta comunicación es definir un lenguaje para mejorar la potencia expresiva de L permitiendo una instrucción de bucle indefinido (while). El lenguaje resultante se denominará XL. Como se menciona en [7], la principal característica de L (o XL) en relación con otros lenguajes de programación borrosa (por ejemplo, L.P.L [1], [2]; FRIL[3]) es su simplicidad. Tiene un conjunto reducido de instrucciones y un solo tipo de datos. En principio L y XL están diseñados como lenguajes de propósito general. En la sección 3 definimos el lenguaje XL y mostramos algunos ejemplos; en concreto, uno de ellos es la descripción en nuestro lenguaje de una regla de control borroso.

Por último, en la sección 4 reseñamos algunas conclusiones generales y trabajos futuros relacionados con la calculabilidad borrosa y los lenguajes de programación borrosa.

2 Notación y resultados previos

En el desarrollo que sigue, consideraremos que los grados se evalúan en un semianillo ordenado finito W $(\{l_0, ..., l_r\}, \oplus, \otimes, \leq)$, siendo las operaciones calculables y el orden decidible [12]. Supondremos que los elementos neutros son l_0 para $\oplus$ y l_r para $\otimes$.

Usaremos W-conjuntos sobre $(Z^+)^n$, $n \geq 1$, con la representación estandar: $A = \mu_A(a) / a + ... + \mu_A(z) / z$. Si $a = (a_1, ..., a_n) \in (Z^+)^n$ entonces π_i representa la *i-proyección* $\pi_i(a) = a_i$

Sean U y V conjuntos. Una *W-función* f de U a V es una función de UxV en W. Si $f(u,v)=\mu$, entonces decimos que el valor de la W-función f en u es v con grado μ.

2.1 Una reseña del lenguaje L

El lenguaje L opera sobre datos que son subconjuntos borrosos de $(Z^+)^n$. Hay dos instrucciones de asignación nítida (X:=0, X:=suc(Y)) y una instrucción de asignación borrosa. La única estructura de control es un bucle tipo "FOR"; la instrucción de control condicional "IF" se puede programar en términos de asignaciones y bucles "FOR" .

Debido a entradas borrosas o asignaciones borrosas, el programa puede alcanzar un punto a partir del cual cada instrucción debe ejecutarse de manera borrosa. Esto es especialmente importante al considerar estructuras repetitivas o condicionales. Cuando acaba la ejecución se obtiene un conjunto de salidas cada una con su grado. Es decir, el programa produce como salida un conjunto borroso.

La ejecución de un programa nítido se puede ver como la generación de una sucesión de "configuraciones nítidas". Cada configuración representa el valor de cada variable del programa. En el caso de programas en L, la sucesión no es única; a partir de cierto punto, hay varias ramas de cálculo cada una con su grado. De esta forma, las configuraciones generadas por la ejecución de un programa en L son subconjuntos borrosos de "configuraciones nítidas".

En lo que sigue usaremos el término "elemento de la configuración" para referirnos a estas "configuraciones nítidas".

El significado de un programa en L está definido en [7] mediante una semántica operacional, esto es, viendo el programa como un dispositivo que calcula una función. Cada programa en L tiene dos conjuntos de variables distinguidas - variables de entrada y variables de salida- y calcula una W-función de las variables de entrada en las variables de salida.

Ejemplo 1. El siguiente programa calcula, para un número natural n, el W-conjunto que se puede considerar "previo" a n. El concepto de previo es dependiente del contexto, y esto se refleja en la elección de la W-función para FUZZ1 en la asignación borrosa.

```
Programa previous  P = (1,1,Q)
        PREV;
        begin
                X2:=X1;
                repeat X2 times
                        X1:= FUZZ1(X3);
                        X3:=suc(X3)
                semit
        end.
```

Vemos como un programa se define como una terna, número de variables de entrada (en este caso, una: X1), número de variables de salida (en este caso, una: X1) y el código. Este código tiene un identificador y la parte ejecutable está parentizada entre begin y end.

3 El lenguaje XL

3.1 Presentación del lenguaje XL

El lenguaje XL se obtiene añadiendo una estructura de bucle indefinido al lenguaje L de la sección anterior. La condición de control es del tipo $x\neq 0$. La variable x, como es habitual, se denomina variable de control del bucle.

Sea Xr la variable de control del bucle. Para cada elemento en la configuración con $Xr\neq 0$, el cuerpo del bucle se ejecuta una vez. Los resultados se agrupan con los elementos de la configuración de partida con $Xr=0$, formando una configuración intermedia. Este proceso se repite hasta que todos los elementos en la configuración tengan valor de $Xr=0$. Si una tal configuración no se alcanza, el proceso continua indefinidamente.

Más formalmente, sea P un programa en XL que usa k variables. La configuración en un instante de la ejecución del mismo viene dada por un conjunto borroso C en $(Z^+)^k$, donde cada elemento de la configuración es un vector de la forma $b=(b_1, b_2, ..., b_k)$ que representa el contenido actual de cada una de las k variables y su grado asociado, $\mu_C(b)$ es el grado con que se ha alcanzado ese elemento de la configuración.

Supongamos que C es la configuración alcanzada antes de ejecutar un bucle indefinido con variable de control Xr. Para cada elemento b, $\mu_C(b)=\alpha$, cumpliendo $\pi_r(b)\neq 0$ (esto es $Xr\neq 0$), se ejecuta el bucle indefinido que partiendo de α/b produce la configuración $\mathcal{B}_1{}^b$. Como $C=\Sigma\ (\alpha/b)$, la configuración obtenida tras esta primera ejecución es

$$B_1 = \bigcup_b B_1^b = B_{10} \cup B_{11}$$

donde

$$B_{10} = \Sigma(\beta/b) \text{ con } \pi_r(b)=0$$

esto es, la configuración formada por los elementos b de la configuración de partida C en los que $\pi_r(b)=0$, junto con los nuevos elementos obtenidos cuya componente r también es 0; para todos ellos el bucle no se vuelve a ejecutar y

$$B_{11} = \Sigma(\beta/b) \text{ con } \pi_r(b)\neq 0$$

que es la parte de la configuración obtenida, cuyos elementos tienen en la componente r un valor no nulo. Para los elementos de B_{11} el bucle se vuelve a ejecutar, obteniendose como resultado la configuración B_2.

De manera similar a la primera ejecución, $B_2 = B_{20} \cup B_{21}$ donde

$B_{20} = \Sigma\ (\beta/b)$ con $\pi_r(b)=0$

donde está incluido B_{10} y

$B_{21} = \Sigma\ (\beta/b)$ with $\pi_r(b)\neq 0$,

Repitiendo este proceso se pueden dar dos casos:

a) En un número finito de pasos alcanzamos una configuración B tal que $B=B_0$, esto es, para todo b con $\mu_B(b)\neq 0$ se verifica que $\pi_r(b)=0$. Entonces, el bucle acaba y desde la configuración C alcanzamos la configuración B.

b) Una tal configuración B nunca se alcanza. En este caso el programa calcula indefinidamente.

Nótese que esta instrucción "while" borrosa, cuando se usa con datos nítidos, se comporta como un bucle indefinido nítido.

Por otra parte, con una condición de control de bucle de la forma $Xr\neq 0$ se puede representar cualquier otra, haciendo uso de la suma, el predecesor (pred) y la resta definida en Z^+ ($x-y=0$ si $x<y$).

Una condición C de la forma $X<Y$ se puede representar mediante la expresión E dada por $(Y-X)-pred(Y-X)$, que toma el valor 1 si la condición se cumple y 0 en caso contrario.

Sean dos condiciones C_1 y C_2 representadas por expresiones E_1 y E_2. Entonces, la expresión para $C_1 \wedge C_2$ es $pred(E_1 + E_2)$ y la expresión para $C_1 \vee C_2$ es $(E_1 + E_2) - pred(E_1 + E_2)$, y para $\neg C_1$ es $1-E_1$.

Ahora a partir de la expresión para $X<Y$ tenemos expresiones para $X\neq Y$ ($X<Y \vee Y<X$), para $X=Y$ ($\neg(X\neq Y)$), etc.; y combinando estas podemos tener cualquier tipo de condición relativa a mas de dos variables. Mas precisamente, sea un bucle de la forma

while C(X1, X2,..., Xk) do S od

y sea E es la expresión asociada a la condición C. Entonces el bucle se expresa en nuestro lenguaje mediante:

```
Xk+1 := E;
while Xk+1 ≠0 do
      S;
      Xk+1 := E
od
```

3.2 W-función calculada por un programa

En [7] se demostró que cada programa en L define una W-función total, en el sentido de que para cada configuración inicial dada se alcanza una y sólo una configuración final. Con programas en XL la situación es diferente; la presencia de un bucle indefinido da origen a programas en los que no se alcanza una configuración final. Estos programas no terminan para ciertos datos de entrada. Más formalmente, tendremos las siguientes definiciones:

Definición 1. Sea P = (j,m,Q) un k-programa, esto es, que usa k variables. Asociada a P definimos la W-función $\Omega p:(Z^+)^k \times (Z^+)^k \rightarrow W$ como sigue: $\Omega p(a,b)=\beta$ sii a partir de una configuración inicial nítida a, el valor de la configuración final es un W-conjunto que contiene al elemento b con grado β.

Esta función da el grado asociado a la salida b cuando la entrada es el valor nítido a.

Definición 2. La W-función semántica para P es $\phi p : (Z^+)^j \times (Z^+)^m \rightarrow W$ definida como sigue:
Sea la configuración inicial $1/(a_1 ,...,a_j ,0,..,0) = 1/a$, entonces

$$\phi p(a_1 ,...,a_j ,b_1 ,...,b_m)= \Sigma\, \Omega p\ (a,a')$$

donde Σ denota la operación $\oplus$ extendida a todo a', tal que sus m primeras componentes coinciden con $b_1, ..., b_m$.

Informalmente, si un programa tiene menos variables de entrada de las que usa, entonces:

A) Las variables de entrada son las primeras; y

B) Las restantes variables se inicializan implícitamente a cero.

Por otra parte, si el programa tiene menos variables de salida de las que usa, entonces:

C) La configuración final se obtiene proyectando sobre las primeras variables, acumulando los grados aditivamente por medio de la operación $\oplus$ del semianillo.

Los programas en XL se hacen largos e ilegibles. Para evitar este problema, definimos en [7] el concepto de *macro*. Una macro-instrucción tiene la forma

Xi := <ident>(Xi$_1$, ..., Xi$_s$)

donde <ident> es el identificador asociado con un programa: R=(s,1,T).

Las macro-instruciones se expanden de manera natural de acuerdo con el código del programa que la define.

Mediante expresiones adecuadas hemos definido más arriba las condiciones de comparación entre variables (X<Y, X=Y, ...), las conectivas lógicas ($\wedge$, $\vee$, $\neg$). Análogamente [7], puede expresarse la instrucción condicional IF C THEN P FI, IF C THEN P ELSE Q FI. En los siguientes ejemplos haremos uso de estas macros.

3.3 Ejemplos de programas en XL

Ejemplo 2: Sea el siguiente programa en XL: P=(2,1,Q), siendo Q como sigue:

```
EJEMPLO;
begin
        while X2≠0 do
                X1:= suc(X1);
                X2:= FUZZ1(X2)
        od
end
```

donde la W-función asociada a FUZZ1 es $f_1:Z^+xZ^+\rightarrow W$ definida por:

$f_1(x,y)=$ 1 si y=x - 1;
0.5 si y=x - 2
0 en otro caso.

Supongamos que $\oplus$=max y $\otimes$=min

Sea la configuración inicial 1/(a,b). Suponemos que b≠0.

Tras la primera ejecución del bucle, la configuración resultante es:

1/(a+1,b-1) + 0.5/(a+1,b-2)

Después de la siguiente ejecución la configuración obtenida es:

1/(a+2,b-2) + 0.5/(a+2,b-3)+ 0.5/(a+2,b-4).

Tras $\underline{b/2}$ ejecuciones (donde $\underline{c}$ significa el menor entero mayor o igual que c), la configuración resultante es

$$1/(a+\underline{b/2},\ b-\underline{b/2}) + \sum_{i=\underline{b/2}+1}^{2\underline{b/2}} 0.5/(a+\underline{b/2},\ b-i)$$

En el segundo sumando, el último valor de i alcanza el valor b; la ejecución ha acabado para los elementos con segunda componente cero, pero debe continuar para los elementos que tienen segunda componente b-i≠0.

En un número finito de pasos se obtiene la siguiente configuración final:

$$1/(a+b,0) + \sum_{i=\underline{b/2}}^{b-1} 0.5/(a+i,\ 0)$$

Entonces la W-función asociada a P, Ωp: $(Z^+)^2x\ (Z^+)^2 \rightarrow W$, está definida por:

$\Omega p((a, b), (c, d))$ = 1 si c=a+b y d=0
0.5 si c=a+i, i=$\underline{b/2}$, ..., b-1 y d=0
0 en otro caso

y la W-función semántica $\phi p:(Z^+)^2 \times Z^+ \rightarrow W$ es:

$\phi p((a, b), c)=$ 1 si c=a+b
0.5 si c+a+i, i=b/2, ..., b-1
0 en otro caso.

Ejemplo 3: Sea el XL-programa EJEMPLO definido en el ejemplo 2:

```
EJEMPLO;
begin
        while X2≠0 do
                X1:= suc(X1);
                X2:= FUZZ2(X2)
        od
end
```

pero considerando la W-función asociada con FUZZ2, f_2 definida por:

$f_2(x,y)=$ 1 si y=x - 1;
0.5 si y=x +1
0 en otro caso.

Supongamos que $\oplus$ = max y $\otimes$ = min.

Sea la configuración inicial 1/(a,b). Suponemos que b≠0.

Tras la primera ejecución del bucle, la configuración resultante es:

1/(a+1,b-1) + 0.5/(a+1,b+1)

Después de la siguiente ejecución, la configuración obtenida es:

1/(a+2,b-2) + 0.5/(a+2,b)+ 0.5/(a+2,b+2).

Nótese que las siguientes iteraciones dan origen a configuraciones con X2=b+1, X2=b+2, etc. Como el valor inicial de X2 es b≠0, siempre habrá elementos con X2≠0 y el programa no termina.

Supongamos ahora que W=[0,1] y $\otimes = \times$. Como la función asociada a FUZZ2 tiene el grado 0.5 para ciertos valores, la acumulación de estos grados da origen a grados cada vez mas pequeños. En esta situación, y en la idea propuesta por Ostasiewicz [9] podríamos definir ejecución con "umbral" , ignorando un elemento de configuración cuando su grado esté por debajo de cierto valor. Por ejemplo, consideremos el umbral μ=0.26. Entonces tras la primera ejecución tenemos la misma configuración de antes: 1/(a+1,b-1) + 0.5/(a+1,b+1).

Después de la siguiente ejecución, la configuración obtenida es:

1/(a+2,b-2) + 0.5/(a+2,b)

y los elementos 0.25/(a+2, b) + 0.25/(a+2,b+2) se omiten.

Tras la tercera ejecución, la configuración resultante es:

1/(a+3, b-3) + 0.5/(a+3, b-1)

y los elementos 0.25/(a+3, b-1) + 0.25/(a+3, b+1) se omiten.

Es fácil comprobar que la ejecución termina tras un número finito de iteraciones del bucle.

Ejemplo 4: Controlador borroso.

Veamos como programar en el lenguaje XL la activación de una regla para controlar cierto proceso abstracto.

En este ejemplo resaltaremos fudamentalmente los aspectos prácticos.

Sea una regla de la forma:

Si X=S, entonces Y=M

donde S y M son los conjuntos borrosos representados en las figuras 1 y 2 respectivamente.

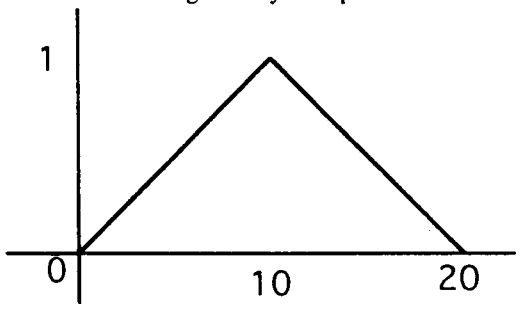


Figura 1: Conjunto borroso S

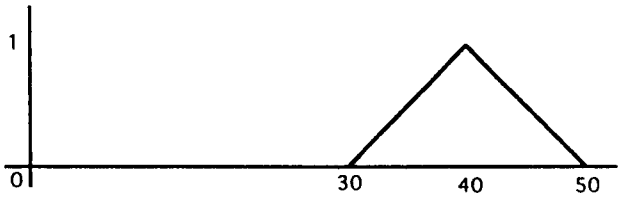


Figura 2: Conjunto borroso M

Estos conjuntos se pueden ver como W-funciones con grafo recursivo [5], al tomar para μ_A: $Z^+ \rightarrow [0, 1]$, la W-función f: $Z^+ \times \{1\} \rightarrow [0, 1]$ de manera que $f(a,1) = \mu_A(a)$. Así pues, podemos dar como entrada un elemento de Z^+ y, mediante un programa nítido, obtener como salida una codificación del grado (codificando los grados 0, 0.1, ..., 0.9, 1.0, respectivamente mediante los naturales 0, 1, ..., 9, 10).

En nuestro caso concreto, los programas correspondientes para S y M son como sigue:

S_set= (1, 1, T), siendo T como sigue:

```
S;
begin
    IF X1>0 AND X1≤10 THEN X2:= X1 FI;
    IF X1>10 AND X1≤20 THEN X2:= 20-X1 FI;
    X1:= X2
end
```

M_set= (1, 1, N), siendo N como sigue:

```
M;
begin
    IF X1>30 AND X1≤40 THEN X2:= X1-30 FI;
    IF X1>40 AND X1≤50 THEN X2:= 50-X1 FI;
    X1:= X2
end
```

En el programa que realiza el control aparecen algunas instrucciones borrosas que tienen asociadas W-funciones. Veamos ahora cúales son:

En la instrucción X4:= FUZZ0(X1), la W-función asociada es ψ_{01} definida por:

$\psi_{01}(x, y)=$ 1 si y=0 ó y=1
0 en otro caso.

En la macroinstrucción X3:=FLATTEN(X1, X2) la macro FLATTEN está asociada al programa R=(2,1, V) cuyo código V es:

```
FLATTEN;
begin
     IF X2=1 THEN X1:=FUZZ1(X1) FI;
     IF X2=2 THEN X1:=FUZZ2(X1) FI;
     ............
     IF X2=r THEN X1:=FUZZr(X1) FI
end
```

Las W-funciones asociadas a FUZZi (i=1,..,r) son I_i definidas como sigue:

$I_i(x, y)=$ l_i si y=x
0 en otro caso

(l_i son los elementos de W, definido en la sección 2)

El programa que simula la regla de control difuso es P=(1, 1, Q), siendo Q :

```
   FUZZCON;
   begin
1      X2:= S(X1);
2      IF X2≠0 THEN X1:=FLATTEN(X2, X1) FI;
3      X1:=0; X2:=0;
4      WHILE M(X1) = 0 DO
5             X1:= suc(X1)
6      OD;
7      WHILE M(X1) ≠ 0 DO
8             IF X4 ≠ 1 THEN
9                    X2:= M(X1);
10                   X4:= FUZZ0(X1);
11                   IF X4 = 1 THEN
12                         X3:=FLATTEN(X1, X2)
13                   FI
14            FI;
15            X1:= suc(X1);
16     OD;
17     IF X4≠1 THEN
18            X1:= pred(X1);
19            X2:= M(X1);
20            X3:=FLATTEN(X1, X2)
21     FI;
22     X1:= X3
   end
```

Veamos cómo es su funcionamiento para un valor concreto. Sea el valor de entrada X1 igual a 12, entonces la configuración inicial es 1/(12, 0, 0, 0)

Tras la ejecución de la línea 1, a X2 le corresponde el valor 20-12=8, obteniendo la configuración: 1/(12, 8, 0, 0).

Como X2≠0, se ejecuta la macro FLATTEN y resulta: 0.8/(12, 8, 0, 0) y tras la línea 3 tenemos: 0.8/(0, 0, 0, 0)

Continuamos ahora para obtener el conjunto borroso de salida.

Inicialmente, con el bucle 4-6 avanzamos hasta el primer valor con grado no nulo, llegando a la configuración: 0.8/(31, 0, 0, 0).

Comienza ahora el bucle 7-16 para obtener los elementos con grado no nulo:

Como X4 es 0≠1 se ejecuta la línea 9, resultando: 0.8/(31, 1, 0, 0) y tras la línea 10 tenemos la configuración 0.8/(31, 1, 0, 0) + 0.8/(31, 1, 0, 1).

La instrucción condicional 11-13 sólo afecta al segundo sumando y obtenemos:

0.8/(31, 1, 0, 0) + 0.1/(31, 1, 31, 1)

Nótese que en sucesivas ejecuciones, este segundo sumando no volverá a ser modificado ni en su grado ni en la variable X3, pues la condición de la línea 8 (X4 ≠ 1) no se va a cumplir nunca más.

Tras la ejecución de la línea 15 resulta: 0.8/(32, 1, 0, 0) + 0.1/(32, 1, 31, 1)

Repitiendo el proceso anterior llegamos de nuevo a la línea 15 con la configuración:

0.8/(32, 2, 0, 0) + 0.2/(32, 2, 32, 1) + 0.1/(32, 1, 31, 1)

Las sucesivas repeticiones de este proceso llevan a obtener una configuración en la que cada configuración elemental tiene, en la variable X3, valores entre 31 y 49 y con grados entre 0.1 y 0.8.

Tras la ejecución de la sentencia condicional 17-21, la asignación de la línea 22 y la proyección de salida sobre la variable X1, obtenemos como salida el siguiente conjunto borroso de Z^+ :

0.1/31 + 0.2/32 + ... +0.7/37 + 0.8/38 + 0.8/39 + 0.8/40 + 0.8/41+ 0.8/42 + 0.7/43 + ... + 0.2/48 + 0.1/49

cuya representación gráfica es la correspondiente a la figura 3:

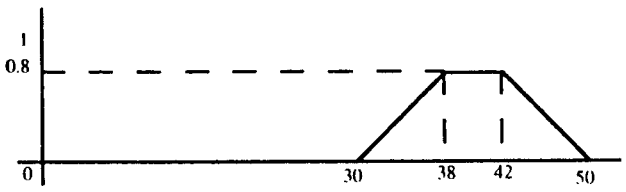


Figura 3: Conjunto borroso de salida para la entrada 12

que se correponde con la interpretación habitual de la activación de una regla como la propuesta (modus ponens generalizado para el caso de regla fuzzy con premisa nítida).

4 Conclusiones y trabajos futuros

Los resultados brevemente expuestos en esta comunicación son parte de nuestro trabajo sobre los fundamentos de los lenguajes imperativos borrosos [6], [7], [8]. La investigación se centra en dos aspectos: definir e implementar lenguajes de programación borrosa

de auténtico propósito general para especificar algoritmos borrosos [13], y estudiar la calculabilidad borrosa desde un punto de vista teórico.

En relación con el primer aspecto, un primer paso se dio al definir e implementar el lenguaje L [7]. Ahora hemos presentado el lenguaje XL como una extensión de L, incluyendo las potencialidades que proporciona una instrucción de bucle indefinido. Desde un punto de vista práctico, los programas en L y XL se ejecutan demasiado lentamente. Hay dos razones para esta lentitud: los recursos usados son secuenciales y las primitivas implementadas con muy elementales. Quizás los programas serían mas rápidos implementando directamente las macro-instrucciones sobre un hardware paralelo.

Desde el punto de vista teórico, teniendo en cuenta que cualquier función W-calculable (en el sentido de [11]) es válida para la asignación borrosa, la semántica de L (y XL) queda algo indeterminada. Por tanto, tiene sentido estudiar el caso en que sólo se permite, en la asignación borrosa, un subconjunto restringido de funciones borrosas. De esta manera nos acercamos más a un lenguaje práctico. Para ello es necesario definir la familia de lenguajes XL(H) obtenida restringiendo la asignación borrosa a las W-funciones pertenecientes al conjunto H [10].

Referencias

[1] Adamo, J. M., L.P.L. A fuzzy programming language: 1. Syntactic aspects. *Fuzzy Sets and Systems* 3 (1980), 151-179.

[2] Adamo, J. M., L.P.L. A fuzzy programming language: 2. Semantic aspects. *Fuzzy Sets and Systems* 3 (1980), 262-289.

[3] Baldwin, J. F, Zhou, S. Q., A fuzzy relational inference language. *Fuzzy Sets and Systems* 14 (1984), 155-174.

[4] Clares, B., Delgado, M., Introduction of the concept of recursiveness of fuzzy functions, *Fuzzy Sets and Systems* 21 (1987) 301-310.

[5] Gerla, G., Turing L-Machines and Recursive Computability for L-Maps, *Studia Logica* XLVIII, 2 (1989) 179-192.

[6] Morales-Bueno, R., Clares, B., Calculabilidad mediante gramáticas difusas. *Primer Congreso Español sobre Tecnologías y Lógica fuzzy*, Granada 1991, pp. 167-172.

[7] Morales-Bueno, R. et al., An elementary fuzzy programming language, *Fuzzy Sets and Systems* 58 (1993), 55-73.

[8] Morales-Bueno, R. et al., Estudio de una clase de W-funciones calculables, *III Congreso Español sobre Tecnologías y Lógica fuzzy*, Santiago de Compostela 1993, pp. 153-160.

[9] Ostasiewicz, W., A new approach to fuzzy programming, *Fuzzy Sets and Systems* 7 (1982), 139-152.

[10] Pérez de la Cruz, J. L. et al., Caracterización mediante programas de ciertas clases de funciones W-calculables, *V Congreso Español sobre Tecnologías y Lógica fuzzy*, Murcia 1995.

[11] Santos, E.S., Fuzzy algorithms. *Inform. Contr.*, 17 (1970), 326-339.

[12] Santos, E.S., Fuzzy and probabilistic programs. En Gupta, M. M., Saridis, Gaines, G. N. (eds.), *Fuzzy automata and decision processes*, North-Holland, 1977, pp.133-148.

[13] L. A. Zadeh, Fuzzy algorithms, Inf. & Control 12 (1968), pp. 94-102.

Caracterización mediante programas de ciertas clases de funciones difusas

José-Luis Pérez-de-la-Cruz
Dpto. Lenguajes y Ciencias de la Computación
Facultad de Informática
Universidad de Málaga
e-mail: cruz@tecma1.ctima.uma.es

Rafael Morales-Bueno
Dpto. Lenguajes y Ciencias de la Computación
Facultad de Informática
Universidad de Málaga
e-mail: morales@tecma1.ctima.uma.es

Ricardo Conejo
Dpto. Lenguajes y Ciencias de la Computación
Facultad de Informática
Universidad de Málaga
e-mail: conejo@tecma1.ctima.uma.es

Buenaventura Clares
Dpto. Lenguajes y Sistemas Informáticos
Facultad de Ciencias
Universidad de Granada

Resumen

En este trabajo estudiamos, desde el punto de vista de la Teoría de la Calculabilidad, algunas posibilidades que surgen cuando al escribir programas difusos en el lenguaje XL se emplean en las asignaciones únicamente funciones difusas pertenecientes a ciertos subconjuntos finitos. De está forma conseguimos caracterizar las clases WRG de Gerla y CONS de Morales et al.

Palabras clave: Lenguaje, programación, calculabilidad, difuso.

1 Introducción

En recientes trabajos hemos definido el lenguaje de programación XL. Puesto que se permite emplear en la sentencia de asignación cualquier W-función calculable (en el sentido de [8]), la semántica de XL no está completamente definida. En este trabajo estudiamos algunas posibilidades que surgen cuando solamente se permite en las asignaciones el empleo de ciertos subconjuntos de W-funciones. Más concretamente, mostramos en la sección 4 una familia finita de funciones cuyo empleo en las sentencias de asignación de XL es suficiente para generar precisamente la clase de W-funciones construibles [5], y análogamente en la sección 5 para la clase de W-funciones con grafo recursivo [2].

Para que este trabajo resulte autocontenido, introducimos en secciones previas algunos conceptos ya aparecidos en la literatura: en la sección 2 realizamos una presentación parcialmente informal del lenguaje XL, y en la sección 3 definimos las W-máquinas de Turing multicinta y recordamos las definiciones de las clases de W-funciones construibles y con grafo recursivo.

2 El lenguaje XL

El lenguaje XL opera sobre datos que son subconjuntos borrosos de $(Z^+)^n$. Hay dos instrucciones de asignación nítida (X:=0, X:=suc(Y)) y una instrucción de asignación borrosa. Hay dos estructuras de control: a)un bucle tipo "FOR" que representamos por "REPEAT x TIMES ... SEMIT"; b)un bucle indefinido con una condición de control del tipo $x\neq 0$ que representamos WHILE $x\neq 0$ DO ... OD. La variable x, como es habitual, se denomina variable de control del bucle. La instrucción de control condicional "IF" se puede programar en términos de asignaciones y bucles "FOR" .

Debido a entradas borrosas o asignaciones borrosas, el programa puede alcanzar un punto a partir del cual cada instrucción debe ejecutarse de manera borrosa. Esto es especialmente importante al considerar estructuras repetitivas o condicionales. Cuando acaba la ejecución se obtiene un conjunto de salidas cada una con su grado. Es decir, el programa produce como salida un conjunto borroso.

La ejecución de un programa nítido se puede ver como la generación de una sucesión de "configuraciones nítidas". Cada configuración representa el valor de cada variable del programa. En el caso de programas en XL, la sucesión no es única; a partir de cierto punto, hay varias ramas de cálculo cada una con su grado. De esta forma, las configuraciones generadas por la ejecución de un programa en XL son subconjuntos borrosos de "configuraciones nítidas". En lo que sigue usaremos el término "elemento de la configuración" para referirnos a estas "configuraciones nítidas".

El significado de un programa en XL está definido en [6] mediante una semántica operacional, esto es, viendo el programa como un dispositivo que calcula una función. Cada programa en XL tiene dos conjuntos de variables distinguidas - variables de entrada y variables de salida- y calcula una W-función de las variables de entrada en las variables de salida.

Sea Xr la variable de control del bucle. Para cada elemento en la configuración con $Xr\neq 0$, el cuerpo del bucle se ejecuta una vez. Los resultados se agrupan con los elementos de la configuración de partida con Xr=0,

formando una configuración intermedia. Este proceso se repite hasta que todos los elementos en la configuración tengan Xr=0. Si una tal configuración no se alcanza, el proceso continua indefinidamente. Nótese que esta instrucción "while" borrosa, cuando se usa con datos nítidos, se comporta como un bucle indefinido nítido.

En [4] se demostró que cada programa en XL sin bucles indefinidos define una W-función total, en el sentido de que para cada configuración inicial dada se alcanza una y sólo una configuración final. Con programas con bucles indefinidos la situación es diferente; puede haber programas en los que no se alcanza una configuración final. Estos programas no terminan para ciertos datos de entrada.

La asignación difusa en XL es Xi:=FUZZn(Xj). Una W-función f_n de Z^+ en Z^+ está asociada a esta asignación. Sin embargo, es demasiado optimista suponer que cualquier W-función está disponible para las asignaciones.

Definición 1. Sea H un conjunto de W-funciones recursivas. Una W-function f es XL(H)- programable sii existe un XL-programa P que calcula f, supuesto que todas las asignaciones en P, Xi:=FUZZn(Xj), están asociadas a W-funciones $f_n \in H$. La clase de las funciones XL(H)-programables se denotará por F(XL(H)). Nótese que una función programable puede ser parcial. Definimos también FT(XL(H)) = {fp / P es un XL(H) programa con una ata para toda configuración inicial}. Por tanto, si f∈ FT(XL(H)), entonces existe un XL(H)-programa P que termina para toda entrada.

El caso más simple será H={I}, donde I es la identidad de Z^+ en Z^+. XL({I}) es la versión nítida de XL, es decir, el lenguaje SAL definido y estudiado en [9].

3 W-Máquinas de Turing multicinta, CONS y WRG

En esta sección definimos el concepto de W-TM multicinta. Hablando informalmente, una máquina multicinta lee y escribe en varias cintas. Estas construcciones son bien conocidas en la teoría de la calculabilidad nítida.

Definición 2. Una W-TM de k-cintas es una cuádrupla Z=(V, S, p, s_0) donde V y S son conjuntos finitos no vacíos, V∩S = ∅; $s_0 \in S$ y p es una W-función de $S\times(V)^k$ en $(V\cup\{L, R\})^k\times(S\cup\{h\})$; L,R∈ V, h∈ S.

Una descripción elemental de Z es una k+1-upla (q, $u_1\underline{a}_1v_1$, ..., $u_k\underline{a}_kv_k$) donde $q\in S\cup\{h\}$ y $u_i\underline{a}_iv_i \in (V-\{\#\})V^*)\underline{V}(V^*(V-\{\#\})\cup\{e\})$

Una descripción elemental es final cuando q=h.

Los conceptos de descripción, descripción final, cálculo y parada se definen de manera enteramente análoga a la del caso de una cinta [10].

Definición 3. Sea f una W-función de $(I^*)^r$ en $(O^*)^s$. Diremos que T=(V,S,p,s_0) *construye* f sii:

a) I,O ⊆V

b) Para cada $x\in (I^*)^r$, T se para con entrada x y la descripción final que se alcanza es D tal que $f(x,y)=\Sigma m_D(d)$, donde Σ representa la aplicación de ⊕ al conjunto de todas las descripciones elementales d tales que $m_D(d)\neq 0$, d=(h, c_1, ...,c_k) y $c_i=y_i\underline{\#}$, i=1,..,s, y=(y_1,..,y_s)

Una W-función f es *construible* sii existe una W-TM que construye f. CONS(r,s) denotará la clase de las W-funciones construibles de $(Z^+)^r$ en $(Z^+)^s$. CONS será la unión de todos los CONS(r,s) (r,s∈ Z^+, s>0). Nótese que esta definición coincide con la dada en [5] a partir de máquinas de una cinta.

Definición 4. Sea f una W-función de I^* en O^*. Sea $Im(f,x)=\{y\in O^* : f(x,y)\neq 0\}$. Sea sup(f,x)=sup(Im(f,x)) respecto al orden lexicográfico. Definimos la función nítida s_f: $I^*\rightarrow O^*$ como $s_f(x)=sup(f,x)$.

Definición 5. [2] Sea f una W-función de I^* en O^*. f∈ WRG (f es una función con grafo recursivo) sii su función de pertenencia $\mu_f: I^*\times O^*\rightarrow W$ es recursiva total.

Proposición 1. [5] La condición necesaria y suficiente para que una W-función con grafo recursivo sea construible es que la función nítida s_f sea recursiva.

Entre los XL-programas y las W-máquinas se puede establecer una correspondencia natural. Nos concentraremos en las máquinas con alfabeto V={I,#}, es decir, que calculan la clase CONS(r,s).

Notación. Sea $c=(c_1, ..., c_k)\in (Z^+)^k$. El par (q, c) representará una descripción elemental de la W-TM con k cintas (q, $c'_1\underline{\#}$, $c'_2\underline{\#}$, ..., $c'_k\underline{\#}$) donde c'_i= I...I (c_i veces).

Definición 6. Sea C una configuración de un k-programa en XL y sea D la descripción de una W-TM con k cintas. C and D son equivalentes sii existen un subconjunto finito $\{b_1,...,b_n\}\subset (Z^+)^k$, una sucesión ($a_1$, ..., a_n) en W, y un estado q tales que

$C = a_1/b_1+a_2/b_2+...+a_n/b_n$ y

$D = a_1/(q,b_1)+a_2/(q,b_2)+...+a_n/(q,b_n)$

Definición 7. Sea P un k-programa en XL y sea M una W-TM con k cintas, M=(V,S,p,s_0). P y M son equivalentes sii para todo a, b∈ $(Z^+)^k$ se tiene i) o ii):

i) Wp(a,b) no está definida y M no se para con entrada a.

ii) Wp(a,b)=b y M se para con entrada a alcanzando una descripción final D tal que $m_D(b)=b$

Comentario. Si P y M son equivalentes, entonces hay una correspondencia biunívoca entre los pares (descripción inicial, descripción final) en M y los pares ($val_{coin}(e)$, $val_{cofi}(e)$) en el árbol de derivación de P. En otras palabras, si $val_{coin}(e)$ y la descripción inicial son equivalentes, entonces P y M o bien se embarcan ambos en una computación sin fin o bien se paran ambos y, si

se paran ambos, $val_{cofi(e)}$ y la descripción final de M son equivalentes.

4 Caracterización de CONS mediante programas XL

Necesitaremos para las demostraciones de esta sección un teorema formulado y probado en [9] cuyas raíces se remontan a los trabajos de Wang [10] y Böhm-Jacopini [1] (vd. por ejemplo [7]). Con nuestra notación, el teorema es el siguiente:

Proposición 2. [9] Una función (nítida) f es recursiva sii f es XL({I})-programable.

Definición 8.- Sean I_1, ..., I_r y y_{01} las siguientes W-funciones Z^+ en Z^+:

$$I_i(x,y)=\begin{cases} I_i & \text{si } y=x \\ 0 & \text{en otro caso} \end{cases}$$

$$\psi_{01}(x,y)=\begin{cases} 1 & \text{si } (y=0) \text{ o } (y=1) \\ 0 & \text{en otro caso} \end{cases}$$

$$(i=1,..,r)$$

Definimos la familia C como $C = \{ I_1, ..., I_r, \psi_{01}\}$. Nótese que

a) C es finita y tiene r+1 elementos;

b) $C \subset CONS$;

c) $\{I\} \subset C$.

Proposición 3. Sea f una W-función construible de Z^+ en Z^+. Entonces f es XL(C)- programable.

Demostración: Sea f una W-función construible de Z^+ en Z^+; entonces, según se probó en [5], $f \in WRG$, y por la definición 4, f es una función recursiva total de $Z^+ \times Z^+$ en W, y hay un XL({I}) programa (nítido) que calcula f. Sea $P_1=(2,1,Q_1)$ este programa; P_1 es también un XL(C) programa. Considerando P1 como identificador de este programa, puede usarse como una macro en la forma

Xi := P1(Xj, Xk)

cuya ejecución produce el siguiente resultado: dados $a,b \in Z^+$, valores para Xj y Xk respectivamente, Xi almacena el grado f(a,b) codificado por un entero positivo 0,..,r.

Por otra parte, la función s_f: $Z^+ \to Z^+$ es nítida y recursiva y hay otro XL({I}) programa que la calcula. Sea $P_2=(1,1,Q_2)$ este programa. Usaremos P2 para identificarlo como macro.

Ahora es posible escribir un XL(C)-programa $P_3=(1,1,Q_3)$ que calcula f (el concepto de macro expansion se describe por completo en [4]):

```
P3;
  begin
1       X2:= P2(X1);
2       REPEAT X2 TIMES
3               IF X5≠1 THEN
4                  X4:= P1(X1,X3);
5                  IF X4≠0 THEN
6                     X5:= FUZZ0(X4);
7                     IF X5=1 THEN
8                        X6:= FLATTEN(X3,X4)
9                     FI
10                 FI
11              FI;
12              X3:=suc(X3)
13      SEMIT;
14      IF X5≠1 THEN
15              X4:= P1(X1,X3);
16              X6:=FLATTEN(X3,X4)
17      FI;
18      X1:=X6
  end
```

En este programa, la función asociada a FUZZ0 es ψ_{01}. La macro FLATTEN está asociada al programa P=(2,1, Q) cuyo código Q es:

```
FLATTEN;
begin
        IF X2=1 THEN X1:=FUZZ1(X1) FI;
        IF X2=2 THEN X1:=FUZZ2(X1) FI;
        ..............
        IF X2=r THEN X1:=FUZZr(X1) FI
end
```

Las W-funciones asociadas a FUZZi (i=1,...r) son I_i.

Una demostración formal de la equivalencia de este programa con la W-función f es tediosa. En lugar de ella, mostraremos cuáles son las configuraciones al principio, en un punto intermedio, y al final del cómputo de P3 aplicado a un ejemplo. Señalemos primeramente el papel de cada variable de P3:

X1: almacena la entrada y la salida (línea 18).

X2: almacena el valor de s_f(X1) y controla el ciclo principal (2-13)

X3: toma todos los valores desde 0 a s_f(X1); aquellos con f(X1,X3)≠0 constituyen la salida.

X4: almacena el grado f(X1,X3) codificado como un entero positivo.

X5: almacena 0 ó 1 para controlar una bifurcación.

X6:almacenamiento auxiliar para la salida.

Ejemplo 1. Sea f una W-función construible $f:Z^+ \times Z^+ \to W$ tal que para el argumento 9 la imagen es: f(9,0)=0, f(9,1)=0, f(9,2)=I_4, f(9,3)=I_6, f(9,4)=I_1, f(9,5)=0, f(9,6)=I_3, para todo x>6, f(9,x)=0 . Mostremos que f_P (9)=f(9).

La configuración inicial es: 1/(9,0,0,0,0,0)

Tras ejecutar 1, es 1/(9,6,0,0,0,0)

Ahora, el bucle definido (2-13) se repite 6 veces para X3=0,..,5 (instrucción 12). El valor final de X3 es 6.

Según 3, X5=0≠1, por tanto se ejecuta 4; ya que f(9,0)=0, la configuración no se cambia. Puesto que X4=0, la siguiente sentencia debe ser 12, cuya ejecución produce

1/(9,6,1,0,0,0)

y volvemos a 3. Como f(9,1)=0, el proceso se repite y tras 12 tenemos:

$1/(9,6,2,0,0,0)$

De nuevo en 3, se tiene que X5≠1, se ejecuta 4, se calcula para X4 el valor 4 (que codifica el grado l_4), y la configuración queda:

$1/(9,6,2,4,0,0)$

Ahora, en 5 se tiene queX4≠0, se ejecuta 6 y queda:

$1/(9,6,2,4,1,0) + 1/(9,6,2,4,0,0)$

La condición de 7 (X5=1) es satisfecha por la primera configuración elemental y tras ejecutar 8 queda:

$l_4/(9,6,2,4,1,2) + 1/(9,6,2,4,0,0)$

Sólo X3 cambiará en la primera configuración elemental; sin embargo, el grado y el valor de X6 no cambian. Ahora es la segunda configuración elemental la que cambia. Tras 12 tenemos:

$l_4/(9,6,3,4,1,2) + 1/(9,6,3,4,0,0)$

Para X5≠1, tras la ejecución de 4, X4 almacena el valor 6 (que codifica el valor l_6), y tenemos:

$l_4/(9,6,3,4,1,2) + 1/(9,6,3,6,0,0)$

X4≠0, y tras ejecutar 6 tenemos:

$l_4/(9,6,3,4,1,2) + 1/(9,6,3,6,1,0)+ 1/(9,6,3,6,0,0)$

Para la segunda configuración elemental X5=1, luego la línea 8 se ejecuta dando:

$l_4/(9,6,3,4,1,2) + l_6/(9,6,3,6,1,3)+ 1/(9,6,3,6,0,0)$

Tras la siguiente ejecución del ciclo tendremos:

$l_4/(9,6,4,4,1,2) + l_6/(9,6,4,6,1,3)+ l_1/(9,6,4,1,1,4) + 1/(9,6,4,1,0,0)$

Ahora X3 se incrementa en 12, y será:

$l_4/(9,6,5,4,1,2) + l_6/(9,6,5,6,1,3)+ l_1/(9,6,5,1,1,4) + 1/(9,6,5,1,0,0)$

Sólo la última configuración satisface X5≠1. Tras 4 será:

$l_4/(9,6,5,4,1,2) + l_6/(9,6,5,6,1,3)+ l_1/(9,6,5,1,1,4) + 1/(9,6,5,0,0,0)$

y X4≠0 no se satisface; la línea 12 se ejecuta y la configuración queda:

$l_4/(9,6,6,4,1,2) + l_6/(9,6,6,6,1,3)+ l_1/(9,6,6,1,1,4) + 1/(9,6,6,0,0,0)$

El bucle principal (2-13) se ha ejecutado 6 veces; en la línea 14 se cumple que X5≠1 en la última configuración elemental. Se ejecuta la línea 15 y tenemos:

$l_4/(9,6,6,4,1,2) + l_6/(9,6,6,6,1,3)+ l_1/(9,6,6,1,1,4) + 1/(9,6,6,3,0,0)$

Tras 16 obtenemos:

$l_4/(9,6,6,4,1,2) + l_6/(9,6,6,6,1,3)+ l_1/(9,6,6,1,1,4) + l_3/(9,6,6,3,0,6)$

Finalmente, tras 18:

$l_4/(2,6,6,4,1,2) + l_6/(3,6,6,6,1,3)+ l_1/(4,6,6,1,1,4) + l_3/(6,6,6,3,0,6)$

La única variable de salida es X1 y por tanto la salida del programa es

$l_4/2 + l_6/3+ l_1/4 + l_3/6$

Por tanto $fp(9,2)=l_4$, $fp(9,3)=l_6$, $fp(9,4)=l_1$, $fp(9,6)=l_3$; y en otro caso fp(9,x)=0; el programa produce los mismos valores que f para entrada 9.

Proposición 4.-CONS ⊆ FT(XL(*C*))

Demostración: Sea f una W-función construible de $(Z^+)^n$ en $(Z^+)^k$. Consideremos las codificaciones de Cantor (vd. por ejemplo [9]) c_n: $(Z^+)^n \rightarrow Z^+$ y c_k:$(Z^+)^k \rightarrow Z^+$. Tanto estas como sus inversas son funciones recursivas nítidas. Entonces podemos considerar f' de Z^+ en Z^+ definida por $f'(x,y)=f(c_n^{-1}(x),c_k^{-1}(y))$. Obviamente, f' es una función construible y por el teorema 4, f' es XL(*C*)-programable, por tanto $f(u,v) = f'(c_n(u),c_k(v))$ es también XL(*C*)-programable, es decir, f∈ F(XL(*C*)). Puesto que toda f∈ CONS es total, f∈FT(XL(*C*))

Ahora probaremos la inclusión contraria, es decir, que FT(XL(*C*))⊆ CONS. Para cada programa P, mostraremos una máquina equivalente M, en el sentido de la definición 7 (lema 4). La prueba del lema 4 se hará por inducción estructural sobre P, a partir de los programas con una sola instrucción (lema 3), simulando la secuencia (lema 1) y el anidamiento (lema 2). Puesto que los ciclos "repeat ... times" pueden sustituirse con ciclos "while", consideraremos únicamente éstos.

Lema 1. Sea P_1 un k-programa en XL con código

```
P1; begin B1 end.
```

Sea P_2 un k-programa en XL con código

```
P2; begin B2 end.
```

Supongamos que hay dos W-máquinas T_1 y T_2 equivalentes respectivamente a P_1 y P_2. Entonces hay una máquina T equivalente al XL-programa P cuyo código es

```
P12;begin B1;B2 end
```

Demostración: Sea $T_i=(V,S_i,p_i,s_{0i})$ (i=1,2) la W-MT equivalente al programa P_i.

Entonces la W-TM $T=(V,S,p,s_0)$ se define como sigue:

$S=S_1 \cup S_2$

$p(q,c,c',q') = p_1(q,c,c',q')$ si $q,q' \in S_1$ y $q' \neq h$

$p(q,c,c',s_{02}) = p_1(q,c,c',h)$ si $q \in S_1$

$p(q,c,c',q') = p_2(q,c,c',q')$ si $q,q' \in S_2$

$s_0=s_{01}$

Las transiciones de T son, aproximadamente, las de T_1 más las de T_2: sin embargo, en las transiciones de parada de T_1, h se sustituye por s_{02}. T comienza en el estado inicial de T_1; opera como T_1 hasta que T_1 se pararía (simulando así P1); luego opera como T_2 (simulando así P2). Es fácil verificar que T es equivalente a P.

En lo sucesivo, la máquina T así definida se representará por $T_1 o T_2$ (T_1 concatenada con T_2)

Lema 2. Sea P_1 un k-programa en XL con código

```
P1; begin B1 end.
```

Supongamos una máquina T_1 equivalente a P_1. Entonces hay una máquina T equivalente al k-program P cuyo código es

`P1i; begin while Xi≠0 do B1 od end`

Demostración: Sea $T_1=(V,S_1,p_1,s_{01})$ la W-MT equivalente al programa P_1. Entonces la W-TM $T=(V,S,p,s_0)$ se define como sigue:

$S= S_1 \cup \{s_0, s'\}$, $(s_0, s' \in S_1)$

$p(s_0,\#(i), L(i), s')=1$

$p(s',l(i),R(i),s_{01})=1$

$p(s',\#(i),R(i),h)=1$

$p(q,c,c',q')= p_1(q,c,c',q')$ si $q,q' \in S_1$ y $q' \neq h$

$p(q,c,c',s_0)= p_1(q,c,c',h)$ si $q \in S_1$

Es decir, T comprueba si el contenido de la i-ésima cinta es ≠0; en ese caso, el estado de T es el estado s_{01} -estado inicial de T_1- y T opera como T_1 hasta que T_1 se hubiera parado, sustituyendo el estado de parada h por el estado inicial s_0; en otro caso, T se para. Evidentemente, T y P son equivalentes : T termina sii P se para y, si P se para, la descripción final alcanzada por T es equivalente a la configuración final de P tras la ejecución de

`begin while Xi≠0 do B1 od end`

En lo sucesivo, la máquina T así definida se representará por w(T,i).

Lema 3. Para todo programa P sin bucles, hay una W-TM equivalente.

Demostración: Consideremos primeramente programas de una instrucción, y posteriormente secuencias de instrucciones. Supongamos que P es

`ident; begin <asignación> end`

y `<asignación>` es `Xi:=0`. T es equivalente a P:

$T=(V,S,p,s_0)$,

$S = \{s_0, s_1, s_2\}$,

y p se define como sigue:

$p(s_0, \#(i), L(i), s_1) = 1$ $\quad p(s_1, l(i), \#(i), s_2)= 1$

$p(s_1, \#(i), \#(i), h) = 1$ $\quad p(s_2, \#(i), L(i), s_1)=1$

Supongamos ahora que `<asignación>` es `Xi:=suc(Xj)`; la máquina T definida más abajo es equivalente al programa P:

$T=(V,S,p,s_0)$, $S = \{s_0, ..., s_6\}$

$p(s_0, \#(i), L(i), s_1) =$ $\quad p(s_1, l(i), \#(i), s_2) =1$

$p(s_1, \#(i), \#(i), s_3) = 1$ $\quad p(s_2, \#(i), L(i), s_1)=1$

$p(s_3, \#(j), L(j), s_4)= 1$ $\quad p(s_4, \#(j), R(j),s_5)=1$

$p(s_4, l(j), L(j), s_4) = 1$ $\quad p(s_5, l(j), l(i), s_6) =1$

$p(s_5, \#(j), l(i), s_6) = 1$ $\quad p(s_6, l(j), R(i,j),s_5)=1$

$p(s_6, l(i), R(i), h) = 1$

Las primeras 4 líneas borran la i-ésima cinta. Las transiciones con estados s_3 y s_4 ponen la máquina al comienzo del contenido a_j de la j-ésima cinta. Las transiciones con estados s_5 y s_6 copian el contenido de la j-ésima cinta en la i-ésima. Finalmente, otro "|" se escribe en la i-ésima cinta.

Sea ahora `<asignación> Xi:=FUZZ0(Xj)`. Recordemos que la función asociada a FUZZ0 es ψ_{01}. La máquina T es equivalente a P:

$T=(V,S,p,s_0)$, $S = \{s_0, ..., s_3\}$

$p(s_0, \#(i), L(i), s_1) = 1$ $\quad p(s_1, l(i), \#(i), s_2) =1$

$p(s_1, \#(i), \#(i), h) = 1$ $\quad p(s_1, \#(i), l(i), s_3) =1$

$p(s_2, \#(i), L(i), s_1) = 1$ $\quad p(s_3, l(i), R(i), h) =1$

La W-función ψ_{01} siempre produce las imágenes 0 (grado 1) y 1 (grado 1). En efecto, T borra la i-ésima cinta y entonces a) se para con grado 1 y b) escribe un solo "|" y se para con grado 1.

Supongamos ahora que `<asignación>` es `Xi:=FUZZk(Xj)`. Recordemos que la W-función asociada a FUZZk is I_k. La máquina T es equivalente a P:

$T=(V,S,p,s_0)$, $S = \{s_0, ..., s_6\}$

$p(s_0, \#(i), L(i), s_1) = 1$ $\quad p(s_1, l(i), \#(i), s_2) =1$

$p(s_1, \#(i), \#(i), s_3) = 1$ $\quad p(s_2, \#(i), L(i), s_1)=1$

$p(s_3, \#(j), L(j), s_4)= 1$ $\quad p(s_4, \#(j), R(j),s_5)=1$

$p(s_4, l(j), L(j), s_4) = 1$ $\quad p(s_5, l(j), l(i), s_6) =1$

$p(s_5, \#(j), \#(i), h) = I_k$ $\quad p(s_6, l(j), R(i,j),s_5)=1$

La W-función I_k produce x (grado I_k) con entrada x. En efecto, las únicas acciones que T realiza son copiar x_j en la i-ésima cinta y pararse con grado I_k.

Supongamos ahora que P es

```
<ident>;
begin
   <asignación>;
   ...
   <asignación>;
end
```

Nótese que basta probar el lema para secuencias de dos instrucciones. Pero por el lema 1 ello está probado. Por consiguiente, el lema 3 está también demostrado.

Lema 4. Para todo XL(C)-programa existe una W-TM equivalente.

Demostración: Supongamos que el programa no tiene bucles anidados. Por el lema 3, tal máquina existe.

Supongamos que la proposición es verdadera para todos los programas con bucles anidados b o menos niveles. Debemos demostrar que el teorema es verdadero para los programas con b+1 niveles de anidamiento. Sea P uno de ellos; P puede escribirse como

`<ident>; begin` S_1; S_2 ;... S_n `end`

donde cada S_i es o una asignación o un bucle

`while Xi≠0 do` Q_i `od`

y el programa P_i

`<ident> begin Qi end`

tiene bucles anidados a b o menos niveles. Si S_i es una asignación entonces, por el lema 3, hay una máquina M_i equivalente a

`<ident> begin` S_i `end`

si S_i es un bucle while, entonces, por el lema 2 y la hipótesis de inducción, hay también una máquina $T_i=w(T,i)$ equivalente a

`<ident> begin while Xi≠0 do Qi od end.`

En cualquier caso, un razonamiento similar al del lema 1 muestra que $T=T_1 o T_2 o ... o T_n$ es equivalente a P.

Proposición 5. $FT(XL(C)) \subseteq$ CONS

Demostración: Sea f una W-función de FT(XL(*C*)). Por el lema 4, hay una W-TM M equivalente a P. Como f es total, P termina con cualquier entrada. Por tanto, M se para siempre en un número finito de pasos y f∈CONS.

5. Caracterización de la clase WRG mediante programas XL.

Definiremos una familia finita de W-funciones *G* y probaremos que es suficiente para programar cualquier función f∈WRG.

Definición 9. Sean I_1, ..., I_r las W-funciones de Z^+ en Z^+ de la definición 10, y sea ψ_∞ la siguiente W-función de Z^+ en Z^+: $\forall x,y\in Z^+$ $\psi_\infty(x,y)=1$.

La familia *G* es $G = \{ I_1, ..., I_r, \psi_\infty \}$

Nótese que

a) *G* es finita y tiene r+1 elementos;

b) $G\subset$WRG;

c) $\{I\}\subset G$

Proposición 6. Sea f una W-función con grafo recursivo de Z^+ en Z^+. Entonces, f es XL(*G*)-programable.

Demostración: Sea f una W-función con grafo recursivo; en ese caso, por la definición 9, f es una función recursiva total de $Z^+\times Z^+$ en $W_{,}$, y existe un XL({I}) programa que calcula f. Usando los mismos identificadores de la sección anterior, el siguiente XL(*G*)-programa $P_3=(1,1,Q_3)$ calcula f:

```
P3;
begin
    X3:= FUZZ0(X1);
    X4:= P1(X1,X3);
    X6:= FLATTEN(X3,X4);
    X1:= X6
end
```

donde la única diferencia con la sección anterior es la W-función asociada a FUZZ0, que es ψ_∞.

El papel jugado por las variables es el mismo; sin embargo, nótese que X2 y X5 no se usan en el programa y sus valores son 0 a lo largo de toda la ejecución. Sea $f(a,i)=a_i\in W$, $i\in Z^+$.

Con entrada a, la configuración inicial es 1/(a,0,0,0,0,0). Tras la ejecución de X3:= FUZZ0(X1), la configuración es:

$1/(a,0,0,0,0,0) + 1/(a,0,1,0,0,0) + ...+ 1/(a,0,k,0,0,0)+ ...$

Entonces se ejecuta X4:= P1(X1,X3), produciendo la configuración:

$1/(a,0,0,n_0,0,0) + 1/(a,0,1,n_1,0,0) + ...+ 1/(a,0,k,n_k,0,0)+ ...$

y tras ejecutar la siguiente instrucción X6:= FLATTEN(X3,X4) tendremos:

$a_0/(a,0,0,n_0,0,0) + a_1/(a,0,1,n_1,0,1) + ...+ a_k/(a,0,k,n_k,0,k)+ ...$

Finalmente, se ejecuta X1:=X6 y la configuración final es:

$a_0/(0,0,0,n_0,0,0) + a_1/(1,0,1,n_1,0,1) + ...+ a_k/(k,0,k,n_k,0,k)+ ...$

Por tanto, la W-función semántica asociada con P3, $f_p:Z^+\times Z^+\rightarrow W$, satisface $f_p(a,i)=a_i = f(a,i)$,$i\in Z^+$, es decir, f =f_p y f is XL(*G*)-programable.

Proposición 7.- WRG$\subseteq$ F(XL(*G*))

Demostración: Como en el corolario 1.

6 Conclusiones

En este trabajo estudiamos la caracterización mediante programas de ciertas aproximaciones al concepto de "W-función calculable" anteriormente propuestas en la literatura. Desde un punto de vista teórico, la calculabilidad difusa es una materia algo compleja que se ha tratado desde varios puntos de vista [3], [8], algunos de ellos demasiado extensivos. En este trabajo consideramos los dos enfoques más restringidos: la clase WRG de G. Gerla [2] y nuestra clase CONS [5]. Considerando la familia de lenguajes de programación XL(H) generada al restringir la asignación a funciones de H, se han definido dos familias de funciones F(XL(*C*)) y F(XL(*G*)) tales que $C\subset$CONS, CONS = FT(XL(*C*)), y $G\subset$ WRG, WRG$\subseteq$ F(XL(*G*)).

Referencias

[1] Böhm, C., Jacopini, G., Flow diagrams, Turing machines and languages with only two formation rules. *Comm. ACM* 9 (1966), 366-371.

[2] Gerla, G., Turing L-Machines and Recursive Computability for L-Maps. *Studia Logica* XLVIII (2) (1989), 179-192.

[3] Harkleroad, L., Fuzzy Recursion, Ret's and Isols. *Zeitschr. f. math. Logik und Grundlagen d. Math*, 30 (1984), 425-436.

[4] Morales-Bueno, R. et al., An elementary fuzzy programming language. *Fuzzy Sets and Systems* 58 (1993), 55-73.

[5] Morales-Bueno, R. et al., Estudio de una clase de W-funciones calculables, *III Congreso Español sobre Tecnologías y Lógica fuzzy*, Santiago de Compostela 1993, pp. 153-160.

[6] Morales-Bueno, R. et al., El lenguaje de programación borrosa XL, *V Congreso Español sobre Tecnologías y Lógica fuzzy*, Murcia 1995.

[7] Odifreddi, P., *Classical Recursion Theory*. North-Holland, 1992.

[8] Santos, E.S., Fuzzy algorithms. *Inform. Contr.* 17 (1970), 326-339.

[9] Sommerhalder, R., van Westrhenen, S.C., *The Theory of Computability*. Addison-Wesley, 1988.

[10] Wang, H., A variant of Turing's theory of calculating machines. *Journal ACM* 4(1957) 63-92.

Métodos intuicionistas de inferencia: Comparación por compatibilidades

Pedro Burillo López
Dep. de Matemática e Informática
Universidad Pública de Navarra
Campus de Arrosadía s/n
31006 Pamplona, España
e-mail: pburillo@si.upna.es

Humberto Bustince Sola
Dep. de Matemática e Informática
Universidad Pública de Navarra
Campus de Arrosadía s/n
31006 Pamplona, España
e-mail: bustince@si.upna.es

Victoria Mohedano Salillas
Dep. de Matemática e Informática
Universidad Pública de Navarra
Campus de Arrosadía s/n
31006 Pamplona, España
e-mail: vmohedano@si.upna.es

Resumen

En este trabajo estudiamos el modus ponens generalizado para conjuntos intuicionistas fuzzy. Por una parte analizamos la composición de un conjunto intuicionista fuzzy con una relación intuicionista y por otra, el grado de compatibilidad intuicionista. Seguidamente estudiamos las principales propiedades de dos grados de compatibilidad intuicionista particulares. A continuación, exponemos un algoritmo que nos permite obtener una conclusión del modus ponens generalizado para intuicionistas utilizando los grados de compatibilidad anteriores. Finalizamos el trabajo comparando los resultados obtenidos cuando se utiliza el algoritmo, con los obtenidos por medio de la regla composicional mediante dos relaciones intuicionistas fuzzy concretas.

Palabras clave: *Reglas de inferencia, modus ponens generalizado, conjuntos intuicionistas fuzzy, relaciones intuicionistas, grado de compatibilidad intuicionista.*

1 Introducción

Sabemos que en sistemas expertos una regla tiene la forma:

$$Si\ x\ es\ A\ entonces\ y\ es\ B$$

donde "x"e"y" representan variables linguísticas y A, B son respectivamente conjuntos difusos sobre los referenciales $X \neq \phi$ e $Y \neq \phi$.

En el marco del cálculo de Zadeh [9, 10, 11] una regla se interpreta como una relación R entre los conjuntos difusos A y B. Más precisamente si $FSs(X)$, $FSs(Y)$ y $FSs(X \times Y)$ representan los conjuntos de difusos sobre los referenciales $X, Y, X \times Y$, entonces

$$R : FSs(X) \times FSs(Y) \rightarrow FSs(X \times Y)$$

La relación R puede ser definida a partir de los operadores implicación, es decir, aplicaciones

$$L : [0,1] \times [0,1] \rightarrow [0,1]$$

verificando

$$L(0,0) = L(0,1) = L(1,1) \text{ y } L(1,0) = 0$$

que generalizan la implicación lógica clásica [6], y de los que se conocen abundantes ejemplos en la bibliografía. Con estos operadores, R se obtiene de la siguiente forma:

$$\mu_{R(A,B)}(x,y) = L(\mu_A(x), \mu_B(x)).$$

No obstante es necesario tener en cuenta que a partir de una regla no podemos adquirir ningún conocimiento, por este motivo es necesario tener información adicional de alguna de las variables relacionadas en la regla *Si x es A entonces y es B*. Una de las mayores ventajas de los sistemas expertos

fuzzy es que no es necesario tener una correspondencia perfecta entre el antecedente de la regla y la información dada sobre una de las variables, es decir, la información adicional puede ser de la forma x *es* A', donde A' es un conjunto fuzzy sobre el mismo referencial que el conjunto fuzzy A.

En estas condiciones, las dos premisas:

$$\begin{array}{c} \textit{Si } x \textit{ es } A, \textit{ entonces } y \textit{ es } B \\ x \textit{ es } A' \end{array}$$

con $A, A' \in FSs(X)$, $B \in FSs(Y)$, permiten obtener nueva información para la variable "y", dada por:

$$y \text{ es } B', \text{ con } B' \in FSs(Y)$$

constituyendo este proceso el método de inferencia llamado *modus ponens generalizado*, consistente en obtener B' por medio de la regla composicional [4, 5]

$$B' = A' \circ R(A, B).$$

En otro orden de ideas la utilización de los conjuntos fuzzy puede suponer, en algunos casos, una pérdida de información al tener que determinar con precisión los valores de las funciones de pertenencia $\mu_A(x), \mu_B(x)$, etc; por esta razón Zadeh [7, 8, 11] planteó la conveniencia de utilizar conjuntos difusos intervalo-valorados para desarrollar el modus ponens generalizado. Más adelante, en 1986, K. Atanassov [1, 2] introdujo los conjuntos intuicionistas fuzzy y demostró que para determinadas interpretaciones estos conjuntos y los intervalo-valorados son isomorfos. En este trabajo estudiamos el *modus ponens generalizado* con conjuntos intuicionistas fuzzy y presentamos un algoritmo para calcular la conclusión de tal regla a la vez que comparamos los resultados obtenidos con este algoritmo y los que se obtendrían con la regla composicional de Zadeh adaptada a conjuntos intuicionistas fuzzy.

Denotaremos con X e Y a los conjuntos referenciales, siendo estos siempre finitos y no vacíos, con $IFSs(X)$ al conjunto de conjuntos intuicionistas fuzzy sobre X y con $IFRs(X \times Y)$ al conjunto de relaciones intuicionistas fuzzy sobre $X \times Y$.

Empezamos el artículo recordando las propiedades que vamos a utilizar de los $IFSs(X)$, a continuación definiremos el grado de compatibilidad de conjuntos intuicionistas y analizamos las propiedades más importantes de dos ejemplos de tal grado. Seguidamente presentamos un algoritmo para obtener la conclusión del modus ponens generalizado.

Terminamos estudiando los resultados que se obtienen con este algoritmo y con la regla de inferencia composicional establecida por Zadeh cuando se utilizan dos relaciones intuicionistas fuzzy concretas.

2 Conjuntos intuicionistas fuzzy

Un estudio detallado y completo de estos conjuntos puede encontrarse en [1, 3]. Aquí mostramos las propiedades de los IFS que vamos a utilizar más adelante.

Definición 1. Sea $X \neq \phi$ un conjunto dado. Un conjunto intuicionista fuzzy sobre X es una expresión A dada por

$$A = \{< x, \mu_A(x), \nu_A(x) > | x \in X\} \qquad \text{donde}$$
$$\mu_A : X \longrightarrow [0, 1]$$
$$\nu_A : X \longrightarrow [0, 1]$$

verifican $\mu_A(x) + \nu_A(x) \leq 1$, para todo $x \in X$

Las funciones μ_A y ν_A reciben el nombre de funciones de pertenencia y no pertenencia respectivamente.

Las siguientes expresiones son definidas en [1, 3] para todo $A, B \in IFSs(X)$

1. $A \leq B$ si y sólo si
   $\mu_A(x) \leq \mu_B(x)$ y $\nu_A(x) \geq \nu_B(x)$ $\forall x \in X$
2. $A = B$ si y sólo si $A \leq B$ y $B \leq A$
3. $A_c = \{< x, \nu_A(x), \mu_A(x) > | x \in X\}$
4. $\max(A, B) = \{< x, \max(\mu_A(x), \mu_B(x)),$
   $\min(\nu_A(x), \nu_B(x)) > | x \in X\}$
5. $\min(A, B) = \{< x, \min(\mu_A(x), \mu_B(x)),$
   $\max(\nu_A(x), \nu_B(x)) > | x \in X\}$

Debido a la importancia que tendrán más adelante, denotaremos con S_k y S_n a los siguientes conjuntos intuicionistas fuzzy:

$$\begin{aligned} S_k(A, B) &= \max(A_c, B) = \\ &= \{< x, \max(\nu_A(x), \mu_B(x)), \\ &\quad \min(\mu_A(x), \nu_A(x) > | x \in X\} \\ S_n(A, B) &= \min(A_c, B) = \\ &= \{< x, \min(\nu_A(x), \mu_B(x)), \\ &\quad \max(\mu_A(x), \nu_A(x) > | x \in X\} \end{aligned}$$

2.1 Relaciones

A continuación damos la definición de relación intuicionista fuzzy y exponemos la composición de un

conjunto intuicionista con una relación intuicionista fuzzy, también damos la expresión de la proyección. Un análisis de tales relaciones se encuentra en [3].

Definición 2. Llamamos relación intuicionista fuzzy a todo subconjunto intuicionista fuzzy de $X \times Y$, es decir, a toda expresión R de la forma:

$$R = \{< (x,y), \mu_R(x,y), \nu_R(x,y) > | x \in X, y \in Y\}$$

Definición 3. Sea $A' \in IFSs(X)$ y $R \in IFRs(X \times Y)$, el conjunto $A' \circ R \in IFSs(Y)$ está definido de la manera siguiente:

$$\mu_{A' \circ R}(y) = \operatorname*{Max}_{x \in X}\Big[\min\{\mu_{A'}(x), \mu_R(x,y)\}\Big]$$
$$\nu_{A' \circ R}(y) = \operatorname*{Min}_{x \in X}\Big[\max\{\nu_{A'}(x), \nu_R(x,y)\}\Big]$$

Definición 4. A cada relación intuicionista fuzzy entre X e Y podemos asociarle un subconjunto intuicionista fuzzy de Y mediante

$$R^Y = \{< y, \mu_{R^Y}(y), \nu_{R^Y}(y) > | y \in Y\}$$
$$\mu_{R^Y}(y) = \operatorname*{Max}_x\Big(\mu_R(x,y)\Big)$$
$$\nu_{R^Y}(y) = \operatorname*{Min}_x\Big(\nu_R(x,y)\Big)$$

a R^Y le llamaremos proyección de R sobre Y.

2.2 Grado de compatibilidad entre conjuntos intuicionistas fuzzy

En lo sucesivo utilizaremos la siguiente notación:

$$\widetilde{1} = \{< x, 1, 0 > | x \in X\}$$
$$\widetilde{0} = \{< x, 0, 1 > | x \in X\}$$
$$D = \{(x,y) \in [0,1] \times [0,1] | x + y \leq 1\}$$

Definición 5. Una función

$$\Gamma : IFSs(X) \times IFSs(Y) \to D$$

es un *grado de compatibilidad intuicionista*, si satisface las siguientes propiedades:

1. $\Gamma(\widetilde{0}, \widetilde{1}) = (1,0)$
2. $\Gamma(A,B) = \Gamma(B_c, A_c)$

Dos ejemplos de grado de compatibilidad intuicionista son los siguientes:

$$\Gamma_k(A,B) = \Big(\operatorname*{Max}_{x \in X}(\max(\nu_A(x), \mu_B(x)),$$
$$\operatorname*{Min}_{x \in X}(\min(\mu_A(x), \nu_B(x))\Big) =$$
$$= \Big(\operatorname*{Max}_{x \in X}(\mu_{S_k(A,B)}(x)), \operatorname*{Min}_{x \in X}(\nu_{S_k(A,B)}(x))\Big)$$

$$\Gamma_n(A,B) = \Big(\operatorname*{Max}_{x \in X}(\min(\nu_A(x), \mu_B(x)),$$
$$\operatorname*{Min}_{x \in X}(\max(\mu_A(x), \nu_B(x))\Big) =$$
$$= \Big(\operatorname*{Max}_{x \in X}(\mu_{S_n(A,B)}(x)), \operatorname*{Min}_{x \in X}(\nu_{S_n(A,B)}(x))\Big)$$

en todo lo que sigue emplearemos exclusivamente estos dos grados de compatibilidad intuicionista.

Los siguientes teoremas muestran las propiedades más importantes de Γ_k y Γ_n respectivamente.

Teorema 1.

1) Si $A = B$ conjuntos intuicionistas fuzzy normales, entonces $\Gamma_k(A,B) = (1,0)$.

2) $\Gamma_k(\widetilde{0}, \widetilde{0}) = \Gamma_k(\widetilde{1}, \widetilde{1}) = (1,0)$

3) $\Gamma_k(A,B) = (0,1)$ sí y sólo si $A = \widetilde{1}$ y $B = \widetilde{0}$

4) $\Gamma_k(\widetilde{0}, B) = \Gamma_k(A, \widetilde{1}) = (1,0)$

5) $\Gamma_k(\widetilde{1}, B) = \Big(\operatorname*{Max}_{x \in X}(\mu_B(x), \operatorname*{Min}_{x \in X}(\nu_B(x))\Big)$

6) $\Gamma_k(A, \widetilde{0}) = \Big(\operatorname*{Max}_{x \in X}(\nu_A(x), \operatorname*{Min}_{x \in X}(\mu_A(x))\Big)$

Teorema 2.

1) $\Gamma_n(\widetilde{0}, \widetilde{0}) = \Gamma_n(\widetilde{1}, \widetilde{1}) = (0,1)$

2) Si $A = \widetilde{1}$ y $B = \widetilde{0}$, entonces $\Gamma_n(A,B) = (0,1)$

3) $\Gamma_n(\widetilde{0}, B) = \Big(\operatorname*{Max}_{x \in X}(\mu_B(x), \operatorname*{Min}_{x \in X}(\nu_B(x))\Big)$

4) $\Gamma_n(A, \widetilde{1}) = \Big(\operatorname*{Max}_{x \in X}(\nu_A(x), \operatorname*{Min}_{x \in X}(\mu_A(x))\Big)$

2.3 Construcción de $IFSs$ constantes

La definición anterior de grado de compatibilidad asocia a cada pareja de conjuntos intuicionistas fuzzy un elemento de

$$D = \{(x, y) \in [0,1] \times [0,1] | x + y \leq 1\},$$

es precisamente la condición $x + y \leq 1$ la que nos permite construir un conjunto intuicionista fuzzy constante, siendo para todo elemento la pertenencia igual a "x" y la no pertenencia igual a "y". A tales conjuntos los denotaremos con la letra Φ_i, donde el subíndice indica el grado de compatibilidad utilizado en la construcción del $IFSs$, es decir $i = k, n$.

Teorema 3. Para cada $A, B \in IFSs(X)$ se verifica:

1) $\max(\Gamma_k(A, C), \Gamma_k(B, C)) =$
$= \Gamma_k(\min(A, B), C) = \Gamma_k(A, \max(B_c, C))$

2) $\max(\Gamma_n(A, C), \Gamma_n(B, C)) =$
$= \Gamma_n(\min(A, B), C)$

3) $\Gamma_n(\max(A, B), C) = \Gamma_n(A, \min(B_c, C))$

3 Construcción algorítmica y composicional

Vamos a estudiar el modus ponens generalizado para el caso particular en que $A, A' \in IFSs(X)$ y $B' \in IFSs(Y)$. Además, para la abtención de la consecuencia vamos a utilizar dos métodos diferentes:

I) La regla composicional de Zadeh adaptada a conjuntos intuicionistas fuzzy.

II) Un método algorítmico dado por los pasos siguientes:

[*A*]

1) Calcular el grado de compatibilidad entre A y A', es decir $\Gamma_i(A, A')$ con $i = k, n$.
2) Construir un conjunto intuicionista fuzzy constante sobre el referencial Y utilizando el grado de compatibilidad anterior, es decir

$$\Phi_i(A, A') =$$
$$= \{< y, \mu_{\Phi_i(A,A')}(y), \nu_{\Phi_i(A,A')}(y) > | y \in Y\}$$

donde

$\mu_{\Phi_i(A,A')}(y)$ = primera componente de $\Gamma(A, A')$

$\nu_{\Phi_i(A,A')}(y)$ = segunda componente de $\Gamma(A, A')$

[*B*]

$$\mathbf{B}' = \min\Big(\Phi_i(A, A'), B\Big).$$

La diferencia fundamental entre estos dos métodos radica en la siguiente idea: " mientras que con el método I) primero obtenemos la relación entre A y B dada por la proposición condicional *Si x es A estonces y es B*, y luego componemos con A' para obtener la consecuencia B'; con el método II) primero relacionamos A con A' mediante el grado de compatibilidad y después construimos la consecuencia B' utilizando el resultado de la comparación anterior y B".

3.1 Comparación de resultados para Γ_k

La regla de inferencia composicional cuando utilizamos conjuntos intuicionistas fuzzy y la relación intuicionista fuzzy $R_k(A, B)$ dada por

$$\mu_{R_{k(A,B)}}(x, y) = \max(\nu_A(x), \mu_B(y))$$
$$\nu_{R_{k(A,B)}}(x, y) = \min(\mu_A(x), \nu_B(y))$$

establece que: $B'_k = A' \circ R_k(A, B)$, y por tanto

$$\mu_{B'_{k(A,B)}}(y) =$$
$$= \operatorname*{Max}_{x \in X}\left[\min\left\{\mu_{A'}(x), \max(\nu_A(x), \mu_B(y))\right\}\right]$$
$$\nu_{B'_{k(A,B)}}(y) =$$
$$= \operatorname*{Min}_{x \in X}\left[\max\left\{\nu_{A'}(x), \min(\mu_A(x), \nu_B(y))\right\}\right]$$

Tomando $\Gamma_k(A, A')$, el algoritmo anterior nos permite construir la siguiente consecuencia:

$$\mathbf{B}'_k = \min\Big(\Phi_k(A, A'), B\Big), \quad \text{donde}$$

$$\mu_{\mathbf{B}'_k}(y) =$$
$$= \min\left\{\operatorname*{Max}_{x \in X}\left[\max(\nu_A(x), \mu_{A'}(x))\right], \mu_B(y)\right\}$$
$$\nu_{\mathbf{B}'_k}(y) =$$
$$= \max\left\{\operatorname*{Min}_{x \in X}\left[\min(\mu_A(x), \nu_{A'}(x))\right], \nu_B(y)\right\}$$

Teorema 4.

1) Si $A'' \leq A'$, entonces $\mathbf{B}''_k \leq \mathbf{B}'_k$ y $B''_k \leq B'_k$

2) Si A' es normal, entonces $\mathbf{B}'_k = B$

3) Si A_c es normal y $A' = A$, entonces $\mathbf{B}'_k = B$

3.2 Comparación de resultados para Γ_n

Utilizando la regla de composición de conjuntos intuicionistas fuzzy con la relación intuicionista fuzzy $R_n(A, B)$ dada por:

$$\mu_{R_{n(A,B)}}(x,y) = \min(\nu_A(x), \mu_B(y))$$
$$\nu_{R_{n(A,B)}}(x,y) = \max(\mu_A(x), \nu_B(y))$$

tenemos $B'_n = A' \circ R_n(A,B)$, donde

$$\mu_{B'_n}(y) =$$
$$= \operatorname*{Max}_{x\in X}\left[\min\left\{\mu_{A'}(x), \min(\nu_A(x), \mu_B(y))\right\}\right]$$
$$\nu_{B'_n}(y) =$$
$$= \operatorname*{Min}_{x\in X}\left[\max\left\{\nu_{A'}(x), \max(\mu_A(x), \nu_B(y))\right\}\right]$$

La consecuencia obtenida por el algoritmo para $\Gamma_n(A,A')$ es:

$$\mathbf{B}'_n = \min\left(\Phi_I(A,A'), B\right), \quad \text{donde}$$

$$\mu_{\mathbf{B}'_n}(y) =$$
$$= \min\left\{\operatorname*{Max}_{x\in X}\left[\min(\nu_A(x), \mu_{A'}(x))\right], \mu_B(y)\right\}$$
$$\nu_{\mathbf{B}'_n}(y) =$$
$$= \max\left\{\operatorname*{Min}_{x\in X}\left[\max(\mu_A(x), \nu_{A'}(x))\right], \nu_B(y)\right\}$$

Teorema 5. $\mathbf{B}'_n = B'_n$

Este teorema establece que la conclusión obtenida cuando en la construcción algorítmica utilizamos $\Gamma_n(A,A')$ es la misma que la obtenida por la regla composicional para conjuntos intuicionista, hecho que no ocurre cuando empleamos en el algoritmo el grado de compatibilidad intuicionista $\Gamma_k(A,A')$.

Teorema 6.

1) Si $A'' \leq A'$, entonces $\mathbf{B}''_n \leq \mathbf{B}'_n$ y $B''_n \leq B'_n$

2) Si A_c es normal y $A' = A_c$, entonces $B'_n = B$.

Notemos que para Γ_n no se verifica que si A' es normal, entonces $\mathbf{B}'_n = B$, este hecho pone de manifiesto una diferencia importante entre la conclusión obtenida por medio del algoritmo cuando utilizamos el grado de compatibilidad $\Gamma_n(A,A')$ o cuando utilizamos $\Gamma_k(A,A')$.

3.4 Comparación general de métodos

El siguiente teorema presenta la relación existente entre las conclusiones deducidas con el algoritmo para las dos expresiones de grado de compatibilidad intuicionista estudiadas, así como la relación entre las conclusiones cuando se utiliza la regla de inferencia composicional con las relaciones intuicionistas fuzzy R_k y R_s.

Teorema 7.

1) $\mathbf{B}'_k \geq \mathbf{B}'_n$

2) $B'_n \leq B'_k$

Anteriormente hemos visto que la conclusión obtenida por medio del algoritmo cuando se emplea $\Gamma_n(A,A')$ y la deducida por la regla composicional al utilizar la relación $R_n(A,B)$ coinciden. El siguiente teorema pone de manifiesto la relación existente entre las conclusiones obtenidas por medio del algoritmo para Γ_k y la obtenida por la regla de inferencia composicional cuando se utiliza la relación intuicionista fuzzy $R_k(A,B)$.

Teorema 8. $\mathbf{B}'_k \leq \max\left(R_n^Y, B'_k\right)$

4 Conclusiones

Hemos dicho en la introducción que los conjuntos intuicionistas fuzzy son isomorfos a los conjuntos intervalo-valorados fuzzy en el caso en que éstos se estudien exclusivamente por medio de los extremos de los intervalos. El algoritmo que presentamos aún siendo diferente al desarrollado por M. B. Gorzalczany en [4] para conjuntos intervalo-valorados fuzzy, da resultados similares. Como dice este autor, la mayor ventaja del algoritmo frente a la regla de composición es que para secuencias del tipo: Si A_1 entonces B_1 o Si A_2 entonces B_2 o $\cdots$ o Si A_m entonces B_m, la complejidad computacional es menor que cuando se utiliza la regla composicional.

Es necesario tener en cuenta que el algoritmo presentado depende directamente del grado de compatibilidad elegido en el proceso, por este motivo se hace necesario realizar en sucesivos trabajos un análisis conciso de dicho grado y su influencia en el algoritmo presentado. También se impone estudiar la regla de inferencia composicional cuando se utiliza cualquier relación inticionista, así como la composición para cualquier t-norma y t-conorma [3].

Referencias

[1] K. Atanassov , Intuitionistic fuzzy sets, *Fuzzy sets and Systems* 20 (1986) 87-96.

[2] K. Atanassov and G. Gargov, Interval valued intuitionistic fuzzy sets, *Fuzzy Sets and Systems*, 31 (1989) 343-349.

[3] II. Bustince, Conjuntos Intuicionistas e Intervalo valorados Difusos: Propieda- des y Construcción. Relaciones Intuicionistas Fuzzy. *Thesis, Universidad Pública de Navarra*, Spain, 1994.

[4] M. B. Gorzalczany. "A method of inference in approximate reasoning based on interval-valued fuzzy sets". *Fuzzy Sets and Systems*, 21 (1987) 1-17.

[5] M. Mizumoto and II.-J. Zimmermann, Comparison of fuzzy reasoning methods, *Fuzzy Sets and Systems* 8 (1982) 253-238.

[6] D. Ruan and E. E. Kerre, Fuzzy implication operators and generalized fuzzy method of cases. *Fuzzy Sets and systems* 54 (1993) 23-37.

[7] R. Sambuc. Functions Φ-Floues. Application à l'aide au Diagnostic en Pathologie Thyroidienne. *Thèse de Doctorat en Médecine*, Marseille, 1975.

[8] I. B. Turksen, Approximate reasoning for production planning. *Fuzzy Sets and Systems*, 26 (1988) 23-27.

[9] Zadeh L.A. Calculus of fuzzy restrictions. In L. A. Zadeh, K. S. Fu, K. Tanaka and M. Shimura (eds):*Fuzzy Sets and Their Applications to Cognitive and Decision Processes*. New York: Academic Press, pp. 1-39. 1974.

[10] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences* 8 (1975) 301-357.

[11] L. A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. Systems Man Cybernet.* 1 (1973) 28-44.

Un Análisis Modal de la Lógica de la Similitud

Francesc Esteva, Pere Garcia y Lluís Godo

Institut d'Investigació en Intel.ligencia Artificial (IIIA)
Consejo Superior de Investigaciones Científicas (CSIC)
e-mails: {esteva,pere,godo}@iiia.csic.es

Resumen

La Lógica de la Similitud es un formalismo lógico para el razxonamiento aproximado que se basa en el uso de relaciones de similitud borrosas sobre el conjunto de mundos posibles y de varias medidas sobre proposiciones definibles a partir de dichas realciones de similitud. Algunos aspectos de dicho formalismo, como las relaciones de deducción graduadas o su semántica basada en sistemas de esferas han sido ya tratados en trabajos previos. En este trabajo proponemos un marco modal donde relacionamos cada relación de similitud con un sistema centrado de esferas que, a la vez, induce una familia de relaciones de accesibilidad y, por tanto, también de operadores modales. Por último, mostramos cómo la deducción posibilística puede ser capturada en el marco modal que proponemos como una deducción entre algunas fórmulas modales particulares.

Palabras Clave: Relaciones de Similitud, Sistemas de Esferas de Lewis, Lógica modal, Lógica Posibilística.

1. Introducción

Para dar semántica a una lógica basada en la Similitud, quizás el marco más general esté definido por los sistemas de Lewis [5]. En su libro, al definir y estudiar los contrafactuales, Lewis plantea tres tipos de sistemas que él denomina Sistemas de Esferas, Sistemas de Similitud comparativa y Sistemas de Posibilidad comparativa. Los sistemas de esferas y de similitud compartiva están definidos en el conjunto W de mundos posibles o interpretaciones del cálculo proposicional clásico, mientras que los sistemas de posibilidades comparativa están definidos sobre el conjunto de proposiciones. Lewis demuestra que, caso de cumplirse unas condiciones muy razonables, los tres tipos de sistemas son equivalentes. En los tres tipos de sistemas la idea básica es la misma: para cada interpretación o mundo posible w existe una forma de ordenar el conjunto de mundos por el hecho de ser más o menos "similares" o "cercanos" a w, o bien por ser más o menos "preferidos" tomando a w como origen o referencia. Veamos de forma resumida cuál es el fundamento de cada uno de estos sistemas:

Un **sistema de esferas** es aquel que asocia a cada mundo w una familia de subconjuntos anidados de W que reciben el nombre de esferas. Cada esfera de la familia asociada a un mundo w (w-esfera) nos dice que los mundos que pertenecen a ella son más similares o cercanos a w que (o son preferidos a) los que no pertenecen a dicha esfera.

Un **sistema de Similitud comparativa** está definido cuando tenemos asociado a cada mundo w un preorden total $\leq_w$ en el conjunto W de forma que $w_1 \leq_w w_2$ significa que w_1 es "más similar o cercano" a w que w_2, o que desde w, w_1 es "preferido" a w_2.

Un **sistema de Posibilidad comparativa** es aquel que asocia a cada mundo w un preorden total $\ll_w$ sobre el conjunto de proposiciones o subconjuntos de W de forma que $p \ll_w q$ representa que una proposición p es "más posible" que q respecto al mundo w.

No es difícil imaginar la equivalencia entre estos tipos de sistemas ya que la familia de esferas asociadas a un mundo w definen un preorden en W:

$w' \leq_w w''$ si toda w-esfera que contiene a w' también contiene a w'',

y un preorden entre proposiciones:

$p \ll_w q$ si en toda w-esfera se cumple: si existen mundos que satisfacen p, también los hay que satisfacen q ,

que en lenguaje de lógica modal, diría

si p es "posible" en una w-esfera, q también lo es .

El recíproco también es cierto en casi todos los casos como es fácil deducir. Tampoco es difícil imaginar que el marco lógico idóneo para la interpretación semántica a partir de los sistemas de Lewis sea el marco de las lógicas multi-modales en las que la relación multivaluada

o fuzzy de accesibilidad a partir de cada mundo w viene dada por la familia anidada de w-esferas.

En este marco se puede interpretar, por ejemplo, la lógica posibilística cualitativa (comparativa) de Dubois y Prade [1]. Así en [4] se demuestra que la posibilidad cualitativa se corresponde con un sistema absoluto de Similitud comparativa de Lewis, en clara concordancia con el hecho de que la teoría de la posibilidad cualitativa se define como un sistema absoluto de preferencias. Ello ha permitido dar una axiomática para la posibilidad comparativa a partir del correspondinte sistema VA de Lewis.

También en el caso de medidas cuantitativas, el marco de la semántica de esferas de Lewis es utilizable. Tal es el caso de las Similitudes numéricas que dan lugar a la lógica de la Similitud (ver [7] y [4]). En este trabajo veremos que la definición de un sistema de esferas que represente adecuadamente la cercanía entre mundos —definida a partir de una T-Similitud— nos permitirá un análisis multi-modal de dichos sistemas al considerar las relaciones de accesibilidad entre mundos inducidas por dichas esferas. Cabe citar que la lógica posibilística también puede interpretarse en este marco como un caso particular en el que los preórdenes asociados a cada mundo en el sistema de similitud o posibilidad compartiva todos coinciden (ver [4] para un primer análisis multi-modal de la lógica posibilística).

En este trabajo, después de esta introducción, presentamos en la sección 2 los conceptos básicos que serán necesarios en el resto del trabajo. En la sección 3 se da una intepretación multi-modal de la lógica de la Similitud. Por último, en la sección 4 proponemos una interpretación de la deducción posibilística en el marco de los sistemas multi-modales descritos en la sección 3. Finalizamos con unas breves conclusiones.

2. Preliminares

Distintos aspectos de la llamada lógica de la Similitud han sido desarrollados en [3] y [2], a partir de los trabajos de Ruspini para dar semántica a la lógica *fuzzy* [7]. Sea L el lenguaje proposicional de la lógica clásica y W el conjunto de mundos posibles o interpretaciones de la misma. La base de la lógica de la similitud es la existencia, para alguna t-norma T, de una T-Similitud S (ver [6]) definida sobre el conjunto W, es decir, S es una relación borrosa que cumple las propiedades reflexiva, i.e. S(w, w) = 1, simétrica, i.e. S(w, w') = S(w', w), y T-transitiva, i.e. $T(S(w_1,w_2), S(w_2,w_3)) \leq S(w_1,w_3)$.

Por otra parte, Ruspini considera que la información sobre el sistema viene modelizada por un conjunto evidencial E de mundos o interpretaciones posibles ($E \subseteq W$). A partir de la T-Similitud S y del conjunto evidencial E se definen las medidas de implicación y consistencia sobre proposiciones como

$$I_{S,E}(p) = \mathrm{Inf}_{w\in E}\, \mathrm{Sup}_{w'\models p}\, S(w,w'),$$
$$C_{S,E}(p) = \mathrm{Sup}_{w\in E}\, \mathrm{Sup}_{w'\models p}\, S(w,w')$$

y las condicionadas de la forma siguiente

$$I_{S,E}(q \mid p) = \mathrm{Inf}_{w\in E}\, \{R_T(I_{S,w}(p), I_{S,w}(q))\},$$
$$C_{S,E}(q \mid p) = \mathrm{Sup}_{w\in E}\, \{R_T(I_{S,w}(p), I_{S,w}(q))\}.$$

donde R_T es la función de implicación definida por residuación a partir de la t-norma T.

Dada una proposición $p \in L$, es fácil comprobar que la medida de consistencia no condicionada es una medida de posibilidad en el sentido de Dubois&Prade, mientras que la medida de implicación no condicionada no es, en general, una medidad de necesidad. Por otra parte, Ruspini a partir de dichas medidas condicionadas propone una regla de inferencia que denomina Modus Ponens generalizado, que permitiría un cálculo deductivo con estas medidas.

En [3] se relaciona esta teoría con los conjuntos difusos puesto que, si S nos da una representación del grado de *semejanza* o *cercanía* entre mundos, entonces toda proposición clásica (subconjunto de W) tiene asociado un fuzzy set sobre W en el que el grado de pertenecia de un mundo representa el grado de semejanza entre este mundo y aquellos en que la proposición se satisface (prototipo de p). Es decir, a cada proposición p le asociamos el fuzzy set p* cuya función de pertenencia μ_{p^*}: $W \rightarrow [0,1]$ viene dada por ,

$$\mu_{p^*}(w') = \mathrm{Sup}_{w\models p}\, S(w,w')$$

Obtenemos así una familia de fuzzy sets asociados al conjunto de proposiciones clásicas. Desgraciadamente este conjunto de fuzzy sets no cubre todo el conjunto de fuzzy sets definibles sobre el universo W, y además, su cálculo no es funcional. De hecho es funcional tan sólo respecto a la unión, i.e.

$$\mu_{(p\vee q)^*}(w) = \mathrm{máx}(\mu_{p^*}(w), \mu_{q^*}(w))$$

pero no lo es ni respecto a la intersección ni respecto a la negación. En general sólo se tienen las siguientes desigualdades:

$$\mu_{(p\wedge q)^*}(w) \leq \mathrm{mín}(\mu_{p^*}(w), \mu_{q^*}(w))$$
$$\mu_{(\neg p)^*}(w) \geq 1 - \mu_{p^*}(w)$$

Pero lo que a nosotros nos interesa en este trabajo es definir el sistema de esferas asociado a una T-Similitud S y la interpretación semántica de las medidas de implicación en este contexto. Parece lógico en este caso definir el sistema de las w-esferas como los α-cortes cerrados (Closed-sphere CS_α) y abiertos (Open-Sphere OS_α) del fuzzy set w*, denotando la "aproximación de w", y que, de acuerdo con la definición anterior, su función de pertenencia viene dada por $\mu_{w*}(w') = S(w,w')$. Es decir, las w-esferas de nivel α son:

$$CS_\alpha(w) = \{w' \mid S(w,w') \geq \alpha\},$$
$$OS_\alpha(w) = \{w' \mid S(w,w') > \alpha\}$$

De forma similar, podemos considerar las α-esferas cerradas y abiertas para cualquier proposición p, que denotaremos de la manera siguiente:

$$CS_\alpha(p) = \cup_{w \models p} CS_\alpha(w),$$
$$OS_\alpha(p) = \cup_{w \models p} OS_\alpha(w)$$

Nótese que $OS_\alpha(p)$ coincide con el α-corte abierto del fuzy set p*, i.e. $OS_\alpha(p) = \{w' \mid \mu_{p*}(w') > \alpha\}$, mientras que para el caso de las esferas cerradas en general sólo tenemos la inclusión $CS_\alpha(p) \subseteq \{w' \mid \mu_{p*}(w') \geq \alpha\}$, valiendo la igualdad en el caso finito.

3. Análisis multi-modal de la lógica de la Similitud

Aumentamos ahora el lenguaje proposicional L con los operadores modales $\Box$ y $\Diamond$. Sea L* tal extensión. Usaremos en el resto del trabajo los símbolos p, q, ... para referirnos a proposiciones de L, y símbolos φ, ψ, ... para referirnos a proposiciones en general (modales o no) de L*. Recordemos que en el caso de la lógica modal clásica un modelo de Kripke M es una estructura (W, R, $\models$), donde W es el conjunto de mundos, R es una relación de accesibilidad entre mundos, y $\models$ es la relación de satisfacción, que para los operadores de **necesidad** y **posibilidad** viene dada por:

$(M, w) \models \Box\varphi$ si_{DEF} para cada $w' \in W$ tal que $R(w,w')$, entonces $w' \models \varphi$

$(M, w) \models \Diamond\varphi$ si_{DEF} existe $w' \in W$ tal que $R(w,w')$ y $w' \models \varphi$

Es decir, si $R(w) = \{w' \in W \mid R(w,w')\}$ y $[\varphi] = \{w \in W \mid w \models \varphi\}$, entonces, podemos expresar las relaciones anteriores como:

$$(M, w) \models \Box\varphi \text{ ssi } R(w) \subseteq [\varphi], \text{ y}$$
$$(M, w) \models \Diamond\varphi \text{ ssi } R(w) \cap [\varphi] \neq \emptyset.$$

Dentro de un modelo de Kripke M = (W, R, $\models$), nos es de interés considerar los siguientes casos particulares de deducción semántica entre proposiciones:

$\psi \models \Box\varphi$ si_{DEF} $(\forall w \in W)$(si $w \models \psi$ entonces $w \models \Box\varphi$), o
si_{DEF} $R(w) \subseteq [\varphi]$, $\forall w \in W$ tal que $w \models \psi$)

o equivalentemente,

$$\psi \models \Box\varphi \text{ ssi } R(\psi) \subseteq [\varphi].$$

Para el caso de la posibilidad, de forma análoga, tenemos,

$\psi \models \Diamond\varphi$ si_{DEF} $(\forall w)(w \models \psi,\ w \models \Diamond\varphi)$, o bien
si_{DEF} $(\forall w)(w \models \psi,\ R(w) \cap [\varphi] \neq \emptyset)$,

que no es equivalente a $R(\psi) \cap [\varphi] \neq \emptyset$.

En el caso en que R(w) venga definida por las w-esferas de nivel α asociadas a una relación de Similitud S, lo que obtenemos en realidad es un sistema multi-modal de forma que, para cada α, tenemos dos pares de operadores modales duales, correspondientes a las esferas cerradas y abiertas de nivel α. Más concretamente, dada una relación de Similitud S, definimos un modelo multi-modal de Kripke como una estructura $\Sigma_S = (W, R_S, \models)$ donde:

- W es el conjunto de interpretaciones o mundos posibles,
- $R_S = \{R_{c\alpha}, R_{o\alpha}\}_{\alpha \in [0,\ 1]}$ es una familia de relaciones de accesibilidad, donde $R_{c\alpha}(w)$ es la w-esfera cerrada de nivel α, es decir $R_{c\alpha}(w) = CS_\alpha(w)$, y $R_{o\alpha}(w)$ es la w-esfera abierta de nivel α, es decir, $R_{o\alpha}(w) = OS_\alpha(w)$.

Denotaremos por Σ la clase de los modelos multi-modales de Kripke definidos a partir del conjunto W de mundos y de las T-Similitudes S sobre W, i.e.

$$\Sigma = \{\Sigma_S = (W, R_S, \models) \mid S \text{ es una T-Similitud sobre W}\}.$$

Consideremos ahora L^* como la extensión del lenguaje L con la familia de operadores modales $\{\Box_{c\alpha}, \Box_{o\alpha}\}_{\alpha \in [0,\ 1]}$ y sus duales $\{\Diamond_{c\alpha}, \Diamond_{o\alpha}\}_{\alpha \in [0,\ 1]}$. Dado un modelo $(W, R_S, \models)$, los operadores modales $\Box_{c\alpha}$ y $\Box_{o\alpha}$, asociados a $R_{c\alpha}$ y $R_{o\alpha}$ respectivamente, tienen la siguiente semántica:

$$(S, w) \models \Box_{c\alpha}\varphi \text{ } si_{DEF} \text{ } CS_\alpha(w) \subseteq [\varphi]$$
$$(S, w) \models \Box_{o\alpha}\varphi \text{ } si_{DEF} \text{ } OS_\alpha(w) \subseteq [\varphi]$$

De forma usual, diremos que una formula φ es válida en un modelo Σ_S, y escribiremos $\Sigma_S \models \varphi$, cuando φ es satisfecha en todo mundo w de Σ_S, y diremos que φ es válida en la clase de modelos Σ, y escribiremos $\Sigma \models \varphi$, cuando es válida en cada modelo Σ_S de la clase.

En cuanto a la noción de consecuencia lógica, nos interesa considerla a tres niveles. A continuación Φ representará un conjunto arbitrario de fórmulas y φ una fórmula.

1) en un modelo Σ_S:
$\Phi \models_S \varphi$ ssi $(\forall w \in W)[(S, w) \models \psi$ para toda $\psi \in \Phi$ implica $(S, w) \models \varphi]$

2) en la clase de modelos Σ, en sentido local:
$\Phi \models_{l\Sigma} \varphi$ ssi $(\forall \Sigma_S \in \Sigma)(\Phi \models_S \varphi)$

3) en la clase de modelos Σ, en sentido global:
$\Phi \models_{g\Sigma} \varphi$ ssi $(\forall \Sigma_S \in \Sigma)(\Sigma_S \models \Phi$ implica $\Sigma_S \models \varphi)$

Obviamente, la noción de consecuencia local es más fuerte que la global. Como veremos más adelante, nos interesará principalmente la noción de consecuencia local.

A continuación vamos a estudiar la relación de los operadores modales anteriormente introducidos con las medidas de implicación y consistencia de la lógica de la Similitud presentadas en la Introducción. Empezaremos por estudiar los operadores de necesidad.

3.1 Operadores modales de necesidad.

Para un modelo particular $\Sigma_S = (W, R_S, \models)$, es interesante considerar los siguientes casos de consecuencia lógica entre conjuntos de fórmulas no modales y formulas modales. En particular, tenemos:

$$\psi \models \Box_{c\alpha}\varphi \text{ ssi } \forall\, w \models \psi,\ CS_\alpha(w) \subseteq [\varphi], \text{ es decir,}$$
$$\psi \models \Box_{c\alpha}\varphi \text{ ssi } CS_\alpha(\psi) \subseteq [\varphi]$$

$$\psi \models \Box_{o\alpha}\varphi \text{ ssi } \forall\, w \models \psi,\ OS_\alpha(w) \subseteq [\varphi], \text{ es decir,}$$
$$\psi \models \Box_{o\alpha}\varphi \text{ ssi } OS_\alpha(\psi) \subseteq [\varphi]$$

donde los conjuntos $CS_\alpha(\psi)$ y $OS_\alpha(\psi)$ son la generalización usual de las definiciones dadas para el caso de fórmulas modales y donde hemos suprimido el símbolo S cómo subíndice en el símbolo de deducción $\models$. Adoptaremos esta simplificación siempre que no induzca a confusión.

La siguiente proposición relaciona los operadores de necesidad $\Box_{c\alpha}$ y $\Box_{o\alpha}$ con medidas de necesidad de la Lógica Posibilística. Recordemos que, dada una distribución de posibilidad π, la medida de necesidad inducida por π se define por

$$Nec_\pi(p) = Inf\{\ 1 - \pi(w) \mid w \models \neg p\ \}.$$

A notar que usaremos la expresión "caso finito" para referirnos al caso en el que S(WxW) lo sea, es decir, los casos donde sup e inf se convierten en máx y mín.

Proposición 1. Si p y q denotan proposiciones no modales, entonces se cumplen las siguientes relaciones:
(i) $p \models \Box_{o\alpha}\, q$ ssi $Nec_{\pi_{S,p}}(q) \geq 1-\alpha$.
(ii) si $Nec_{\pi_{S,p}}(q) > 1-\alpha$ entonces $p \models \Box_{c\alpha}\, q$
Además para el caso finito se cumple también:
(iii) si $p \models \Box_{c\alpha}\, q$ entonces $Nec_{\pi_{S,p}}(q) > 1-\alpha$.

En esta proposición, $\pi_{S,p}$ denota la distribución de posibilidad asociada a la proposición p a través de S, es decir, $\pi_{S,p}(w) = \mu_p{}^*(w) = \sup\{\ S(w, w') \mid w' \models p\ \}$.

3.2 Operadores modales de posibilidad.

De forma análoga al caso anterior, en una estructura de Kripke Σ_S, definimos los operadores modales $\Diamond_{c\alpha}$ y $\Diamond_{o\alpha}$:

$$w \models \Diamond_{c\alpha}\varphi \text{ si}_{DEF}\ CS_\alpha(w) \cap [\varphi] \neq \emptyset$$
$$w \models \Diamond_{o\alpha}\varphi \text{ si}_{DEF}\ OS_\alpha(w) \cap [\varphi] \neq \emptyset$$

y por tanto, para la estructura Σ_S, tendremos la siguiente noción de consecuencia

$$\psi \models \Diamond_{c\alpha}\, \varphi \text{ ssi } \forall\, w \models \psi,\ CS_\alpha(w) \cap [\varphi] \neq \emptyset$$
$$\psi \models \Diamond_{o\alpha}\, \varphi \text{ ssi } \forall\, w \models \psi,\ OS_\alpha(w) \cap [\varphi] \neq \emptyset$$

Obsevemos que, al ser S simétrica, $CS_\alpha(w) \cap [\varphi] \neq \emptyset$ ssi $w \in CS_\alpha(\varphi)$ (y de forma análoga para las esferas abiertas), y por tanto, la extensión de la deducción a proposiciones se puede expresar de la forma siguiente.

Proposición 2. Para toda proposición φ y ψ, se cumplen las siguientes equivalencias:
(i) $\psi \models \Diamond_{c\alpha}\varphi$ ssi $[\psi] \subseteq CS_\alpha(\varphi)$
(ii) $\psi \models \Diamond_{o\alpha}\varphi$ ssi $[\psi] \subseteq OS_\alpha(\varphi)$

Corolario. Si p y q denotan proposiciones no modales, entonces se cumple:
(i) si $p \models \Diamond_{c\alpha}q$ entonces $I_{S,p}(q) \geq \alpha$.
(ii) si $I_{S,p}(q) > \alpha$ entonces $p \models \Diamond_{o\alpha}q$.
Además para el caso finito también se tiene la equivalencia en (i) y en (ii), es decir:
(iii) $p \models \Diamond_{c\alpha}q$ ssi $I_{S,p}(q) \geq \alpha$
(iv) $p \models \Diamond_{o\alpha}q$ ssi $I_{S,p}(q) > \alpha$.

El tipo de consecuencias lógicas (iii) corresponden a las que se estudiaron en [2] aunque allí estaba definida para el caso finito y como una relación de consecuencia graduada entre formulas (i.e. $p \models^{\alpha} q$ ssi $I_{S,p}(q) \geq \alpha$).

A partir de la Proposición 1 y del corolario de la Proposición 2, es obvio que la medida de implicación $I_{S,p}$ y la medida de necesidad $Nec_{\pi_{Sp}}$ son duales en el sentido de que caracterizan a los operadores modales duales de posibilidad y necesidad respectivamente.

Este resultado pudiera parecer que está en contradicción con el hecho de que las medidas duales de las medidas de necesidad son las de posibilidad y no las de implicación. Sin embargo, nótese que que cuando la proposición p corresponde a exactamente un sólo mundo w, entonces la medida de implicación $I_{S,w}$ es también una medidad de posibilidad. Es decir, localmente hay dualidad como medidas, pero ésta se pierde en el caso general.

4 Interpretación multi-modal de la deducción posibilística en el marco de la Lógica de la Similitud

El fragmento de la Lógica Posibilística que se refiere a formulas con cotas inferiores para su medida de necesidad tiene las siguientes características principales:

• las **fórmulas** son pares de la forma (φ, α), donde $\varphi \in L$ y $\alpha \in [0, 1]$ representa una cota inferior del grado de necesidad de φ.

• los **modelos posibilísticos** son del tipo $\pi = (W, \pi, \models_{LP})$, donde π: W ----> [0, 1] es una distribución de posibilidad sobre el conjunto de interpretaciones de L. La clase de los modelos posibilísticos la notaremos por Π.

• la **satisfactibilidad** en un modelo π viene definida por

$$\pi \models_{LP} (\varphi, \alpha) \text{ ssi } Nec_{\pi}(\varphi) \geq \alpha$$

Si $\Gamma = \{(\varphi_i, \alpha_i) \mid i \in I\}$ entonces:

$$\pi \models_{LP} \Gamma \text{ ssi } \pi \models_{LP} (\varphi_i, \alpha_i), \text{ para todo } i \in I$$

• **consecuencia lógica** en la clase Π:
$\Gamma \models_{LP} (\varphi, \alpha)$ ssi para todo $\pi \in \Pi$, si $\pi \models_{LP} \Gamma$ entonces $\pi \models_{LP} (\varphi, \alpha)$

El principal resultado de este apartado es mostrar que la consecuencia lógica en Π se puede interpretar en términos de consecuencia lógica (en sentido local) en la clase Σ de los sistemas multi-modales basados en Similitudes estudiados en el apartado 2. Para ello, necesitamos unos resultados previos.

En primer lugar debemos recordar que en [3] se vió que:

1) A toda distribución de posibilidades π se le puede asociar una Similitud S_{π} y un conjunto evidencial E_{π} definidos por,

$$S_{\pi}(w,w') = Mín(R_T(\pi(w),\pi(w')),R_T(\pi(w'),\pi(w))),$$
$$E_{\pi} = Core(\pi).$$

2) A todo par (S, E) se le puede asociar la distribución de posibilidades $\pi_{S,E}$ dedinida por

$$\pi_{S,E}(w) = Sup_{w' \in E} S(w, w'),$$

cumpliéndose la relación $\pi_{S_{\pi},E_{\pi}} = \pi$ y además, para todo $w \in Core(\pi)$, $\pi_{S_{\pi},w} = \pi$.

En segundo lugar, los siguientes resultados son fácilmente demostrables.

Lema 1. Sea π una distribución de posibilidades. Entonces las dos afirmaciones siguientes son equivalentes:

(i) $\pi_{S,w} \models_{LP} (\psi, \beta)$ implica $\pi_{S,w} \models_{LP} (\varphi, \alpha)$, para todo w y para toda S

(ii) $\pi \models_{LP} (\psi, \beta)$ implica $\pi \models_{LP} (\varphi, \alpha)$, para todo π

Lema 2. Sea una estructura de Kripke $S = (W, R_S, \models)$ y $\varphi \in L$. Entonces se cumple:

$$(S, w) \models \Box_{o\alpha}\varphi \text{ ssi } \pi_{S,w} \models_{LP} (\varphi, 1-\alpha)$$

Como consecuencia de todos estos resulatdos, podemos enunciar el siguiente teorema.

Teorema. La deducción posibilística de formulas valoradas con necesidad se puede interpretar como deducción de formulas modales en el marco de los sistemas multimodales basados en Similitud. En concreto, sea $\Gamma = \{(\varphi_i, 1-\alpha_i) \mid i \in I\}$ un conjunto de formulas posibilísticas y sea $\Gamma^* = \{\Box_{o\alpha i}\varphi_i \mid (\varphi_i, 1-\alpha_i) \in \Gamma\}$ el correspondiente conjunto de formulas modales. Entonces se cumple:

$$\Gamma \models_{LP} (\varphi, 1-\alpha) \text{ ssi } \Gamma^* \models_{I\Sigma} \Box_{o\alpha}\varphi$$

Demostración. Se cumple que:

$\Gamma^* \models_{I\Sigma} \Box_{o\alpha}\varphi$
- ssi $\Gamma^* \models_S \Box_{o\alpha}\varphi$, para todo $S \in \Sigma$
- ssi $(S, w) \models \Gamma^*$ implica $(S, w) \models \Box_{o\alpha}\varphi$, para todo w y toda S
- ssi (por el lema 2) $\pi_{S,w} \models_{LP} (\varphi_i, 1-\alpha_i)$ implica $\pi_{S,w} \models (\varphi, 1-\alpha)$, para todo w y toda S
- ssi (por el lema 1) $\pi \models_{LP} (\varphi_i, 1-\alpha_i)$ implica $\pi \models (\varphi, 1-\alpha)$, para toda $\pi \in \Pi$
- ssi $\Gamma \models_{LP} (\varphi, 1-\alpha)$, c.q.d.

5. Conclusiones

En este trabajo se ha presentado un primer análisis multimodal de la lógica de la Similitud. El hecho de que una Similitud defina, para cada mundo, un sistema de esferas correspondiente a los diferentes niveles de similitud entre aquel mundo y los restantes, parece indicar que el marco adecuado para su estudio semántico sea un marco multi-modal. Pensamos que el presente trabajo reafirma esta creencia y demuestra que este análisis es de sumo interés. Además este análisis permite subsumir el análisis multi-modal realizado para la lógica posibilística y se demuestra que la deducción en la lógica de la necesidad no es más que un caso particular de la deducción en el marco multi-modal que se presenta en este artículo. Creemos, por último, que este trabajo permite pensar en la lógica de la Similitud como una "lógica posibilística local" en el sentido de que, en cada modelo de Kripke, tenemos de hecho asociada una distribución de posibilidad distinta para cada mundo posible. La lógica posibilística sería así el caso particular en el que todas estas distribuciomes de posibilidad locales, cada una de ellas asociada a cada mundo posible, fueran exactamente la misma. El desarrollo de este marco desde el punto de vista sintáctico, proponiendo una axiomatización completa, y el desarrollo de una lógica de la Similitud comparativa son temas interesantes que pensamos continuar investigando en el futuro inmediato.

Agradecimientos

Los autores agradecen las facilidades que encontraron en el Fuzzy Demonstrations Centrum Dortmund, en particular a su director el Dr. Rudolf Felix, donde se realizó en parte este trabajo en el marco de la Acción Integrada hispano-alemana HA93-095. Por otra parte este trabajo se inscribe también en el marco del proyecto nº PB91-0334 de la DGICYT.

Referencias

[1] Dubois D., Lang J. and Prade H. (1994) Possibilistic logic. In : *Handbook of Logic in Artificial Intelligence and Logic Programing* (Gabbay D. et al., eds.), Oxford Univ. Press, 439-513.

[2] Dubois D., Esteva F., Garcia P., Godo L. and Prade H. (1995). Similarity-based Consequence Relations. *IIIA Research Report 95/1*, Barcelona, Spain. Una versión reducida aparecerá en un pròximo *Lecture Notes in Computer Sciences.*

[3] Esteva F, Garcia P, Godo L. (1994a) Relating and extending semantical approaches to possibilistic reasoning. *Int. Journal of Approximate Reasoning* 10, 311-344.

[4] Fariñas del Cerro L., Herzig A. (1991). A modal analysis of possibility theory. *Lecture Notes in Computer Sciences*, Vol. 535, Springer Verlag, Berlin, 11-18.

[5] Lewis D.(1973) *Counterfactuals*. Basil Blackwell. London.

[6] Trillas E., Valverde Ll. (1984) On Implication and Indistinguishability in the setting of Fuzzy Logic. In *Management Decission Support Systems using Fuzzy Set and Possibility Theory* (R.R.Yager and J. Kacprzyk eds).Verlag TÜV, 198-212.

[7] Ruspini E. (1991). On the semantics of fuzzy logic. *Int. Journal of Approximate Reasoning* , 5, 45-88.

UNA CONTRIBUCIÓN A LA APROXIMACIÓN LINGÜÍSTICA UTILIZANDO PROTOTIPOS Y SUPERVALORACIONES

Santiago Fernández Lanza
Universidad de Santiago de Compostela

Cristina del Campo Campos
Universidad de Santiago de Compostela

Alejandro Sobrino Cerdeiriña
Área de Lógica y Filosofía de la Ciencia
Universidad de Santiago de Compostela
e-mail: lflgalcx@correo.usc.es

Resumen

Se escoge un term-set de nueve etiquetas y se representa de forma usual aunque arbitraria. A continuación se observan todos los casos problemáticos de etiquetas que, como resultado de algún tipo de operación con las originariamente definidas, ocupan un área entre dos o más de ellas. Usando la teoría de prototipos de Zadeh y el concepto de supervaloración (que proviene de las lógicas supervaloracionales) se sugiere una verbalización sencilla para dichos casos.

Palabras clave: *aproximación lingüística, prototipos, supervaloración.*

1. Introducción

Como es sabido, una de las características más señaladas de la lógica borrosa es que opera con valores de verdad lingüísticos. Es comúnmente admitido que 7 ± 2 valores es un rango razonable para representar cualquier problema propio del razonamiento humano. Fijado el universo del problema y el contexto en el que se enuncia el predicado vago, un conjunto de términos o term-set encierra las formas de matizar la atribución de la propiedad vaga a alguno de los individuos del universo.

Usualmente un term-set se construye fijando el término medio con un adjetivo y su antónimo (alto y bajo), y añadiendo modificadores a uno y otro de forma gradualmente extrema (ligeramente alto, ligeramente bajo; bastante alto, bastante bajo; muy alto, muy bajo; extraordinariamente alto, extraordinariamente bajo) de manera que cada par tienda a ser un adjetivo modificado y su antónimo. Este term-set, que hace alusión a la variable 'altura' y que se constituye por especificaciones de la misma, será utilizado en este trabajo como ejemplo.

Así como en lógica bivaluada 'verdadero' y 'falso' tienen como correlatos numéricos a 1 y 0, en lógica borrosa se le hace corresponder a cada etiqueta lingüística un número borroso. Un número borroso tiene asociada una función de pertenencia continua a trozos, cuyo caso más habitual es un trapecio. Esta función marca dónde comienza la pertenencia al concepto, dónde se estabiliza, dónde decrece y dónde, por fin, desaparece. Conforme a lo relatado anteriormente, el term-set del ejemplo podría representarse por la siguiente figura:

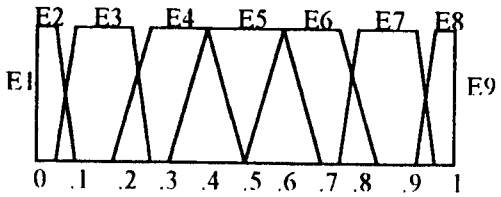


Figura 1

La aproximación lingüística representa todavía un problema abierto dentro de la lógica borrosa. Por ello se le denomina frecuentemente como 'el problema de la aproximación lingüística'. Brevemente presentada, la aproximación lingüística surge, en el contexto de los interfaces persona-máquina, al menos en dos situaciones:

(a) Cuando un usuario del sistema informático que utiliza lógica borrosa da una cualificación de la pertenencia de un objeto al concepto vago y ésta es diferente a las recogidas en el term-set que el ingeniero de conocimiento tiene definido.

(b) Cuando, admitido un valor de entrada por coincidir con alguno de los previamente definidos, una operación con dos o más de esos valores da como resultado un nuevo

número borroso que no sea posible nombrar por ninguna de las etiquetas originales.

En este trabajo sólo nos ocuparemos de la aproximación lingüística en el aspecto aquí señalado como (b).

Una buena eficiencia en la comunicación persona-máquina demandaría una verbalización gramaticalmente aceptable para estos casos problemáticos. La teoría de prototipos de Zadeh [3] constituye, a nuestro juicio, un marco teórico sugerente para abordar un proceso que conduzca a nombrar de manera bastante natural los casos difíciles. A continuación presentamos una breve reseña de esta teoría.

La teoría de prototipos de Zadeh surge como una reacción a la teoría clásica o ingenua de prototipos, donde se definía un prototipo como el ejemplar que reunía todas las notas positivas y ninguna negativa de un concepto. La teoría de Zadeh deja entrever, en cambio, que la comprensión real del significado de un concepto vago no se reduce a establecer una similitud con su estereotipo, o prototipo en sentido clásico. La comprensión habitual del significado de un concepto vago pasa, no sólo por ser capaz de mostrar a su mejor ejemplar, sino tambien de hacer lo propio con los ejemplares mediocres y malos del concepto. Un marciano reconocería mejor cualquier queso terráqueo si en su mapa conceptual figurase el requesón (ejemplar de prototipo malo de queso), el queso fresco gallego (ejemplar de un prototipo mediocre de queso) y el idiazàbal (ejemplar de un prototipo bueno de queso), que no sólo si conociese el queso manchego (prototipo absoluto en el sentido de que éste sea considerado como mejor prototipo de queso que el idiazàbal). Estas reflexiones se apoyan en la igualdad que, según Zadeh, define el prototipo de un concepto vago:

$$PT(A) = High/PT(A_{Good}) + Medium/PT(A_{Borderline}) + Low/PT(A_{Poor})$$

Esta definición justifica una agrupación de los valores del term-set inicial en tres categorías: ejemplares buenos, mediocres y malos. Una de las posibles agrupaciones es la propuesta en [2] tal y como muestra la siguiente figura:

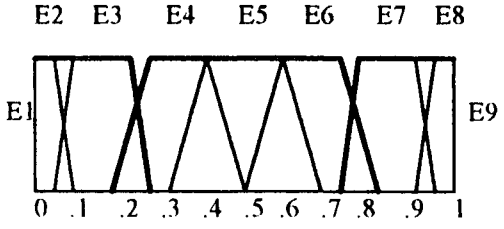


Figura 2

Esto permite una re-evaluación de los valores anómalos para el term-set de 'alto' utilizando recursos lingüísticos derivados de la definición de prototipo, como la noción de 'ejemplar' (en el sentido de 'caso que se pondría como ejemplo'). Se distinguirán así 'buenos ejemplares de alto', para aquellos casos que caigan dentro del *prototipo bueno*, 'ejemplares mediocres de alto' para aquellos casos que caigan dentro del *prototipo mediocre* y, por fin, 'malos ejemplares de alto' para aquellos casos que caigan dentro del *prototipo malo*.

Para los casos que todavía fuesen anómalos, esto es, que no se incluyan en el área demarcada por alguno de los prototipos, sino que solapen a dos o más de ellos, se utilizará una estrategia de verbalización basada en la negación. En el caso de los valores verbalizados de manera adecuada por la técnica basada en los prototipos de Zadeh, se utilizarán valores de valores (supervalores borrosos), como ocurre en *es un buen ejemplar de alto*. Para los valores que no pueden ser verbalizados adecuadamente usando supervalores, creemos que es más ilustrativo definirlos por negación, como ocurre por ejemplo en *no es extremadamente alto*. Con ello se agota sólo una de las secuencias de acciones que caracterizan a las lógicas supervaloracionales. Dar el paso de agruparlos en un supervalor (borroso) sería tanto como negar que la vaguedad permanece todavía. Creemos que es mejor la inespecificidad que acompaña a la negación que ignorar la vaguedad.

2. Objetivo

El objetivo de este trabajo es proponer un algoritmo que facilite una verbalización inteligible para aquellos casos conflictivos dentro del term-set original. Para ello se utilizan técnicas lingüísticas propias de la teoría de prototipos de Zadeh y de las lógicas supervaloracionales (modificadas).

3. Prototipos solapados

Decimos que dos prototipos se solapan cuando, tomados los dos primeros y los dos últimos, cada una de estas parejas posee un espacio común que coincide con una de las etiquetas del term-set inicial, como se observa en la Fig. 3.

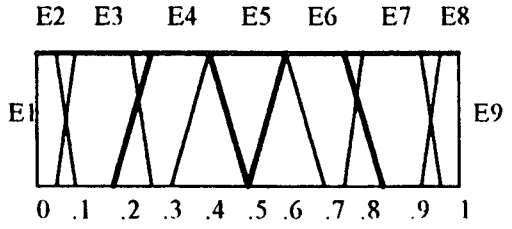


Figura 3

De esta manera todo número borroso que quede entre dos de los prototipos podrá ser verbalizado mediante esa

etiqueta del term-set inicial que queda en el medio, siempre y cuando no supere las dimensiones de ésta.

4. Niveles de valoración o verbalización

Si el número borroso supera las dimensiones de la etiqueta intermedia fruto del solapamiento de los prototipos, el número borroso a verbalizar está entre al menos dos de los prototipos y además posee unas dimensiones superiores a esa etiqueta intermedia. En este grupo hay algunos casos conflictivos. Para verbalizar este tipo de casos proponemos un segundo nivel de prototipos (super-prototipos) sobre el primero utilizando el mismo mecanismo de solapamiento, de tal forma que tengamos solamente dos super-prototipos: el de los números que están entre los malos y los mediocres, que verbalizaremos por negación como "no es un buen ejemplar de alto", y el de los números que están entre los mediocres y los buenos, como "no es un buen ejemplar de bajo", quedando como espacio intermedio fruto del solapamiento, el prototipo de los mediocres definido en el nivel anterior.

Pero todavía se pueden dar casos conflictivos en este nivel, aunque creemos que son muy extraños, por eso proponemos como alternativa otro nivel más de verbalización. Éste supone un número borroso que se encuentra entre los dos prototipos anteriores y cuyas dimensiones superan a las del prototipo de los mediocres, resultado del solapamiento de los dos prototipos presentados en el nivel anterior. En el punto 5 de este trabajo esto se verá con más claridad.

El primer nivel (term-set inicial) posee 9 etiquetas, el segundo 3 prototipos, el tercero 2 super-prototipos y el cuarto agota todas las posibilidades de seguir introduciendo niveles.

4.1. Nivel A

El nivel A viene dado por el term-set inicial de nueve etiquetas distintas E1, E2, E3, E4, E5, E6, E7, E8 y E9, las cuales, en nuestra propuesta, deben cumplir las siguientes condiciones[1]:

$$(1)\ a_{E1} = 0;\ b_{E1} = 0$$

$$(2)\ c_{E9} = 1;\ d_{E9} = 1$$

$$(3)\ a_{E_n} \le a_{E_{n+1}}\ \forall n \text{ tal que } 1 \le n \le 8$$

$$(4)\ b_{E_n} \le b_{E_{n+1}}\ \forall n \text{ tal que } 1 \le n \le 8$$

$$(5)\ c_{E_n} \le c_{E_{n+1}}\ \forall n \text{ tal que } 1 \le n \le 8$$

$$(6)\ d_{E_n} \le d_{E_{n+1}}\ \forall n \text{ tal que } 1 \le n \le 8$$

4.2. Nivel B

Viene dado por tres etiquetas B1, B2 y B3 obtenidas cada una de ellas mediante la unión[2] de alguna de las etiquetas del nivel A, de tal forma que:

$$B1 = E1 \cup E2 \cup E3 \cup E4$$

$$B2 = E4 \cup E5 \cup E6$$

$$B3 = E6 \cup E7 \cup E8 \cup E9$$

Con lo cual tenemos:

$$B1 = (a_{E1}, b_{E1}, c_{E4}, d_{E4}) = (0, 0, c_{E4}, d_{E4})$$
$$B2 = (a_{E4}, b_{E4}, c_{E6}, d_{E6})$$
$$B3 = (a_{E6}, b_{E6}, c_{E9}, d_{E9}) = (a_{E6}, b_{E6}, 1, 1)$$

4.3. Nivel C

Viene dado por dos etiquetas C1 y C2 obtenidas cada una de ellas por la unión de alguna de las etiquetas del nivel B, de tal forma que:

$$C1 = B1 \cup B2$$

$$C2 = B2 \cup B3$$

Con lo cual tenemos:

$$C1 = (a_{E1}, b_{E1}, c_{E6}, d_{E6}) = (0, 0, c_{E6}, d_{E6})$$
$$C2 = (a_{E4}, b_{E4}, c_{E9}, d_{E9}) = (a_{E4}, b_{E4}, 1, 1)$$

4.4. Nivel D

Viene dado por una etiqueta D1 obtenida por la unión de las etiquetas del nivel C, de tal forma que:

$$D1 = C1 \cup C2$$

Con lo cual tenemos:

$$D1 = (a_{E1}, b_{E1}, c_{E9}, d_{E9}) = (0, 0, 1, 1)$$

[1] Las condiciones (3)-(6) permiten que dos etiquetas del term-set sean iguales o prácticamente iguales. Algunos de los *a*, *b*, *c*, *d* de dos etiquetas pueden coincidir, pero es absurdo que coincidan todos, pues en este caso tendríamos dos etiquetas iguales. Por otra parte, no es usual que el lenguaje permita distinguir entre dos etiquetas prácticamente iguales. Los casos normales poseen etiquetas claramente diferentes como sucede en nuestro ejemplo. Véase figura 1.

[2] Sean dos números borrosos X e Y tal que X = (a, b, c, d) e Y = (e, f, g, h), llamaremos *unión de dos números borrosos*, denotado X ∪ Y, a otro número (i, j, k, l) tal que i = min (a, e); j = min (b, f); k = max (c, g) y l = max (d, h). El procedimiento puede generalizarse a una cantidad finita de números borrosos.

5. Algoritmo de valoración o verbalización

Antes de hablar del algoritmo de verbalización resulta necesario dar la siguiente definición:

Sea X un número borroso con la forma (a, b, c, d), tal que $a \le b \le c \le d$, y sea En una etiqueta de la forma (a_{En}, b_{En}, c_{En}, d_{En}) tal que $a_{En} \le b_{En} \le c_{En} \le d_{En}$.

Diremos que X está incluido en la etiqueta En ($X \subseteq En$) si y sólo si:

$$(1)\ a_{En} \le a \le d_{En}$$
$$(2)\ b_{En} \le b \le c_{En}$$
$$(3)\ b_{En} \le c \le c_{En}$$
$$(4)\ a_{En} \le d \le d_{En}$$

El algoritmo funciona desde el nivel A hacia el nivel D, y nunca en sentido contrario. A partir del momento en que se da la solución, el proceso debe pararse, ya que un número borroso que es verbalizable en el nivel A puede ser problemático en los demás niveles. El algoritmo es el siguiente:

- Entrada: Un número borroso X.
- Salida: Una etiqueta lingüística (que a su vez nos indica el nivel de verbalización que posee).
- Procedimiento:

NIVEL A

Si X está incluido en E1 entonces X se verbaliza diciendo "extraordinariamente bajo".

Si X está incluido en E2 entonces X se verbaliza diciendo "muy bajo".

Si X está incluido en E3 entonces X se verbaliza diciendo "bastante bajo".

Si X está incluido en E4 entonces X se verbaliza diciendo "ligeramente bajo".

Si X está incluido en E5 entonces X se verbaliza diciendo "ni alto ni bajo".

Si X está incluido en E6 entonces X se verbaliza diciendo "ligeramente alto".

Si X está incluido en E7 entonces X se verbaliza diciendo "bastante alto".

Si X está incluido en E8 entonces X se verbaliza diciendo "muy alto".

Si X está incluido en E9 entonces X se verbaliza diciendo "extraordinariamente alto".

Si no se da ninguno de los casos anteriores, pasar al nivel B.

NIVEL B

Si X está incluido en B1 entonces X se verbaliza diciendo "Es un mal ejemplar de alto".

Si X está incluido en B2 entonces X se verbaliza diciendo "Es un ejemplar mediocre de alto".

Si X está incluido en B3 entonces X se verbaliza diciendo "Es un buen ejemplar de alto".

Si no se da ninguno de los casos anteriores, pasar al nivel C.

NIVEL C

Si X está incluido en C1 entonces X se verbaliza diciendo "No es un buen ejemplar de alto".

Si X está incluido en C2 entonces X se verbaliza diciendo "No es un buen ejemplar de bajo".

Si no se da ninguno de los casos anteriores, pasar al nivel D

NIVEL D

Verbalícese X por negación dentro del term-set original.

5.1. Verbalización por negación

La verbalización por negación es útil para verbalizar números borrosos que caen dentro de la mayor parte de las etiquetas, de tal forma que la única manera de verbalizarlos es nombrando las etiquetas dentro de las que no cae dicho número. Por ejemplo, supongamos que un número borroso cae dentro de E1, E2, E3, E4, E5, E6, E7, en este caso diríamos que "no es muy alto". En el punto 6 se analiza detalladamente la vebalización de estos casos excepcionales.

6. Comprobación de la eficiencia del algoritmo

Según el conjunto de 9 etiquetas del que partimos tendremos los siguientes tipos de números borrosos:

Tipo 1: $X \subseteq En$[3] donde $1 \le n \le 9$
Tipo 2: $X \subseteq (En \cup En+1)$ donde $1 \le n \le 8$
Tipo 3: $X \subseteq (En \cup En+1 \cup En+2)$ donde $1 \le n \le 7$
Tipo 4: $X \subseteq (En \cup ... \cup En+3)$ donde $1 \le n \le 6$
Tipo 5: $X \subseteq (En \cup ... \cup En+4)$ donde $1 \le n \le 5$
Tipo 6: $X \subseteq (En \cup ... \cup En+5)$ donde $1 \le n \le 4$
Tipo 7: $X \subseteq (En \cup ... \cup En+6)$ donde $1 \le n \le 3$
Tipo 8: $X \subseteq (En \cup ... \cup En+7)$ donde $1 \le n \le 2$
Tipo 9: $X \subseteq (E1 \cup ... \cup E9)$

donde $n \in N$, X es un número borroso cualquiera, y Ea es una etiqueta de las del conjunto de partida donde a es el número que indica cuál de ellas es.

Las tablas siguientes nos indican todas las posibilidades que se pueden dar dentro de cada tipo. A cada una de estas posibilidades o sub-tipos le corresponde una verbalización. Podemos, por tanto, clasificar los tipos y

[3] Un número borroso X que, por ejemplo, está incluido en la unión de dos de las etiquetas también estará incluido en la unión de tres, cuatro o más de las etiquetas; con lo cual, a un numero borroso le corresponderían varios tipos, un tipo mínimo y todos los tipos superiores. Para salvaguardar la univocidad del procedimiento seleccionaremos el tipo más pequeño al que pertenece el número.

sub-tipos con números naturales de dos cifras, ordenados de tal forma que la primera cifra nos indique el tipo de número borroso de que se trata (1-9), y la segunda cifra la primera de las etiquetas que necesitamos unir mediante ($\cup$) para que nos resulte el trapecio más pequeño dentro del cual el número borroso X está incluido. De esta forma tendremos:

Tipo	Verbalización
11	Extraordinariamente bajo (A)
12	Muy bajo (A)
13	Bastante bajo (A)
14	Ligeramente bajo(A)
15	Ni alto ni bajo (A)
16	Ligeramente alto (A)
17	Bastante alto (A)
18	Muy alto (A)
19	Extraordinariamente alto (A)
21	Es un mal ejemplar de alto (B)
22	Es un mal ejemplar de alto (B)
23	Es un mal ejemplar de alto (B)
24	Es un ejemplar mediocre de alto (B)
25	Es un ejemplar mediocre de alto (B)
26	Es un buen ejemplar de alto (B)
27	Es un buen ejemplar de alto (B)
28	Es un buen ejemplar de alto (B)
31	Es un mal ejemplar de alto (B)
32	Es un mal ejemplar de alto (B)
33	No es un buen ejemplar de alto (C)
34	Es un ejemplar mediocre de alto (B)
35	No es un buen ejemplar de bajo (C)
36	Es un buen ejemplar de alto (B)
37	Es un buen ejemplar de alto (B)
41	Es un mal ejemplar de alto (B)
42	No es un buen ejemplar de alto (C)
43	No es un buen ejemplar de alto (C)
44	No es un buen ejemplar de bajo (C)
45	No es un buen ejemplar de bajo (C)
46	Es un buen ejemplar de alto (B)
51	No es un buen ejemplar de alto (C)
52	No es un buen ejemplar de alto (C)
53	No es ni muy alto ni muy bajo (D)
54	No es un buen ejemplar de bajo (C)
55	No es un buen ejemplar de bajo (C)
61	No es un buen ejemplar de alto (C)
62	No es muy alto pero tampoco extraordinariamente bajo (D)
63	No es muy alto pero tampoco extraordinariamente bajo (D)
64	No es un buen ejemplar de bajo (C)
71	No es muy alto (D)
72	No es ni extraordinariamente alto ni extraordinariamente bajo (D)
73	No es muy bajo (D)
81	No es extraordinariamente alto (D)
82	No es extraordinariamente bajo (D)
91	Imposible de verbalizar (D)

7. Conclusiones

Usando la teoría de prototipos de Zadeh y el concepto de supervaloración se ha facilitado una sencilla verbalización de las etiquetas problemáticas dentro del term-set original. Para ello se ha diseñado un algoritmo y se ha comprobado su eficiencia.

Referencias

[1] Aguilar-Martin, J. Grado de recubrimiento de particiones borrosas, *Congreso Español sobre Tecnologías y Lógica Fuzzy*, Santiago de Compostela, pp. 137-144, 1993.

[2] Olivas, J. A., Sobrino, A.,Aproximación lingüística y teoría de prototipos de Zadeh. *Congreso Español sobre Tecnologías y Lógica Fuzzy*, Granada, pp. 179-183, 1991.

[3] Zadeh, L. A. , A note on prototype set theory and fuzzy sets, *Cognition*, 12 (1982), pp. 291-297.

Agradecimientos

Este trabajo ha sido realizado con la ayuda del proyecto PB92-0394 de la D.G.I.C.Y.T.

Ideales T-S-Difusos. Análisis de sus propiedades

Alfonso Garmendia Salvador
Univ. Complutense de Madrid

María Molero Aparicio
Univ. Alcalá Henares Madrid

Adela Salvador Alcaide
Dto. Matemáticas e Informática Aplicadas
E. T. S. de Ingenieros de Caminos
Universidad Politécnica de Madrid

Abstract

This paper has got as a global aim the study of T-S-fuzzy ideals of a ring defined starting from subsets type Zadeh. We define T-S-fuzzy ideals by substituting in the classical definition of fuzzy ideals the minimum by a triangular norm and the maximum by any triangular conorm, adding a condition that imposes that the value of belonging of one element x by a T-S-fuzzy ideal coincides with the of -x.

Keywords: Algebra, Fuzzy ideal, T-S-fuzzy ideal, Noetherian ring, Artinian ring.

1. Introducción

Definimos los ideales T-S-difusos. La primera diferencia encontrada entre ideales T-S-difusos e ideales difusos está en la propia definición.

Otra diferencia importante entre ideales T-S-difusos y difusos es que estos últimos estan caracterizados por el criterio de que sus subconjuntos de nivel no vacíos son ideales, mientras que los ideales T-S-difusos no verifican necesariamente esta condición. Definimos una relación de equivalencia en el conjunto de los ideales T-S-difusos.

2. Preliminares

Definición 2.1: Una **norma triangular** es una aplicación T de $[0,1]x[0,1]$ en $[0,1]$ tal que:

T1) $T\{x,1\}=T\{1,x\}=x$;
T2) Si $x\geq x'$, $y\geq y' \Rightarrow T\{x,y\}\geq T\{x',y'\}$;
T3) $T\{x,y\}=T\{y,x\}$;
T4) $T\{x,T\{y,z\}\}=T\{T\{x,y\},z\}$

Definición 2.2: Una **conorma triangular** es una aplicación S de $[0,1]x[0,1]$ en $[0,1]$ que verifica las propiedades:
S1) $S\{0,x\}=S\{x,0\}=x$ y S2) S3) S4) como las correspondientes de T.

Definición 2.3: Una norma triangular T^* (o una conorma triangular) **predomina** sobre otra norma triangular T (o conorma triangular), y se denota por $T^* >> T$, si:
$T^*\{T\{x,y\},T\{z,w\}\}\geq T\{T^*\{x,z\},T^*\{y,w\}\}$.

3. Ideal T-S-difuso

Definición 3.1: Sea A anillo, I subconjunto difuso de A, $I\neq\phi$, T una norma triangular. Decimos que I es un **ideal a la izquierda T-difuso** de A si y sólo si verifica:

(i) I es un subgrupo T-difuso:
$I(x-y)\geq T\{I(x),I(y)\}$, para todo $x,y\in A$
$I(x)=I(-x)$, para todo $x\in A$

(ii) $I(x.y)\geq I(y)$, para todo $x,y\in A$

Si sustituimos la condición (ii) por: (iii) $I(x.y)\geq I(x)$, para todo $x,y\in A$, I es un **ideal a la derecha T-difuso** de A.

Si la sustituimos por: (iv) $I(x.y)\geq S\{I(x),I(y)\}$, para todo $x,y\in A$, definimos a I como **ideal (bilátero) T-S-difuso**

de A, siendo S una conorma triangular.

Si la conorma triangular S es la conorma max al ideal T-max-difuso lo llamamos **ideal T-difuso**. Si la norma triangular T es la norma min llamamos al ideal min-S-difuso **ideal S-difuso**. Estos ideales particulares no son contemplados en la definición de S. Sessa [10] ya que la norma triangular y la conorma triangular que los definen no son duales. Llamamos al ideal min-max-difuso simplemente **ideal difuso**.

Si además se verifica: (v) $I(0)=1$ decimos que I es un **ideal fuerte (a la izquierda, a la derecha) T-S-difuso** de A.

Naturalmente todo ideal fuerte T-S-difuso de A es un ideal T-S-difuso de A.

4. Ejemplos

Ejemplos 1 y 2: Dos ideales fuertes T-S-difusos particulares de un anillo A son:

(i) El propio anillo A, definido por $A(x)=1$, para todo $x\in A$.

(ii) El ideal trivial, denotado **0**, definido por $\mathbf{0}(x)=0$, para todo $x\neq 0$ y $\mathbf{0}(0)=1$.

Ejemplo 3: En el anillo A de las matrices cuadradas triangulares de orden tres de números reales con la norma $T(\alpha,ß)=\alpha.ß$ y la conorma $S(\alpha,ß)=\alpha+ß-\alpha.ß$ el siguiente subconjunto difuso es un ideal fuerte T-S-difuso de A.

$$I(x)=\begin{cases} 1 & \text{si } x=\begin{pmatrix}0,0,0\\0,0,0\\0,0,0\end{pmatrix} \\ 0,6 & \text{si } x=\begin{pmatrix}0,0,0\\0,0,0\\0,0,0\end{pmatrix}\ a\neq 0 \\ 0,3 & \text{si } x=\begin{pmatrix}0,b,a\\0,0,c\\0,0,0\end{pmatrix}\ b\neq 0 \text{ o } c\neq 0 \\ 0 & \text{si } x\in A-\begin{pmatrix}0,a,b\\0,0,c\\0,0,0\end{pmatrix}\ a,b,c\in R \end{cases}$$

Ejemplo 4: En el anillo $Z/4=\{0,1,2,3\}$ con la norma $T(a,b)=a.b$ y la conorma $S(a.b)=a+b-a.b$ el subconjunto difuso definido por:

$$I(x)=\begin{cases} 1 & \text{si } x=0 \\ 0.5 & \text{si } x=2 \\ 0 & \text{si } x\in\{1,3\} \end{cases}$$

I es un ideal fuerte T-S-difuso de Z/4.

Ejemplo 5: Si A es un anillo y J un ideal de A entonces el subconjunto difuso I definido por:

$$I(x)=\begin{cases} s & \text{si } x\in J \\ t & \text{si } x\in A-J \end{cases}$$

con $s,t\in[0,1]$, $s\geq t$, es un ideal difuso de A y por lo tanto, para toda norma T, I es un ideal T-difuso de A, aunque no necesariamente T-S-difuso.

Como caso particular cuando $s=t$, I es constante y es un ideal difuso y por tanto T-difuso. No podemos asegurar que sea T-S-difuso salvo cuando I coincide con el anillo A o la conorma S es el máximo.

5. Propiedades

Proposición 5.1: Si A es un anillo, T una norma triangular, S una conorma triangular e I un subconjunto difuso de A, tal que $I(0)=1$, $I(x-y)\geq T\{I(x),I(y)\}$ para todo $x,y\in A$ y $I(x.y)\geq S\{I(x),I(y)\}$ para todo $x,y\in A$ entonces I es un ideal fuerte T-S-difuso de A.

Demostración:

Por la definición de ideal fuerte T-S difuso basta demostrar que $I(x)=I(-x)$. Para todo $x\in A$, $I(-x)=I(0-x)\geq T\{I(0),I(x)\}=T\{1,I(x)\}=I(x)=I(0-(-x))\geq T\{I(0),I(-x)\}=T\{1,I(-x)\}=I(-x)$. ■

Proposición 5.2: Si A es un anillo con elemento unidad, T una norma triangular, S una conorma triangular e I un subconjunto difuso de A, tal que $I(x-y)\geq T\{I(x),I(y)\}$ para todo $x,y\in A$ e $I(x.y)\geq S\{I(x),I(y)\}$ para todo $x,y\in A$ entonces I es un ideal T-S-difuso de A.

Demostración:

Por la definición de ideal T-S difuso basta demostrar que $I(x)=I(-x)$. Para todo $x\in A$,

$I(-x)=I((-1).x)\geq S\{I(-1),I(x)\}\geq \max\{I(-1),I(x)\}\geq I(x)= I((-1).(-x))\geq S\{I(-1),I(-x)\}\geq \max\{I(-1),I(-x)\}\geq I(-x)$.■

Así, si el anillo A es unitario o si I(0) es igual a 1, bastaría utilizar como criterio para reconocer si I es un ideal T-S-difuso que para todo $x,y\in A$: $I(x-y)\geq T\{I(x),I(y)\}$, $I(x.y)\geq S\{I(x),I(y)\}$. Aunque, en general, es necesario imponer además que $I(x)=I(-x)$ como indica el siguiente contraejemplo:

Ejemplo 7: En el anillo $A=\{0,2,4,6\}\subset Z/8$ consideremos el subconjunto difuso I definido por:

$$I(x)=\begin{cases}0.7 & \text{si } x\in\{0,2,4\}\\ 0.5 & \text{si } x=6\end{cases}$$

Con la norma $T(a,b)=a.b$ y la conorma $S=\max$, el subconjunto difuso I verifica que $I(x-y)\geq T\{I(x),I(y)\}$ para todo $x,y\in A$ y además $I(x.y)\geq \max\{I(x),I(y)\}$ para todo $x,y\in A$ y sin embargo $I(2)=0.7\neq I(-2)=I(6)=0.5$.

Proposición 5.3: Sea A anillo. Si I es ideal de A entonces, para toda norma triangular T y para toda conorma triangular S, la función característica de I, que notaremos σ_I, es un ideal T-S-difuso de A.

Proposición 5.4: Si A es un anillo e I un ideal T-S-difuso de A entonces

(i) $I(0)=S\{I(0),I(x)\}$ para todo $x\in A$.

(ii) Si $x=y.z$ entonces $I(x)\geq I(z)$.

Demostración:

(i) $I(0)=I(x.0)\geq S\{I(0),I(x)\}\}\geq S\{I(0),0\}=I(0)$

(ii) Si $x=y.z$ entonces $I(x)\geq S\{I(y),I(z)\}\geq \max\{I(y),I(z)\}\geq I(z)$.■

Corolario 5.5: Si A es un anillo entonces para todo I ideal T-S-difuso de A y para todo $x\in A$ se verifica que $I(0)\geq I(x)$.

$I(0)= S\{I(0),I(x)\}\geq \max\{I(0),I(x)\}\geq I(x)$.■

Proposición 5.6: Si A es un anillo e I es un ideal fuerte T-S-difuso de A entonces si $I(x-y)=I(0)$ entonces $I(x)=I(y)$.

Demostración:

$I(x)=I((x-y)+y)\geq T\{I(x-y),I(y)\}= T\{1,I(y)\}= I(y)= I((y-x)+x)\geq T\{I(y-x),I(x)\}= T\{1,I(x)\}= I(x)$.■

Con ideales difusos, como A. Rosenfeld [9] prueba para subgrupos difusos, no es necesario imponer la condición de que el ideal difuso sea fuerte.

Proposición 5.7: Si A es un anillo e I es un ideal difuso de A entonces si $I(x-y)=I(0)$ entonces $I(x)=I(y)$.

Sin embargo no es cierto que si A es un anillo, I un ideal T-S-difuso de A, pero no un ideal fuerte T-S-difuso y $x,y\in A$ tales que $I(x-y)=I(0)$ entonces $I(x)=I(y)$, como comprueba el siguiente contraejemplo:

Ejemplo 8: Sea Z el anillo de los enteros, T la norma producto y S la conorma max. Si I es el ideal T-S-difuso definido por:

$$I(x)=\begin{cases}0.4 & \text{si } x\in(2)-\{2,-2\}\\ 0.2 & \text{si } x\in\{2,-2\}\\ 0.16 & \text{si } x\in Z-(2)\end{cases}$$

$I(6-2)=I(0)$ y sin embargo $I(6)\neq I(2)$.

Proposición 5.8: Si A es un anillo con elemento unidad para todo ideal T-S-difuso I de A y para todo $x\in A$ se verifica que $I(x)\geq I(1)$.

Demostración:

Para todo $x\in A$, $I(x)=I(1.x)$ y como I es un ideal difuso $I(x)= I(1.x)\geq S\{I(1),I(x)\}\geq \max\{I(1),I(x)\}\geq I(1)$.■

Si I es un ideal T-S-difuso y A un anillo con elemento unidad tenemos que, para todo $x\in A$, $I(0)\geq I(x)\geq I(1)$.

Proposición 5.9: Si A es un anillo con elemento unidad e I un ideal T-S-difuso de A entonces $I(1)=S\{I(1),I(1)\}$.

Demostración:

$I(1)=I(1.1)\geq S\{I(1),I(1)\}\geq S\{I(1),0\}=I(1)$.■

Proposición 5.10: Si A es un anillo con elemento unidad y x es una unidad de A entonces en todo I ideal T-S-difuso de A se verifica que $I(x)=I(1)$.

Demostración:

Si x es una unidad de A entonces existe un $y\in A$ tal que $x.y=1$, luego $I(1)= I(x.y)\geq S\{I(x),I(y)\}\geq \max\{I(x),I(y)\}\geq I(x)$, y como $I(x)\geq I(1)$ tenemos que $I(x)=I(1)$.■

Corolario 5.11: Si A es un anillo con elemento unidad e I un ideal T-S-difuso de A tal que $I(1)=1$ entonces $I=A$.

Corolario 5.12: Si A es un cuerpo no

conmutativo e I es un ideal T-S-difuso entonces $I(x)=I(1)\leq I(0)$, para todo $x\in A$, $x\neq 0$

Aunque sea cierto que $I(x)=I(1)\leq I(0)$, para todo $x\in A$, $x\neq 0$, no podemos asegurar que I sea un ideal T-S-difuso, para toda norma triangular T y para toda conorma triangular S.

Ejemplo 9: Si Q es el cuerpo de los racionales, T=min y S la conorma definida por S(a,b)=a+b-a.b, si I es el subconjunto difuso de Z definido por:

$$I(x)=\begin{cases} 0.8 & \text{si } x=0 \\ 0.4 & \text{si } x\neq 0 \end{cases}$$

$I(x.0)=I(0)=0.8$; $S\{I(x),I(0)\}=I(x)+I(0)-I(x).I(0)=0.88$. Luego I no es un ideal T-S-difuso de Q.

Los únicos ideales T-S-difusos con Card(Iml)=2 para la conorma S(a,b)=a+b-a.b son las funciones características de los ideales ordinarios. Sin embargo, cuando el Card(Iml)>2, el ejemplo siguiente nos muestra un ideal T-S-difuso que no es la función característica de un ideal ordinario.

Ejemplo 10: En el anillo A de las matrices cuadradas triangulares de orden 2 de números reales con la norma T(a.b)=a.b y la conorma S(a.b)=a+b-a.b el siguiente subconjunto difuso es un ideal fuerte T-S-difuso de A.

$$I(x)=\begin{cases} 1 & \text{si } x=\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ 0.5 & \text{si } x=\begin{pmatrix} 0 & a \\ 0 & 0 \end{pmatrix} a\neq 0 \\ 0 & \text{si } x\in A-\left\{\begin{pmatrix} 0 & a \\ 0 & 0 \end{pmatrix} a\in R\right\} \end{cases}$$

S. Sessa [10] (Teorema 2.5) demuestra que en un cuerpo no conmutativo K cuando T y S son duales un subconjunto difuso I es un ideal T-S-difuso si y sólo si: (i) $I(x)=I(1)$ para todo $x\in K-\{0\}$; (ii) $I(0)=S\{I(0),I(x)\}\geq T\{I(1),I(1)\}$ para todo $x\in A$; (iii) $I(1)=S\{I(1),I(1)\}$. En la demostración utiliza que T y S son duales sin embargo utilizando las proposiciones anteriores y su demostración es fácil ver que dicho resultado es cierto siendo T y S normas cualesquiera.

Proposición 5.13: Sea A un anillo. Si I es un ideal difuso de A tal que $I(x)\neq I(y)$, $x,y\in A$, entonces $I(x+y)=\min\{I(x),I(y)\}$.

Demostración:

Al ser $I(x)\neq I(y)$ suponemos que $I(x)>I(y)$. Por ser I ideal difuso de A $I(x+y)\geq \min\{I(x),I(y)\}= I(y)= I((x+y)-x)\geq \min\{I(x+y),I(-x)\}=\min\{I(x+y),I(x)\}=I(x+y)$ ya que si $\min\{I(x+y),I(x)\}=I(x)$ entonces $I(y)\geq I(x)$ en contra de la hipótesis. ■

Ejemplo 11: Sea Z el anillo de los enteros, T la norma producto y S la conorma max. Sea I el ideal T-S-difuso definido por:

$$I(x)=\begin{cases} 0.8 & \text{si } x=0 \\ 0.4 & \text{si } x\in(2)-\{2,-2,0\} \\ 0.2 & \text{si } x\in\{2,-2\} \\ 0.16 & \text{si } x\in Z-(2 \end{cases}$$

Para todo $x\neq 0$, $I(x)\neq I(0)$ y sin embargo $I(0+x)>T\{I(0),I(x)\}=0.8.I(x)<I(x)$. Luegono siempre se verifica que si I es un ideal T-S-difuso de A, $x,y\in A$ tal que $I(x)\neq I(y)$ $I(x+y)=T\{I(x),I(y)\}$.

Proposición 5.14: Si I es un ideal T-S-difuso de un anillo A, $T\geq T^*$ y $S\geq S^*$ entonces I es un ideal T^*-S^*-difuso.

Demostración:

$I(x-y)\geq T\{I(x),I(y)\}\geq T^*\{I(x),I(y)\}$

$I(-x)=I(x)$

$I(x.y)\geq S\{I(x),I(y)\}\geq S^*\{I(x),I(y)\}$.

Luego I es un ideal T^*-S^*-difuso. ■

Como consecuencia de esta proposición, al verificarse que $S\geq \max\geq \min\geq T$ para toda norma triangular T y para toda conorma S, tenemos que los ideales T-max-difusos generalizan a los ideales difusos, mientras que los ideales min-S-difusos son un caso particular.

6. Conjuntos de nivel.

Llamamos subconjunto de nivel t de un subconjunto difuso I y lo representamos por I_t, al subconjunto ordinario formado por los

elementos cuyo valor de pertenencia es mayor o igual que t.

Proposición 6.1: Sea A un anillo, T una norma triangular y S una conorma triangular. Si I es un ideal T-S-difuso de A entonces el subconjunto de nivel 1 de I o es el vacío o es un ideal ordinario de A y el subconjunto de nivel 0 es un ideal ordinario de A.

Demostración:

Si $I_1 \neq \phi$, entonces para todo $x,y \in I_1$ $I(x)=I(y)=1$, luego se verifica que $I(x-y) \geq T\{I(x),I(y)\}=T\{1,1\}=1$, por lo que $I(x-y)=1$ y por tanto $x-y \in I_1$.

Para todo $x \in I_1$, para todo $y \in A$, $I(x)=1$, $I(x.y) \geq S\{I(x),I(y)\}=S\{1,I(y)\}=1$, luego $I(x.y)=1$ y por tanto $x.y \in I_1$. El conjunto de nivel 1 es siempre un ideal de A.
El conjunto de nivel 0 es el propio anillo. ■

D. S. Malik [6] demuestra que los subconjuntos de nivel de los ideales difusos que son distintos del vacio, son ideales.

En el ejemplo siguiente podemos observar que los subconjuntos de nivel de los ideales T-S-difusos no son necesariamente ideales.

Ejemplo 12: Sea Z el anillo de los enteros, T la norma producto y S la conorma max. Sea I el subconjunto difuso definido por:

$$I(x)=\begin{cases} 0.8 & \text{si } x=0 \\ 0.4 & \text{si } x \in (2)-\{2,-2,0\} \\ 0.2 & \text{si } x \in \{2,-2\} \\ 0.16 & \text{si } x \in Z-(2) \end{cases}$$

I es un ideal T-S difuso y sin embargo $I_{0.4}$ no es un ideal de Z.

En el conjunto de los ideales T-S-difusos de un anillo vamos a definir una relación de equivalencia de forma que dos ideales que pertenezcan a la misma clase tengan los mismos subconjuntos de nivel. Vemos que cada clase no tiene necesariamente un ideal fuerte T-S-difuso que pertenezca a ella.

Definición 6.1: Si I, J son ideales T-S-difusos de un anillo A decimos que $I\mathbb{R}J$ si para todo $x,y \in A$, $I(x)>I(y)$ si y sólo si $J(x)>J(y)$. Si I es un ideal T-S-difuso de un anillo A designamos por $[\mathbb{I}]$ la clase de equivalencia a la que pertenece el elemento I.

Ejemplo 13: Sea Z el anillo de los enteros, T la norma producto y S la conorma max. Sea I el ideal T-S-difuso definido por:

$$I(x)=\begin{cases} 0.4 & \text{si } x \in (2)-\{2,-2\} \\ 0.2 & \text{si } x \in \{2,-2\} \\ 0.16 & \text{si } x \in Z-(2) \end{cases}$$

Supongamos que existe un ideal fuerte T-S-difuso F de A tal que $F \in [\mathbb{I}]$. Por ser F ideal fuerte $F(0)=1$. Por otra parte como para todo $x \in (2)-\{2,-2\}$ se verifica que $I(x)=I(0)$ entonces $F(x)=F(0)=1$ y además como $I(x)>I(2)$ entonces $F(x)>F(2)$ y por lo tanto $1>F(2)$. También se verifica que para todo $x \in (2)-\{2,0,-2\}$ $I(0)=I(x)=I(x+2)$ luego $F(x)=1$ y $F(x+2)=1$ y por tanto $T\{F(x+2),F(x)\}=1$. Y por otra parte $F(2)=F((2+x)-x)<T\{F(2+x),F(x)\}=1$ lo que contradice que F sea un ideal T-S-difuso de A.

Si I es un ideal T-S-difuso y la imagen de I es finita, con $Iml=\{t_0,t_1,\ldots,t_n\}$, verificando la siguiente relación de orden $t_0>t_1>\ldots>t_n$, entonces $I_{t0} \subset I_{t1} \subset \ldots \subset I_{tn}=A$ donde los contenidos son estrictos. A esta familia de subconjuntos de nivel de A la llamaremos **cadena de subconjuntos de nivel** del ideal T-S-difuso I y la notaremos $\mathbb{C}_I(A)$.

Proposición 6.2: Si I, J son ideales T-S-difusos de un anillo A entonces $I\mathbb{R}J$ si y sólo si $\mathbb{C}_I(A)=\mathbb{C}_J(A)$.

Si A es un anillo e I un ideal T-S-difuso de A entonces podemos formar la cadena $\mathbb{C}_I(A)$ de los conjuntos de nivel del ideal T-S-difuso I. De igual forma, si A es un anillo e I un ideal difuso de A, entonces podemos formar la cadena $\mathbb{C}_I(A)$ de los ideales conjuntos de nivel del ideal difuso I. Recíprocamente dada una cadena de ideales $\mathbb{C}=\{I_i\}$ con $I_{I(0)} \subset I_1 \subset \ldots \subset A$, donde los contenidos son estrictos, entonces existe una clase de equivalencia $[\mathbb{I}]$ de ideales difusos de A tales que $\mathbb{C}_I(A)=\mathbb{C}$.

Proposición 6.3: Si I es un ideal difuso de un anillo A e $[\mathbb{I}]$ su clase de equivalencia entonces existe F ideal fuerte difuso de A, tal que $F \in [\mathbb{I}]$.

Demostración:

Sea I un ideal difuso de A y sea $I_{I(0)}=\{x\in A; I(x)=I(0)\}$. Si F es el ideal difuso de A definido por:

$$F(x)=\begin{cases} 1 & \text{si } x\in I_{I(0)} \\ I(x) & \text{si } x\notin I_{I(0)} \end{cases}$$

F es un ideal fuerte difuso que $F\in[\mathbb{I}]$.

Si el ideal T-S-difuso I es constante la cadena de subconjuntos de nivel de I se reduce al anillo y por lo tanto todos los ideales T-S-difusos constantes pertenecen a la misma clase de equivalencia. Si $Im I=\{t,s\}$, $t\neq s$, la cadena de subconjuntos de nivel de I tiene dos elementos un ideal propio y el anillo y por lo tanto en cada una de las clases de equivalencia existe un elemento que es la función característica de un ideal ordinario.

Si todas las cadenas de ideales de un anillo son finitas entonces la imagen de cualquier ideal difuso es también finita. Sin embargo, al no ser los subconjuntos de nivel de un ideal T-S-difuso necesariamente ideales no podemos, asegurar esta propiedad, para los ideales T-S-difusos.

T. K. Mukherjee y M. K. Sen [7] (Teorema 7) demuestran que si A es un anillo con unidad y cada ideal difuso de A está finitamente valorado entonces A es noetheriano. Posteriormente [8] estos mismos autores caracterizan los anillos artinianos en el Teorema 8 por la condición de que cada ideal difuso esté finitamente valorado y demuestran en el Teorema 3 que un anillo es noetheriano si y sólo si el conjunto de sus valores de pertenencia es un conjunto bien ordenado de [0,1].

Observamos dos hechos en un anillo A con unidad.

(1) Si todo ideal T-S-difuso de A es finitamente valorado, entonces A es un anillo artiniano.

(2) Si el conjunto de valores de cualquier ideal T-S-difuso de A es un conjunto bien ordenado de [0,1], entonces A es un anillo noetheriano.

Referencias

[1] Anthony, J. M., Sherwood, H., Fuzzy groups redefined. *J. Math. Anal. Appl.* 69 (1979), 124-130.

[2] Anthony, J. M., Sherwood, H., A characterization of fuzzy subgroups. *Fuzzy Sets and Systems* 7 (1982), 297-305.

[3] Das, P. S., Fuzzy Vector Space Under Triangular Norms, *Fuzzy Sets and Systems* 25 (1988), 73-85.

[4] Dubreil, P., Dubreil-Jacotin, M. L., *Lecciones de álgebra moderna.* Reverté, Barcelona, 1971.

[5] Liu, W. J., Fuzzy invariant subgroups and fuzzy ideals. *Fuzzy Sets and Systems* 8 (1982), 133-139.

[6] Malik, D. S., Fuzzy ideals of artinian rings. *Fuzzy Sets and Systems* 37 (1990), 111-115.

[7] Mukherjee, T. K., Sen, M. K., On fuzzy ideals of a ring I. *Fuzzy Sets and Systems* 21 (1987), 99-104.

[8] Mukherjee, T. K., Sen, M. K., Rings with chain conditions. *Fuzzy Sets and Systems* 39 (1991), 117-123.

[9] Rosenfeld, A., Fuzzy groups. *J. Math. Anal. Appl.* 35 (1971), 512-517.

[10] Sessa, S., On fuzzy subgroups and fuzzy ideals under triangular norms. *Fuzzy Sets and Systems* 13 (1984), 95-100.

Ideal Semiprimo T-S-Difuso. Nil-Radical T-S-Difuso

Luis Garmendia Salvador
Univ. Complutense de Madrid

María Molero Aparicio
Univ. Alcalá Henares Madrid

Adela Salvador Alcaide
Dto. Matemáticas e Informática Aplicadas
E. T. S. de Ingenieros de Caminos
Universidad Politécnica de Madrid

Abstract

This paper is a continuation of the work of Garmendia [4] on T-S-fuzzy ideals. We define and characterize the T-S-fuzzy semiprime ideals. We define and study the T-S-fuzzy nil radical of a fuzzy subset.

Keywords: Algebra, T-S-Fuzzy ideal; T-S-fuzzy semiprime ideal; fuzzy semiprime ideal; T-S-fuzzy nil radical; fuzzy nil radical.

1. Introducción

Este trabajo es continuación del trabajo de Garmendia [4]. En él se definen y estudian los ideales T-S-difusos y en este se definen, estudian y caracterizan los ideales semiprimos T-S-difusos y los nil-radicales T-S-difusos.

Definimos los ideales semiprimos T-S-difusos haciendo que el valor de pertenencia de un elemento coincida con el valor de pertenencia de su cuadrado. Cuando Kuroki [9] estudia los subsemigrupos semiprimos de un semigrupo propone como condición para definir los subconjuntos semiprimos difusos la propiedad de que el valor de pertenencia de un elemento sea mayor o igual que el valor de pertenencia de su cuadrado, que podemos probar que es equivalente a la nuestra para determinar los ideales T-S-difusos que son semiprimos. Caracterizamos los ideales semiprimos T-S-difusos mediante sus subconjuntos de nivel.

Definimos el nil-radical T-S-difuso de un subconjunto difuso como la clausura de la familia difusa de Moore de los ideales semiprimos T-S-difusos que contienen a dicho subconjunto. Analizamos cuales de las propiedades del nil-radical en álgebra clásica verifica el nil-radical T-S-difuso. Así comprobamos como consecuencia inmediata de la definición, que es una aplicación extensiva, isótona e idempotente.

2. Preliminares

Definición 2.1: Sea A anillo, I subconjunto difuso de A, $I \neq \varnothing$, T una norma triangular y S una conorma triangular. Decimos que I es un **ideal T-S-difuso** de A si y sólo si se verifican las siguientes condiciones:

(i) I es un subgrupo T-difuso:

$I(x-y) \geq T\{I(x),I(y)\}$, para todo $x,y \in A$

$I(x) = I(-x)$, para todo $x \in A$

(ii) $I(x.y) \geq S\{I(x),I(y)\}$, para todo $x,y \in A$.

Si la conorma triangular S es la conorma max al ideal T-max-difuso lo llamamos **idcal T-**

difuso. Si la norma triangular T es la norma min llamamos al ideal min-S-difuso **ideal S-difuso**. En el caso particular de ser ambas min y max llamamos al ideal min-max-difuso simplemente **ideal difuso**. Si además se verifica que $I(0)=1$ decimos que I es un **ideal fuerte T-S-difuso** de A.

Definición 2.2: Sean A, B ideales T-S-difusos de A y T^* una norma triangular:
$(A+_{T^*}B)(x)=\sup_{x=y+z}\{T^*\{A(y),B(z)\}\}$,
$(A\cap_{T^*}B)(x)=T^*\{A(x),B(x)\}$,
$(A._{T^*}B)(x)=\sup_{x=y.z}\{T^*\{A(y),B(z)\}\}$,
$(A._{S^*\text{-}T^*}B)(x)=S^*_{x=y.z}\{T^*\{A(y),B(z)\}\}$,
$A^2_{T^*}=A._{T^*}A$; $A^n_{T^*}=A._{T^*}\ldots._{T^*}A$
$A^*_{T^*}B$ es el mínimo ideal T-S-difuso que contiene a $A._{T^*}B$.
$A^*_{S^*\text{-}T^*}B$ es el mínimo ideal T-S-difuso que contiene a $A._{S^*\text{-}T^*}B$.

Se llama $A+B$, $A\cap B$, A.B, A^n, A*B si, en la definición anterior T es igual al mínimo.

3. Ideales Semiprimos T-S-difusos

Definición 3.1: Sea I un ideal T-S-difuso de un anillo A. Decimos que I es un **ideal semiprimo T-S-difuso** de A si para todo $x\in A$ $I(x)=I(x^2)$.

Recordemos que si B es un subconjunto de un anillo A, B es una **parte semiprima** o **subconjunto semiprimo** si y sólo si para todo $x^m\in B$, $x\in A$, $m\in N \Rightarrow x\in B$. (P. Dubreil [3], pag 46).

Proposición 3.1: Sea I un ideal T-S-difuso de un anillo A. Si para todo $x\in A$ $I(x)=I(x^2)$ entonces, para todo $m\in N$ y para todo $x\in A$, $I(x)=I(x^m)$.

Demostración:

$I(x^4)=I((x^2)^2)=I(x^2)=I(x)$ y para todo $k\in N$, si $m=2^k$ aplicando que $I(x)=I(x^2)$ obtenemos que $I(x^m)=I(x)$.

$I(x)=I(x^4)\geq S\{I(x),I(x^3\}\geq \max\{I(x),I(x^3\}\geq I(x^3)\geq S\{I(x),I(x^2)\}\geq \max\{I(x),I(x^2)\}\geq I(x)$.

Si $q\geq p$, $I(x^q)\geq I(x^p)$ ya que $I(x^q)\geq S\{I(x^p),I(x^{q-p})\}\geq \max\{I(x^p),I(x^{q-p})\}\geq I(x^p)$ para todo $p,q\in N$.

Para todo $n\in N$, $m=2^n>n\geq 1$ se verifica que $I(x)\leq I(x^n)\leq I(x^m)=I(x)$. ■

Como consecuencia inmediata de esta proposición tenemos la siguiente caracterización de ideales semiprimos T-S-difusos.

Corolario 3.2: Sea I un ideal T-S-difuso de un anillo A. I es ideal semiprimo T-S-difuso de A si y sólo si para todo $x\in A$ y para todo $m\in N$, $I(x)=I(x^m)$.

Proposición 3.3: Sea I un ideal T-S-difuso de un anillo A. I es un ideal semiprimo T-S-difuso de A si y sólo si para todo $t\leq I(0)$ los conjuntos de nivel I_t son subconjuntos semiprimos de A.

Demostración:

$\Rightarrow$) Si $x^m\in I_t$ entonces $I(x^m)\geq t$ y como por hipótesis $I(x)=I(x^m)\geq t$, se verifica que $x\in I_t$ luego I_t es un subconjunto semiprimo para todo $t\leq I(0)$.

$\Leftarrow$) Sea $t=I(x^m)$, $m\in N$, $t\leq I(0)$ entonces $x^m\in I_t$ y por ser I_t subconjunto semiprimo entonces $x\in I_t$, luego se verifica que $I(x)\geq t=I(x^m)$. Por otra parte en un ideal T-S-difuso $I(x^m)\geq I(x)$. ■

M. M. Zahedi, V. N. Dixit, R. Kumar y N. Ajmal caracterizan los ideales semiprimos difusos por la condición enunciada en el siguiente corolario. Esta caracterización es probada por Zahedi [12] en anillos conmutativos y unitarios; por Kumar [5] en ideales finitamente valorados en anillos conmutativos y unitarios; y Ajmal [2] para cualquier ideal difuso.

Corolario 3.4: Sea I un ideal difuso de un anillo A, I es un ideal semiprimo difuso de A si y sólo si para todo $t\leq I(0)$ los conjuntos de nivel I_t son ideales semiprimos de A.

Corolario 3.5: Si I es un ideal semiprimo no vacio de un anillo A y σ_I su función característica entonces σ_I es un ideal semiprimo T-S-difuso de A.

Nos indica que los ideales semiprimos ordinarios son un caso particular de los ideales semiprimos T-S-difusos.

Proposición 3.6: Si T^* es una norma triangular generalizada que predomina sobre T y sobre S, la T^*-intersección difusa de una

familia cualquiera $\{I_i\}$ de ideales semiprimos T-S-difusos de un anillo A, tal que $\cap_{T^*} I_i \neq \varnothing$, es un ideal semiprimo T-S-difuso.

Demostración:

Se puede probar que $\cap_{T^*} I_i$ es un ideal T-S-difuso. Y es semiprimo ya que $\cap_{T^*} I_i(x^2) = T^*\{I_i(x^2)\} = T^*\{I_i(x)\} = \cap_{T^*} I_i(x)$. ■

Corolario 3.7: La intersección de una familia cualquiera $\{I_i\}$ de ideales semiprimos T-S-difusos de un anillo A, tal que $\cap I_i \neq \varnothing$, es un ideal semiprimo T-S-difuso.

4. Nil-radical T-S-difuso

Definición 4.1: Sea I un subconjunto difuso de un anillo A. Llamamos **nil-radical T-S-difuso** de I y notaremos $\mathbf{Rad_{T\text{-}S}I}$ a la intersección de todos los ideales semiprimos T-S-difusos que contienen a I.

Denotamos por **radI** al radical de un conjunto ordinario y por $\mathbf{Rad_{T\text{-}S}I}$ al nil-radical T-S-difuso de un subconjunto difuso. Si T es la norma min y S la conorma max llamaremos **RadI** al $\mathrm{Rad_{min\text{-}max}I}$ de un subconjunto difuso.

Sabemos que rad(I) es igual a $\{x \in A;$ existe $n \in N,\ x^n \in I\}$.

Ejemplo 1: Sea Z el anillo de los enteros, T la norma producto y S la conorma max, $p \in Z$ primo e I el ideal T-S-difuso de Z definido por:

$$I(x) = \begin{cases} 0.8 & x \in (p^2) - \{p^2, -p^2\}, \\ 0.7 & x \in \{p^2, -p^2\}, \\ 0.2 & x \in (p) - (\{p, -p\} \cup (p^2)), \\ 0.16 & x \in \{p, -p\}, \\ 0 & x \in Z - (p), \end{cases}$$

$$\text{Entonces } \mathrm{Rad_{T\text{-}S}I}(x) = \begin{cases} 0.8 \text{ si } x \in (p), \\ 0 \quad \text{si } x \in Z - (p). \end{cases}$$

Si llamamos $\mathcal{F}$ al conjunto de los ideales semiprimos T-S-difusos J de A, tales que $J \supset I$, entonces $\mathcal{F}$ es una familia difusa de Moore y $\mathrm{Rad_{T\text{-}S}I}$ es la clausura difusa de Moore de I. Como consecuencia tenemos:

Corolario 4.1: Si I es un ideal T-S-difuso de un anillo A entonces $\mathrm{Rad_{T\text{-}S}I}$ es el mínimo ideal semiprimo T-S-difuso de A que contiene a I.

Corolario 4.2: Sea I un ideal T-S-difuso de un anillo A. I es un ideal semiprimo T-S-difuso de A si y sólo si $\mathrm{Rad_{T\text{-}S}I} = I$.

Proposición 4.3: Sea I un ideal T-S-difuso de un anillo A. Se verifican las siguientes propiedades:

(i) $I \subset \mathrm{Rad_{T\text{-}S}I}$

(ii) Si J es un subconjunto T-S-difuso tal que $I \subset J$ entonces $\mathrm{Rad_{T\text{-}S}I} \subset \mathrm{Rad_{T\text{-}S}J}$

(iii) $\mathrm{Rad_{T\text{-}S}(Rad_{T\text{-}S}I)} = \mathrm{Rad_{T\text{-}S}I}$

Proposición 4.4: Si I es un subconjunto de un anillo A y σ_I su función característica entonces $\sigma_{radI} = \mathrm{Rad_{T\text{-}S}}\sigma_I$.

A continuación vamos a realizar un estudio de los conjuntos de nivel de $\mathrm{Rad_{T\text{-}S}I}$:

Proposición 4.5: Si I es un ideal T-S-difuso de un anillo A, entonces para todo $t \leq I(0)$, $(\mathrm{Rad_{T\text{-}S}I})_t \supset \mathrm{rad}(I_t)$.

Demostración:

Si $t \leq I(0)$, para todo $x \in \mathrm{rad}(I_t)$ existe $n \in N$ tal que $x^n \in I_t$, luego $I(x^n) \geq t$. $\mathrm{Rad_{T\text{-}S}I}$ es un ideal semiprimo T-S-difuso y $\mathrm{Rad_{T\text{-}S}I} \supset I$, luego para todo $n \in N$ $\mathrm{Rad_{T\text{-}S}I}(x) = \mathrm{Rad_{T\text{-}S}I}(x^n) \geq I(x^n) \geq t$ por lo que $x \in (\mathrm{Rad_{T\text{-}S}I})_t$ y por tanto $\mathrm{rad}(I_t)$ $(\mathrm{Rad_{T\text{-}S}I})_t$. ■

Proposición 4.6: Si I es un ideal difuso finitamente valorado de un anillo A entonces $(\mathrm{RadI})_t = \mathrm{rad}(I_t)$, para todo $t \leq I(0)$.

Demostración:

Aplicando la proposición anterior con la t-norma min y la conorma max $\mathrm{rad}(I_t)$ $(\mathrm{RadI})_t$. Pero además se verifica la igualdad, ya que si construimos J como el ideal difuso definido por: Si x $\mathrm{rad}(I_t)$ y para todo $s > t$ x $\mathrm{rad}(I_s)$ entonces $J(x) = t$. J es un ideal semiprimo difuso de A, ya que sus subconjuntos de nivel son ideales semiprimos puesto que para todo $t \leq I(0) = J(0)$ $J_t = \mathrm{rad}(I_t)$ es un ideal semiprimo de A. Además $J \supset I$, pues para todo $x \in A$ tal que $I(x) = t$ entonces $x \in I_t$ y $I_t \subset \mathrm{rad}(I_t) = J_t$ luego $J(x) \geq t = I(x)$. Por lo tanto J es un ideal semiprimo difuso de A y $J \supset I$, luego $J \supset \mathrm{RadI}$ y por tanto $J_t = \mathrm{rad}(I_t) \supset (\mathrm{RadI})_t$ para todo $t \leq I(0)$. ■

Ejemplo 2: Sea Z el anillo de los enteros, I el ideal difuso de Z definido por:

$$I(x)=\begin{cases}1 & x=0,\\ 1-1/n & x\in(2^{n-1})-(2^n),\ \text{para todo } n\geq 1.\end{cases}$$

Proposición 4.7: Sean I, J ideales difusos finitamente valorados de un anillo A. Se verifica que $RadI=A$ si y sólo si $I=A$.

Demostración:

Si $RadI=A$ entonces $(RadI)_{I(0)}$ es igual a $\{x;RadI(x)\geq RadI(0)\}=A \Rightarrow$ $(RadI)_{I(0)}=rad(I_{I(0)})=A \Rightarrow I_{I(0)}=\{x;I(x)\geq I(0)\}=A$

Y como $I(x)\leq I(0)$, para todo x entonces $I(x)=I(0)$. Luego $I(x)=k$, para todo x. I es una función constante. Pero además $I=A$, pues si no lo fuera sería $k\neq 1$ y por tanto $I(x)=k$, para todo x, $k\in[0,1)$. Pero entonces I es un ideal semiprimo difuso, luego $RadI=I$ y por tanto $RadI(x)=k\neq 1$ y $RadI\neq A$ lo que contradice la hipótesis. La otra implicación es evidente.■

Proposición 4.8: Sea I un ideal difuso finitamente valorado de un anillo A. Si $I(x^n)=0$ para todo $n\in N$ entonces $RadI(x^n)=0$.

Demostración:

Sea $x\in A$. Si $I(x^n)=0$ para todo $n\in N$ entonces para todo $t\neq 0$ y para todo $n\in N$, $x^n\notin I_t$ luego por la definición de radical ordinario para todo $t\neq 0$ y para todo $n\in N$, $x^n\notin rad(I_t)$ y aplicando la proposición 4.6 tenemos que $x^n\notin(RadI)_t$ para todo $n\in N$ y para todo $t\neq 0$ luego $RadI(x^n)\neq t$ para todo $t\neq 0$ y para todo $n\in N$ y por lo tanto $RadI(x^n)=0$ para todo $n\in N$.■

El resultado de la siguiente proposición lo enuncia R. Kumar [5,6].

Proposición 4.9: Sea I un ideal difuso de un anillo A entonces $RadI(0)=I(0)$.

Proposición 4.10: Si I es un ideal difuso de un anillo A finitamente valorado entonces $Im(RadI)\subset Im(I)$

Demostración:

Sea $t\in Im(RadI)$ entonces, existe $x\in A$ tal que $RadI(x)=t$, luego $x\in(RadI)_t=rad(I_t)$, por tanto existe n tal que $I(x^n)\geq t$.

Por otra parte como RadI es un ideal semiprimo difuso $RadI(x^n)=RadI(x)=t$ y al estar $I\subset RadI$ entonces $I(x^n)\leq RadI(x^n)=t$ luego de las dos desigualdades obtenemos $I(x^n)=t$. Por lo que $t\in Im I$ y por tanto $Im I\supset Im(RadI)$.

Este contenido, en general, es estricto. Incluso un ideal con infinito número de valores puede tener un nil-radical finitamente valorado como demuestra el ejemplo 2.

5 Operaciones

Proposición 5.1: Si I, J son ideales fuertes T-difusos de un anillo A entonces: $Rad_T(I+_TJ)=Rad_T(Rad_TI+_TRad_TJ)$.

Demostración:

I, J son ideales T-difusos de un anillo A luego por definición de nil-radical $I\subset Rad_TI$ y $J\subset Rad_TJ$, entonces $I+_TJ\subset Rad_TI+_TRad_TJ$, luego $Rad_T(I+_TJ)\subset Rad_T(Rad_TI+_TRad_TJ)$.

Por otra parte si I, J son ideales fuertes T-difusos de A: $I\subset I+_TJ$ y $J\subset I+_TJ \Rightarrow$ $Rad_TI\subset Rad_T(I+_TJ)$ y $Rad_TJ\subset Rad_T(I+_TJ) \Rightarrow$ $Rad_TI+_TRad_TJ\subset Rad_T(I+_TJ)$

Y como $Rad_T(I+_TJ)$ es un ideal semiprimo T-difuso y contiene a $Rad_TI+_TRad_TJ$ contiene a su nil-radical T-difuso, por tanto: $Rad_T(Rad_TI+_TRad_TJ)\subset Rad_T(I+_TJ)$.■

Corolario 5.2: Si I, J son ideales fuertes difusos de un anillo A, entonces $Rad(I+J)=Rad(RadI+RadJ)$.

Proposición 5.3: Si I es un ideal semiprimo T-S-difuso de un anillo A entonces para todo J ideal T-S-difuso tal que $J^n\subset I$, $n\in N$, implica que $J\subset I$.

Demostración:

Si J es un ideal difuso T-S-difuso tal que $J^n\subset I$ siendo I un ideal semiprimo T-S-difuso, entonces para todo $x\in A$, $J(x)\leq J^n(x^n)\leq I(x^n)=I(x)$. Por lo tanto $J\subset I$.■

Proposición 5.4: Si I, J son ideales T-S-difusos de un anillo A, se verifican las siguientes propiedades:

(i) $Rad_{T\text{-}S}(I*_{T*}J)=Rad_{T\text{-}S}(I._{T*}J)\subset Rad_{T\text{-}S}(I\cap_{T*}J)\subset Rad_{T\text{-}S}(I._{S*\text{-}T*}J)=Rad_{T\text{-}S}(I*_{S*\text{-}T*}J)$

(ii) $Rad_{T\text{-}S}(I*_{T*}J)=Rad_{T\text{-}S}(I\cap_{T*}J)$

(iii) $Rad_{T\text{-}S}I\cap Rad_{T\text{-}S}J\supset Rad_{T\text{-}S}(I\cap_{T*}J)$.

Demostración:

(i) Por la definición de nil radical T-S-difuso $Rad_{T\text{-}S}(I*_{T*}J)=Rad_{T\text{-}S}(I._{T*}J)$. Por otra parte para todo $x\in A$:

$Rad_{T\text{-}S}(I._{S*\text{-}T*}J)(x)=Rad_{T\text{-}S}(I._{S*\text{-}T*}J)(x^2)\geq$

$I._{S^*.T^*}J(x^2) = S^*_{x2=y.z}\{T^*\{I(y),J(z)\}\} \geq \sup_{x2=y.z}\{T^*\{I(y),J(z)\}\} \geq T^*\{I(x),J(x)\} = I \cap_{T^*} J(x)$.

$Rad_{T\text{-}S}(I._{S^*.T^*}J) \supset Rad_{T\text{-}S}(I \cap_{T^*} J)$ puesto que $Rad_{T\text{-}S}(I._{S^*.T^*}J)$ es un ideal semiprimo T-S-difuso.

Por otra parte, ya que $I._{T^*}J \subset I \cap_T J$, tenemos por la proposición 4.3 que: $Rad_{T\text{-}S}(I._{T^*}J) \subset Rad_{T\text{-}S}(I \cap_{T^*} J)$.

Luego $Rad_{T\text{-}S}(I._{T^*}J) \subset Rad_{T\text{-}S}(I \cap_{T^*} J) \subset Rad_{T\text{-}S}(I._{S^*.T^*}J)$.

(ii) Es consecuencia inmediata de (i), cuando S=sup, por la doble inclusión

(iii) $I \cap_{T^*} J \subset I \subset Rad_{T\text{-}S}I$, $I \cap_{T^*} J \subset J \subset Rad_{T\text{-}S}J$ luego $I \cap_{T^*} J$ está contenido en $Rad_{T\text{-}S}I \cap Rad_{T\text{-}S}J$ y como $Rad_{T\text{-}S}I \cap Rad_{T\text{-}S}J$ es un ideal semiprimo T-S-difuso de A, al contener a $I \cap_{T^*} J$ contiene a su nil-radical luego: $Rad_{T\text{-}S}(I \cap_{T^*} J) \subset Rad_{T\text{-}S}I \cap Rad_{T\text{-}S}J$. ■

Corolario 5.5: Si $\{I_i\}$ $i \in \{1...n\}$ es una familia de ideales T-S-difusos de un anillo A entonces $\cap_{i\in N} I_i \subset Rad_{T\text{-}S}(\cap_{T^*} I_i) = Rad_{T\text{-}S}(I_1 *_{T^*} \ldots *_{T^*} I_n) \subset \cap Rad_{T\text{-}S}I_i$.

Proposición 5.6: Si I, J son ideales difusos finitamente valorados de un anillo A entonces se verifica que $Rad(I \cap J) = Rad(I.J) = RadI \cap RadJ$.

Demostración:

Por la proposición 5.4 particularizada para la norma min y la conorma max tenemos que $Rad(I \cap J) = Rad(I.J) \subset RadI \cap RadJ$. Por otra parte para todo $x \in A$, si $(RadI \cap RadJ)(x) = t$ entonces $\min\{RadI(x), RadJ(x)\} = t \Rightarrow$
$RadI(x) \geq t$ y $RadJ(x) \geq t \Rightarrow$
$x \in (RadI)_t$ y $x \in (RadJ)_t \Rightarrow$
$x \in rad(I_t)$ y $x \in rad(J_t) \Rightarrow$ existe $n \in N$, $x^n \in I_t$, existe $m \in N$, $x^m \in J_t \Rightarrow$
$I(x^n) \geq t$ y $J(x^m) \geq t \Rightarrow$
$\min\{I(x^n), J(x^m)\} \geq t \Rightarrow$

$Rad(I.J)(x) = Rad(I.J)(x^{n+m}) \geq I.J(x^{n+m}) = \sup\{\min\{I(y),J(z)\}; x^{n+m} = y.z\} \geq \min\{I(x^n), J(x^m)\} \geq t \Rightarrow$
$(RadI \cap RadJ)(x) \leq Rad(I.J)(x)$. ■

Proposición 5.7: Si I_i, $i \in \{1..n\}$, $n \in N$ son ideales difusos finitamente valorados de un anillo A, entonces:

$Rad(I_1..I_n) = Rad(I_1 *..* I_n) = \cap Rad_{i\in N}(I_i)$.

Es claro que la proposición anterior no se verifica para una familia infinita de ideales difusos ya que no se verifica para una familia infinita de ideales ordinarios, que son un caso particular de los ideales difusos.

Corolario 5.8: Si I es un ideal difuso finitamente valorado de un anillo A entonces $Rad(I^n) = RadI$.

Proposición 5.9: Sean I, J ideales T-S-difusos de un anillo A. Si $I \supset Rad_{T\text{-}S}J$, con $I(x) > Rad_{T\text{-}S}J(x)$ para todo $x \in A$ entonces:

(i) $J:I \subset Rad_{T\text{-}S}J$

(ii) $Rad_{T\text{-}S}(J:I) = Rad_{T\text{-}S}J$

Demostración:

(i) Ya que $(J:I).I \subset J$ tenemos que para todo $x \in A$ $((J:I).I)(x^2) \leq J(x^2)$.

Por lo que $\min\{(J:I)(x), I(x)\} \leq \sup_{x2=y.z}\{\min\{(J:I)(y), I(z)\}\} = ((J:I).I)(x^2) \leq J(x^2) \leq Rad_{T\text{-}S}J(x^2) = Rad_{T\text{-}S}J(x)$. Por lo tanto $\min\{(J:I)(x), I(x)\} \leq Rad_{T\text{-}S}J(x)$ para todo $x \in A$.

Como por hipótesis $I(x) > Rad_{T\text{-}S}J(x)$, entonces se verifica que $(J:I)(x) \leq Rad_{T\text{-}S}J(x)$ para todo $x \in A$ y por lo tanto $J:I \subset Rad_{T\text{-}S}J$

(ii) Ya que $J \subset J:I$ por (i) tenemos que $J \subset J:I \subset Rad_{T\text{-}S}J \Rightarrow$
$Rad_{T\text{-}S}J \subset Rad_{T\text{-}S}(J:I) \subset Rad_{T\text{-}S}J \Rightarrow$
$Rad_{T\text{-}S}(J:I) = Rad_{T\text{-}S}J$. ■

Corolario 5.10: Si J es un ideal semiprimo T-S-difuso de un anillo A, I es un ideal T-S-difuso de A tal que $I \supset J$, con $I(x) > J(x)$ para todo $x \in A$, entonces $J:I = J$.

$J:I \subset Rad_{T\text{-}S}J = J$ y como $J \subset J:I$ se tiene la igualdad.

Corolario 5.11: Si I es un ideal fuerte semiprimo T-S-difuso de un anillo A, entonces el anillo cociente T-S-difuso $A/I_{I(0)}$ no tiene elementos nilpotentes y el ideal cociente T-S-difuso I^* asociado a I también es un ideal fuerte semiprimo T-S-difuso de $A/I_{I(0)}$.

La primera observación es una consecuencia inmediata de los resultados de álgebra clásica y la segunda una conclusión obvia de las definiciones de ideal semiprimo T-S-difuso y de ideal cociente T-S-difuso.

Proposición 5.12: Si A es un anillo noetheriano cada ideal difuso finitamente valorado de A contiene a una potencia de su nil-radical difuso.

Demostración:

Si I es un ideal difuso finitamente valorado de un anillo noetheteriano, sea $I_{t_0} \subset I_{t_1} \subset \ldots \subset I_{t_n}$ la cadena de ideales del ideal difuso I, por ser A un anillo noetheriano cada ideal de A contiene una potencia de su nil-radical, luego para todo $i \in \{0,1,\ldots,n\}$ existe mi tal que I_{t_i} contiene a $(\text{rad}(I_{t_i}))^{m_i}$.

Sea m el mínimo común multiplo de mi, $i \in \{0,1,\ldots,n\}$, entonces $I_{t_i} \supset (\text{rad}(I_{t_i}))^m$, para todo $i \in \{0,1,\ldots,n\}$. Como $\text{rad}(I_{t_i}) = (\text{RadI})_{t_i}$ por la proposici⊃on 4.6 tenemos $I_{t_i} \supset ((\text{RadI})_{t_i})^m$ y $((\text{RadI})_{t_i})^m = ((\text{RadI})^m)_{t_i}$ tenemos pues que para todo i, $1 \leq i \leq n$ $I_{t_i} \supset ((\text{RadI})^m)_{t_i}$, luego $I \supset (\text{RadI})^m$. ■

Corolario 5.13: Si A es un anillo artiniano cada ideal difuso de A contiene a una potencia de su nil-radical difuso.

Este corolario es una consecuencia inmediata de la proposición anterior y de que en un anillo artiniano cada ideal difuso es finitamente valorado.

Bibliografía

[1] Anthony J. M.; Sherwood H., Fuzzy groups redefined. *J. Math. Anal. Appl.* 69, (1979), 124-130.

[2] Dixit, V. N., Kumar R., Ajmal, N., On Fuzzy ring. *Fuzzy Sets and Systems*, 49, (1992), 205-213.

[3] Dubreil, P., Dubreil-Jacotin, M. L., *Lecciones de álgebra moderna*. Reverté, Barcelona, 1971.

[4] Garmendia, A., Molero, M., Salvador, A., Ideales T-S-difusos. Análisis de sus propiedades. Congreso ESTYLF'95. Murcia.

[5] Kumar, R., Fuzzy nil radicals and fuzzy primary ideals. *Fuzzy Sets and Systems* 43, (1991), 81-93.

[6] Kumar, R., Certains fuzzy ideals of rings redefined. *Fuzzy Sets and Systems* 46, (1992), 251-260.

[7] Kumar, R., Fuzzy cosets and some fuzzy radicals. *Fuzzy Sets and Systems* 46, (1992), 261-265.

[8] Kumar, R., Fuzzy nil radicals and fuzzy semiprimary ideals of ring. *Fuzzy Sets and Systems* 51, (1992), 345-350.

[9] Kuroki, N., Fuzzy semiprime ideals in semigroups. *Fuzzy Sets and Systems* 8, (1982), 71-79.

[10] Rosenfeld, A., Fuzzy groups. *J. Math. Anal. Appl.* 35, (1971), 512-517.

[11] Sidky F. I., Khatab, S. A., Nil radical of fuzzy ideal. *Fuzzy Sets and Systems* 47, (1992), 117-120.

[12] Zahedi, M. M., A note on L-fuzzy primary and semiprime ideals. *Fuzzy Sets and Systems* 51, (1992), 243-247.

Comparación de definiciones de antisimetría fuzzy. Propiedades.

M. Luisa Eraso Goicoechea
Dep. de Estadística e Inv. Operativa
Universidad Pública de Navarra
Campus de Arrosadía s/n
31006 Pamplona. España
e-mail: leraso@si.upna.es

Miren I. Portilla Manjón
Dep. de Estadística e Inv. Operativa
Universidad Pública de Navarra
Campus de Arrosadía s/n
31006 Pamplona. España
e-mail: mportilla@si.upna.es

Resumen

Se estudia la extensión de la propiedad de antisimetría de relaciones binarias ordinarias a relaciones binarias fuzzy, según diversos autores como Dubois & Prade [1], Kaufmann [2], Klir & Folger [3], Ovchinnikov [4], Prevot [5] y Zadeh [6], analizando las distintas condiciones que imponen para realizar una clasificación y agrupación de las definiciones. Por último se estudian los tipos de antisimetría tales que para una relación fuzzy antisimétrica sus subconjuntos de nivel R_α y la relación ordinaria inducida $\succeq$ son antisimétricas.

Palabras clave: Relación binaria fuzzy, antisimetría, subconjuntos de nivel, relación inducida.

1 Introducción

Las propiedades que verifican las relaciones ordinarias se pueden generalizar para obtener definiciones de propiedades análogas en relaciones fuzzy. La mayor parte de la bibliografía sobre relaciones fuzzy extiende de forma similar dichas propiedades, salvo en lo correspondiente a la antisimetría, en la que los diversos autores imponen distintas condiciones. Enunciaremos las definiciones más usuales para, posteriormente, analizar si son o no equivalentes.

Señalemos en primer lugar la notación que utilizaremos en lo sucesivo para relaciones binarias ordinarias y fuzzy.

Dado un conjunto ordinario E, una relación binaria ordinaria R en E es un subconjunto del producto cartesiano $E \times E$, de forma que dos elementos $x, y \in E$ se dice que están relacionados si $(x,y) \in$ R .

Para representar el conjunto de relaciones binarias fuzzy R entre dos elementos del mismo referencial E, utilizaremos la notación $FR(E)$, denotando por $\mu_R(x,y)$ la función de pertenencia en la relación fuzzy R de cada par de elementos $(x,y) \in E \times E$.

Definición 1 .- *Una relación binaria ordinaria R en E es* **antisimétrica** *si* $\forall x, y \in E$ con $x \neq y$

$$(x,y) \in \mathrm{R} \quad \Rightarrow \quad (y,x) \notin \mathrm{R}$$

La definición se puede expresar de forma equivalente por: $\forall x, y \in E$

$$(x,y) \in \mathrm{R} \quad (y,x) \in \mathrm{R} \quad \Rightarrow \quad x = y$$

Las definiciones más usuales para extender la propiedad de antisimetría de relaciones ordinarias a relaciones fuzzy depende del autor y son las siguientes:

Definición 2.1.- *Una relación* $R \in FR(E)$ *es* **antisimétrica según Dubois & Prade** [1] *si* $\forall x, y \in E$ con $x \neq y$

$$\mu_R(x,y) = 0 \quad \text{ó} \quad \mu_R(y,x) = 0$$

Definición 2.2 .- *Una relación* $R \in FR(E)$ *es* **antisimétrica según Kaufmann** [2] *si* $\forall x, y \in E$ con $x \neq y$

$$\mu_R(x,y) \neq \mu_R(y,x) \quad \text{ó} \quad \mu_R(x,y) = \mu_R(y,x) = 0$$

Definición 2.3 .- *Una relación* $R \in FR(E)$ *es* **antisimétrica según Zadeh** [6] **y Klir & Folger** [3] *si* $\forall x, y \in E$

$$\mu_R(x,y) > 0 \quad \mu_R(y,x) > 0 \quad \Rightarrow \quad x = y$$

Esta definición es equivalente a

$$\forall x, y \in E \quad \text{con} \quad x \neq y$$
$$\mu_R(x,y) > 0 \quad \Rightarrow \quad \mu_R(y,x) = 0$$

expresión a la que Kaufmann [2], Prevot [5] y Zimmermann [7], identifican con el nombre de **antisimétrica perfecta**.

Definición 2.4 .- *Una relación* $R \in FR(E)$ *es* **antisimétrica según Ovchinnikov** [4] *si*

$$\forall x, y \in E \quad \mu_R(x,y) \wedge \mu_R(y,x) = 0$$

Observemos que esta definición obliga a que R sea antirreflexiva.

Definición 2.5 .- *Una relación* $R \in FR(E)$ *es* **antisimétrica según Prevot** [5] *si* $\forall x, y \in E$ con $x \neq y$

$$\mu_R(x,y) \neq \mu_R(y,x)$$

Observemos que las cinco definiciones anteriores pueden clasificarse en dos tipos. Por un lado aquellas que generalizan al caso fuzzy la definición de antisimetría ordinaria, como las de Dubois & Prade y Ovchinnikov, que generalizan la primera expresión de la definición 1, y la de Zadeh y Klir & Folger, que ofrecen la extensión de las dos expresiones equivalentes. Por otro lado tenemos las definiciones de Kaufmann y Prevot, cuyas expresiones provienen de la negación para cada elemento de $E \times E$ de la simetría fuzzy

$$\forall x, y \in E \quad \mu_R(x,y) = \mu_R(y,x)$$

teniendo además en el caso de Kaufmann la formulación fuzzy de no pertenencia a la relación fuzzy de ambos pares $(x,y),(y,x) \in E \times E$

$$\mu_R(x,y) = \mu_R(y,x) = 0$$

que está incluída implícitamente en la definición 1 de antisimetría ordinaria.

2 Comparación de antisimetrías

Para analizar las implicaciones y equivalencias entre las diversas definiciones de antisimetría de los autores señalados anteriormente, utilizaremos una notación conjuntista, denotando con $\mathcal{D}$, $\mathcal{K}$, $\mathcal{Z}$, $\mathcal{O}$ y $\mathcal{P}$ los conjuntos de relaciones fuzzy que verifican la definición de antisimetría de Dubois & Prade, Kaufmann, Zadeh (y Klir & Folger), Ovchinnikov y Prevot, respectivamente.

Proposición 1 .- $\mathcal{O} \subset \mathcal{Z}$

Demostración.

$\mathcal{O} \subset \mathcal{Z}$: Sean $x, y \in E$ con $x \neq y$ tales que $\mu_R(x,y) > 0$. Entonces por la antisimetría de Ovchinnikov

$$\mu_R(x,y) \wedge \mu_R(y,x) = 0$$

luego tenemos $\mu_R(y,x) = 0$ como queríamos demostrar.

$\mathcal{O} \not\supseteq \mathcal{Z}$: Contraejemplo: Sea $R \in FR(E)$ dada por

$$\forall x, y \in E \text{ con } x \neq y \quad \mu_R(x,y) \wedge \mu_R(y,x) = 0$$
$$\text{y} \quad \exists x \in E \text{ tal que } \mu_R(x,x) \neq 0$$

R es una relación fuzzy antisimétrica según la definición de Zadeh pero no lo es según la de Ovchinnikov, ya que éste no exige $x \neq y$. □

Observemos que

$$\mathcal{Z} \cap \mathcal{O} = \mathcal{Z} \cap \{R \in FR(E) | \forall x \in E \quad \mu_R(x,x) = 0\}$$

es decir, las relaciones antisimétricas según Zadeh que además sean antirreflexivas serán antisimétricas según la definición de Ovchinnikov.

Proposición 2 .- $\mathcal{Z} \equiv \mathcal{D}$

Demostración.

$\mathcal{Z} \subseteq \mathcal{D}$: Supongamos, por reducción al absurdo, que la relación no es antisimétrica según Dubois & Prade, es decir:

$$\exists x, y \in E \text{ con } x \neq y \quad \text{tal que}$$
$$\mu_R(x,y) > 0 \quad \text{y} \quad \mu_R(y,x) > 0$$

Entonces por ser R antisimétrica según la definición de Zadeh y $\mu_R(x,y) > 0$ tendríamos $\mu_R(y,x) = 0$ que es contradictorio con el supuesto inicial de $\mu_R(y,x) > 0$.

$\mathcal{Z} \supseteq \mathcal{D}$: Sean $x, y \in E$ con $x \neq y$ tales que $\mu_R(x,y) > 0$. Entonces por verificar R la antisimetría de Dubois & Prade necesariamente $\mu_R(x,y) = 0$ como queríamos demostrar. □

Proposición 3 .- $\mathcal{Z} \subset \mathcal{K}$

Demostración.

$\mathcal{Z} \subset \mathcal{K}$: Si $\forall x, y \in E$ con $x \neq y$ tenemos

$$\mu_R(x,y) = \mu_R(y,x) = 0$$

se cumple la proposición de forma trivial.

Si en cambio tenemos que

$$\exists x, y \in E \text{ con } x \neq y \text{ tales que } \mu_R(x,y) > 0$$

entonces por ser antisimétrica de Zadeh $\mu_R(y,x) = 0$ luego $\mu_R(x,y) \neq \mu_R(y,x)$ como queríamos demostrar.

$\mathcal{Z} \not\supseteq \mathcal{K}$: Contraejemplo: Sea $R \in FR(E)$ dada por

$$\forall x, y \in E \text{ con } x \neq y$$
$$\mu_R(x,y) \neq \mu_R(y,x) \quad \mu_R(x,y) \wedge \mu_R(y,x) \neq 0$$

R es una relación fuzzy antisimétrica según la definición de Kaufmann pero no lo es según la de Zadeh. □

Observemos que en este caso

$$\mathcal{K} \setminus \mathcal{Z} = \{R \in FR(E) | \forall x, y \in E \quad x \neq y$$
$$\mu_R(x,y) \neq \mu_R(y,x) \quad \mu_R(x,y) \wedge \mu_R(y,x) \neq 0\}$$

Proposición 4 .- $\mathcal{P} \subset \mathcal{K}$

Demostración.

$\mathcal{P} \subset \mathcal{K}$: Trivial, ya que la condición que impone la definición de Prevot

$$\mu_R(x,y) \neq \mu_R(y,x)$$

es una de las condiciones que impone la de Kaufmann.

<table>
<tr>
<td rowspan="3">Kaufmann
$\forall x,y \in E$ con $x \neq y$
$\mu_R(x,y) \neq \mu_R(y,x)$
o bien
$\mu_R(x,y) = \mu_R(y,x) = 0$</td>
<td rowspan="2">Prevot
$\forall x,y \in E$ con $x \neq y$
$\mu_R(x,y) \neq \mu_R(y,x)$</td>
<td colspan="2">$\exists\, x,y \in E$ con $x \neq y$ t.q. $\mu_R(x,y) \wedge \mu_R(y,x) \neq 0$</td>
</tr>
<tr>
<td rowspan="2">Zadeh
Dubois y Prade
Klir y Folger
$\forall x,y \in E$ con $x \neq y$
$\mu_R(x,y) \wedge \mu_R(y,x) = 0$</td>
<td>$\forall x,y \in E$ con $x \neq y$
$\mu_R(x,y) \wedge \mu_R(y,x) = 0$
$\exists x \in E$ t.q. $\mu_R(x,x) \neq 0$</td>
</tr>
<tr>
<td>$\exists\, x,y \in E$ con $x \neq y$
$\mu_R(x,y) = \mu_R(y,x) = 0$</td>
<td>Ovchinnikov
$\forall x,y \in E$ con $x \neq y$
$\mu_R(x,y) \wedge \mu_R(y,x) = 0$
$\forall x \in E \quad \mu_R(x,x) = 0$</td>
</tr>
</table>

Tabla 2.1.- Clasificación de las relaciones fuzzy antisimétricas.

$\mathcal{P} \not\supseteq \mathcal{K}$: Contraejemplo: Sea $R \in FR(E)$ dada por
$\forall x,y \in E$ con $x \neq y \quad \mu_R(x,y) = \mu_R(y,x) = 0$
R es una relación fuzzy antisimétrica según la definición de Kaufmann pero no lo es según la de Prevot. □

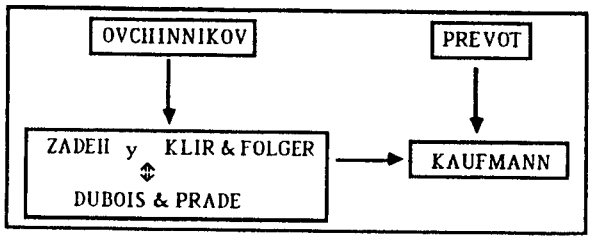
Además la diferencia entre ambos conjuntos es
$\mathcal{K} \setminus \mathcal{P} = \{R \in FR(E) | \forall x,y \in E \quad x \neq y$
$\mu_R(x,y) = \mu_R(y,x) = 0\}$

Observemos que la definición de antisimetría de Prevot no implica la de Zadeh ni viceversa, es decir, $\mathcal{P} \not\subseteq \mathcal{Z}$ ni $\mathcal{P} \not\supseteq \mathcal{Z}$, como muestran los siguientes ejemplos:
$\mathcal{P} \not\subseteq \mathcal{Z}$: Sea $R \in FR(E)$ dada por
$\forall x,y \in E$ con $x \neq y$
$\mu_R(x,y) \neq \mu_R(y,x) = 0 \quad \mu_R(x,y) \wedge \mu_R(y,x) \neq 0$
R es una relación fuzzy antisimétrica según la definición de Prevot pero no lo es según la de Zadeh.
$\mathcal{P} \not\supseteq \mathcal{Z}$: Sea $R \in FR(E)$ dada por
$\forall x,y \in E$ con $x \neq y \quad \mu_R(x,y) = \mu_R(y,x) = 0$
R es una relación antisimétrica según la definición de Zadeh pero no lo es según la de Prevot.
Sin embargo la intersección no es vacía, ya que:
$\mathcal{P} \cap \mathcal{Z} = \{R \in FR(E) | \forall x,y \in E \quad x \neq y$
$\mu_R(x,y) \neq \mu_R(y,x) \quad \mu_R(x,y) \wedge \mu_R(y,x) = 0\}$

Análogamente las definiciones de Prevot y Ovchinnikov presentan la misma situación $\mathcal{P} \not\subseteq \mathcal{O}$ $\mathcal{P} \not\supseteq \mathcal{O}$, que se comprueba con los ejemplos siguientes:
$\mathcal{P} \not\subseteq \mathcal{O}$: Sea $R \in FR(E)$ dada por
$\forall x,y \in E$ con $x \neq y$
$\mu_R(x,y) \neq \mu_R(y,x) \quad \mu_R(x,y) \wedge \mu_R(y,x) \neq 0$
$\exists x \in E$ tal que $\mu_R(x,x) \neq 0$
R es una relación antisimétrica según la definición de Prevot pero no lo es según la de Ovchinnikov.
$\mathcal{P} \not\supseteq \mathcal{O}$: Sea $R \in FR(E)$ dada por
$\forall x,y \in E \quad \mu_R(x,y) \wedge \mu_R(y,x) = 0$
$\exists x_o, y_o \in E$ con $x_o \neq y_o$ tal que
$\mu_R(x_o,y_o) = \mu_R(y_o,x_o) = 0$
R es una relación antisimétrica según la definición de Ovchinnikov pero no cumple la de Prevot para x_o, y_o.
Sin embargo las definiciones no son excluyentes, ya que:
$\mathcal{P} \cap \mathcal{O} = \mathcal{P} \cap \{R \in \mathcal{Z} | \forall x \in E \quad \mu_R(x,x) = 0\}$

Como señalamos anteriormente la definición de Ovchinnikov implica la antirreflexividad en relaciones antisimétricas. De esta forma para relaciones antirreflexivas las definiciones de Zadeh y Ovchinnikov son equivalentes.(ver tabla 2.1)

Tabla 2.2.- Implicaciones entre las definiciones de antisimetría.



Por tanto en los resultados posteriores consideraremos como definiciones de antisimetría significativamente distintas las de Kaufmann, Prevot y Zadeh.

3 Relaciones inducidas en el referencial

Dada una relación binaria fuzzy $R \in FR(E)$ sobre un referencial E, se pueden definir, a partir de dicha R, relaciones binarias ordinarias inducidas en el referencial como señala Kaufmann [3]:

Definición 3.- *Dada una relación binaria fuzzy R y $\alpha \in (0,1]$, llamaremos* **subconjunto de nivel** α *de dicha relación fuzzy a la relación ordinaria*

$$R_\alpha = \{(x,y) \in E \times E | \mu_R(x,y) \geq \alpha\}$$

Definición 4.- *Dada una relación binaria fuzzy $R \in FR(E)$, llamaremos* **relación inducida** *a la relación ordinaria, que denotaremos $\succeq$, dada por*

$$\forall (x,y) \in E \times E$$
$$x \succeq y \quad \text{si} \quad \mu_R(y,x) \geq \mu_R(x,y)$$

De esta forma es conveniente estudiar si la propiedad de antisimetría de las relaciones fuzzy conlleva que las relaciones ordinarias inducidas por ellas son también antisimétricas, y analizar si la conservación o no de dicha propiedad depende de la definición del autor.

Proposición 5 .- *Sea $R \in \mathcal{Z}$ con R_α sus subconjuntos de nivel con $0 < \alpha \leq 1$.*
Entonces R_α es antisimétrica en $E \quad \forall \alpha \in (0,1]$
Demostración.
Sean $x, y \in E$ con $x \neq y$ y $\alpha \in (0,1]$ y supongamos $(x,y) \in R_\alpha$. Entonces $\mu_R(x,y) \geq \alpha$ con $\alpha > 0$ luego $\mu_R(x,y) > 0$
Por la antisimetría perfecta de R, $\mu_R(y,x) = 0$ y por tanto

$$(y,x) \notin R_\alpha \quad \forall \alpha \in (0,1]$$

lo que significa que R_α es antisimétrica para cada $\alpha \in (0,1]$ □

Para relaciones fuzzy antisimétricas según la definición de Kaufmann o de Prevot los R_α con $\alpha \in (0,1]$ pueden tener diversas propiedades como muestran los siguientes resultados.

Proposición 6 .- *Sea $\alpha \in (0,1]$. Entonces:*
i) *Si* $R \in \mathcal{K}_1 = \{R \in \mathcal{K} | \forall x, y \in E \quad x \neq y$

$$\text{o bien } \mu_R(x,y) \wedge \mu_R(y,x) \geq \alpha$$
$$\text{o bien } \mu_R(x,y) \vee \mu_R(y,x) < \alpha\}$$

su subconjunto de nivel R_α es una relación simétrica.
ii) *Si* $R \in \mathcal{K}_2 = \{R \in \mathcal{K} | \forall x, y \in E \quad x \neq y$

$$\text{o bien } \mu_R(x,y) \vee \mu_R(y,x) \geq \alpha$$
$$\text{y } \mu_R(x,y) \wedge \mu_R(y,x) < \alpha$$
$$\text{o bien } \mu_R(x,y) \vee \mu_R(y,x) < \alpha\}$$

su subconjunto de nivel R_α es una relación ordinaria antisimétrica.
iii) *Si $R \in \mathcal{K} \setminus \mathcal{K}_1$, su subconjunto de nivel R_α es una relación asimétrica.*
Demostración.
i) Sean $x, y \in E$ con $x \neq y$ tal que $(x,y) \in R_\alpha$. Entonces $\mu_R(x,y) \geq \alpha$, con lo cual tenemos

$$\mu_R(x,y) \vee \mu_R(y,x) \geq \alpha$$

Entonces, para $R \in \mathcal{K}_1$ tendremos

$$\mu_R(x,y) \wedge \mu_R(y,x) \geq \alpha$$

con lo cual $\mu_R(y,x) \geq \alpha$, luego $(y,x) \in R_\alpha$ y entonces R_α es simétrica.
ii) Sean $x, y \in E$ con $x \neq y$ tal que $(x,y) \in R_\alpha$. Entonces $\mu_R(x,y) \geq \alpha$, con lo cual

$$\mu_R(x,y) \vee \mu_R(y,x) \geq \alpha$$

Entonces, para $R \in \mathcal{K}_2$ tendremos además

$$\mu_R(x,y) \wedge \mu_R(y,x) < \alpha$$

con lo cual $\mu_R(y,x) < \alpha$, luego $(y,x) \notin R_\alpha$ y entonces R_α es antisimétrica.
iii) Trivial por eliminación del caso i) ya que simetría y asimetría son excluyentes. □

Observemos que la intersección de los conjuntos $\mathcal{K}_1$ y $\mathcal{K}_2$ anteriores es no vacía:

$$\mathcal{K}_1 \cap \mathcal{K}_2 = \{R \in \mathcal{K} | \forall x, y \in E \quad x \neq y$$
$$\mu_R(x,y) \vee \mu_R(y,x) < \alpha\}$$

ya que en ese caso $(x,y) \notin R_\alpha$ y $(y,x) \notin R_\alpha$

Proposición 7 .- *Sea $\alpha \in (0,1]$. Entonces:*
i) *Si $R \in \mathcal{P}_1 = \mathcal{P} \cap \mathcal{K}_1$, su subconjunto de nivel R_α es una relación simétrica.*
ii) *Si $R \in \mathcal{P}_2 = \mathcal{P} \cap \mathcal{K}_2$, su subconjunto de nivel R_α es una relación antisimétrica.*
iii) *Si $R \in \mathcal{P} \setminus \mathcal{P}_1$, su subconjunto de nivel R_α es una relación asimétrica.*
Demostración.
Basta aplicar la proposición anterior, ya que $\mathcal{P} \subset \mathcal{K}$ por la proposición 4. □

Además se obteniene una observación análoga:

$$\mathcal{P}_1 \cap \mathcal{P}_2 = \{R \in \mathcal{P} | \forall x, y \in E \quad x \neq y$$
$$\mu_R(x,y) \vee \mu_R(y,x) < \alpha\}$$

ya que en ese caso $(x,y) \notin R_\alpha$ y $(y,x) \notin R_\alpha$

Proposición 8 .- *Sea $R \in \mathcal{P}$. Entonces la relación $\succeq$ es antisimétrica.*
Demostración.
Sean $x, y \in E$ tal que $x \succeq y \quad y \succeq x$. Entonces

$$\mu_R(y,x) \geq \mu_R(x,y) \quad \text{y} \quad \mu_R(x,y) \geq \mu_R(y,x)$$

Por tanto $\mu_R(y,x) = \mu_R(x,y)$ y aplicando la definición de antisimetría de Prevot, por la cual

$$\mu_R(x,y) \neq \mu_R(y,x) \quad \forall x, y \in E \text{ con } x \neq y$$

tenemos que

$$\mu_R(x,y) = \mu_R(y,x) \quad \Rightarrow \quad x = y$$

luego $\succeq$ es una relación ordinaria antisimétrica. □

Notemos que para relaciones fuzzy que sean antisimétricas según la definición de Zadeh o bien según la de Kaufmann la propiedad anterior solo es cierta si además cumple la definición de Prevot, ya que las relaciones $R \in \mathcal{K} \setminus \mathcal{P} = \mathcal{Z} \setminus \mathcal{P}$ son tales que

$$\mu_R(x,y) = \mu_R(y,x) = 0 \quad \forall x,y \in E \quad x \neq y$$

y entonces $x \succeq y$, $y \succeq x$ con $x \neq y$, lo que significa que $\succeq$ no es antisimétrica.

Sin embargo Kaufmann enuncia en [3] un resultado en el que la relación ordinaria inducida por una relación fuzzy antisimétrica según su definición, es antisimétrica, lo cual es erróneo en su enunciado pero no en su demostración, ya que en ella Kaufmann hace referencia a la definición de grafo asociado a una relación de orden fuzzy, en la que impone para el grafo G la condición

$$\forall x,y \in E \text{ con } x \neq y$$
$$\mu_R(x,y) = \mu_R(y,x) = 0 \quad \Rightarrow \quad (x,y) \notin G \quad (y,x) \notin G$$

lo que aplicado a la relación inducida $\succeq$ significa

$$\forall x,y \in E \text{ con } x \neq y$$
$$\mu_R(x,y) = \mu_R(y,x) = 0 \quad \Rightarrow \quad x \not\succeq y \quad y \not\succeq x$$

de forma que obliga a su relación antisimétrica fuzzy a verificar la definición de antisimetría de Prevot para cualesquiera $x,y \in E$ tal que $x \succeq y$, y por tanto lo que demuestra es equivalente a la proposición 8 anterior.

4 Conclusiones y trabajos futuros

La comparación de las diversas condiciones impuestas por varios autores en las definiciones de antisimetría nos llevan a clasificarlas en tres categorías, coincidentes para cierto tipo de relaciones, pero excluyentes en otros. Como consecuencia las definiciones de orden de dichos autores también se pueden clasificar, y en general la mayor parte de la teoría desarrollada para relaciones fuzzy puede enriquecerse analizando las propiedades que se derivan de cada uno de los tipos de antisimetría.

En esta línea nuestro trabajo se encamina hacia el análisis de la influencia de la anterior clasificación en la construcción de relaciones fuzzy a partir de familias de relaciones ordinarias (estudio preparado para enviar al *Journal of fuzzy Mathematics*) y en las propiedades de las ecuaciones relacionales, así como al estudio de la adecuación de cada uno de los tipos de antisimetría a distintas aplicaciones.

5 Agradecimientos

Agradecemos al Dr. Pedro Burillo su orientación y supervisión de los resultados, así como su interés y apoyo en futuros trabajos.

Referencias

[1] D. DUBOIS y H. PRADE. *Fuzzy sets and Systems. Theory and Applications.* Academic Press, 1980.

[2] A. KAUFMANN. *Introduction a la theorie des sous-ensembles flous a l'usage des ingénieurs.* Tomo I. Masson, 1977.

[3] G. J. KLIR y T.A. FOLGER. *Fuzzy sets, uncertainty and information.* Prentice-Hall International, 1988.

[4] S. OVCHINNIKOV. Similarity relations, fuzzy partitions and fuzzy orderings. *Fuzzy Sets and Systems* **40** (1991), 107-126.

[5] M. PREVOT. *Sous-ensembles flous. Une approche theorique.* Institut de Mathematiques economiques, 1977.

[6] L. A. ZADEH. Similarity relations and fuzzy orderings. *Information Sciences* **3** (1971), 177-200.

[7] H. J. ZIMMERMANN. *Fuzzy set theory and its applications.* Kluwer Academic Publishers, 1991.

Control

Fuzzy Logic Supervisory Control of Induction Motor Under Full Range of Operating Conditions

M.A. Brdys
Dpto. ESAII-ETSEIT
Universitat Politècnica de Catalunya.
E-mail brdysm@tiger.upc.es
On leave from School of Electronic & Electrical Engineering
the University of Birmingham
Birmingham, UK.

Joseba Quevedo
Dpto. ESAII-ETSEIT
Universitat Politècnica de Catalunya
E-mail joseba@esaii.upc.es

W.L. Sim
School of Electronic & Electrical Engineering
the University of Birmingham
Birmingham, UK.

Abstract

A two-layer controller is proposed for speed control of induction motor. Full range of motor operating conditions is considered which are due to normal loading and also overloading of the motor. These motor load torques imply different operational models of the motor and consequently different structures of the controlled variables. Within each of the operational modes two operational phases are distinguished in order to account better for the motor nonlinearities. A normal operational mode, that is under a load torque which can be met by an electrical torque produced by the motor was considered by a recent paper [4]. A fuzzy logic based supervisory control layer was proposed to tune the weights of the modified adaptive GPC which was used as the direct layer controller. The paper uses this control system structure and develops it further so that the overloading situation can be handled as well. A set of controlled variables by the GPC is now extended to include also the variables to be controlled under overloading conditions. The transfers from one set of controlled variables to the other one required due to changes of the motor loading are performed by manipulating the GPC weights. This is done by the supervisory control layer. A fuzzy logic is employed here within a gain scheduling form and provides for a mechanism to tune the weights gradually so that the smooth transients can be achieved.

A feasibility of the approach is verified by computer simulation.

Key words: *AC drives, predictive control, adaptive control, fuzzy logic supervision.*

1 Introduction

Presently, typical control strategy for induction motor is the field-oriented control. It was first introduced by Blaschke [1] and followed by many others. An excellent presentation of different field-oriented control strategies is given in [8]. The method involves transformation of vectors in fixed stator frame into vectors in a frame (x, y) which rotates with either rotor or stator flux vectors y_r or y_s respectively. Suitable transformation of control inputs allow linear dynamic model and partly linear rotor speed dynamic model to be obtained. The two subsystems are de coupled if constant flux is maintained. Two independent PI control loops are, therefore, used to control the flux and the rotor speed. An interpretation of the field-oriented control, from modern control view point, that uses state transformation and nonlinear state feedback is presented in [7]. It was shown in [6] that field-oriented control can improved by achieving exact input-output decoupling and linearization if scalars such as rotor speed, ω_r, electrical torque, T_e and energy of the motor magnetic field, $|y_r|^2$, are taken as the state variables. Again, a nonlinear state feedback control law is required with not greater degree of complexity as compared to the field-oriented one. The fundamental drawback of the decoupling approach is that

highly accurate values of the motor parameters such as rotor resistance or inductance and the rotor flux are required in order to produce required compensating signals. The controllers performance is sensitive to "inaccurate decoupling". An adaptive version of the scheme of [6] was proposed in [7]. However, the parameters and load torque, T_L are only estimated while the rotor flux is assumed to be known. It is well known [2] that joint y_r, rotor resistance, and T_L estimation is not as of high performance as separate estimation of parameters and T_L.

Another approach which does not require decoupling has been recently proposed in [3]. It uses Generalized Predictive Control (GPC) technique [4] as the basic control strategy. The GPC needs the motor model in order to generate on-line control inputs by optimizing a performance index. The performance index constitutes a weighted sum of squares of errors between current and reference trajectories of the controlled variables over suitably chosen time horizon. A trade off between interacting control loops is achieved by adjusting the weights. The induction motor response is relatively fast and a simple model is required to solve the optimization problem on a typical DSP. The model is the key issue in this approach. In [3] a stator flux reference frame is taken and the multi-input multi-output (MIMO) model of ARIMAX type is used to relate controlled outputs and control inputs. The model is time discrete, linear but its parameters are functions of the motor operating point and the physical parameters. The model parameters are updated recursively on-line based on available measurements. Standard RLS with forgetting factor identification algorithm turns out to be adequate. The resulting controller is, therefore, a MIMO interacting adaptive GPC algorithm. It uses the stator field coordinates to obtain a suitable model.

The controller was further developed in [4] in order to improve the transient properties due to strong nonlinearities of the induction motor. A supervisory control layer was introduced in order to tune the GPC weights according to changes of the motor operating point. The supervisory controller was designed based on a fuzzy logic philosophy. The approach was motivated by the fact that relationship between weights and the transient properties can not be described by suitable mathematical model while sufficient qualitative knowledge exists. The two-layer controller of [4] was designed only for normal loading conditions of the motor. In this paper we develop further that controller in order to handle also a situation when the load torque can not be met by the electrical torque produced by the motor.

An interesting control approach to the torque tracking was proposed in [9]. The field orientation is achieved but only asymptotically. The state feedback control law is combined with the rotor currents observer. However, the machine parameters are assumed to be known.

A it will be explained in the subsequent section the controlled variables structure should be different in such situation. In order to cover full operational range the supervisory controller must be able to change on-line the direct controller structure. This is still very much open problem although much research work was done to solve it. We shall show that two-layer integration of GPC and fuzzy logic constitutes a successful approach in the case of the induction motor case-study.

2 Control Structure

If in the stator supply neutral point is isolated, the three-phase stator windings can be transformed, using Parks transformation into equivalent quadrature-phase windings (SD, SQ).

The induction motor dynamics in the stationary d-q coordinates is described a fifth-order nonlinear model:

$$\frac{d\omega_r}{dt} = C_1 (\psi_{rd} i_{sq} - \psi_{rq} i_{sd}) - C_O T_L$$

$$\frac{di_{sd}}{dt} = C_2 \psi_{rd} + C_3 \omega_r \psi_{rq} - C_4 i_{sd} + C_5 u_{sd}$$

$$\frac{di_{sq}}{dt} = C_2 \psi_{rq} + C_3 \omega_r \psi_{rd} - C_4 i_{sq} + C_5 u_{sq}$$

$$\frac{d\psi_{rd}}{dt} = C_6 i_{sd} + C_7 \psi_{rd} - C_8 \omega_r \psi_{rq}$$

$$\frac{d\psi_{rq}}{dt} = C_6 i_{sq} + C_7 \psi_{rq} - C_8 \omega_r \psi_{rd}$$

where C0, ..., C8 are coefficients depending upon electromechanical parameters of the motor and i, u, ψ denote respectively current, voltage and flux linkage. Subscripts r and s stand for rotor and stator respectively.

A major controller task is to maintain the required rotor speed ω_r^* or to track the prescribed speed trajectory regardless of the load torque. Important physical constraints are due to a magnetic saturation of the induction machine and the thermal limitations of the electrical circuitry. Larger stator flux-linkage phasor modulus higher motor efficiency. Therefore, although a

primary control objective is to control the rotor speed the stator flux phasor modulus $|y_S|$ is taken as the second controlled variable. The set-point value is chosen at the saturation boundary so that the tracking error in the flux loop must be kept very small. The thermal constraints can be transformed into the constraints imposed on the stator current which is drawn from the supply, that is $|i_S| \leq |i_{Smax}|$. The stator flux rotating frame is chosen for the modeling purposes. It means that the Cartesian set of coordinates is chosen where the x axis is determined by actual position of the vector ψ_S. This is a rotating set of coordinates known as the stator vector field coordinates [8]. Such representation of the induction motor variables allows for suitable identification of the variables operational functions and also leads to relatively simple input-output model. The latter is very important for any controller which utilizes the model on on-line basis.

The i_{SY} component of the stator current phase changes according to the load torque T_L or ω_r changes in order to deliver required value of electrical torque T_e. These changes imply changes of the second component i_{SX} as the $|y_S|$ is kept constant. If therefore $|i_S|$ tends to exceed its upper limit in "steady-state" it means that the required T_e, $T_e > T_{emax}$ and thus it can not be delivered. The motor is *overloaded* and the corresponding operational mode is called *emergency mode* as opposed to the *normal mode*, that is when $T_e \leq T_{emax}$. The rotor speed can not be then maintained any longer and must keep decreasing. What one can do, however, is to slow down this decrease as much as possible maintaining the $|i_S|$ at its upper limit. This is implemented by changing the set of controlled variables from ($|y_S|$, ω_r) to (i_{SX}, i_{SY}) and applying the corresponding reference values $ins_2 \overset{\Delta}{=} i^*_{sy} - i_{sy} i^*_{sx}$ and i^*_{sx} calculated from T_{emax} (i^*_{sy}) and $|i_{Smax}|$ (i^*_{sx}) limits. Clearly, these reference values constitute the limits for i_{SY} and i_{SX}.

In this way the machines delivers T_{emax} and constraints are satisfied. If the rotor speed reaches zero value then the supply is switched off. If the excessive load torque is removed before the speed reaches zero the normal operational mode is restored and the corresponding structure ($|\psi_s|$, ω_r) is restored as well. The machine is then, however, not in the *steady-state phase* of the normal operational mode but in a *recovery phase* of this mode. The last phase of the normal mode is if the motor starts-up and it is called a *start-up phase*. Clearly, due to the induction motor nonlinearities the best sets of the controller parameters are different for different phases.

Wanted fast responses of the above individual control loops may lead to instantaneous large variations of i_{SY} (dynamics of i_{SX} is much slower). This means that apart from the "steady-state" violation of the thermal constraints we may face also the instantaneous variations of these constraints. A suitable control mechanism is, therefore, needed to trade between quality of the main loop responses and the instantaneous violation of the current constraints. The derivative $\frac{d_{sy}}{dt}$ of the current i_{SY} is proposed to be the additional controlled variable. Its set-point value to be maintained is taken as the derivative value in a steady-state, that is zero.

In this paper the Generalized Predictive Controller is applied to implement the proposed control strategy at a direct level. The GPC weights are very suitable instruments to compromise the transient responses of this highly interacting MIMO system within each of the operational modes. The rotor speed control loop can be implemented directly (ω_r^* being compared with the stator flux phasor rotating speed, w_{ms}). The corresponding controllers are called *closed-loop* and *open-loop*, respectively. In the latter case ω_r differs from ω_r^* due to a slip. An obvious advantage of the open-loop controller can be then utilized only if the slip is small. Both controllers are investigated in the paper. The parameters of the model used by GPC must be updated on-line according to changes in the motor operating point. The adaptation is also needed to accommodate natural changes in time of certain parameters such as the rotor resistance and to compensate an initial error in assessment of constant parameters. As a result of modeling in the rotating frame the controller outputs are (x,y) coordinates u_{SX} and u_{SY} of the stator voltage phasor u_S.

As the induction motor dynamics is relatively fast a mathematical model which is used by GPC must be simple so that the corresponding optimization problem can be solved on-line on a typical DSP. We shall choose a linear structure. It was found that due to physical nonlinearities of induction motor the on-line update of the model parameters may not be sufficient if the changes of the TL and/or ω_r^* are large or if the motor moves through two complete operational phases, e.g. start-up and normal. It is proposed to utilize the *error weights* of the GPC performance index as the controller parameters to be tuned according to the current operating point and/or operating phase of machine. Mathematical relationships between the weights and the controlled motor transient responses are highly nonlinear and not available. However, *good qualitative* knowledge on how the variations of these weights affect the re-

sponses exist. Therefore, it is proposed in this paper to employ a fuzzy logic approach to design a control layer responsible for tuning of the weights. The corresponding control unit is called a *supervisory controller*. Applications of fuzzy logic to direct control design for an induction motor can be found in, e.g. [11], [10] and [12]. We believe, however, that our approach offers better integration of the quantitative and qualitative information, what leads to better performance of the resulting controller. The supervisory controller is of a gain scheduling type. The weight values are produced based on current values of selected physical variables.

Similar integration of the GPC at a direct control layer with the fuzzy logic at the supervisory control layer was used in [13] for path tracking of mobile robots.

So far we have considered the supervisory controller operation only for fixed operational model. Its key activity is also to supervise change of structure of the controlled variables implied by change of the operational mode. The GPC weights are excellent instruments for the purpose of implementing the structure changes. Indeed, the wanted control loops can be activated by applying non-zero values of the corresponding weights while the unwanted loops can be switched off by applying zero values of the associated weights. The desired operational mode is detected by the difference $ins_2 \overset{\Delta}{=} i^*_{sy} - i_{sy}$ as the current i_{sy} can be taken as an approximate measure of the load torque. A crisp implementation of this strategy may lead, however, to very rapid changes of the controlled variables driven by uncontrolled dynamics. The use of fuzzy logic in the supervisory control mechanism enables us to implement the zero-non zero changes of the weights gradually and therefore, to achieve smooth transient responses.

The overall control structure is illustrated in figure 1.

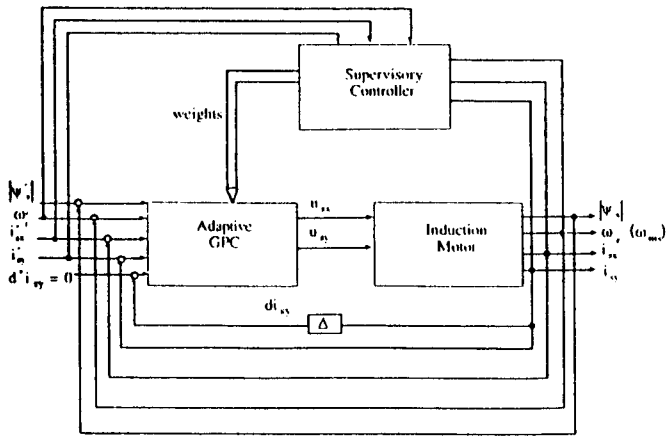


Figure 1

3 Control Algorithms

We shall now describe briefly the algorithms used in the control layers.

3.1 Modified Adaptive Generalized Predictive Control Algorithm

Let use denote by ω_1, ω_2, ω_3, ω_4 and ω_5 the weights associated with the $|y_s|$, ω_r, i_{sx}, i_{sy} and di_{sy} loops, respectively. In order to simplify an implementation of the GPC algorithm we shall assume only crisp membership functions for the weights ω_4 and ω_3 and allocate numerical zero to symbolic ZE for ω_1, ω_2, ω_3 and ω_4. This practically implies that the GPC is run with too control inputs and three controlled outputs. However the modification reported in [3] which uses the fact that the zero set-point value of the third controlled output di_{sy} is always reached in a steady-state allows to maintain the computational effort comparable with a case when the number of controlled outputs is not greater than the number of control inputs. A standard discrete-time input-output model of ARIMAX structure is used. The model parameters are updated by using standard least square method with a forgetting mechanism due to nonstationarity of the parameters.

3.2 Fuzzy Logic Supervisory Control Algorithm

The supervisory controller input variables are defined as follows:

$$ins_1 = i^*_{sx} - i_{sx}$$

$$ins_2 = i^*_{sy} - i_{sy}$$

$$ins_3 = \omega_r^* - \omega_r\ (\omega_{ms})$$

$$dins_2 = i_{sy}\ (i) - i_{sy}\ (i\text{-}1)$$

$$dins_3 = \omega_r\ (i) - \omega_r\ (i\text{-}1) \text{ or } \omega_{ms}\ (i) - \omega_{ms}\ (i\text{-}1)$$

The outputs are weights ω_1, ..., ω_5.

The input variables are fuzzified using crisp and triangular membership functions corresponding to the linguistic values negative medium (NM), negative small (NS), zero (ZE), positive small (PS) and positive medium (PM). Analogously, five membership functions are employed to define the weights linguistic values: ZE, PS, PM, PL and very large (VL). All membership functions describing ω_5 are assumed crisp. True fuzzifica-

tion of the ω_5 complicates significantly design of the supervisory controller and is still under the research.

The membership functions for ins_2, $dins_1$, ω_5 and ω_1 are illustrated in figures 2, 3, 4 and 5, respectively.

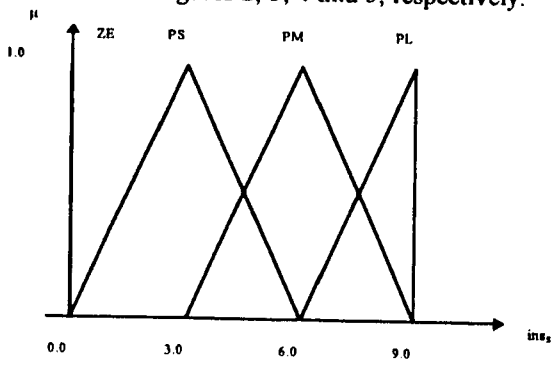


Figure 2

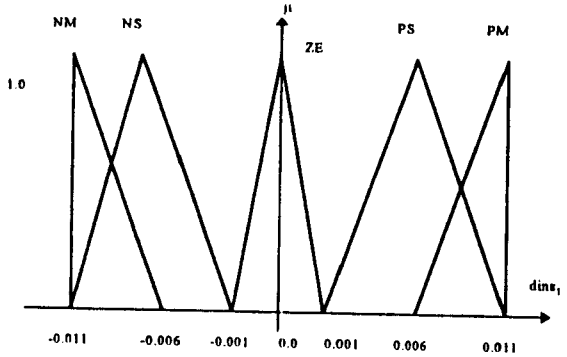


Figure 3

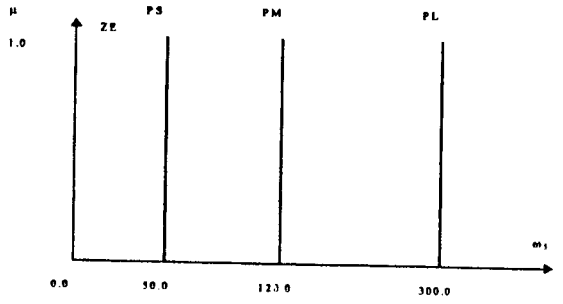


Figure 4

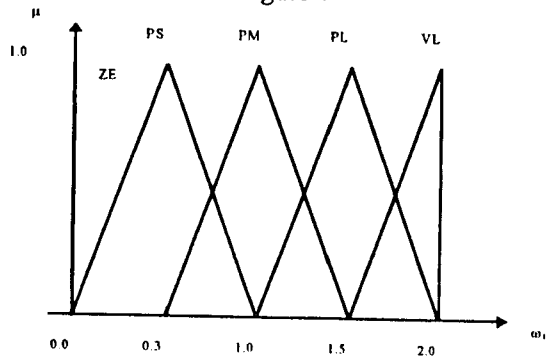


Figure 5

The variable ins_2 is very important as it defines the mode of operation of the controller. Load torque is generally approximated by the value of ins_2 and the torque overloading is indicated as $ins_2 = 0$. This defines the emergency mode and the supervisory control layer will react accordingly by switching "on" and "off" the corresponding loops.

The variable ins_1 is used to monitor and limit the value of i_{sx} during emergency mode of operation. The variable ω_r (ω_{ms}) is mainly used during the startup and recovery phase because it tracks the amount of deviation of the actual value from the set-point. The main purpose of using $dins_2$ is to detect the presence of excessive torque during the emergency mode of operation. In this case, a positive value for $dins_2$ would indicate that the speed is recovering the excessive load torque has been removed.

The membership functions were formulated using the knowledge and experience gathered from past experiments and simulation trials.

The inference engine linguistic rules depend on the operational phase and mode of the motor. the rules are designed to provide for a desired compromise between the rise time and the overshoots of the control loop responses.

The rules for ω_1 and ω_2 during startup, steady state and recovery are shown in tables 1, 2,...

The steady-state phase is the final phase of supervisory control algorithm. The supervisory control layer will remain in the steady-state phase as long as overloading does not occur during operation.

FAM-STARTUP

ω_1	ins_3 / ins_2	NM	NS	ZE	PS	PM
	ZE	ZE	ZE	ZE	ZE	PS
	PS	PM	PS	PM	PM	PS
	PM	PL	PM	PM	PM	PS
	PL	PL	PM	PM	PM	PS

Table 1

ω_2	ins_3 / ins_2	NM	NS	ZE	PS	PM
	ZE	ZE	ZE	ZE	ZE	PS
	PS	PM	PS	PM	PM	PS
	PM	PL	PM	PM	PM	PS
	PL	PL	PM	PM	PM	PS

Table 2

FAM-STEADY STATE

ω_1 \ ins_3	NS	NM	ZE	PS	PM
ZE	ZE	ZE	ZE	ZE	ZE
ins_2 PS	PL	PM	VL	PL	PM
PM	VL	PL	VL	PL	PM
PL	VL	PL	VL	PL	PM

Table 3

ω_2 \ ins_3	NS	NM	ZE	PS	PM
ZE	ZE	ZE	ZE	ZE	ZE
ins_2 PS	PL	PM	VL	PL	PM
PM	VL	PL	VL	PL	PM
PL	VL	PM	VL	PL	PM

Table 4

FAM-RECOVERY

ω_1 \ ins_3	NS	NM	ZE	PS	PM
ZE	ZE	ZE	ZE	ZE	ZE
ins_1 PS	PM	PS	PS	PS	PS
PM	PM	PM	PS	PS	PS
PL	PL	PM	PS	PS	PS

Table 5

ω_1 \ ins_3	NS	NM	ZE	PS	PM
ZE	ZE	ZE	ZE	ZE	ZE
ins_2 PS	PM	PS	PS	PS	PS
PM	PM	PM	PS	PS	PS
PL	PL	PM	PS	PS	PS

Table 6

Notice that 'soft' weights are used during the recovery phase in order to keep the unwanted transients to a minimal.

Tables 7, 8 and 9 illustrate the rules for ω_1 and ω_2 and ω_3 during the emergency mode.

FAM- EMERGENCY

ω_1 \ $dins_3$	NS	NM	ZE	PS	PM
ZE	ZE	ZE	ZE	PS	PS
ins_2 PS	X	X	X	X	X
PM	X	X	X	X	X
PL	X	X	X	X	X

Table 7

ω_2 \ $dins_3$	NS	NM	ZE	PS	PM
ZE	ZE	ZE	ZE	PS	PS
ins_2 PS	X	X	X	X	X
PM	X	X	X	X	X
PL	X	X	X	X	X

Table 8

Notice that the flux and speed loops are activated when the speed increases.

ω_3 \ $dins_3$	NM	NS	ZE	PS	PM
ZE	PL	PL	PL	ZE	ZE
ins_1 PS	X	X	X	X	X
PM	X	X	X	X	X
PL	X	X	X	X	X

Table 9

Finally, table 10 shows the values for the weight ω_5.

ω_5 \ ins_3	NM	NS	ZE	PS	PM
NM	PL	PL	PL	PL	PL
$dins_2$ NS	PL	PL	PM	PL	PL
ZE	X	X	PS	X	X
PS	PM	PM	PM	PM	PL
PM	PM	PM	PM	PM	PL

Table 10

A standard Max-Min method and the Mamdani's interpretation of implication are used to solve the fuzzy relational equations [14]. A centroide method is applied to produce the numerical weight values.

4 Simulation Results

The proposed two-layer controller was verified by a computer simulation. The values of booth the stator flux and the rotor speed were assumed available from either measurements or estimation. The results obtained corresponding to selected scenario of T_L and initial conditions of the motor are illustrated in Figure 6 which shows the full scenario of the steady-state and recovery phases of the normal mode and the emergency mode. During the time period [10.0; 10.2] (sec) the induction motor is overloaded.

5 Conclusions

An integration of adaptive GPC and fuzzy logic supervisory control into two-layer controller is proposed to control and induction motor under full range of operat-

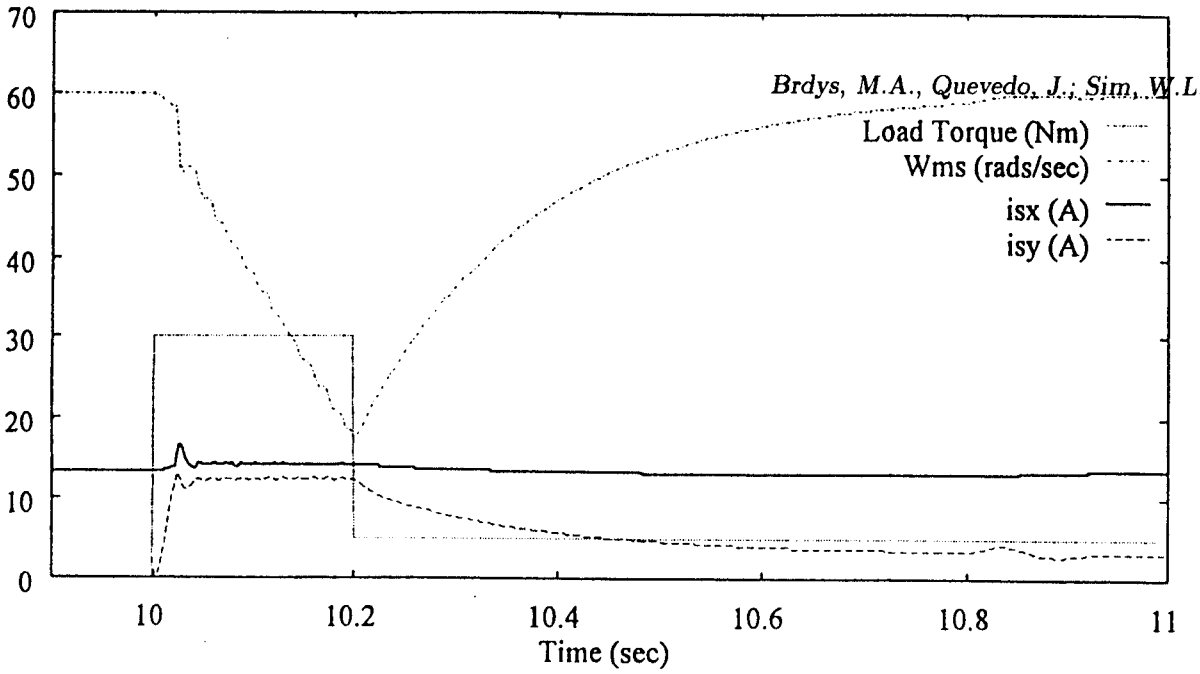


Figure 6

ing conditions. The simulation results confirm feasibility of the approach. Full fuzzification of all input and output variables which would further improve the controller performance particularly during changes of the controlled variable structure is under research.

Acknowledgment

The paper has been prepared during first author sabbatical leave at UPC, which has been supported by the 'Dirección General de Investigación Científico y Técnica (Ref. SAB94-0238). The support is greatly appreciated.

References

[1] Blaschke, F., 1972. Siemens-Rev. 39, 217-220.

[2] Du, T. and Brdys, M.A., *5th EPE Conference*, 1993.

[3] Brdys, M. A., Ogonowski, Z. and Swider, Z., *ECC'95*, 1995.

Brdys, M. A. and Sim, W. L., *6th European Power Electronics Conference*, 1995.

[4] Clarke, J. W., Mohtadi, C. and Tuffs, P. S., *Automatica*. 23, 137-160, 1987.

[5] Krzeminski, Z., *10th IFAC World Congress*, 349-354, 1987.

[6] Marino, R., Persada, S. and Valigi, P., *IEEE Trans. on Aut. Contr.* 38, 1993.

[7] Vas, P.,. Vector Control of AC Machines, *Claredon Press*, Oxford, UK, 1990.

[8] Ortega, R., Canudas, C. and Seleme, S. J., *IEEE Trans. on Aut. Contr.*, 38, 1675-1680, 1993.

[9] Vas, P., Chen, J. and Stronach, A. J., *IEE Control'94 Conference*, 1501-1506, 1994.

[10] Sayeed, A. M., Zinger, D. S., and Elbuluk,. *IEEE Trans. Ind. Appl.*, 30, 1994.

[11] Brdys, M. A., and Schär, T., 1995. Adaptive Fuzzy Logic Control of Induction Motor, *Research Report*, School of EE, The University of Birmingham.

[12] Ollero, A., Garcia-Cerézo, A. and Martinez, J., *IFAC Symposium on Intelligent Autonomous Vehicles*, Southampton, 1993.

[13] Mamdani, E.H. and Assilan,. An experiment in linguistic synthesis with a fuzzy logic controller, *Int. Journal of Man-Machine Studies*, 7 (1), 1-13, 1975.

Controlador Borroso con Aprendizaje mediante la Evolución de su Base de Conocimiento

Luis Magdalena y **Félix Monasterio**
ETSI Telecomunicación
Universidad Politécnica de Madrid
e-mail: llayos@mat.upm.es,fmh@tat.upm.es

Resumen

Los Controladores Borrosos son sistemas basados en conocimiento, teniendo como uno de sus componentes fundamentales una Base de Conocimiento constituida, entre otros elementos, por una Base de Reglas y por un conjunto de Funciones de Pertenencia. La definición del contenido de la Base de Reglas, y de las expresiones de las Funciones de Pertenencia, representa un problema clave en el diseño de Controladores Borrosos. Plantear mecanismos que posibiliten la obtención o el ajuste de estos elementos en base a criterios objetivos, constituye por tanto un tema de gran interés para los diseñadores de Controladores Borrosos. El presente trabajo propone un posible enfoque en este sentido, mostrando los resultados obtenidos en una aplicación concreta.

Palabras clave: *Control borroso, Algoritmos genéticos, Aprendizaje.*

1 Introducción

Los Controladores Borrosos ([1]) son sistemas basados en conocimiento, teniendo como uno de sus componentes fundamentales una Base de Conocimiento constituida, entre otros elementos, por una Base de Reglas (BR) y por un conjunto de Funciones de Pertenencia (FPs). Uno de los principales problemas pendientes en el estudio y aplicación de Controladores Borrosos, es la correcta selección de las Funciones de Pertenencia, y la definición de las reglas que constituyen la Base de Reglas. En pocas palabras, la adecuada definición del conocimiento del sistema.

Recientemente se ha producido una cierta tendencia hacia la aplicación en estas tareas de métodos de adaptación y/o aprendizaje, que permitan obtener o modificar tanto las reglas como las FPs. En este sentido se han producido diferentes intentos de incorporación a los sistemas borrosos de técnicas relacionadas con las Redes Neuronales [2] y con los Algoritmos Genéticos [3]. Las ideas que van a desarrollarse a continuación se encuadran dentro de la segunda línea, la simbiosis entre conceptos borroso y conceptos genéticos, aunque posiblemente fuera más adecuado emplear el término *evolutivos*.

La definición de un problema de aprendizaje como el que aquí se está planteando, presenta tres puntos fundamentales a analizar: como representar cada una de las posibles soluciones al problema, como evaluar cada una de las posibles soluciones al problema, y finalmente, como obtener nuevas soluciones a partir de las ya existentes. En los sistemas genéticos, el conocimiento es codificado como una cadena de ceros y unos de dimensión fija. Recientemente ([4]), el concepto de adaptar la información del sistema a las estructuras binarias empleadas por el sistema genético, ha dejado paso a una tendencia en sentido inverso: modificar los operadores genéticos para poder mantener la estructura de datos del sistema. La sección 3 tratará la codificación del contenido de la Base de Conocimiento, ya que es este contenido el que representa las posibles soluciones. La sección 4 describirá el método empleado para obtener nuevas soluciones a partir de las existentes, dejando el tema de la evaluación para ser abordado en relación con una aplicación concreta (sección 5).

2 Control Borroso y Algoritmos Genéticos

Los algoritmos genéticos [5] son métodos de búsqueda adaptativa, caracterizados por:

1. Un esquema de codificación de las posibles soluciones al problema en forma de cadena finita de bits. Cada cadena (solución) recibe el nombre de cromosoma, y cada bit el de gen.
2. Una función de evaluación que permita caracterizar cada una de las soluciones que constituyen el conjunto de soluciones o población, en base a un criterio de calidad, coste, bondad, etc.
3. Un conjunto inicial de soluciones al problema (población inicial, G(0)).
4. Un conjunto de operadores que empleando la información contenida en una cierta población (definida como generación G(t)) y una serie de parámetros de aprendizaje, creen una nueva población (la siguiente generación, G(t+1)).
5. Un criterio de finalización que nos permita definir en que momento debe detenerse el proceso.

Los operadores genéticos básicos son tres: reproducción, cruce y mutación. La reproducción consiste en la obtención de un conjunto de progenitores, generados mediante copias de individuos de *G(t)*, sobre los que posteriormente actuarán los restantes operadores. Los individuos de *G(t)* que han obtenido una mejor

evaluación (fuerza), generan un mayor número de copias en el conjunto de progenitores. El cruce, mezcla los atributos de los distintos progenitores representados por las correspondientes cadenas binarias. La mutación es la alteración ocasional de un gen en una cierta posición de la cadena.

La bibliografía presenta diferentes enfoques a la hora de aplicar Algoritmos Genéticos a Controladores Borrosos, algunos de los cuales son comentados en [3]. Cada uno de ellos plantea su propio esquema de codificación, función de evaluación, y composición de la población inicial. En unos casos los algoritmos genéticos se dirigen a la modificación de las Funciones de Pertenencia, mientras que en otros trabajan sobre la Base de Reglas, o incluso, sobre ambos elementos de forma simultánea.

Las características de los métodos planteados no parecen ajustarse a problemas en los que la dimensión del espacio de decisión sea relativamente grande. El presente enfoque trabaja de forma simultánea sobre la Base de Reglas y sobre las Funciones de Pertenencia. Emplea BRs expresadas en forma de conjuntos de reglas para conseguir una reducción del volumen de información manejado. A su vez, las FPs se modifican a través de los límites de normalización de las variables, reduciendo de esta forma la dimensión del espacio de búsqueda.

3 Codificando la Base de Conocimiento

El sistema de aprendizaje va a ser aplicado sobre Controladores Borrosos cuya BR se define mediante un conjunto de reglas. Las variables de entrada y salida del controlador se normalizarán linealmente en el intervalo [-1,1] (partiendo de su rango real $[v_{min}, v_{max}]$). Las FPs tendrán forma trapezoidal definida a partir de cuatro parámetros.

La información contenida en la Base de Conocimiento del Controlador Borroso se estructura normalmente en dos unidades fundamentales, la Base de Datos (BD) y la Base de Reglas.

3.1 Codificación de la Base de Datos

Desde el punto de vista de la codificación, y en relación con el proceso de aprendizaje, la BD contiene tres informaciones diferentes: un conjunto de parámetros del controlador, un conjunto de límites de normalización para las diferentes variables y la descripción de los diferentes conjuntos borrosos.

El conjunto de parámetros se encarga de definir las dimensiones del sistema: número de variables de entrada (N) y salida (M), y a través de dos vectores (n y m) el número de variables lingüísticas (conjuntos borrosos) asociadas a cada una de las variables de entrada y salida del sistema. La componente i-ésima del vector n ($n = \{n_1, \ldots, n_N\}$) representa el número de variables lingüísticas asociadas a la variable i-ésima de entrada. La j-ésima componente del vector m ($m = \{m_1, \ldots, m_M\}$) indica el número de variables lingüísticas asociadas a la j-ésima variable de salida.

El conjunto de límites de normalización queda representado por una matriz de $(N + M) \times 2$ números reales. Cada fila de esta matriz contiene los límites de normalización de una variable de entrada o salida del sistema ($\{v_{min}, v_{max}\}$).

El número total de conjuntos borrosos definidos es igual a L:

$$L_a = \sum_{i=1}^{N} n_i \ , \ L_c = \sum_{j=1}^{M} m_j \ , \ L = L_a + L_c. \quad (1)$$

Cada conjunto borroso queda definido mediante una FP de tipo trapezoidal, caracterizada mediante cuatro parámetros. De esta forma las funciones de pertenencia que definen los L conjuntos borrosos del controlador, dan lugar a una matriz de $L \times 4$ números reales cuyo valor está comprendido en el intervalo [-1,1] como consecuencia de la utilización de variables normalizadas. Cada una de las filas de la matriz define una FP.

3.2 Representación de la Base de Reglas

La Base de Reglas va a quedar definida mediante un conjunto de reglas borrosas combinadas a través de la conectiva *also* [1]:

$$R = \ also\ (R_1, R_2, \ldots, R_k).$$

La estructura de las reglas de control, considerando los anteriores parámetros $\{N, M, n, m\}$, es:

$$\begin{array}{ll} \textbf{Si} & x_i \text{ es } C_{io} \text{ and } \ldots \text{and } x_k \text{ es } C_{kp} \\ \textbf{entonces} & y_j \text{ es } D_{jq} \text{ and } \ldots \text{and } y_l \text{ es } D_{lr} \end{array} \quad (2)$$

siendo x_i e y_j variables de entrada y salida respectivamente, y C_{io} y D_{jq}, conjuntos borrosos asociado con dichas variables. Múltiples conjuntos borrosos asociados a una misma variable pueden ser agrupados en una misma regla: **Si** $x_i$es$(C_{io}$ or $C_{ip})\ldots$**entonces** $y_j$es$(D_{jq}$ or $D_{jr})\ldots$

Las tablas de decisión generadas por controladores con múltiples variables de entrada, son multidimensionales. El número de reglas codificadas (L_r) en una tabla de este tipo, puede ser obtenido a partir del previamente definido vector n:

$$L_r = \prod_{i=1}^{N} n_i \quad (3)$$

A cada una de estas reglas la denominaremos regla elemental.

Una regla elemental debe contener entradas borrosas para cada una de las variables de entrada, y en ninguna de ellas podrá aparecer el operador 'or'. La regla descrita en (2) no es una regla elemental. Una regla que contenga dos conjuntos borrosos correspondientes a una misma variable, y asociados mediante el operador 'or', es equivalente a dos reglas elementales, siempre que la definición de la conectiva 'also' y del operador 'or' sean similares. Una regla en la que no aparece una determinada variable de entrada, representa la agrupación de tantas reglas elementales como conjuntos borrosos se asocian con dicha variable. La regla equivaldrá a la agrupación de dichas reglas elementales, si además de la condición impuesta anteriormente, se satisface que la unión de todos los conjuntos borrosos asociados a la variable da lugar a un conjunto borroso cuya función de pertenencia es siempre uno. El número de reglas elementales que

son reemplazadas por una determinada regla se obtiene multiplicando el número de reglas reemplazadas en cada una de las variables. Por ejemplo, si se considera un Controlador Borroso con los siguientes parámetros $N = 3, M = 1, n = \{5, 3, 5\}$ y $m = \{7\}$ ($L_a = 13, L_c = 7$), la regla

$$\begin{array}{ll} \textbf{Si} & x_1 \text{ es } (C_{13} \text{ or } C_{14}) \text{ and } x_3 \text{ es } (C_{31} \text{ or } C_{32}) \\ \textbf{entonces} & y_1 \text{ es } (D_{14} \text{ or } D_{15}) \end{array} \tag{4}$$

reemplaza a un total de $2 \times 3 \times 2 = 12$ reglas elementales.

En sistemas con un reducido número de variables de entrada, el empleo de conjuntos de reglas en lugar de tablas de decisión puede no suponer ninguna mejora; pero en aquellos casos en los que el número de variables de entrada es elevado, el efecto es notable. Cuando el número de variables de entrada aumenta, el número de reglas elementales contenidas en la tabla de decisión aumenta de forma exponencial. Por otro lado, si se analiza el tipo de reglas generadas por los expertos a la hora de definir un Controlador Borroso, se puede comprobar como por lo general, estas reglas sólo contienen un subconjunto de las variables de entrada al sistema (normalmente no más de cuatro o cinco variables por regla), tal y como se muestra en [6, 7]. Como consecuencia, el número de reglas elementales reemplazadas por cada regla del conjunto, crece también de forma exponencial. El resultado, es que la proporción de crecimiento en el conjunto de reglas, es muy inferior a la presentada por la tabla de decisión.

3.3 Codificación de la Base de Reglas

En nuestro sistema, cada regla queda codificada a través de dos cadenas binarias: la primera encargada de codificar el antecedente y cuya longitud es L_a, y la segunda codificando el consecuente y con una longitud L_c. La codificación del antecedente se inicia con una cadena de longitud L_a, y cuyos bits toman valor inicial *0*. Si el antecedente de la regla contiene una condición del tipo x_i es C_{ij}, situaremos un *1* en la posición p de la cadena:

$$p = j + \sum_{k=1}^{i-1} n_k. \tag{5}$$

Este proceso se repetirá para cada una de las condiciones de la premisa. La codificación del consecuente se realiza de forma similar, tomando m en lugar de n.

Según este esquema de codificación, una regla como la descrita en la expresión 4 es codificada como

$$0011000011000 - 0001100. \tag{6}$$

Cada regla de un controlador tal quedará representada por una cadena de 20 bits (longitud fija). La BR contendrá un número indeterminado de éstas, hasta un número máximo de $L_r = 75$, codificándose en consecuencia la BR como una cadena de hasta 75 elementos, cada uno de ellos con un total de 20 bits.

3.4 El código

El código que refleja el contenido de la Base de Conocimiento, puede resumirse en los siguientes puntos:

1. Una cadena de $2 + N + M$ números enteros que reflejan las dimensiones del controlador.
2. Una cadena de $(N + M) \times 2$ números reales que definen los límites de normalización de las diferentes variables del sistema.
3. Una cadena de $(L_a + L_c) \times 4$ números reales, reflejando las funciones de pertenencia de cada uno de los conjuntos borrosos definidos.
4. Una cadena de hasta L_r reglas, en la que cada regla es a su vez una cadena de $L_a + L_c$ bits.

4 Los operadores evolutivos

Aunque se podrían aplicar mecanismos evolutivos sobre cada uno de los fragmentos de código anteriormente descritos, a partir de este momento sólo trabajaremos con los límites de normalización (2) y la BR (4). Los efectos producidos mediante la modificación de los límites de normalización, están limitados frente a los que se podrían conseguir si se modificasen las FPs, pero a cambio se reduce considerablemente la longitud del código empleado, acelerando el proceso de aprendizaje. Los operadores de evolución que se van a describir son básicamente adaptaciones de los operadores genéticos clásicos al código empleado.

4.1 Reproducción

El operador de reproducción arranca con la aplicación de un mecanismo de élite que puede quedar definido en base a un número de miembros, un porcentaje o un proceso de umbralización. Este mecanismo permite que un subconjunto de $G(t)$, definido como la élite de la generación t ($E(t)$), sea directamente copiado o reproducido en $G(t+1)$.

A continuación, ciertos individuos de $G(t)$ son seleccionados como progenitores en base a un criterio probabilístico definido a partir de la fuerza de cada individuo. Siguiendo el método clásico, una mayor fuerza queda asociado con un mayor número de copias en el conjunto de reproductores. Junto a este operador clásico se ha definido un operador modificado, en base al cual, la mayor fuerza define mayores probabilidades de incluir una copia en el conjunto de reproductores, no generándose más de un reproductor a partir de un mismo miembro de la población.

Una vez finalizada la selección de la élite y de los reproductores, el número de miembros de la nueva generación deberá ser ajustado a la población máxima, mediante la inclusión de nuevos individuos en la élite, o la eliminación de progenitores.

4.2 Cruce

El cruce consiste en la combinación de las Bases de Datos y Reglas de dos progenitores ($x_i = (r_i, d_i)$ y $x_j = (r_j, d_j)$), para generar dos nuevas Bases de Conocimiento (x_u, x_v). El proceso de cruce se inicia por las BRs.

$$\begin{array}{ll} r_i = & \{r_{i1}, \ldots, r_{ik}\} \\ r_j = & \{r_{j1}, \ldots, r_{jl}\} \end{array} \tag{7}$$

En primer lugar se definen puntos de corte sobre las cadenas de reglas de cada uno de los progenitores. Tomando los puntos de corte en las posiciones α y γ, se obtiene

$$\begin{array}{ll} r_i = & \{r_{i1}, \ldots, r_{i\alpha} | r_{i\alpha+1}, \ldots, r_{ik}\} \\ r_j = & \{r_{j1}, \ldots, r_{j\gamma} | r_{j\gamma+1}, \ldots, r_{jl}\} \end{array} \tag{8}$$

Resultando tras la realización de los cruces

$$\begin{array}{rl} r_u = & \{r_{i1},\ldots,r_{i\alpha}|r_{j\gamma+1},\ldots,r_{jl}\} \\ r_v = & \{r_{j1},\ldots,r_{j\gamma}|r_{i\alpha+1},\ldots,r_{ik}\} \end{array} \quad (9)$$

Es importante recordar que aunque las nuevas reglas son copias de las existentes en los progenitores, su significado va a verse modificado en función de las nuevas BDs que se creen. Esta dependencia ha inducido a tratar de otorgar cierta predominancia a los elementos de la BD cuyas reglas aporten mayor contenido al nuevo controlador. La determinación de las aportaciones realizadas por r_i y r_j, a r_u y r_v, se ha analizado variable a variable, seleccionando por tanto para cada variable la información contenida en d_i o d_j. La obtención de los elementos de d_u correspondientes a cada una de las diferentes variables del sistema, se inicia con el cálculo de dos enteros (v_{hi} y v_{hj}) que representan el número de veces que aparece la variable v_h en las reglas aportadas a r_u por parte de r_i y r_j respectivamente. La relación entre estos dos enteros refleja la probabilidad de emplear en d_u, la información referida a la variable v_h procedente de d_i o de d_j. Este proceso se repite de forma independiente para obtener d_v, de forma que ambas Bases de Datos pueden duplicar parte de la información contenida en d_i o d_j, descartando la parte equivalente de la base contraria.

4.3 Reordenación de reglas

La reordenación de una BR (r_i) modifica el orden en que las reglas de dicha Base se sitúan en la cadena genética, determinando un punto de corte (β) e invirtiendo la posición de los fragmentos generados para dar lugar a una nueva BR (r_u):

$$\begin{array}{rl} r_i = & \{r_{i1},\ldots,r_{i\beta}|r_{i\beta+1},\ldots,r_{ik}\} \\ r_j = & \{r_{i\beta+1},\ldots,r_{ik}|r_{i1},\ldots,r_{i\beta}\} \end{array} \quad (10)$$

Mediante este operador se hace posible obtener diferentes agrupaciones de reglas en cruces posteriores. El operador no afecta a la Base de Datos.

4.4 Mutación

La mutación puede afectar a los límites de normalización (rangos), o a la Base de Reglas. Los rangos son pares de números reales que determinan los extremos superior e inferior del intervalo de normalización de cada una de las diferentes variables. Sobre dichos extremos actúa el operador de mutación, produciendo desplazamientos proporcionales a la amplitud del rango, y a un factor aleatorio de distribución uniforme. A la hora de aplicar el desplazamiento sobre los límites de normalización, es preciso considerar la existencia de dos tipos de variables diferenciadas, unas en las que las propiedades de la variable se reflejan en comportamientos simétricos respecto del origen, y otras en las que los rangos no tienen por que ser simétricos. A la hora de desplazar los límites de normalización, se mantendrán simétricos aquellos rangos que previamente lo fueran.

Reuniendo todo ello en una única expresión, y tomando una variable cuyo rango sea $[\lambda_i, \lambda_s]$, la mutación se ajusta a la siguiente definición:

$$\begin{array}{rl} \lambda_i(t+1) = & \lambda_i(t) + KP_1S_1(\lambda_s(t) - \lambda_i(t))/2 \\ \lambda_s(t+1) = & \lambda_s(t) + KP_2S_2(\lambda_s(t) - \lambda_i(t))/2 \end{array} \quad (11)$$

donde $P_1 = P_2$ y $S_1 = -S_2$ cuando $\lambda_i(t) = -\lambda_s(t)$ (rangos simétricos). P_1 y P_2 son valores aleatorios generados en el intervalo $[0, 1]$ con distribución uniforme, S_1 y S_2 toman (equiprobablemente) los valores -1 ó 1, y K es un parámetro del mecanismo de aprendizaje comprendido en el intervalo $[0, 1]$, y que determina la proporción máxima de variación producida por una mutación.

El operador de mutación aplicado a la BR actúa sobre las cadenas de ceros y unos que constituyen las reglas, que a su vez son un elemento de la lista que define la Base de Reglas.

Dado un cierto controlador, una regla cualquiera del mismo estará compuesta por una cadena de $L_a + L_c$ ceros y unos, que podemos representar como:

$$\begin{array}{l} p_{11}\ldots p_{1n_1},\ldots,p_{N1}\ldots p_{Nn_N} \\ c_{11}\ldots c_{1m_1},\ldots,c_{M1}\ldots c_{Mm_M} \end{array} \quad (12)$$

Según este esquema, la cadena global que constituye la regla queda dividida en $N + M$ subcadenas asociadas con cada una de las variables (entrada y salida). Cada una de las subcadenas (gen) es un candidato sobre el que aplicar el operador clásico de mutación genética.

5 Aplicación

Se pretende controlar la generación de trayectorias de articulación de una estructura bípeda bidimensional con cinco grados de libertad actuados. El objetivo es la consecución de una marcha continuada y estable por parte de dicha estructura.

5.1 El sistema bípedo simulado

El sistema simulado es una estructura bidimensional de seis barras, con dos piernas, una cadera y un tronco. Cada pierna presenta un grado de libertad en su rodilla, y un segundo grado de libertad respecto de la cadera. El tronco presenta un grado de libertad respecto de la cadera. Los pies son puntuales. Las dimensiones del modelo son las definidas en la TABLA 1. Se definen en la figura 1 las variables empleadas por el Controlador Borroso (α_1-α_6), y por el modelo matemático ([3]) sobre el que se ha probado el controlador (δ_1-δ_6), relacionadas ambas por la expresión:

$$\vec{\delta} = [R]\vec{\alpha} + \vec{r}. \quad (13)$$

La ecuación dinámica del modelo queda definida por la ecuación diferencial no lineal, de orden dos:

$$\vec{\tau_\delta} = D(\delta)\ddot{\vec{\delta}} + C(\dot{\delta}, \delta)\dot{\vec{\delta}} + \phi(\delta), \quad (14)$$

representando τ las fuerzas generalizadas. Considerando que $\vec{\tau_\alpha} = [R]^T\vec{\tau_\delta}$ e imponiendo la presencia en el apoyo de un grado de libertad no actuado, mediante $\tau_{\alpha_1} = 0$, se obtiene la ecuación diferencial

$$A_1\ddot{\delta_1} + A_2(\dot{\delta_1})^2 + A_3\dot{\delta_1} + A_4\cos\delta_1 + A_5\sin\delta_1 + A_6 = 0 \quad (15)$$

que es resuelta numéricamente mediante el método de Runge-Kuta con un intervalo de integración de 10^{-4} segundos. Es importante resaltar que los coeficientes A_1 a A_6 son funciones de $\delta_2, \dot{\delta_2}, \ddot{\delta_2}, \ldots, \delta_6, \dot{\delta_6}, \ddot{\delta_6}$.

Tabla 1: DIMENSIONES DEL MODELO

Elemento	Longitud (m)	Peso (kg)
Tronco	0.65	35
Cadera	0.10	7
Muslo	0.50	9
Tibia	0.50	5

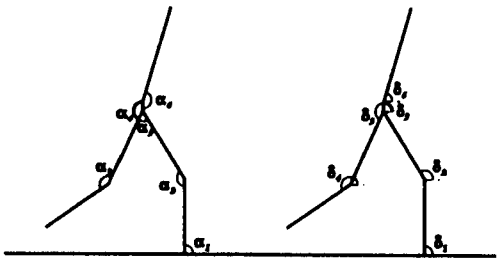


Figura 1: Variables del controlador y del modelo.

5.2 El Controlador Borroso

El Controlador Borroso empleado aplica el operador composicional sup-min, la unión para combinar las salidas borrosas de las diferentes reglas (operador 'also'), y el centro de masas para obtener la salida numérica. Sus parámetros son: 22 variables de entrada (N), 15 variables de salida (M), 9 conjuntos borrosos por variable (n_i, m_j) y un ciclo de control de 0.01 segundos.

Las 22 *variables de entrada* son: las posiciones, velocidades y aceleraciones de cada uno de los grados de libertad del sistema, incluyendo el no actuado (α_1 a α_6 en la figura 1) así como de la componente horizontal del centro de gravedad de la estructura mecánica (CG_x), además se define una variable binaria que indica si el apoyo es simple (un pie) o doble (ambos pies). Las 15 *variables de salida* son: las posiciones, velocidades y aceleraciones de cada uno de los grados de libertad actuados del sistema (α_2 a α_6). Las variables se normalizan en el intervalo [-1,1]. Las variables que son simultáneamente de entrada y salida, emplean los mismos rangos en la normalización de la entrada y la desnormalización de la salida.

5.3 Conocimiento inicial

El sistema emplea nueve conjuntos borrosos trapezoidales, similares, distribuidos regularmente en el intervalo [-1,1], e idénticos para todas las variables. Estos nueve conjuntos son los que aparecen en la figura 2. Los rangos iniciales para las variables articulares se han obtenido a partir de las limitaciones físicas de cada una de ellas y de la información extraída de estudios biomecánicos [8]. Igual método se ha seguido con las reglas iniciales, obteniendo reglas como la descrita a continuación:

Si *El apoyo* es *Simple*
y *La posición del centro de gravedad* (CG_x) es *Negativa* ($C_1 or C_2 or C_3 or C_4$)
y *La posición de la cadera en la pierna de apoyo* (α_3) es *Positiva* ($C_6 or C_7 or C_8 or C_9$)
y *La velocidad de la cadera en la pierna de apoyo* ($\dot{\alpha}_3$) es *Positiva o Nula* ($C_5 or C_6 or C_7 or C_8 or C_9$)
entonces *La aceleración de la cadera en la pierna de apoyo* ($\ddot{\alpha}_3$) debe ser *Negativa extrema* (C_1)

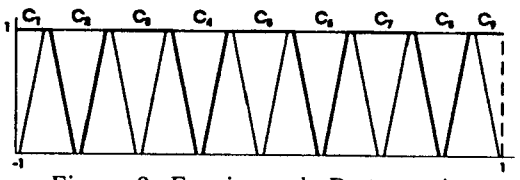


Figura 2: Funciones de Pertenencia.

5.4 Evaluación

Una vez codificadas una serie de Bases de Conocimiento obtenidas tal y como se describe en el apartado anterior, se aplicarán los mecanismos de aprendizaje tomando dichas Bases de Conocimiento como conocimiento inicial del sistema evolutivo, constituyendo cada una, una Descripción de Paso. Estas serán evaluadas mediante simulación sobre el modelo matemático previamente descrito, analizando la estabilidad (caída o no) y regularidad de los pasos (desviación típica en longitud y frecuencia de paso) tras diez segundos de simulación (diez mil ciclos de control). Algunas descripciones de paso darán lugar a marchas correctas, siendo caracterizadas en función de los parámetros fundamentales del paso generado (velocidad de avance, longitud de paso y frecuencia de paso), otras finalizarán con caídas o paradas del sistema simulado. El objetivo del proceso de aprendizaje es obtener descripciones que den lugar a marchas continuadas (sin parada ni caída), y regulares (mínima desviación típica). Un segundo objetivo es la diversidad de las descripciones obtenidas, en lo referente a velocidad, longitud y frecuencia de paso (sólo dos de estos tres parámetros son independientes).

6 Resultados

A continuación se describen algunos de los resultados obtenidos a partir de una población inicial de 27 individuos (5 descripciones validas y 22 que finalizan en caída). La gráfica representa la distribución de los pasos válidos obtenidos (cada cuadrado representa una Base de Conocimiento) sobre el espacio de parámetros, definido éste a través de la relación entre la longitud de paso (m) y la velocidad de avance (m/sg) obtenidas en la simulación.

Los resultados presentados en la figura 3 corresponden a las descripciones de paso obtenidas tras 41 generaciones de población máxima 500 individuos, empleando una proporción de cruces de 0.8, porcentaje de mutaciones 0.05, y fijando el parámetro K de la ecuación (11) en 0.5. Los cuadrados rellenos definen a los cinco pasos válidos que aparecían en la población inicial. En [3] se presentan otros resultados obtenidos, incluyendo los que se han logrado sin aplicar los operadores de mutación.

7 Conclusiones

El presente trabajo propone una nueva aproximación al problema de aprendizaje en Controladores Borrosos, que es analizado mediante su aplicación a un complejo problema de control: la síntesis de la marcha para una estructura bípeda bidimensional.

Los resultados obtenidos demuestran la capacidad del método propuesto para generar nuevo conocimiento. Partiendo de un conjunto de Descripciones de Paso con velocidades en el intervalo [1.05,1.15] m/sg, se

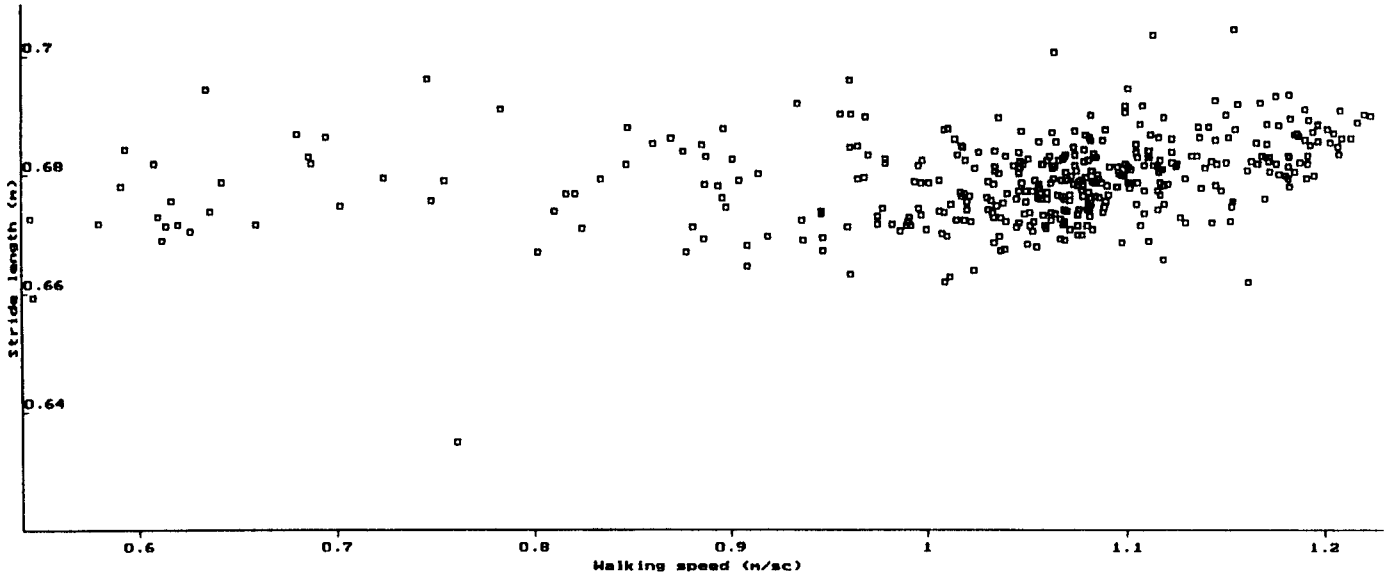


Figura 3: Población final representada en el espacio de parámetros de la marcha.

consigue cubrir el intervalo [0.55,1.23] (figura 3). Así mismo, la figura 4 muestra dos descripciones de paso obtenidas a lo largo de este mismo proceso de aprendizaje, y que presentan longitudes de paso de 0.64 y 0.78 m (esta descripción se ha generado en el proceso citado, pero no se incluye en la figura 3 por encontrarse muy alejada del resto) cuando todas las descripciones iniciales se situaban en el margen [0.67,0.68]. En cuanto a las evaluaciones obtenidos por los diferentes pasos generados, partiendo de una población inicial cuya mejor evaluación era 0.9491 (sobre uno) siendo 0.8 la siguiente evaluación obtenida, la población final reflejada en la figura 3 contiene una descripción con evaluación 0.9817, y otras cincuenta (distribuidas en el rango de velocidades [0.88,1.23]) evaluadas por encima de la mejor evaluación de la población inicial.

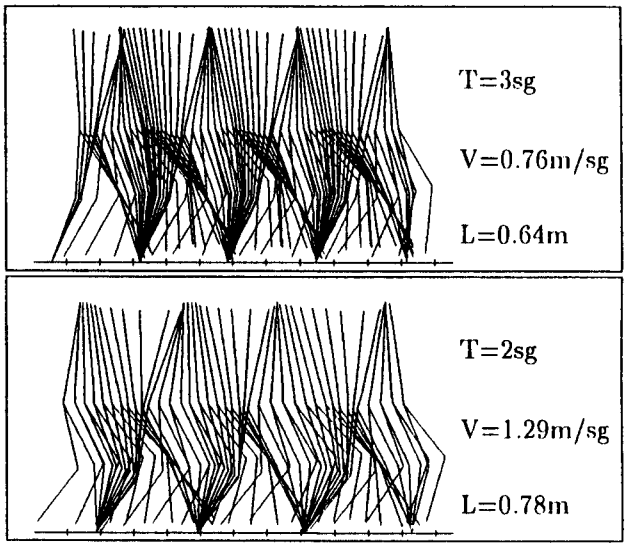


Figura 4: Dos descripciones con diferente longitud.

Agradecimientos

El presente trabajo ha sido parcialmente subvencionado por la CICYT, con referencia TAP93-0971-E.

Referencias

[1] Lee, C.C., Fuzzy logic in control systems: Fuzzy logic controller - part I and II. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2):404–435, Mar/Apr 1990.

[2] Magdalena, L., A first approach to a taxonomy of fuzzy-neural systems. In *IJCAI'95 Workshop on Connectionist-Symbolic Integration: From Unified to Hybrid Approaches*, 1995.

[3] Magdalena, L., *Estudio de la coordinación inteligente en robots bípedos: aplicación de lógica borrosa y algoritmos genéticos*. Tesis doctoral, Universidad Politécnica de Madrid, 1994.

[4] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1992.

[5] DeJong, K. Learning with genetic algorithms: An overview. *Machine Learning*, 3(3):121–138, October 1988.

[6] Magdalena, L., Velasco, J.R., Fernández, G., and Monasterio, F., A control architecture for optimal operation with inductive learning. In *Preprints IFAC Symposium on Intelligent Components and Instrument for Control Applications, SICICA'92*, pp 307–312, May 1992.

[7] Magdalena, L., and Monasterio, F., Fuzzy controlled gait synthesis for a biped walking machine. In R.J. Marks II, editor, *Fuzzy Logic Technology & Applications*, IEEE Technology Update Series, pp 117–122. IEEE Press, 1994.

[8] Plas, F., Viel, E., et Blanc, Y., *La marche humaine*. MASSON, S.A., Paris, 1984.

LA LÓGICA BORROSA EN CONTROL DE SISTEMAS AUTÓNOMOS INTELIGENTES

Fernando Matía Espada
Dpto. Automática, Ingeniería Electrónica
e Informática Industrial (DISAM)
E.T.S.I.Industriales
Universidad Politécnica de Madrid
e-mail: matia@disam.upm.es

Eduardo Moraleda Gil
Dpto. Automática, Ingeniería Electrónica
e Informática Industrial (DISAM)
E.T.S.I.Industriales
Universidad Politécnica de Madrid
e-mail: moraleda@disam.upm.es

Eugenio Andrés Puente
Dpto. Automática, Ingeniería Electrónica
e Informática Industrial (DISAM)
E.T.S.I.Industriales
Universidad Politécnica de Madrid
e-mail: puente@disam.upm.es

Manuel Escriche Vicente
Dpto. Automática, Ingeniería Electrónica
e Informática Industrial (DISAM)
E.T.S.I.Industriales
Universidad Politécnica de Madrid
e-mail: mescriche@disam.upm.es

Resumen

De entre las diversas aplicaciones de la lógica borrosa a control que vienen desarrollándose en nuestro Departamento de Universidad durante los últimos años, como son el control de procesos continuos o la validación de sensores, este trabajo presenta una más: el control de un vehículo autónomo.

Se utiliza una arquitectura de control reactivo como punto de partida, implementándose controladores borrosos para el seguimiento de trayectorias y supervisores borrosos para la toma de decisiones en tiempo real.

La innovación respecto a trabajos anteriores consiste en la fusión de la información procedente de distintos tipos de sensores para mejorar la robustez del control. La coordinación de la navegación del vehículo se logra mediante una satisfactoria integración de los sistemas de control, planificación y percepción sensorial.

Palabras Clave: Robótica Móvil, Control Reactivo, Percepción Sensorial, Aplicaciones.

1 Aplicaciones de la lógica borrosa a control

Desde la aparición de la lógica borrosa, no tarda mucho en surgir la idea de aplicarla a control. En concreto, es un artículo de Bellman y Zadeh [2] el que lanza la idea casi sin saberlo. Desde entonces, son numerosos los grupos que a nivel mundial se dedican a esta (entonces) prometedora rama del control. De hecho, durante años, los resultados obtenidos fueron siempre empíricos y el objetivo de la investigación se centró en la búsqueda de una teoría que permitiera configurar este tipo de sistemas. Y en los últimos años, si bien es cierto que aún no puede hablarse de una teoría de control, sí existen ya las bases necesarias para conseguirlo.

Algo similar ha sucedido en DISAM, donde, de una etapa inicial de aplicaciones industriales [6], se ha pasado a un conocimiento bastante profundo de los sistemas borrosos [10]. Entre las distintas aplicaciones que hemos desarrollado para la industria, pueden citarse:

- El control de procesos continuos: en el proyecto CONEX desarrollado para ASLAND S.A., para el control de un horno de clinker de cemento (1990-1992).

- La validación de datos: en el proyecto ESPRIT 6447 HINT, para la validación de sensores en una refinería de petróleo, cuyo demostrador fue instalado por REPSOL (1992-1994) [9].

- La herramienta de desarrollo FFV-TOOL: en los proyectos MICS (CICYT) y HINT (ESPRIT), para el filtrado, validación de sensores y control mediante lógica borrosa.

2 Control de vehículos autónomos

Asimismo, la robótica móvil es otra de las grandes áreas en las cuales la lógica borrosa ha tenido siempre gran aceptación. DISAM viene trabajando en el desarrollo de vehículos autónomos y en sus aplicaciones industriales durante los últimos 13 años. Desde los primeros experimentos con el prototipo VAD-1 [13], hasta hoy con el vehículo ROBUTER de Robosoft (ver Figura 1), la investigación ha evolucionado mucho.



Figura 1: Plataforma móvil de desarrollo.

Se trata de sistemas autónomos, pues no tienen dependencia externa ni en cuanto a suministro de energía, ni en cuanto a algoritmos de control. Todo el sistema de navegación del vehículo reside sobre un computador a bordo, y se dispone de un sistema de comunicaciones por radio que permite la monitorización de las acciones del vehículo y el envío de nuevas misiones a llevar a cabo.

E inteligentes, puesto que poco a poco se han ido incorporando nuevas técnicas de control, la mayoría procedentes del campo de la inteligencia artificial [4], y cuyo objetivo es dotarles de un alto grado de capacidad de decisión.

En otros trabajos ya hemos mostrado cómo técnicas de control clásico, métodos heurísticos o redes neuronales pueden ser apropiadas para la implementación de algoritmos de control. En este trabajo se muestra cómo es factible y eficiente el empleo de lógica borrosa, presentándose algunas mejoras respecto a trabajos anteriores [11]. Éstas pasan por la utilización de una información sensorial más rica, un supervisor de control más robusto y una más completa integración de los sistemas de planificación, percepción y control.

3 Control reactivo

Como resultado de los últimos años de investigación, se desarrolló una arquitectura propia de control reactivo denominada AFREB (Adaptive Fusion of Reactive Behaviours) [3]. En los sistemas jerárquicos clásicos no es posible hacer frente a situaciones en las cuales las variaciones en el entorno no sean conocidas a priori. En los sistemas reactivos el controlador procesa directamente la información sensorial y reacciona en consecuencia, sin necesidad de disponer de información a priori.

En los sistemas puramente reactivos propuestos por Brooks [7] el control se lleva a cabo en forma de conductas elementales (o primitivas) que se intentan cumplir sin información a priori del entorno:

- Dirigirse a un punto.
- Seguir a una persona.
- Huir de la oscuridad.
- Vagar por el entorno.
- Acercarse hacia regiones calientes.

En nuestro caso hemos ido hacia una arquitectura híbrida, con un módulo de planificación de trayectorias *off-line*, y con conductas primitivas *on-line* para evitar colisiones con obstáculos no previstos en la fase de planificación. Las conductas implementadas son:

- Seguir una trayectoria.
- Dirigirse a un punto.
- Bordear un obstáculo por la derecha.
- Bordear un obstáculo por la izquierda.
- Mantenerse alejado de los obstáculos.
- Buscar espacios abiertos.

Éstas deben ser fusionadas para completar una tarea objetivo:

- Seguir una ruta.
- Alcanzar un destino.

El proceso de fusión consiste en determinar la contribución de cada conducta primitiva al movimiento del vehículo, en función de la misión *off-line* a realizar y de la información sensorial *on-line* disponible. El encargado de hacerlo es el supervisor de control.
El ciclo de control es: sensación -> movimientos elementales -> fusión de movimientos elementales.

En nuestro caso, la fusión se lleva a cabo combinando linealmente los movimientos del vehículo c_i (velocidad

lineal v_i, velocidad angular w_i) propuestos por cada una de las conductas para dar lugar al movimiento real c_0 (v_0, w_0) (comportamiento emergente):

$$v_0 = \sum a_i v_i \quad (1)$$

$$w_0 = \sum a_i w_i \quad (2)$$

En la Figura 2, se muestra el esquema de control reactivo. La misión del supervisor es calcular el valor más adecuado de los pesos a_i.

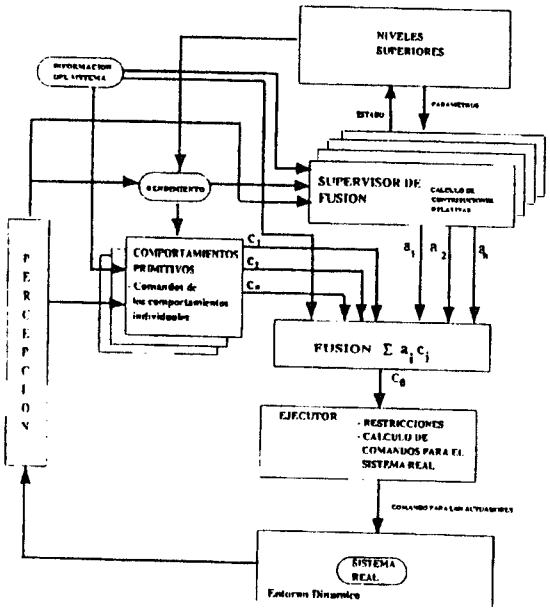


Figura 2: Esquema de control reactivo.

Por ejemplo, si la misión encomendada por el planificador de tareas en seguir una ruta, se dará un peso de 1 a la conducta *seguir una trayectoria* y 0 a las restantes. En el momento en que aparezca un obstáculo imprevisto, el sistema de percepción lo detectará y el supervisor dará más peso a la conducta *bordear el obstáculo*, hasta que el mismo sea rebasado.

4 Control reactivo borroso

La técnica que se utilizó desde un principio en el supervisor, para la determinación de los pesos, fue la lógica borrosa. Asimismo, también se implementó la conducta primitiva seguir una trayectoria mediante lógica borrosa. Entre las ventajas de esta técnica, se encuentran el poder manejar la incertidumbre inherente a la información sensorial de una manera natural, y el permitir establecer unas reglas empíricas con facilidad.

Para la implementación de la conducta, la única información sensorial que emplea es la proveniente de la odometría. Ésta permite disponer de la posición del vehículo dentro del entorno y, por lo tanto, conocer error de posición y orientación respecto al camino deseado. Sin embargo, debido a los grandes errores que proporcionan los sistemas odométricos, los algoritmos presentados no funcionarán correctamente mientras no se complementen con un adecuado sistema de localización.

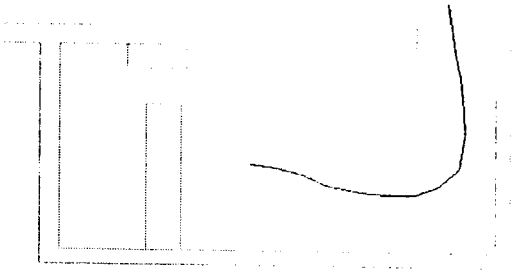


Figura 3: Simulación de la trayectoria.

La experiencia demuestra que un controlador borroso muy sencillo con nueve reglas es suficiente para obtener el comportamiento deseado (Figura 3). Las reglas empleadas son del tipo:

- Si el error es pequeño y su derivada pequeña entonces v_i es grande y w_i es pequeña.
- Si el error es grande y su derivada es pequeña entonces v_i es pequeña y w_i es grande.

A la hora de diseñar el supervisor, hay que considerar que se trata de la parte más crítica de la arquitectura AFREB. Ello se debe principalmente a la necesidad de tomar decisiones con cierto grado de inteligencia, a partir de gran cantidad de información sensorial, pero no por abundante precisa. De hecho, en el caso de utilizar ultrasonidos, dicha información lleva asociada un alto grado de imprecisión e incertidumbre que debe ser tratada adecuadamente. Es por ello que se eligió como arma la lógica borrosa.

El supervisor se diseñó para ejecutar la tarea *seguir una ruta* evitando los obstáculos que encuentren. Dado que se dispone de 24 sensores de ultrasonidos es obligado efectuar un preprocesado de la información con el fin de reducir el número de variables de entrada y que el

número de reglas resulte manejable. Las entradas al supervisor borroso elegidas fueron:

- La trayectoria a seguir.
- La posición actual del robot (odometría).
- El sensor de ultrasonidos que devuelve la distancia mínima.
- Dicha distancia mínima.
- La conducta primitiva dominante en el ciclo anterior.

Con ello, las reglas de fusión son del tipo:

- Si la distancia mínima es muy grande entonces el peso de la conducta *seguir trayectoria* es muy grande y el resto cero.

- Si la distancia mínima es media y el sensor que la proporciona está a la derecha entonces el peso de la conducta *seguir trayectoria* es medio y el de la conducta *bordear obstáculo por la derecha* es medio.

- Si la distancia mínima es pequeña entonces el peso de la conducta *alejarse de los obstáculos* es grande y la del resto pequeño.

El utilizar en cada ciclo únicamente la información sensorial existente en ese instante, no es suficiente para resolver determinadas situaciones geométricas. Ejemplos de esto son los que surgen con obstáculos cóncavos. Para solucionar este problema es necesario introducir también como entrada al supervisor la conducta primitiva dominante en el ciclo anterior:

- Si la distancia mínima es pequeña y el sensor que la proporciona está en la parte delantera y la conducta anterior era bordear el obstáculo entonces el peso de la conducta *bordear el obstáculo* es grande.

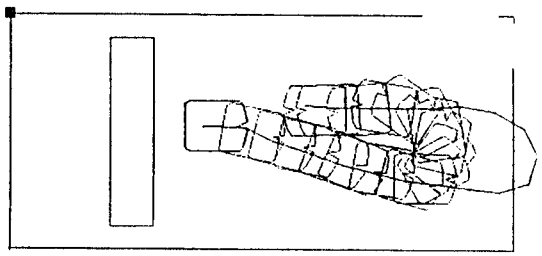


Figura 4: Seguimiento de trayectoria con el robot real.

- Si la distancia mínima es pequeña y el sensor que la proporciona está en la parte delantera y la conducta anterior era seguir la trayectoria entonces el peso de la conducta *alejarse del obstáculo* es grande.

También se diseñó un supervisor para cumplir la tarea *Ir a un destino*. En este caso, la estrategia para el procesamiento de la información sensorial es diferente. En lugar de utilizar únicamente el sensor que da la mínima distancia, se ha dividido el espacio barrido por los sensores en cinco zonas, tomándose la mínima medida en cada una de ellas.

Con ello, se incrementa el número de entradas al supervisor en tres, con lo cual se dispara el número de reglas posibles. A cambio, la información sensorial es mucho más rica.

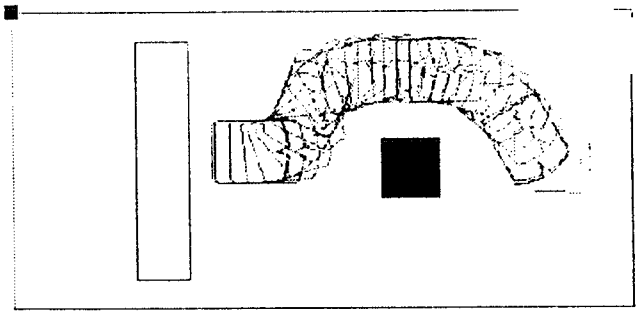


Figura 5. Ir a un destino con el robot real.

5 Mejoras del control reactivo

La utilización exclusiva de sensores de ultrasonidos origina importantes problemas:

- Las medidas que suministran son poco precisas.
- Son sensores muy lentos, especialmente cuando se utilizan los 24.
- En ocasiones se quedan bloqueados.
- Dado que se han calibrado para que no se bloqueen a cortas distancias, *no ven* por debajo de una distancia mínima.

Como consecuencia aparecen dificultades como oscilaciones al pasar por pasos estrechos, velocidad de reacción pequeña, etc.

Con el fin de solucionar estos inconvenientes, se ha mejorado notablemente el supervisor anterior, con la incorporación de la información recibida por un nuevo sistema de percepción: una cámara de infrarrojos situada en la parte delantera del vehículo que recoge la luz estructurada emitida por un diodo láser (Figura 6). Ello permite, entre otras cosas, *ver* con bastante precisión

hacia el frente y por tanto pasar por pasos estrechos.

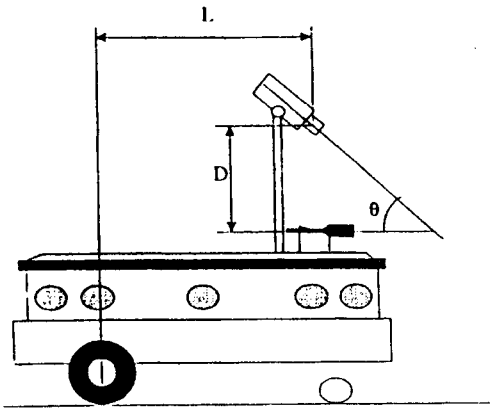


Figura 6: Conjunto diodo láser - cámara infrarrojos.

Este sensor es más rápido y preciso que los ultrasonidos, y barre un ángulo de unos 60°, devolviendo 511 medidas dentro del área iluminada.

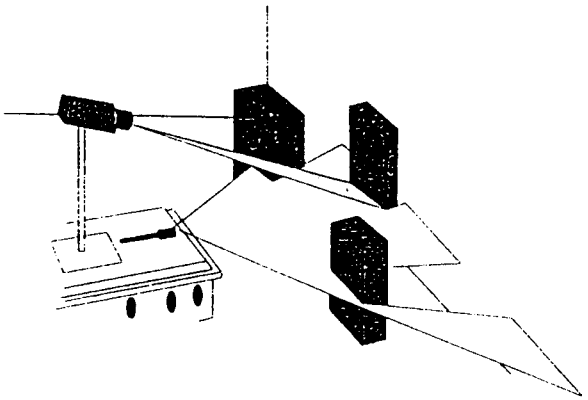


Figura 7. Campo de visión del láser.

Las entradas al supervisor en este caso son:

- La trayectoria a seguir.
- La posición actual del robot (odometría).
- La distancia mínima del lado derecho iluminado por el láser.
- La distancia mínima del lado izquierdo iluminado por el láser.
- El espacio libre a la izquierda del robot (ultrasonidos).
- El espacio libre a la derecha del robot (ultrasonidos).

El nuevo esquema de control reactivo aparece reflejado en la Figura 8. Básicamente el supervisor trata de utilizar la información procedente del láser. Sin embargo, dado el poco campo de visión que proporciona este sensor, es necesario complementarlo con la información que suministran los ultrasonidos en los laterales del vehículo.

Las reglas del supervisor son del tipo:

- Si los obstáculos están situados de manera que constituyen un pasillo suficientemente ancho, el robot intenta posicionarse en medio del pasillo para pasarlo.

- Si la variación de ángulo en la trayectoria es grande y el láser detecta un obstáculo, anticipa la variación en la trayectoria mediante la conducta *cortar el camino*.

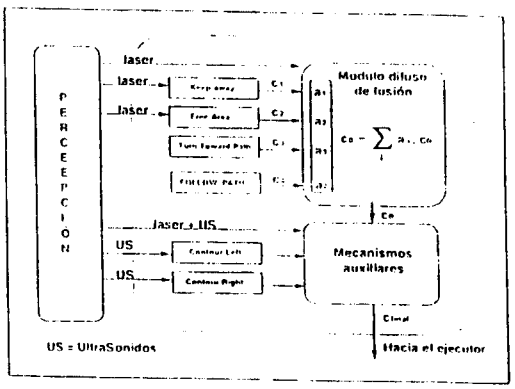


Figura 8: Esquema de control reactivo con fusión de información sensorial.

De esta forma, se fusiona la información sensorial procedente de dos tipos diferentes de sistemas de percepción, con lo que se logra una información más rica del entorno. En la Figura 7, se muestra como se ha enriquecido la visibilidad del supervisor.

Las simulaciones y ensayos realizados muestran las mejoras sustanciales que este nuevo supervisor aporta.

6 Conclusiones

La implementación de una conducta primitiva del tipo *seguir una trayectoria* con lógica borrosa no ofrece mayores problemas y está perfectamente resuelta. La complejidad surge a la hora de dotar de inteligencia al supervisor.

Los resultados obtenidos con los diferentes supervisores son satisfactorios. El ajuste de las reglas resulta intuitivo al quedar fácilmente modelada la incertidumbre inherente a este tipo de objetivos de control.

Sin embargo, el número de situaciones geométricas que pueden darse es tan elevado, que resulta complicado

ajustar el conjunto de reglas. Al aumentar el número de entradas, el diseño se complica rápidamente. Por lo tanto, es necesario llegar a un compromiso entre la cantidad de información sensorial, que para mejorar el control deberá ser lo mayor posible, y el número de entradas al supervisor borroso, que deberá ser lo menor posible para facilitar el diseño del conjunto de reglas.

Agradecimientos

Este trabajo ha sido financiado por la Comisión Interministerial de Ciencia y Tecnología (CICYT) a través del proyecto TAP94-0711-C03-01 "Sistema Central de Control para Sistemas de Transporte Industrial Basados en Múltiples Robots Móviles Autónomos".

Queremos reconocer también el trabajo de los últimos años profesores Miguel Angel Salichs y Luis Moreno, que sentaron las bases de lo que hoy es el área de robótica móvil de DISAM.

Referencias

[1] Anderson, T.L., Donath M., Animal Behavior as a Paradigm for Developing Robot Autonomy, en *Designing Autonomous Agents*, P. Maes Editor, pp. 145-168, MIT Press, 1990.

[2] R.E.Bellman y L.A.Zadeh, *Decision-Making in a Fuzzy Environment*, Management Science 17 B, (1970) 141-164.

[3] Gachet, M.A. Salichs, J.R. Pimentel, L. Moreno y A. de la Escalera, *A Software Architecture for Behavioral Control Strategies of Autonomous Systems*, Proc. IECON'92, pp.1002-1007, San Diego, CA., 1992.

[4] R.Galán y R.Sanz, La Inteligencia Artificial desde el Punto de Vista del Control Automático. En *La Inteligencia Artificial y el Control en Tiempo Real*, REPSOL-IIC Colección "Ensayo", (1992) 97-114.

[5] C. Isic y A.M. Meystel, *Pilot Level of a Hierarchical Controller for an Unmanned Mobile Robot*, IEEE Journal of Robotics and Automation, Vol. 4, No. 3, 1988.

[6] A.Jiménez y F.Matía, Aplicaciones de Fuzzy Logic en Tiempo Real. En *La Inteligencia Artificial y el Control en Tiempo Real*, REPSOL-IIC Colección "Ensayo", (1992) 189-239.

[7] P. Maes y R. A. Brooks, Learning to Coordinate Behaviors, en *Autonomous Mobile Robots: Control, Planning and Architecture*, pp. 224-230, IEEE Computer Society Press, 1991.

[8] Mataric, M.J., *Interaction and Intelligent Behavior*. PhD Thesis, MIT, 1994.

[9] F.Matía, R.Sanz, A.Jiménez y J.P.Oria, *Fuzzy Logic for Sensor Validation: the FFV Tool*, Workshop on Integration in Real-Time Intelligent Control Systems, IRTICS'93. Organizado por el IIC. Madrid (España), 1993.

[10] F. Matía, *Análisis y Diseño de Sistemas de Control de Procesos Basados en Lógica Borrosa*, Tesis Doctoral, Universidad Politécnica de Madrid, 1994.

[11] L. Moreno, E. Moraleda, M. Salichs, J.R. Pimentel y A. de la Escalera, *Fuzzy Supervisor for Behavioral Control of Autonomous Systems*, Proc. IECON'93, Hawaii, USA (1993) 258-261.

[12] L.Moreno, E.Moraleda, E.A.Puente, J.M.Armingol, A. De la Escalera, *Planning and Control for Reactive Control*. 2nd IFAC Conference IAV'95. June 12-14th, 1995.

[13] Puente, E.A., Aracil, R., Balaguer, C., *Vehículos Autoguiados para Transporte en Recintos Industriales. El Prototipo VAD-1*, Regulación y Mando Automático, 136, 1984.

Lógica Difusa a Bordo de un Velero

Jaime Abril Dovalo
Dpto. Física
Facultad de Física
Universidad de las Islas Baleares
e-mail: abril@inodoro.uib.es

Jaime Salom
Dpto. Física
Facultad de Física
Universidad de las Islas Baleares
e-mail: jsalom@inodoro.uib.es

Oscar Calvo Ibáñez
Dpto. Física
Facultad de Física
Universidad de las Islas Baleares
e-mail: oscar@hp1.uib.es

Resumen

Simulaciones con un modelo analógico y experimentos con un prototipo a escala muestran la capacidad de un controlador basado en lógica difusa para gobernar el timón y trimar las velas de un velero.

Palabras Clave: control difuso, control en tiempo real, sistemas marinos, autopilotos, autotimoneles, buques, veleros, sensores y actuadores.

1 Introducción

El gobierno de un velero comprende dos tareas básicas: el gobierno del timón y el trimado (u orientación) de las velas según el rumbo y/o ángulo de viento. Históricamente, la primera tarea que se automatizó fue el gobierno del timón, resultando el autotimonel. Se pretendía conseguir un ahorro de combustible, especialmente en grandes buques mercantes, evitando por una parte las guiñadas y por otra la elongación de la trayectoria [1], [2], [11], [12], [15] y [18]. Sin embargo, la automatización del trimado de las velas es bastante reciente, de hecho, todavía hoy existen pocos estudios sobre el tema. Se persigue, en este caso, alcanzar un régimen de velocidad óptimo, por motivos de tiempo/economía. El motivo principal de este desfase histórico, ha sido la escasa utilización de la vela en la Marina Mercante y en la Armada desde que aparecieron los primeros vapores. No obstante esta situación está cambiando, principalmente debido a la escasez de combustibles fósiles, y en consecuencia, el interés por las energías alternativas está creciendo día a día. Este interés ya se ha trasladado al campo de la Náutica con la recuperación de la vela como forma de propulsión complementaria o alternativa en los buques mercantes (turbovelas, velas rígidas, etc.). Paralelamente ya se están desarrollando sistemas de trimado automático de las velas para conseguir un rendimiento óptimo.

Por otra parte, se conoce, como muestran algunos trabajos sobre modelización de buques [3], [7], [16] y [17], que la dinámica de un buque es no lineal y variante, dependiendo de las condiciones meteorológicas (viento y estado de la mar) y operativas (velocidad, carga y asiento). Si se considera el efecto de las velas, la dinámica es todavía más compleja. Asimismo las perturbaciones como el viento y las olas son de naturaleza estocástica. Por todo ello, resulta difícil obtener un modelo preciso del buque, por lo que, diseñar sistemas de gobierno automático de buques no es una tarea fácil. Sin embargo, se ha podido comprobar a lo largo de los siglos la capacidad y pericia de los hombres para gobernar cualquier tipo de embarcación incluso en las condiciones más adversas. De hecho, todavía hoy en muchas situaciones críticas de la navegación se sustituye el piloto automático por el timonel. Este hecho induce a pensar que la Lógica Difusa, como herramienta probadamente eficaz de modelización del conocimiento y razonamiento humanos, pudiera ser adecuada como base de un sistema de gobierno automático de un buque.

Motivados por todo lo anterior es por lo que hemos desarrollado un sistema de control automático de posición del timón y de las velas de un velero con el fin de obtener un régimen de velocidad óptimo. El sistema ha sido probado con un modelo analógico de un buque, indicando la idoneidad de un controlador difuso para esta aplicación y, actualmente, se está experimentando con un velero a escala con el objeto de obtener resultados definitivos.

2 Gobierno de un buque a vela

La figura 1 muestra las variables principales que intervienen en el gobierno de un buque a vela: R, el rumbo del buque, R_s, el rumbo superficie (o efectivo), Ab, el abatimiento, β, el ángulo de viento aparente, α, el ángulo de ataque del viento, δ, el ángulo de timón y ω, el ángulo de trimado de las velas.

Básicamente existen dos modos de navegación a vela, según que el buque navegue de ceñida (con viento abierto hasta unos 70°) o de través hacia popa (con vientos portantes). En el primer caso, dado que el viento incide con un ángulo adverso, es prioritario mantener un ángulo de viento aparente que permita conseguir un buen régimen de

velocidad. En este caso, el rumbo no interesa más que a efectos de estimar la posición. De hecho, este tipo de navegación se caracteriza por la realización de bordos o navegación en zig-zag con el objeto de seguir un cierto rumbo efectivo. En el segundo caso, el viento incide favorablemente por lo que se navega al rumbo deseado intentando conseguir la máxima velocidad.

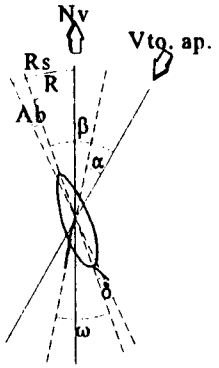


Figura 1: Variables del sistema

Así pues, en el primer caso, el timón se gobierna para mantener un ángulo de viento aparente β^0, mientras que, en el segundo, se gobierna para mantener un rumbo R_{ref}. En cualquier caso, las velas se trimarán a efectos de obtener el máximo empuje de la embarcación. El ángulo óptimo de trimado se puede obtener a partir de la curva polar de la vela conociendo además el ángulo de viento aparente. En la curva de la figura 2, correspondiente a una vela Marconi, de relación de aspecto 6, se puede ver que el ángulo de ataque para el que la fuerza de sustentación es máxima es de 15°. No obstante, una regla de oro es la que dice que *el ángulo de trimado de las velas debe ser la mitad del ángulo de viento aparente.* Por otra parte, la curva polar de velocidad permite determinar la velocidad de la embarcación según el ángulo de viento real. La figura 3 muestra la curva polar de velocidad de un velero en unas determinadas condiciones de viento y estado de la mar. Para ángulos menores de 40° el buque no puede navegar y el mejor rumbo de ceñida, a efectos de velocidad efectiva, es el que está abierto 40° respecto del viento real.

Por último, el gobierno de un buque comprende otros niveles de decisión superiores como la determinación del rumbo (navegador) y la elaboración de la derrota (trazador). El presente trabajo se centra en el nivel más bajo correspondiente a la decisión del ángulo del timón y del ángulo de las velas.

3 La planta: el velero

La planta con la que vamos a trabajar es una reproducción a escala 1:20 de un velero tipo Ketch de la casa Hobbe (fig. 4). Tiene una eslora de 1,030 m, una manga de 0,245 m, una altura total de 1,530 m y un desplazamiento de 4,5 kg. El paño está constituido por dos velas Marconi envergadas a los palos mayor y mesana, y un foque envergado al estay de proa, haciendo una superficie vélica total de 36,6 dm^2. Las escotas de las tres velas se cazan y lascan al unísono mediante un servomotor. Del mismo modo, otro servomotor mueve el timón a una banda o a otra.

El velero en cuestión, como embarcación pequeña, se puede considerar como un sólido rígido afectado de tres movimientos de traslación (avance, lateral y vertical) y tres movimientos de rotación (balance, cabeceo y guiñada), según un sistema de coordenadas (X_b, Y_b, Z_b) solidario al buque (fig. 5). Naturalmente, esta hipótesis es válida independientemente de la escala. Las ecuaciones del movimiento de la embarcación desplazándose en un fluido se obtienen de las leyes de Newton sobre la conservación de los momentos lineal y angular. Expresándolas matricialmente, se tiene:

$$\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} Matriz \\ de\ Masas \\ 6x6 \end{bmatrix}^{-1} \begin{bmatrix} X \\ Y \\ Z \\ L \\ M \\ N \end{bmatrix}$$

Las fuerzas son divididas en diferentes acciones atendiendo a las causas que las producen. En [7] se consideran nueve categorías distintas para un buque de propulsión a motor. Con el objeto de ajustar el modelo a nuestro caso, se ha eliminado la categoría correspondiente al efecto del propulsor y se ha añadido una categoría correspondiente al efecto de las velas. Adicionalmente, se han reorganizado los términos, de modo que, la fuerza total se puede expresar como:

$$\begin{aligned} fuerza\ total = {} & fuerza\ hidrodinámica \\ & + fuerza\ aerodinámica \\ & + perturbaciones\ ambientales \end{aligned}$$

donde el término hidrodinámico se refiere a la acción del fluido sobre el casco, el término aerodinámico se refiere a la acción del viento sobre las velas y las perturbaciones ambientales tienen en cuenta las fluctuaciones en el viento y en el estado de la mar. Las fuerzas hidrodinámicas son funciones no lineales y variantes del movimiento del buque. No obstante, para buques grandes, se pueden obtener aproximaciones lineales de estas funciones que conllevan a modelos lineales de la dinámica del buque [3], [16] y [17].

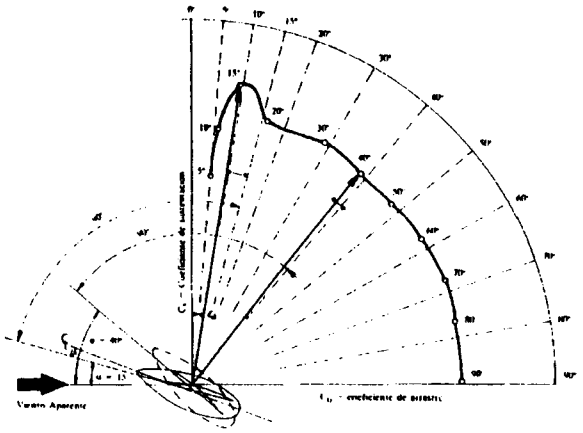


Figura 2: Curva polar de las velas

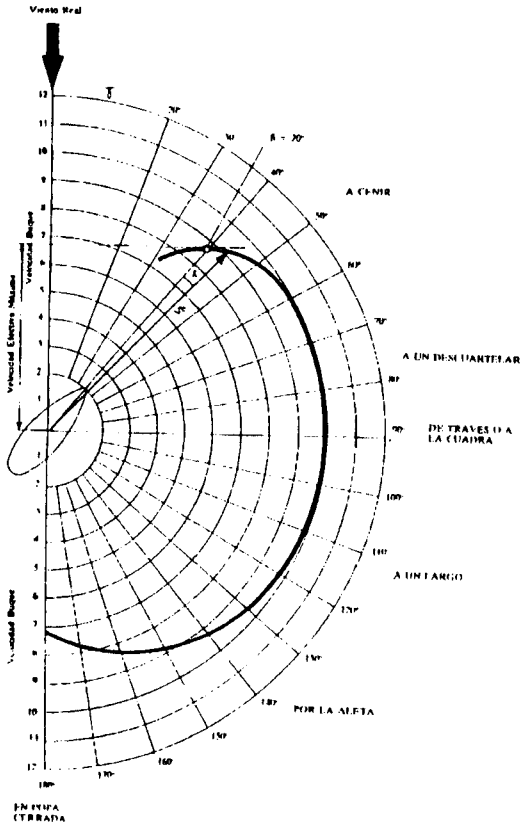


Figura 3: Curva polar de velocidad

Además, en este caso, a efectos de considerar sólo el movimiento en el plano de la mar (el único que nos interesa), se pueden despreciar los acoplamientos con los movimientos en los otros planos con lo que el modelo se simplifica notablemente. Seguidamente se muestra un modelo lineal ampliamente utilizado en el diseño de pilotos automáticos, que utilizaremos nosotros también, en forma de circuito electrónico, para probar la capacidad de un controlador difuso como timonel. La función de transferencia del ángulo de timón, δ, al ángulo de guiñada (o desviación del rumbo), ψ, viene dada por:

$$G_{\psi\delta}(s)=\frac{K(1+sT)}{s(1+sT_1)(1+sT_2)}$$

y la ecuación de estado, respecto de las variables, v, velocidad lateral, r, velocidad angular de guiñada y ψ, ángulo de guiñada, viene dada por:

$$\begin{bmatrix} v \\ r \\ \psi \end{bmatrix} = \begin{bmatrix} \frac{V}{L}a_{11} & Va_{12} & 0 \\ \frac{V}{L^2}a_{21} & \frac{V}{L}a_{22} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} \frac{V^2}{L}b_{11} \\ \frac{V^2}{L^2}b_{21} \\ 0 \end{bmatrix} \delta + \begin{bmatrix} \frac{V^2}{L}g_{11} & \frac{V^2}{L}g_{12} \\ \frac{V^2}{L^2}g_{21} & \frac{V^2}{L^2}g_{22} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

Nótese que en este caso se ha tenido en cuenta el efecto de las perturbaciones como un término aditivo, siendo w_1 y w_2 dos realizaciones de ruido blanco, con matrices de covarianza W_1 y W_2. Esta aproximación se deriva de de considerar tales perturbaciones como fuerzas adicionales independientes del conjunto de fuerzas hidrodinámicas y aerodinámicas, y cuya naturaleza es estocástica [3] y [10].

En nuestro caso, la modelización no es tan simple. La reducida escala del modelo hace más patentes los fenómenos de no linealidad y variancia. Por otra parte, se suman las fuerzas aerodinámicas, de expresiones complejas, mucho menos estudiadas que las fuerzas hidrodinámicas. Todo ello hace pensar que es bastante difícil obtener un modelo de la dinámica de nuestro velero. La mayoría de los sistemas de control convencionales están basados en un modelo del proceso, que permite determinar su estructura y, frecuentemente, también sus parámetros (PID, STR, MRAC, LQG, etc.). En este caso, resultaría harto difícil diseñar alguno de los sistemas de control anteriores por no disponer de un modelo adecuado de la planta. Como veremos más adelante, la Lógica Difusa permite abordar este problema con relativa facilidad.

4 El controlador: fuzzy controller

Los apartados anteriores revelan dos circunstancias que nos llevan a elegir la Lógica Difusa [9], [13], [14], [21], [22] y [23], como la herramienta idónea para resolver nuestro problema: la dificultad para obtener un modelo matemático adecuado del velero y la experiencia humana en el gobierno de embarcaciones de todo tipo. De este modo, se

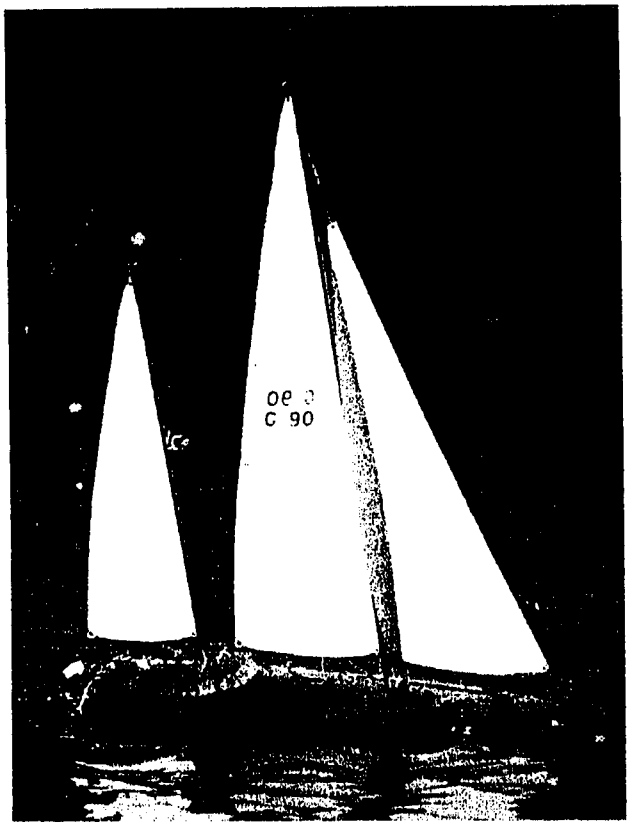


Figura 4: Velero objeto de este trabajo

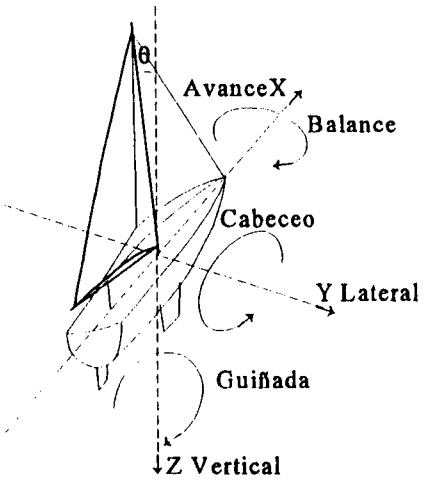


Figura 5: Movimientos del velero

ha desarrollado un sistema de control difuso que permite realizar los dos tipos de navegación: de ceñida y de través hacia popa.

En el diagrama de bloques de la figura 6 se puede ver, de acuerdo con lo expuesto anteriormente, que las variables de control son el ángulo de timón y el ángulo de trimado de las velas y que las variables controladas son el rumbo y el ángulo de viento aparente. Asimismo se puede apreciar la existencia de otras variables secundarias como el ángulo de escora (o inclinación), la velocidad del viento aparente y la velocidad del buque. Conociendo el ángulo de escora (θ) el sistema de gobierno automático puede reaccionar ante golpes de viento o rachas amollando las escotas. Téngase en cuenta que una escora excesiva deja sin efecto al timón perdiéndose el control del velero. Las velocidades del viento aparente (V_{ap}) y del buque (V_b) se pueden utilizar para ajustar los parámetros del controlador (básicamente, ganancias). Estas variables no se han considerado en el presente diseño a efectos de simplificar al máximo su realización.

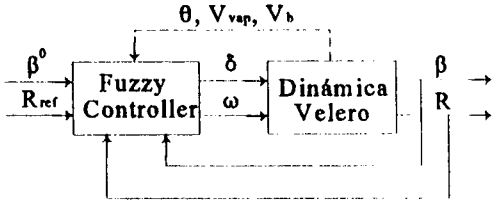


Figura 6: Diagrama de bloques del sistema de control

El controlador que se ha implementado es un tipo Takagi-Sugeno [20], en el que el consecuente de cada regla es un término constante:

$$SI\ A_0\ ENTONCES\ \ z_0$$

y la salida es la media ponderada de todos los consecuentes:

$$z = \frac{\sum_i w_i z_i}{\sum_i w_i}$$

siendo w_i el valor de certeza del antecedente. La razón principal de haber elegido este tipo de razonamiento difuso es su simplicidad, que, entre otras cosas, facilita la labor de implementación como sistema de control en tiempo real. Por otra parte, se puede demostrar que un sistema de inferencia difusa como el anterior permite aproximar cualquier mapeo no lineal definido sobre un conjunto compacto, con un error arbitrariamente pequeño.

Según el modo de navegación, de ceñida o de través hacia popa, se tienen dos tipos de reglas. Se dispararán unas u otras según el valor del ángulo de viento aparente, que aparece en el antecedente de todas las ellas. Algunos ejemplos son:

-SI β *es pequeño*
Y $\beta-\beta^0$ *es positivo*
Y $\Delta(\beta-\beta^0)$ *es positivo*
ENTONCES δ *es* -0.8
-SI β *es pequeño*
Y $\beta-\beta^0$ *es positivo*
Y $\Delta(\beta-\beta^0)$ *es casi cero*
ENTONCES δ *es* -0.4
. . .
-SI β *es grande*
Y $R-R_{ref}$ *es positivo*
Y $\Delta(R-R_{ref})$ *es positivo*
ENTONCES δ *es* -0.8
. . .
-SI β *es pequeño*
O β *es grande*
ENTONCES ω *es* $\beta/2$

donde β^0 es el ángulo de ceñida óptimo, obtenido de la curva polar de velocidad y R_{ref} es el rumbo deseado. Obsérvese la aparición de las derivadas $\Delta(\beta-\beta^0)$, $\Delta(R-R_{ref})$, que dan lugar a una acción derivativa, compensadora, proporcionando una mayor estabilidad al controlador. Esta estrategia, típica del Control Clásico (control PD), se utiliza también en el FLC. Asimismo existe una acción integrativa que resulta del filtrado de las medidas de los sensores. De este modo, se evita que el controlador reaccione ante perturbaciones de alta frecuencia.

Por último, merece comentar los valores lingüísticos o etiquetas utilizados para las diferentes variables. Los universos de discurso de β, $\beta-\beta^0$, $\Delta(\beta-\beta^0)$, $R-R_{ref}$, $\Delta(R-R_{ref})$, δ y ω, han sido normalizados a [-1, +1] (siendo los rangos de valores físicos los siguientes: $0<\beta<180°$, $0<R-R_{ref}<360°$, $-35<\delta<35°$, y $0<\omega<90°$). Por lo que respecta a la variable, β, puede tomar los valores "pequeño" y "grande"; $\beta-\beta^0$, $R-R_{ref}$, al igual que sus derivadas, pueden tomar los valores "negativo", "casi cero" y "positivo"; ω, δ, son valores nítidos.

5 Aspectos tecnológicos [4], [5] y [6]

El sistema de gobierno automático ha sido realizado en base a un sistema basado en el microcontrolador MC68HC11 de Motorola. El sistema sincroniza todas sus operaciones por interrupciones generadas por un reloj de tiempo real. De este modo, se puede controlar la frecuencia de muestreo así como los tiempos de adquisición, procesado y actuación.

El MC68HC11 es un microcontrolador de 8 bits, con una velocidad nominal del bus de 2 Mhz. Incluye un conversor analógico/digital de 8 canales/8 bits/15 Khz, un temporizador con capacidad de recarga, captura y comparación que permite la generación/monitorización de señales PWM (Modulación de Anchura de Pulso) con resolución de 1 µs e interficies de comunicación serie síncrona y asíncrona. El resto del sistema proporciona una memoria SRAM de 32 kB así como diferentes interficies de entrada/salida (motores DC, pantalla LCD, etc.)

El sistema implementado actualmente permite gobernar el velero de forma manual, desde la unidad de radiocontrol, o de forma automática, manteniendo el rumbo y/o el ángulo de viento aparente consignados. Para ello, se sigue la siguiente secuencia de operaciones: 1) sincronización con el RTC; 2) lectura de los datos del compás, de la veleta y monitorización de las señales de la unidad de radiocontrol; 3) filtrado de datos; 4) procesado de datos a través del sistema de inferencia difuso; y 5) generación de las señales de actuación sobre el timón y las velas.

Los sensores del sistema están constituidos por el compás y la veleta. Se prevee en un futuro próximo incluir un clinómetro, un anemómetro y una corredera para medir el ángulo de escora, la velocidad del viento aparente y la velocidad del buque, respectivamente. La veleta mide el ángulo de incidencia del viento aparente, β. El ángulo de orientación de la veleta se determina mediante dos pares de sensores de efecto Hall, dispuestos ortogonalmente, que miden las componentes X-Y del campo magnético producido por un imán solidario al eje del mecanismo. El compás mide el rumbo del buque, R. Dicho ángulo, se determina de un modo análogo al anterior, mediante un dispositivo integrado al efecto, con mayor sensibilidad.

Los actuadores consisten en dos servomotores de la casa Futaba, de 2,5 kg de fuerza y 90° de ángulo de giro, de tipo proporcional, comandados por señales PWM. Uno actúa sobre el eje del timón, haciéndolo girar en un sentido u otro y el otro sobre las escotas de las velas, cazándolas o amollándolas.

Finalmente, cabe citar que se ha utilizado una frecuencia de muestreo de 0.1 s, de acuerdo con el ancho de banda del sistema (velero). A efectos de comparación, las frecuencias de muestreo empleadas en los pilotos automáticos de buques grandes van de 1 s hasta 10 o 15 s [1], [2], [11], [18] y [24]. Se ha considerado una resolución de 8 bits (256 valores), para la adquisición de datos, suficiente para esta aplicación, teniendo en cuenta además la capacidad de la Lógica Difusa para tratar con información imprecisa; y de 32 bits (coma flotante) para el cálculo.

6 Resultados

Se ha probado la capacidad de un controlador difuso como timonel para un buque carguero obteniéndose resultados inmediatos que indican la idoneidad de la Lógica Difusa para abordar este tipo de problemas.

La planta es un un circuito analógico basado en un amplificador operacional, que realiza una función de transferencia de 1er. orden, modelizando así la dinámica de un buque carguero (modelo de Nomoto): $G_{\psi\delta}(s) \; 10/1-3s$. Se puede observar que se trata de una planta inestable (el polo, s=1/3, está en el semiplano derecho del plano complejo), típico de la dinámica de los buques de gran

tamaño.

El controlador difuso es del tipo Takagi-Sugeno, con reglas simplificadas, de acción proporcional-derivativa. Se ha utilizado un conjunto de 9 reglas de la forma: *SI* $R\text{-}R_{ref}$ *es negativo Y* $\Delta(R\text{-}R_{ref})$ *es negativo ENTONCES* δ *es* $+0.8$, extraídas directamente de la práctica. Los universos de discurso de δ, $R\text{-}R_{ref}$ y $\Delta(R\text{-}R_{ref})$, se han normalizado todos a [-1, +1], mediante las constantes K_p, K_d y K_a. Las etiquetas de $R\text{-}R_{ref}$ y $\Delta(R\text{-}R_{ref})$ son "negativo", "casi cero" y "positivo", mientras que δ toma valores nítidos.

Se han realizado diversos experimentos, de regulación (mantenimiento del rumbo de referencia), de seguimiento (seguimiento de formas de onda cuadradas, triangulares y sinusoidales), de sensibilidad frente al ruido y las variaciones paramétricas, todos con éxito. A efectos de comparación se ha probado un controlador convencional, PD con saturación, dando unos resultados similares. Sin embargo, el aspecto más importante que diferencia ambos diseños es que el diseño del controlador difuso se ha llevado a cabo, únicamente, a partir de la experiencia, mientras que el del controlador convencional ha precisado conocer un modelo del buque. Esta característica otorga al controlador difuso, por una parte, una mayor facilidad de diseño, ya que simplemente se ha de escribir en reglas el conocimiento del experto, y por otra, una mayor robustez ante perturbaciones y variaciones de la planta, tanto paramétricas como estructurales, pues apenas se ha tenido en cuenta su dinámica.

También se ha probado el sistema de gobierno automático para el velero con resultados satisfactorios. Actualmente, se está optimizando el funcionamiento del controlador incorporando nuevas reglas, ajustando las ganancias, etc.

7 Conclusiones

En este artículo se ha demostrado como la Lógica Difusa permite resolver de un modo sencillo el problema del gobierno automático de un velero. Para ello, se ha probado la capacidad de un controlador difuso para gobernar un buque, a partir de simulaciones con un modelo analógico y experimentos con un prototipo a escala, en ambos casos, con resultados satisfactorios.

Referencias

[1]Abril, J.A., "Síntesis Comparativa de Diversos Controladores Estocásticos de un Autotimonel para un Buque Carguero", *Proyecto Final de Carrera*, 1993.

[2]van Amerongen, J., "Adaptive Steering of Ships - A Model Reference Approach", *Automática*, vol. 20, pp. 3-14, 1984.

[3]Åström, K.J., and Källström, C.G., "Identification of Ship Steering Dynamics", *Automática*, vol. 12, pp. 9-22, 1976.

[4]Åström, K.J., and Wittenmark, B., *Computer Controlled Systems: Theory and Design*, Prentice-Hall, Inc., 1984.

[5]Auslander, D.M., and Tham, C.H., *Real-Time Software for Control*, Prentice-Hall, Inc., 1990.

[6]Bennet, S., *Real-Time Computer Control, An Introduction*, Prentice-Hall International (UK), Ltd., 1994.

[7]Browning, A.W., "A Mathematical Model to simulate Small Boat Behaviour", *Simulation*, vol. 56:5, pp. 257-264, 1991.

[8]Costa, J.B., *Patrón de Embarcaciones de Recreo*, Capt. J.B. Costa, 1990.

[9]Harris, C.J., Moore, C.G., Brown, M., *Intelligent Control, Aspects of Fuzzy Logic and Neural Nets*, World Scientific Publishing, Co., Pte., Ltd., 1993.

[10] Källström, C.G., "Control of Yaw and Roll Motion by a Rudder/Fin-Stabilization System", *Symp. on Ship Control Systems*, Ottawa, Canada, 1981.

[11] Källström, C.G., Åström, K.J., Thorell, N.E., Eriksson, J., and Sten, L., "Adaptive Autopilots for Tankers", *Automática*, vol. 15, pp. 241-254, 1979.

[12]Layne, J.R., and Passino, K.M., "Fuzzy Model Reference Learning Control for Cargo Ship Steering", *IEEE Control Systems*, December, pp. 23-34, 1993.

[13]Lee, C.C., "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part I", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 20, n. 2, pp. 404-418, March-April, 1990.

[14]Lee, C.C., "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part II", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 20, n. 2, pp. 419-435, March-April, 19: 0.

[15]Minorsky, N., "Directional Stability of Automatically Steered Bodies", *Trans. Soc. Nav. Archit. Mar. Engrs.*, N.Y., U.S.A., 1922.

[16]Nomoto, Taguchi, Honda, and Hirano, "On the Steering Qualities of Ships", *International Shipbuiding Programme*, vol. 4, nº 35, 1957.

[17]Norrbin, N.H., "Theory and Observations on the Use of a Mathematical Model for Ship Manoeuvring in Deep and Confined Waters", *Symp. on Naval Hydrodynamics*, Pasadena, U.S.A., 1970.

[18]Quevedo, J., "Contribución al Estudio y Diseño de Autotimoneles para Buques", *Tesis Doctoral*, Institut de Cibernètica, Barcelona, 1983.

[19]Roger, J-S., "ANFIS: Adaptive-Network Based Fuzzy Inference Systems", *Internet*, 1994.

[20]Takagi, T., and Sugeno, M., "Derivation of Fuzzy Control Rules from Human Operator's Control Actions", *Symp. on Fuzzy Information, Knowledge Representation and Decision Analysis*, Marseilles, France, 1983.

[21]Yamakawa, T., "A Fuzzy Logic Controller", *Journal of Biotechnology*, 24, pp. 1-32, 1992.

[22]Zadeh, L.A., "Fuzzy Sets", *Information and Control*, vol. 8, pp. 338-353, 1965.

[23]Zadeh, L.A., "Outline of a New Approach to the Analysis Complex Systems and Decision Processes", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 3, pp. 28-44, 1973.

Controlador Fuzzy optimizado mediante algoritmos genéticos para el seguimiento de paredes en un robot móvil

Juan Pedro Uribe
Dpto. de Inteligencia Artificial
IKERLAN
Centro de Investigaciones Tecnológicas
E-Mail: robot_movil@ikerlan.es

Jokin Mujika
Dpto. de Robótica
IKERLAN
Centro de Investigaciones Tecnológicas
E-Mail: robot_movil@ikerlan.es

Reinhard Braunstingl*
Dpto. de Robótica
IKERLAN
Centro de Investigaciones Tecnológicas
E-Mail: braun@fmechds01.tu-graz.ac.at.

Resumen

Se presenta un robot para seguimiento de paredes equipado con sensores de ultrasonidos. Este robot utiliza un controlador de lógica difusa y una estrategia de navegación local. La base de la navegación reactiva la suministra el concepto de percepción general, que traspasa la información percibida por los sensores al sistema fuzzy sin necesidad de modelar paredes u obstáculos. Por lo tanto, no es necesario ningún tipo de representación del entorno. La base de reglas del sistema fuzzy fue diseñado manualmente, para aplicar posteriormente un algoritmo con el fin de encontrar las funciones de pertenencia óptimas para que el robot se desplace a una distancia constante de la pared, a gran velocidad y lo más suavemente posible. La optimización se realizó utilizando un robot simulado. Los resultados de esta simulación probaron satisfactoriamente el gran parecido con la realidad cuando fueron aplicados sobre un robot real.

Palabras clave: *Robot móvil, Navegación Local, Lógica Borrosa, Algoritmo Genético.*

1 Descripción del robot móvil

El Vehículo Autónomo Experimental (VEA-1) desarrollado en Ikerlan es un robot móvil de alta maniobrabilidad, capaz de operar autónomamente y realizar misiones establecidas previamente en entornos parcialmente estructurados.

Existen dos partes diferenciadas en el prototipo desarrollado (VEA-1): El vehículo en el sentido estricto de la palabra, con sus mandos, controles y sensores, y la estación central donde, además del plan de trayectos a recorrer, se realiza la supervisión y el control de las acciones asignadas.

* Reinhard Braunstingl pertenece a la Universidad Técnica de Graz (Austria), y ha participado en el proyecto bajo la financiación del Programa Comett de la Comunidad Europea.

1.1 Vehículo

El vehículo móvil es de forma cilíndrica y tiene un diámetro de 0.6 m por una altura de 1.3 m. Está compuesto por cuatro niveles diferentes.

El *nivel básico* contiene el subsistema de locomoción y baterías. El sistema de locomoción consta de cuatro pares de ruedas para tracción y dirección, estando uno de ellos dotado de un sistema de suspensión diseñado para absorber pequeñas irregularidades del terreno, asegurando de esta manera un contacto continuo con el terreno. La acción de tracción está sincronizada mecánicamente, y existe un único actuador para ello. Las ruedas se hacen girar simultáneamente gracias a una correa conectada a un accionador independiente. Cada par de ruedas tiene una unidad diferencial desequilibrada, con la cual consiguen eliminar cualquier influencia generada en la tracción cuando existe un cambio de dirección, debido a que el punto de contacto entre las ruedas y el suelo no coincide con la dirección del eje de giro. Este simple mecanismo supone que la tracción y el guiado del robot son independientes, asegurando de esta manera un extraordinario nivel de maniobrabilidad y facilidad de control.

El *nivel intermedio* contiene los motores y los sensores ultrasónicos. Incluye los dos motores y los reguladores que gobiernan la tracción y la conducción en el subsistema previo y otro actuador que permite variar la orientación de la carcasa del robot. Esta carcasa contiene los sensores de ultrasonidos y el sistema de localización absoluta.

El *nivel superior* contiene el sistema de control de a bordo. Este subsistema es responsable de obtener las funciones en tiempo real para responder a eventos externos. Las funciones que debe realizar la unidad de a bordo son las siguientes:

- Ejecución de las órdenes recibidas desde la estación de control, tales como moverse en línea recta, girar, localizar la posición, etc.
- Informar a la estación de control acerca del estado del robot y su entorno.
- Calcular la interpolación de los trayectos a seguir por el robot y el control de la posición de ejes. Esto se realiza utilizando la realimentación de la posición obtenida mediante los codificadores asociados a los motores.
- Detección de cualquier obstáculo no previsto en el plan de navegación, evitando el obstáculo hasta que recupera la trayectoria predefinida.
- Movimiento a través del entorno siguiendo el contorno de las paredes. Esta función constituye la parte principal de este documento y es explicada con más detalle en las siguientes secciones. Además de estas funciones, el control de a bordo asegura el correcto funcionamiento de todo el sistema.

El *sistema de localización absoluta*, situado en el cuarto nivel, está basado en el método de triangulación. Para lograr esto, se incorporan al entorno marcas luminosas activas, manipulándolo al mínimo y se posiciona el sensor en la cima del robot.

1.2 Estación Central

La estación de control está constituida por la parte del control del robot móvil que no se incluye en las tarjetas de control de a bordo del propio robot. Es responsable de los aspectos del control que no necesitan de una respuesta en tiempo real a variaciones del entorno. Como consecuencia, realiza las tareas de programación y control a un nivel más alto y lleva a cabo una comunicación directa tanto con el usuario como con el control de a bordo vía radio. Las principales funciones de la estación de control son las siguientes:

Establecer un interface con el usuario. A través de este interface, el usuario puede supervisar el estado del robot, introducir comandos simples para ser ejecutados por el mismo, definir el entorno de trabajo, definir y supervisar las misiones a realizar por el robot.

Planificación de trayectos entre puntos del entorno. Principalmente, se trata de adaptar los datos para el entorno real de trabajo a un formato adecuado, de tal manera que sea posible planificar trayectorias entre puntos del espacio accesible, de acuerdo con los diferentes métodos de planificación existentes. Como resultado de esta planificación, se obtiene un secuencia de puntos a recorrer, así como la velocidad y el tipo de trayecto a seguir por el robot entre ellos.

Supervisión del correcto funcionamiento y del estado del robot móvil. Esta función proporciona constantemente información acerca del estado del robot y posibilita supervisar la ejecución de la planificación.

2 Navegación reactiva

El seguimiento de paredes es un típico ejemplo de una misión donde se observa la necesidad de la navegación reactiva. Obviamente no es necesaria la utilización de sensores de alto nivel para el seguimiento de paredes si el entorno es perfectamente conocido. En este caso el robot podría moverse a lo largo de un trayecto óptimo preplanificado simplemente evitando los obstáculos imprevistos. Sin embargo, si el entorno es sólo parcialmente conocido por falta de información acerca de la posición, por ejemplo: *este pasillo conduce a aquella puerta*, el robot puede utilizar una estrategia de seguimiento de paredes para completar la misión rápidamente sin tener que estudiar un entorno desconocido y sin conocer la localización exacta de la puerta. A fin de ser capaz de navegar sin información de la posición global, el robot no intenta modelar el entorno, pero utiliza el concepto de percepción general.

2.1 El concepto de percepción general

El concepto de percepción general se ajusta perfectamente a los robots móviles con sensores de ultrasonidos. Debido a la pobre direccionalidad de este tipo de sensores, la modelización de un entorno no resulta una tarea trivial. El concepto de percepción general evita las dificultades derivadas del esfuerzo de obtener una representación del entorno, suministrando una representación de la percepción en su lugar, un poco imprecisa, y por lo tanto perfectamente procesable mediante lógica difusa. Esta percepción general puede expresarse en términos como *algo a la izquierda* o *totalmente de frente* sin decir absolutamente nada acerca de la dirección de las paredes o la localización exacta de los obstáculos. Un robot móvil es sorprendentemente capaz de seguir una pared de contornos arbitrarios utilizando únicamente esta información. A continuación, se explica brevemente el concepto de percepción general.

A todo sensor de ultrasonidos i del robot móvil se le asigna un vector de percepción p_i. Su dirección es igual a la orientación del eje del sensor y la longitud es función de la distancia d_i medida por el sensor:

$$p_i = \frac{d_{max} - d_i}{d_{max} - d_{min}} \quad , \qquad (1)$$

donde d_{min} y d_{max} designan, respectivamente, la distancia más corta y más larga a la que un objeto puede ser detectado con toda fiabilidad. Es más, p_i está limitado entre 0 y 1. El vector de percepción general p está compuesto por todas las percepciones individuales p_i. Su

dirección es igual a la suma de las percepciones de todos los sensores y su longitud es igual a la percepción individual más fuerte:

$$p = p_{i,\max} \frac{\sum p_i}{\left|\sum p_i\right|} \quad . \tag{2}$$

La variación en el tiempo de la percepción general es representada por $\dot{p}$ y se expresa mediante un escalar. Para este propósito, la variación temporal de la percepción de un sensor i

$$\dot{p}_i = \frac{\mathrm{d}p_i}{\mathrm{d}t} \approx -\frac{\Delta d_i}{\Delta t\,(d_{max} - d_{min})} \tag{3}$$

es relativo a $\dot{p}_{max}$, la variación en el tiempo de la percepción de un acercamiento frontal hacia un obstáculo a la máxima velocidad $\dot{p}_{max} = v_{max}/(d_{max} - d_{min})$, y se llama $\dot{p}_i^*$. Además únicamente se considerarán los valores positivos de $\dot{p}_i^*$, por lo que la variación temporal de la percepción relativa de un sensor i resulta como sigue:

$$\dot{p}_i^* = \begin{cases} \dfrac{\dot{p}_i}{\dot{p}_{max}} = -\dfrac{\Delta d_i}{\Delta t\, v_{max}}, & \text{if } \Delta d_i < 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

El valor máximo para todos los sensores $\dot{p}_i^*$ en el caso de que se dé, es la variación de la percepción general el tiempo $\dot{p}^*$:

$$\dot{p}^* = \dot{p}_{i,\max}^* \quad . \tag{5}$$

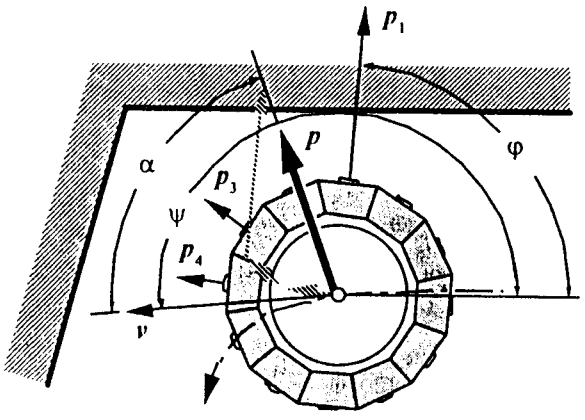


Figura 1: El vector p de percepción general proporciona una descripción cualitativa de la situación sin prestar atención a la localización exacta de las paredes: "Hay algo a la derecha y también cerca del frontal del robot", una información adecuada para ser procesada mediante lógica difusa. Por tanto, no es precisa una representación del entorno.

La figura 1 ilustra la idea básica del concepto de percepción general. Un robot se está desplazando a lo largo de una pared que se encuentra a su derecha y se aproxima a una esquina. Un sensor percibe la pared a la derecha del robot. Otros dos sensores han empezado a percibir la pared que se encuentra en frente. El vector de percepción general, compuesto por la percepción de todos los sensores, apunta a la derecha y un poco hacia adelante, pareciéndose claramente a una descripción lingüística de la situación que podría ser suministrada por un ser humano que observara la imagen sin prestar mucha atención a la forma o a la localización exacta de las paredes: "hay algo en el lado derecho, y también hay algo, pero más lejos, en frente del robot". Cualquiera podría proponer intuitivamente una reacción razonable como "gira a la izquierda". Esta reacción es generada por el controlador fuzzy descrito a continuación.

2.2 Controlador Fuzzy

El controlador fuzzy consta de 33 reglas y utiliza tres variables de entrada, ángulo de percepción α (fig. 1), percepción p (ecuación 2), y variación de percepción $\dot{p}$ (ecuación 5) para generar acciones de control para la variable de salida giro φ, dirección Ψ, y aceleración v (fig. 1). El número de reglas, así como el número de adjetivos para cada variable fueron establecidos a mano y fueron considerados razonables y suficientes. Sin embargo, la forma de las funciones de pertenencia quedó indeterminada para poder hacerlo mediante un algoritmo genético de tal forma que se pueda conseguir el óptimo funcionamiento del robot. Las tablas de 1a a 1c muestran la base de reglas utilizando abreviaciones para los adjetivos, los cuales son mencionados en las figuras 2a a 2c y 3a a 3c, respectivamente.

α / p	RB	RF	LF	LB	RB *or* RF	LB *or* LF
---	LL	LR	LL	LR	---	---
VL	---	---	---	---	L	R

Tabla 1a: Base de reglas de control de la orientación del robot a través de la velocidad de giro $\dot{\varphi}$.

p / α	VL	L	M	H	VH
RB	HR	HR	R	R	C
RF	C	L	L	HL	HL
LF	C	R	R	HR	HR
LB	HL	HL	L	L	C

Tabla 1b: Base de reglas de control de la dirección del movimiento del robot a través de la variación de la dirección $\dot{\psi}$.

$\dot{p}$ \ p	VL *or* VH	L *or* H	ME	---
ZE	ZE	P	P	---
L	EB	B	Z	---
H	---	---	---	EB

Tabla 1c: Base de reglas de control de la velocidad del robot a través de la aceleración $\dot{v}$.

Un operador 'and' combina dos o más antecedentes vía mínimo, y un operador 'or' vía máximo. Se utiliza como estrategia de inferencia la codificación por producto de correlación y combinación de suma. Después se aplica la defuzzificación vía centroide [4]. Se ofrece una descripción más detallada del controlador en [1].

3 Algoritmo genético

Una vez que la base de reglas fue establecida, fuimos capaces de conseguir un comportamiento más o menos satisfactorio del robot cambiando la forma de las funciones de pertenencia de los adjetivos. Sin embargo, incluso en el caso de esta pequeña base de reglas, los resultados de tales cambios no son siempre fácilmente predecibles y por lo tanto investigamos la posibilidad de una puesta a punto del sistema fuzzy mediante un algoritmo genético. Los algoritmos genéticos no tienen ningún requerimiento especial en cuanto a la función objetivo, concediendo por lo tanto máxima libertad a la hora de elegir la función apropiada. Todo lo que necesitan son muchos genes, los cuales representan conjuntos de parámetros codificados, describiendo cada uno de ellos una solución del problema y un valor de evaluación que describa la calidad de la solución. En nuestro caso el problema para el robot era desplazarse a lo largo de una pared. Una posible solución hubiera sido que el robot lo hiciera de cualquier forma. Sin embargo, si golpeara la pared, la calidad de la solución en cuestión sería cero. Si siguiera la pared perfectamente, a gran velocidad y con movimientos suaves, a la solución se le asignaría un valor alto de evaluación.

3.1 Codificación

Para conseguir estas funciones de pertenencia que consisten en triángulos y trapezoides en el caso de las variables de entrada y singletons en el de las variables de salida, se definen inicialmente 20 parámetros y después se codifican en forma de genes mediante codificación binaria multiparamétrica [3] y el esquema de codificación que se muestra en las figuras 2a a 2c, y 3a a 3c. Se realiza un escalado de cada parámetro u_i en el intervalo (0,1), pero cabe destacar que el dominio de algunos de los parámetros depende de otros parámetros, convirtiendo de esta manera las áreas prohibidas, por ser irrelevantes, del parámetro en espacio inaccesible. Estas figuras muestran las funciones miembro halladas por el algoritmo genético. Los 20 parámetros fueron mapeados mediante enteros sin signo en código binario con una resolución de 10 bit y concatenados para formar cadenas de 200 caracteres, los genes o los miembros de población.

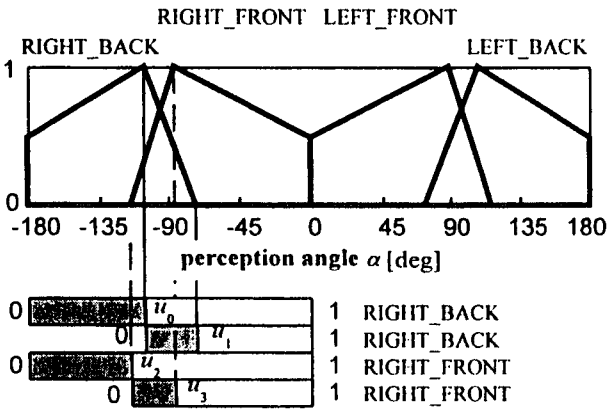


Figura 2a: Codificación de las funciones de pertenencia de los adjetivos del ángulo de percepción. Las funciones de pertenencia de los adjetivos LEFT_FRONT y LEFT_BACK han sido fijadas para que sean simétricas.

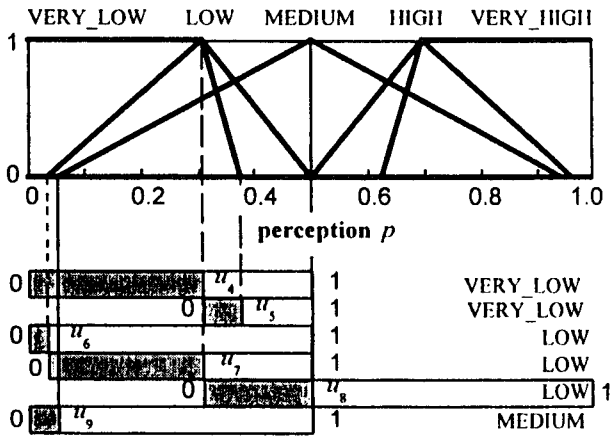


Figura 2b: Codificación de las funciones de pertenencia de los adjetivos de la "percepción". Las funciones de pertenencia de los adjetivos HIGH y VERY_HIGH han sido fijadas para que sean simétricas.

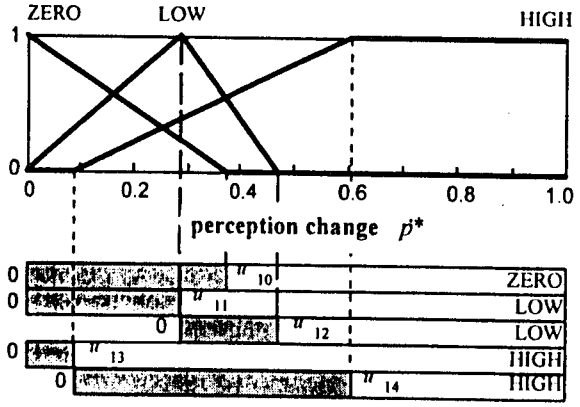


Figura 2c: Codificación de las funciones de pertenencia de los adjetivos del "cambio de la percepción".

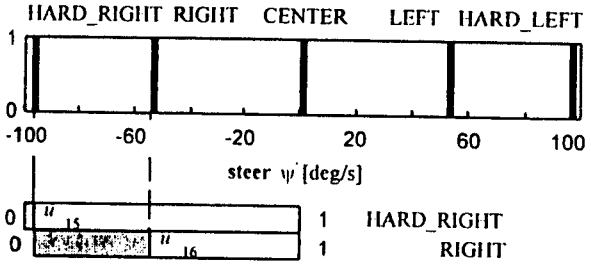


Figura 3a: Codificación de los "singletons" para los adjetivos de dirección. Los adjetivos para LEFT y HARD_LEFT son simétricos, CENTER se mantiene fijo.

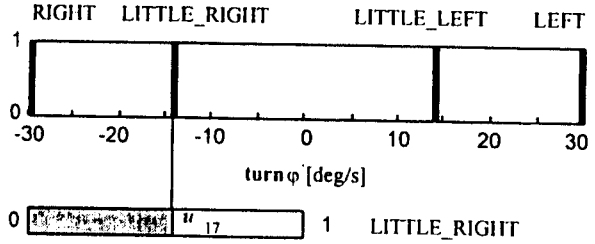


Figura 3b: Codificación de los "singletons" para los adjetivos de aceleración. El adjetivo RIGHT se mantiene fijo y los adjetivos LEFT y LITTLE_LEFT son simétricos.

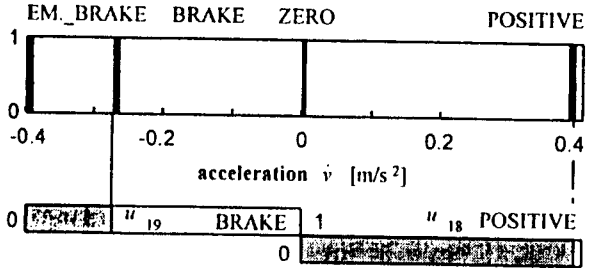


Figura 3c: Codificación de los "singletons" de los adjetivos de aceleración. Los adjetivos para EMERGENCY_BRAKE y ZERO se mantienen fijos.

3.2 Función objetivo o de evaluación

No resulta fácil medir cuantitativamente la actuación de un robot seguidor de paredes. De todas formas, cualquier persona que observara al robot durante cierto tiempo podría ser capaz de juzgar su comportamiento en cuanto a la calidad. Generalmente, un robot es considerado bueno si conserva con precisión una distancia de seguridad respecto a la pared y si se desplaza a gran velocidad pero con el menor número posible de maniobras bruscas. Por otro lado, es inaceptable que el robot choque contra la pared.

Por lo tanto, diseñamos la siguiente función objetivo que sirve como medida de la actuación, teniendo en cuenta estos criterios y tomando como valor de la desviación media ponderada de la percepción ideal 0.5, como valor de la raíz media del cociente de velocidad v/v_{max} y la aceleración a:

$$f = \frac{1}{T}\sum\left[w_1|p-0.5| + w_2\left(1-\frac{v^2}{v_{max}^2}\right) + w_3 a^2 \right]\Delta t \quad . \qquad (6)$$

Los pesos utilizados fueron $w_1 = 2$, $w_2 = 1$, y $w_3 = 20$. Después el robot fue expuesto a cuatro casos diferentes que consideramos importantes en cuanto a la actuación del robot: una pared recta, una esquina, una pared redondeada y un pasillo sin salida. Se generó de nuevo una suma ponderada de cuatro valores de evaluación f_i para obtener un valor de evaluación final f^* de un controlador particular.

$$f^* = d_p \sum_{i=1}^{i=4} s_i f_i \quad . \qquad (7)$$

Se puso un factor de penalización adicional d_p con el valor más pequeño de $2(1-p)$ en el caso de que fueran inferiores a la unidad, $d_p =$ en caso contrario. Este factor asignaba un grado más bajo a los robots que se acercaban demasiado a la pared, aunque fuera sólo durante un instante. Observamos el movimiento de un robot simulado durante 6.5 segundos en cada una de las situaciones mencionadas anteriormente y utilizamos los siguientes valores para los pesos de situación s_i: s_1=1, s_2=2, s_3=2 y s_4=3. Otros investigadores [2], [5] han propuesto similares funciones objetivo, que combinan un criterio análogo mediante multiplicación sin utilizar pesos y exponiendo un controlador a varias condiciones iniciales. Con la función dada en [2] también fuimos capaces de obtener unos resultados aceptables. De todas formas, los pesos ofrecen la posibilidad de diseñar el comportamiento deseado de forma más precisa. Por

ejemplo, si el robot se desplaza demasiado despacio, es suficiente con incrementar el peso w_2.

Experimentar con el robot real hubiera llevado mucho tiempo, es más, hubiera resultado peligroso puesto que el robot debía moverse a gran velocidad sin chocar contra la pared. Por lo tanto, el algoritmo genético utilizó experimentos simulados para determinar la evaluación de cada controlador particular. De todas formas, los resultados fueron verificados probando el mejor controlador con un robot real.

3.3 Optimización

Realizamos varias ejecuciones de optimización utilizando el Algoritmo Genético Simple [3] con diferentes tamaños de población n, desde n=100 hasta n=1000, se estableció un grado de variación p_m=5/n, asumiendo que la importancia de la variación disminuye a medida que aumenta el número de individuos. Un mayor tamaño de población no condujo siempre a unos resultados mejores, presumiblemente porque el grado de variación era ya demasiado bajo, provocando una convergencia prematura y un estancamiento de la población. La figura 4 muestra un desarrollo típico de una población de 250 individuos supervisando la evaluación media de cada generación y la evaluación del mejor individuo.

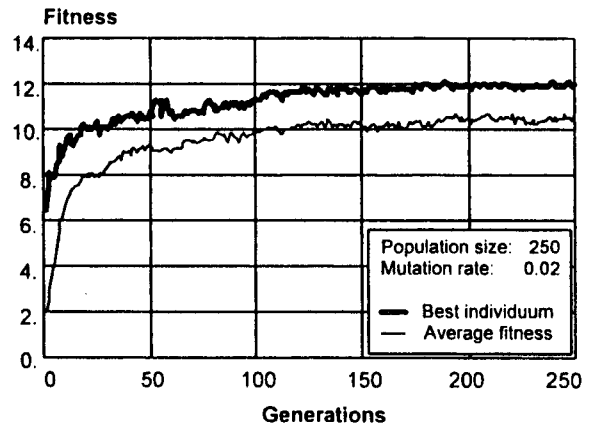


Figura 4: Una típica ejecución del algoritmo genético. El mejor individuo no sobrevive necesariamente.

Referencias

[1] Reinhard Braunstingl, *Fuzzy Logic Wall Following af a Mobile Robot Based on The Concept of General Perception*, pendiente de publicación.

[2] Dario Floreano, Francesco Mondada, *Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-Network Driven Robot*, in D. Cliff, P. Husbands, J. Meyer, and S.W. Wilson, editors, From Animals to Animats III: Proceeding del Third International Conference on Simulation of Adaptive Behavior, Cambridge, MA, MIT Press-Bradford Books, 1994.

[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[4] Bart Kosko, *Neural Networks and Fuzzy Systems*, New Jersey, Prentice-Hall, 1992.

[5] Michael A. Lee, Hideyuki Takagi, *Integrating Design Stages of Fuzzy Systems using Genetic Algorithms*, Second IEEE International Conference on Fuzzy Systems, p.612-617, San Francisco, CA, March, 1993.

[6] Jorge Gasós, *Sistema de navegación basado en lógica borrosa para vehículos autónomos industriales*, Tesis doctoral. Universidad Politécnica de Madrid, Facultad de Informática.

[7] Reinhard Braunstingl, Jokin Mujika, Juan Pedro Uribe, *A Wall Following with a Fuzzy Logic Controller Optimized by a Genetic Algorithm*, FUZZ-IEEE/IFES'95 Fuzzy Robot Competition, p. 77-82, Yokohama, Japan, March 1995.

Supervisor de la Distancia de Seguridad en un Vehículo Basado en Lógica Borrosa

Ramón Ferreiro García
Escuela de la Marina Civil
Dpto. de Electrónica y Sistemas
Universidad de La Coruña
e-mail: ferreiro@des.fi.udc.es

Ramón Juanes Suárez
Joaquín López Lérida
David del Val Latorre
CESAT-UAM Universidad Autónoma de Madrid
e-mail: rjuanes, quino, dval@eii.uam.es

Juan Carlos Sánchez Agrelo
CESAT-UDC Dpto. de Electrónica y Sistemas
Universidad de La Coruña
e-mail: jagrelo@des.fi.udc.es

Resumen

Se propone la implementación de un monitor capaz de informar acerca de la distancia de seguridad a la que debe ser conducido un vehículo para que, en caso de una frenada brusca, no impacte con el vehículo que le precede. Con la aplicación de este tipo de técnicas, se pueden incorporar nuevos elementos de seguridad a los vehículos, lo que puede contribuir a que el número de accidentes de circulación sea menor o éstos sean menos graves.

1 Introducción

La lógica borrosa es una herramienta válida para ser utilizada en la determinación de la distancia de seguridad de un vehículo que circula por una vía pública donde las circunstancias del entorno son diversas y variantes. Se entiende como distancia de seguridad aquella para la cual, accionando el freno, el vehículo recorre una distancia igual o menor a la que le separa del que sigue.

La forma en la que el ser humano razona a la hora de determinar esa distancia de frenado contiene imprecisiones. Ésta depende del estado de humedad del suelo, del estado de las ruedas, de la pendiente de la calzada, de la velocidad del propio vehículo, entre otras causas menos comunes como el estado de rugosidad de la calzada. Es por ello por lo que se propone para el vehículo el empleo de un controlador borroso.

El aspecto más interesante del controlador es la estimación de la distancia de seguridad necesaria en cada instante, en función de las diversas condiciones del entorno. Esta estimación se hace bajo reglas implementadas con lógica borrosa con el fin de usar las misma reglas que emplearía un conductor para determinar esta distancia de seguridad bajo la observación, también borrosamente, de las condiciones del entorno.

Mediante la comparación entre la distancia de seguridad calculada por el controlador y la distancia real al vehículo que le precede se puede avisar al conductor por medio de un testigo digital luminoso de la magnitud de esa diferencia. Por consiguiente, se hace consciente al conductor del grado de peligro en que se encuentra en caso de seguir a esa distancia.

Se piensa que este dispositivo, testigo luminoso digital de la distancia de seguridad al vehículo precedente, puede ser de gran ayuda a la condución de los vehículos actuales, convirtiéndose en una más de las ayudas a la conducción con las que cuenta el automovilista en nuestros días.

2 Planteamiento del Problema

Para la realización de este sistema se tomará como metodología de diseño la propuesta en [1], en donde se ofrece una sistemática para el diseño de controladores borrosos basados en reglas.

La idea de usar lógica borrosa para implementar el sistema es la del deseo de que la distancia mínima de seguridad calculada por el sistema se base en la aplicación de las mismas reglas que emplearía un conductor experto para calcularla, con la ventaja frente al ser humano de que el sistema sería insensible al estado físico y anímico de la persona en cada momento.

La primera etapa de esta metodología consiste en lo que en [1] [3] y [4] se denomina "Adquisición del Conocimiento", en la que se observa el modo de operación de un conductor. La segunda fase es la "Organización de la Base de Conocimiento", donde se identifican las variables utilizadas, se deducen los valores lingüísticos que se emplean, los cuales se usaran para clasificar los valores de las variables medidas, y sus intervalos de medición.

En el problema se tienen como entrada al sistema factores como la distancia real al vehículo que nos precede, la pendiente de la carretera por la que circulamos, el estado del firme, el estado de las ruedas del vehículo, las condiciones climatológicas, y se pueden incluir también aspectos como el estado del conductor, su edad, cuanto tiempo lleva conduciendo, etc.

Para este estudio las variables a emplear y sus intervalos de medición correspondientes serán:

1. Esta investigación ha sido financiada en parte por el plan Nacional de I+D, proyecto TIC92-0267-C02-01.

Ambiente(A):
El estado meteorológico, con los valores Seco o Mojado.
Ruedas(R):
Estado de las ruedas del vehículo, con valores Buen Estado o Mal Estado.
Aparte de estas dos variable binarias, se emplean las dos siguientes:
Pendiente(P):
Pendiente de la pista por la que se circula (Negativa Pronunciada NP, Negativa Suave NS, Nula N, Positiva Suave PS, Positiva Pronunciada PP).
Velocidad(V):
Velocidad del vehículo en cada instante (Baja B, Media M, Alta A).
Estas dos últimas tienen funciones de pertenencia del tipo de las que se muestran en la figura 1, donde también se presenta la función de pertenecia de la Distancia de Seguridad (Muy Pequeña MP, Pequeña P, Justa J, Grande G, Muy Grande MG) :

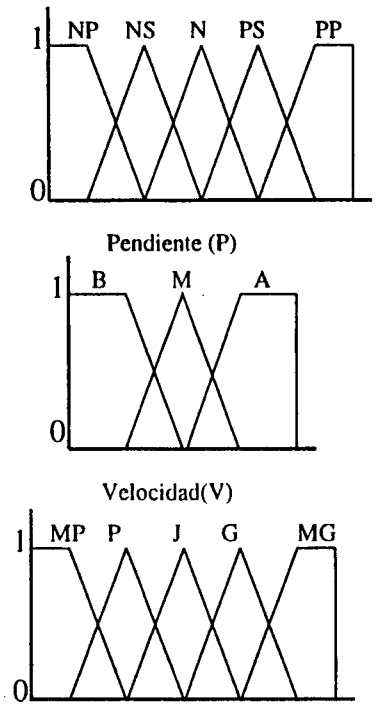


Figura 1 Funciones de pertenencia.

3 Aprendizaje en base al entrenamiento de un modelo

A partir de estas cuatro variables, utilizando un modelo adecuado se obtiene la distancia de seguridad en cada momento. Para la obtención del modelo se debe realizar un entrenamiento, en el que se pueden seguir distintos procedimientos. Uno de ellos consiste en realizar ensayos de frenada sobre un vehículo real. El que se seguirá en este estudio utilizará para la obtención de los datos un modelo de frenada simulado

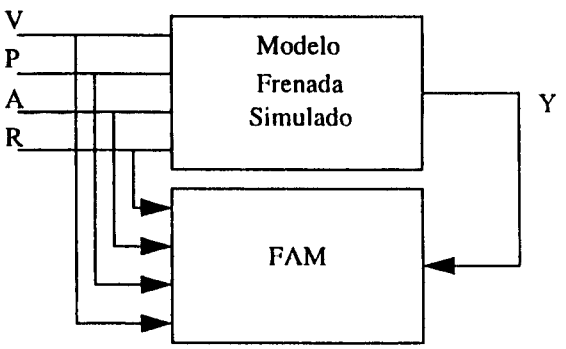


Figura 2 Modelo de frenada simulado.

El modelo utilizado para la simulación de la frenada sigue la ecuación diferencial:

$$M \cdot Y'' + B \cdot Y' = 0$$

En donde $B = f(A, R)$, con lo que se tiene

$$\frac{d^2Y}{dt} = \left(-\frac{1}{M}\right) \cdot f(A, R) \cdot \frac{dY}{dt}$$

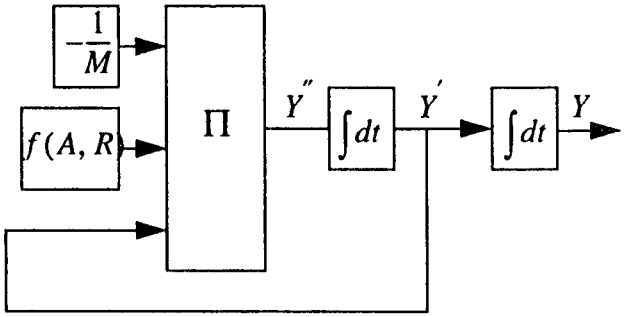


Figura 3 Implementación del sistema.

que está implementado por el sistema de la figura 3
El algoritmo utilizado para entrenar al modelo[4], es el siguiente:

(1) Repetir hasta final de tiempo de entrenamiento
- Leer valores de Entrada/Salida
- Calcular las funciones de pertenencia de cada universo de discurso para las variables de Entrada/Salida. Esto es, la variable de entrada P pertenece al valor j de 1 a 5 y la variable V pertenece al valor k de 1 a 3.
- Almacenar el resultado de sumar el valor actual de la variable de salida al anterior como,

$$Y_T(j,k) = Y_T(j-1,k-1) + Y_T(j,k)$$

$$N(j,k) = N(j-1,k-1) + 1$$

donde j, k son las dimensiones para definir los grados de libertad de la memoria asociativa.

(2) Fin del bucle

(3) Calcular la media aritmética y almacenar los resultados para hacer la base de la regla del hipercubo como,

$$Y_F(j,k) = \frac{Y_T(j,k)}{N(j,k)}$$

Si las variables de Entrada/Salida cumplen los criterios deseados, entonces la base de reglas se considera correcta y puede usarse como base de reglas a usar del sistema.

Una vez realizado el entrenamiento se obtienen las siguientes bases de reglas, que nos proporcionan la distancia de seguridad, para los cuatro valores posibles que toma el par de variables (A,R).

V \ P	-0.05	-0.02	0	0.02	0.05
5	10	9	8	7.5	7
11	22	20	18	16.5	15
17	35	31	28	26	24
23	47	42	38	35	32
29	59	53	48	44	40

Tabla 1: A=300 ; R=300 (Ambiente Seco, Ruedas en buen estado)

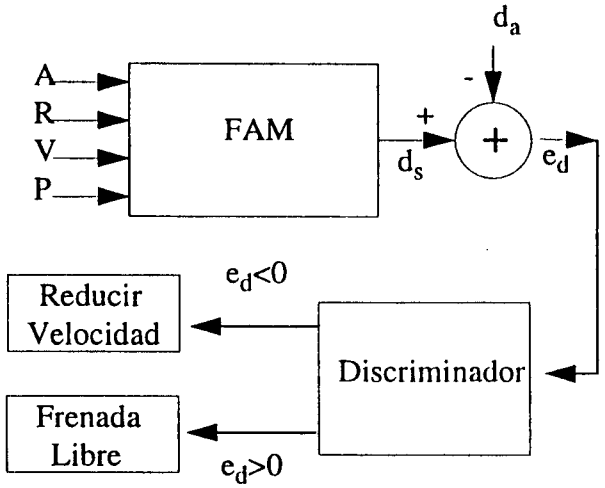
V \ P	-0.05	-0.02	0	0.02	0.05
5	12	11	10	9	8
11	27	24	22	20	18
17	42	37	34	31	28
23	56	50.5	46	42	38
29	71	64	58	52.5	48

Tabla 2:
A=200;R=300(Amb. Húmedo, Ruedas en buen estado)
A=300;R=200 (Ambiente Seco, Ruedas en mal estado)

V \ P	-0.05	-0.02	0	0.02	0.05
5	15	13.5	12	11	10
11	33	30	27	25	23
17	51	46	42	38	35
23	69	62	56	52	47.5
29	87	78	71	65	60

Tabla 3: A=200 ; R=200 (Ambiente Húmedo, Ruedas en mal estado)

Estas reglas se guardan en una FAM [2] que luego se utilizarán en el sistema de monitorización de la distancia minima de seguridad. Este sistema puede tener una forma como la que se aprecia en la siguiente figura.



d_s: distancia de seguridad
d_a: distancia actual
e_d: error de distancia

Figura 4 Sistema de monitorización de la distancia mínima de seguridad.

4 Validación y Conclusiones

Con el modelo propuesto, se ha ensayado el comportamiento de un vehículo de masa 1000 Kg. cuyos resultados se muestran en la base de reglas representada en la figura 3. Cada tipo de vehículo presenta características diferentes, proporcionando así mismo un base de reglas distinta en cada caso. Las condiciones ambientales y el estado de las ruedas contienen incertidumbre en su apreciación y la reducción del problema a considerarlas variables booleanas limita la precisión de los resultados.

Referencias

[1] García Alegre, María C., Inteligencia Artificial en el Control de Procesos. Controladores Borrosos. *Mundo Electrónico*, nº 214, Febrero 1991, pp 42-49.

[2] Bart Kosko, *Neural Networks and Fuzzy Systems*. Prentice Hall International, Inc. Chap 8. pp. 299-335, 1992.

[3] Ferreiro García, R., Associative Memories as Leaning Basis Applied to Roll Control of an Aircraft. *Proceedings of Multivariable System Techniques* (AMST'94), pp. 23-29. Ed. by R.Whalley. University of Bradford. U.K., 1994.

[4] Ferreiro García, R., Fam on Gain Scheduling Control of Highly Nonlinear and Disturbed Process. *Proceedings of Fifth International Conference IMPU* (Information Processing and Management of Uncertainty) in Knowledge-Based Systems. Paris. pp. 463-466. ISBN 2905433-26-4, 1994

Integración de Controladores Difusos en un Robot Autónomo

Vicente Matellán
Lab. de Agentes Inteligentes
Departamento de Ingeniería
Universidad Carlos III de Madrid
e-mail: vmate@ing.uc3m.es

José Manuel Molina
Lab. de Agentes Inteligentes
Departamento de Ingeniería
Universidad Carlos III de Madrid
e-mail: molina@ing.uc3m.es

Camino Fernández Llamas
CETTICO
Universidad Politécnica de Madrid
e-mail: camino@fi.upm.es

José Manuel Blázquez
Lab. de Agentes Inteligentes
Departamento de Ingeniería
Universidad Carlos III de Madrid
e-mail: jblazque@zenon.ing.uc3m.es

Resumen

En este trabajo se propone una metodología que permite integrar distintos comportamientos difusos de un robot autónomo real. Esta integración se realiza a través de una arquitectura formada por dos niveles. El primer nivel está compuesto por diferentes controladores borrosos, cada uno especializado en una tarea distinta. Así definido, el sistema completo se basa en una aproximación reactiva, es decir, la decisión de que acciones realizar se toma de forma continua en función de la información que se recibe de los sensores.

El segundo nivel se encarga de decidir cómo generar el comportamiento global a partir de las propuestas de los distintos módulos del nivel inferior. Las decisiones de este módulo se obtendrán mediante razonamiento borroso. Los resultados obtenidos tanto en simulación como en el robot real muestran un comportamiento más eficiente en este sistema que en uno que seleccione la acción básica más prometedora en cada situación.

Palabras clave: *Robot autónomo, comportamiento difuso, integración, multinivel.*

1 Introducción

Desde los primeros pasos de la Inteligencia Artificial se ha abordado el problema de construir robots móviles autónomos capaces de realizar tareas útiles en el mundo real. Sin embargo, las tentativas basadas en los paradigmas clásicos como la abstracción, planificación, búsqueda heurística, etc., como por ejemplo [9], no tuvieron mucho éxito y este campo permaneció en un estado teórico hasta finales de los años 80.

En los últimos años ha tenido lugar un profundo cambio y la mayoría de los robots se construyen ahora en base a un nuevo paradigma: *reacción*, que sostiene que la "inteligencia"de un sistema es el resultado de la interacción del sistema con el entorno. Así, el comportamiento inteligente de un robot autónomo debería surgir como resultado de su interacción con el mundo real a través de sus sensores y actuadores.

La arquitectura que se presenta se basa en dos premisas. La primera es que los robots autónomos deben confiar sus actividades básicas a una aproximación reactiva [3]. Esto quiere decir que el comportamiento global de un robot debe ser el resultado de una fusión no óptima de diferentes reacciones básicas, que denominaremos *comportamientos*. Los comportamientos están diseñados para operar continuamente sobre los actuadores del robot en función de las informaciones proporcionadas por los sensores, con un objetivo concreto, por ejemplo, evitar obstáculos, seguir un objeto, etc.

La segunda premisa es que la lógica difusa [12] ha demostrado ser un entorno robusto para el desarrollo de controladores eficientes en aplicaciones diversas, como por ejemplo [10], [2]. Por tanto, los comportamientos a los que nos hemos referido en los párrafos anteriores pueden construirse como grupos de reglas difusas. Así, en la literatura han aparecido ultimamente distintas aproximaciones basadas en controladores borrosos al control de robots, como por ejemplo, [1], [6], [7].

Nuestra propuesta se desarrolla en las cinco secciones siguientes. En la segunda sección se utiliza el razonamiento borroso dentro de este nuevo paradigma para la construcción de robots autónomos. En el tercer apartado se define de forma teórica la arquitectura del sistema propuesto. La cuarta sección es una implementación sobre un robot real de la arquitectura anterior. El siguiente apartado muestra los resultados obtenidos con dicha implementación y en el último apartado se presentan las conclusiones y las futuras líneas de trabajo.

2 Robots Reactivos

Se pueden considerar dos aproximaciones reactivas distintas al control de un robot autónomo. La primera de ellas consideraría la totalidad de la información aportada por los sensores para obtener la salida más adecuada a la situación en que se encuentra el robot

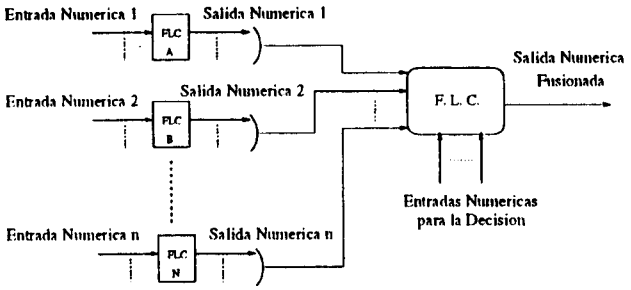


Figura 1: Proceso de Fusión Borrosa

mediante un conjunto de reglas que utilizan las entradas de forma conjunta. Una implementación borrosa del sistema anterior conllevaría la generación de un sistema de reglas muy complejo, tanto en número de reglas (situaciones a considerar) como en complejidad de las mismas, debido a la cantidad de sensores a considerar.

La segunda de las arquitecturas utiliza una descomposición por tareas. Es decir, cada comportamiento debe decidir por sí mismo qué hacer en cada instante utilizando un subconjunto reducido de la información suministrada por los sensores. Una instancia clásica de esta clase de arquitecturas es la *subsumption architecture* [4], [5], donde las salidas de los distintos módulos se integran en base a la inhibición de aquellas menos relevantes. La *subsumption architecture* utiliza máquinas de estado finito para implementar los comportamientos.

La aportación del trabajo que se presenta reside en una aproximación diferente al segundo tipo de arquitectura, basándose en la lógica difusa. En primer lugar, se han diseñado los comportamientos de forma borrosa, debido a que se considera que las reglas difusas permiten utilizar conceptos imprecisos de modo más intuitivo que, por ejemplo, las máquinas de estado finito. En segundo lugar, se ha desarrollado un método difuso para fusionar el conjunto de salidas de los distintos comportamientos, en vez de, por ejemplo, elegir una de ellas mediante inhibiciones.

3 Razonamiento Fuzzy Multinivel

El objetivo de este trabajo es mostrar que un comportamiento complejo en un robot puede emerger de comportamientos simples implementados como conjuntos de reglas difusas. La arquitectura presentada en la figura 1 tendrá n acciones básicas implementadas como controladores fuzzy, FLC (*Fuzzy Logic Controller*) y un módulo encargado de decidir cómo se formará la acción global a partir de las propuestas por las distintas unidades reactivas.

El primer paso para construir el sistema completo es diseñar cada FLC individualmente. Cada uno de ellos deberá ser capaz de controlar el robot para que consiga la tarea de la que se encarga. Para diseñar cada uno de esos comportamientos se necesitará [13]:

- Definir las variables de entrada y salida. Esto es, determinar qué fenómenos se observarán y cuales son las acciones de control que se pueden tomar.

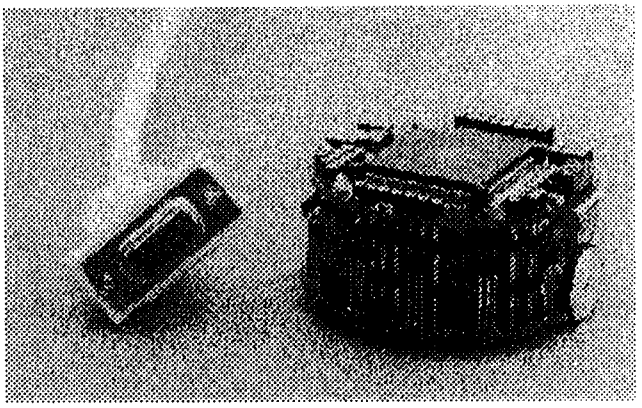


Figura 2: El mini-robot Khepera

- Definir la forma en que las observaciones del mundo se expresan como conjuntos difusos.
- Diseñar la base de reglas.
- Determinar la forma en la que las salidas fuzzy se transforman en acciones de control.

Las entradas del sistema son percepciones del mundo, pero no son las mismas para todos los FLC. Cada comportamiento se basa en una visión particular del mundo, por ejemplo, el comportamiento *evitar obstáculos* tiene que ser consciente de las distancias a los objetos que rodean al robot, mientras que un comportamiento como *clasificar objeto* necesitará reconocer los bordes del objeto para clasificarlo según las formas predefinidas almacenadas en el robot.

Para terminar el primer paso se prueba cada FLC en un entorno preparado para generar únicamente los estímulos que activan uno de los comportamientos. Estos experimentos parciales son necesarios para afinar cada FLC y conseguir que la reacción actue de forma adecuada. En esta fase es posible modificar las entradas, las funciones de pertenencia y la base de reglas.

El segundo paso es construir un sistema capaz de encontrar una buena salida para una determinada situación del robot en el mundo. La primera aproximación sería un sistema que eligiese solamente un comportamiento, es decir, la salida de un comportamiento. Este módulo de decisión ha sido denominado **Sistema 1/n** (elección de 1 sobre n posibles). Por ejemplo:

SI
(Situación **ES** *situación-1*)
ENTONCES
(Comportamiento **ES** *comportamiento-1*)

La dificultad de utilizar esta aproximación radica en establecer los límites entre las distintas situaciones percibidas. Una solución podría ser el uso de una definición difusa de las situaciones. La representación borrosa de las situaciones permite evaluar el grado de correspondencia entre una situación real y la situación predefinda. En este tipo de sistemas, la salida es una combinación de las respuestas predefinidas para cada situación en función del grado de correspondencia. Por

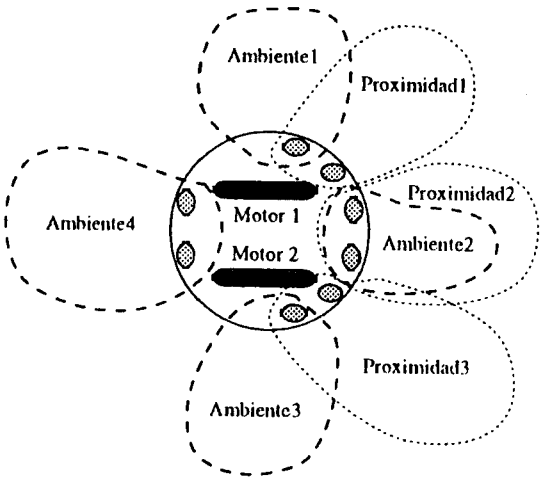


Figura 3: Disposición física de los sensores

ejemplo, en [11] se presenta un sistema donde se realiza una integración fuzzy de acciones predefinidas, no obtenidas mediante razonamiento borroso, para cada una de las situaciones posibles, estas respuestas son del tipo: parar y girar a la izquierda, acelerar y girar a la derecha etc. Este sistema estará implementado mediante reglas del tipo:

SI
(Situation-fuzzy-1 **ES** *Alta*)
ENTONCES
(Parar **Y** Girar-izquierda)

En nuestra arquitectura no hay acciones con valores predefinidos porque cada uno de los comportamientos básicos obtiene la salida apropiada a cada situación mediante un razonamiento borroso. Para ello los comportamientos básicos se implementan como FLCs. La integración de las acciones obtenidas de los FLCs se basará en el grado de correspondencia borroso entre las situaciones reales y las predefinidas. La integración se realizará mediante reglas del tipo:

SI
situation-fuzzy-1 **ES** *Alta*
ENTONCES
(comportamiento-1 **ES** *Alto*) **Y**
(comportamiento-2 **ES** *Bajo*)

⋮

SI
(situation-fuzzy-n **ES** *Alta*)
ENTONCES
(comportamiento-1 **ES** *Bajo*) **Y**
(comportamiento-2 **ES** *Muy-alto*)

Este sistema es capaz de obtener una respuesta que corresponde a un estado intermedio de todos los FLCs, obteniendo la importancia de cada comportamiento en la situación actual. Para mezclar las salidas numéricas es posible hacer una combinación lineal. Por tanto, si cada comportamiento tiene una salida v_i y el módulo de decisión asigna un grado de adecuación s_i del comportamiento i a la situación percibida, la salida fuzzy viene dada por la fórmula 1, donde los pesos α_i se calculan mediante un proceso de normalización de los s_i.

$$v_{total} = \sum_{i=1}^{n} (\alpha_i \times v_i) \qquad (1)$$

Para fijar estos grados de adecuación se tiene que utilizar un conjunto de entradas. Estas entradas dependen del objetivo del robot y del número de comportamientos implementados. En particular, para un robot con dos comportamientos, *perseguir un objeto* y *evitar obstáculos*, la entrada del módulo de decisión será la descripción del entorno. En la siguiente sección presentamos la descripción completa de una implementación real.

4 Una arquitectura real de dos niveles

La arquitectura teórica presentada en la sección anterior se ha desarrollado en un simulador y luego se ha probado, adaptándola, en un robot real. Se ha utilizado un simulador del robot real, denominado SRA (Simulador de Robots Autónomos), desarrollado en la Universidad Carlos III. Este simulador nos permite diseñar un entorno sencillo para probar distintas implementaciones de un comportamiento antes de evaluar las mejores en el robot real.

El robot que se ha utilizado en los experimentos es el mini-robot denominado Khepera [8], desarrollado en el LAMI-EPFL (Suiza). El Khepera, figura 2, es un mini-robot de 5.5 cm. de diámetro diseñado como demostrador tecnológico. Tiene ocho sensores de proximidad basados en infra-rojos, que también sirven como sensores de luz ambiente y dos motores con contadores. Los sensores devuelven un valor numérico entre 0 y 1023, dependiendo de si se encuentra lejos o cerca del objeto. La distribución de los sensores se puede observar en la figura 3. Así mismo, tiene la posibilidad de funcionar de forma autónoma, o bien, conectado a un ordenador mediante un cable serie.

En las siguientes secciones se describe el sistema en dos pasos. En la primera se aborda cada uno de los comportamientos individualmente. En la segunda, se analizan las distintas posibilidades para la realización del módulo de fusión, comenzando con el que denominaremos Sistema 1/n y analizando a continuación el proceso de integración fuzzy.

4.1 Los comportamientos individuales

El primer comportamiento considerado se ha denominado *evitar obstáculos*. Este comportamiento permite al robot deambular por el mundo sin chocar con ningún obstáculo. El sistema borroso desarrollado, que define este comportamiento, se ha probado en primer lugar sobre el simulador SRA. En la figura 4 se muestra la estructura del controlador difuso que implementa el comportamiento, es decir, las funciones de pertenencia de las variables de entrada, las tablas que muestran las reglas de control y las funciones de pertenencia de las variables de salida.

En el proceso de *fuzzificación* se pueden observar los valores concretos de la variable *distancia*, su rango abarca de 0 a 1023 que es el rango de valores devueltos por los sensores, este rango se ha dividido en sólo tres

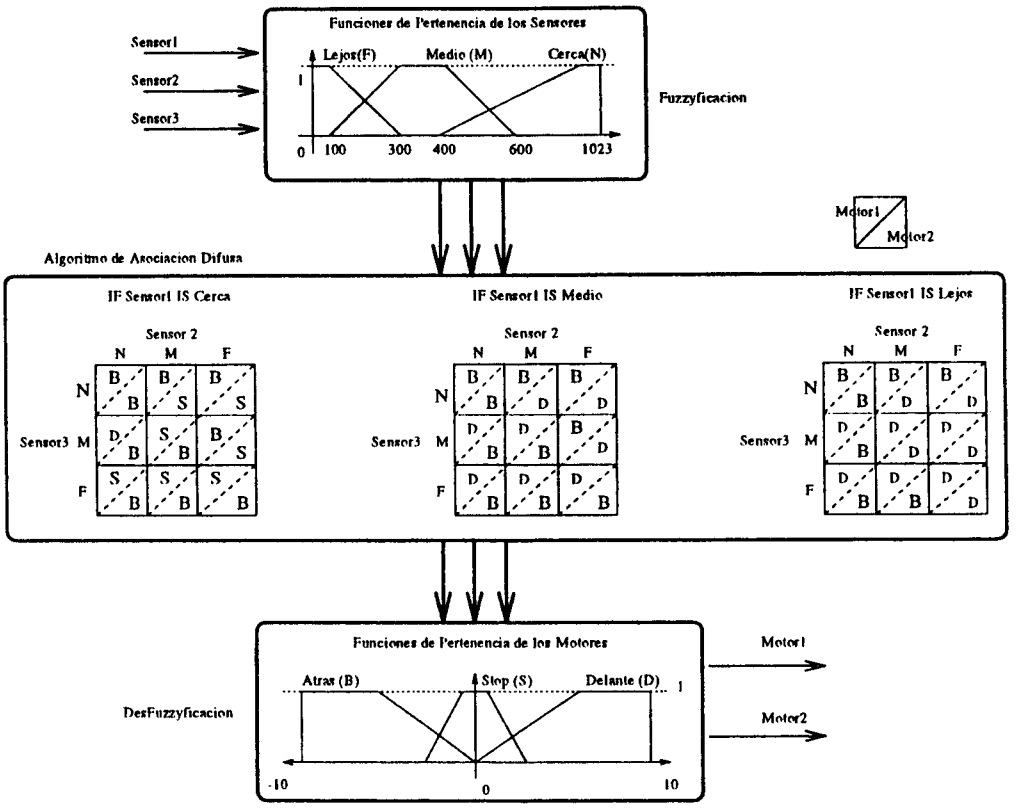


Figura 4: FLC "evitar obstáculos"

etiquetas debido a la pobre discriminación de los sensores. Las reglas se muestran en las tablas del segundo bloque de la figura, así por ejemplo, la regla correspondiente al primer cuadro de la tabla de la izquierda se interpretar como:

SI
(Sensor1 **ES** *Cerca*) **Y**
(Sensor2 **ES** *Cerca*) **Y**
(Sensor3 **ES** *Cerca*)
ENTONCES
(Motor1 **ES** *Atras*) **Y**
(Motor2 **ES** *Atras*)

El rango de valores de la variable *velocidad* va de -10 a 10 que es el rango al que hemos limitado la velocidad de los motores. Sobre dicho rango se han definido tres etiquetas, *Atras, Stop y Delante*, que son suficientes para controlar el robot.

En el caso del comportamiento *evitar obstáculos*, la variable *distancia* se define con las mismas funciones de pertenencia para los valores de entrada suministrados por los tres sensores. En el caso de que la curva de respuesta de alguno de los sensores fuese distinta, por ejemplo, por deficiencias de fabricación, habría que variar las funciones de pertenencia para ajustarlas al valor real.

A continuación, se ha desarrollado otro comportamiento denominado *seguir objeto* que hace que el robot siga un objeto. Para implementar este instinto primario en simulador se han creado dos sensores que no existen en el robot real y que se han denominado distancia y ángulo. Dichos sensores devuelven la distancia al objeto a seguir y el ángulo que presenta con respecto a la dirección del robot. En el robot real se ha utilizado una fuente de luz como objetivo a seguir y se han utilizado los sensores de luz ambiente para determinar la distancia y el ángulo con respecto a dicha fuente.

4.2 Integración de comportamientos

Simplemente con los dos comportamientos descritos en el apartado anterior, ya es necesario un módulo que integre las respuestas. Este ejemplo sirve también para ilustrar cómo los distintos comportamientos tienen distintas entradas. Así, el encargado de evitar obstáculos tiene como entradas las medidas de los sensores de proximidad, mientras que el encargado de seguir un objeto recibe como entradas el ángulo y la distancia al objeto. Las salidas de ambos es la misma; la velocidad de cada una de las ruedas.

Una primera posibilidad de realizar la fusión de las salidas de ambos comportamientos es elegir la mejor salida para cada situación de entre las propuestas por los comportamientos, siguiendo la filosofía de elegir la salida del comportamiento *seguir objeto* mientras no haya peligro de colisión. A esta aproximación, basada en la *subsumption architecture* de R.A. Brooks, es a la que hemos denominado el *Sistema 1/n*. La segunda posibilidad es realizar una fusión fuzzy como se describe en la sección anterior. A esta segunda arquitectura es a la que denominaremos *Sistema Fuzzy*.

5 Experimentación

Para probar la arquitectura se ha programado el sistema sobre el simulador. Se han implementado ambos comportamientos y se ha procedido a generar un conjunto de escenarios mediante la modificación del entorno: tanto las posiciones de partida como el número y la forma de los objetos. En concreto, se han analizado exhaustivamente doce escenarios que incluyen diferentes situaciones. En cada uno de estos escenarios se ha medido la respuesta de los dos sistemas: 1/n y difuso.

Los resultados obtenidos se evalúan a través de la medida del tiempo necesario para alcanzar el objetivo, la distancia recorrida por cada uno de los sistemas y las oscilaciones, medidas como el sumatorio de las variaciones de la dirección. Como ejemplo se presentan las distintas trayectorias, figura 5, en cuatro escenarios distintos. En la figura puede observarse cómo el sistema borroso sigue trayectorias con menos oscilaciones y por consiguiente la distancia recorrida es menor. El menor número de oscilaciones se debe a la transición suave de un comportamiento a otro en el sistema fuzzy. Además este sistema hace que la trayectoria del robot vaya más próxima al contorno de los obstáculos, lo cual hace que el tiempo y la distancia recorrida para alcanzar la meta sean más cortos en el sistema difuso que en el 1/n.

En la implementación sobre el robot real se han utilizado todos los sensores agrupándolos según se muestra en la figura 3. En particular, para el comportamiento *evitar obstáculos* se han utilizado tres agrupaciones de los seis sensores frontales de distancia, que en la figura se denominan como de *proximidad*. Para el comportamiento *seguir objeto*, se han agrupado los ocho sensores de luz ambiente en cuatro: tres frontales y uno trasero, denominados en la figura 3 como de *ambiente*. La integración de las salidas de ambos comportamientos hace que el robot sea capaza de seguir una fuente de luz, evitando los obstáculos que encuentra en su movimiento.

Escenarios	escenario1		escenario2	
Sistema	Fuzzy	1/n	Fuzzy	1/n
Tiempo (seg.)	205	211	159	160
Distancia (mm.)	508	432	348	339
Oscilaciones	9	17	6	7

Escenarios	escenario3		escenario4	
Sistema	Fuzzy	1/n	Fuzzy	1/n
Tiempo (seg.)	192	195	235	215
Distancia (mm.)	474	409	543	457
Oscilaciones	8	16	8	15

Tabla 1: Resultados de los experimentos reales.

Sobre el robot real los resultados se evalúan en base a los mismos criterios que en el simulador (distancia, tiempo y oscilaciones), obteniéndose los datos a partir de los sensores internos que reflejan el desplazamiento real del robot; odómetros de cada rueda. A partir de los datos obtenidos en las pruebas experimentales realizadas con el Khepera, análogos a los resultados co-

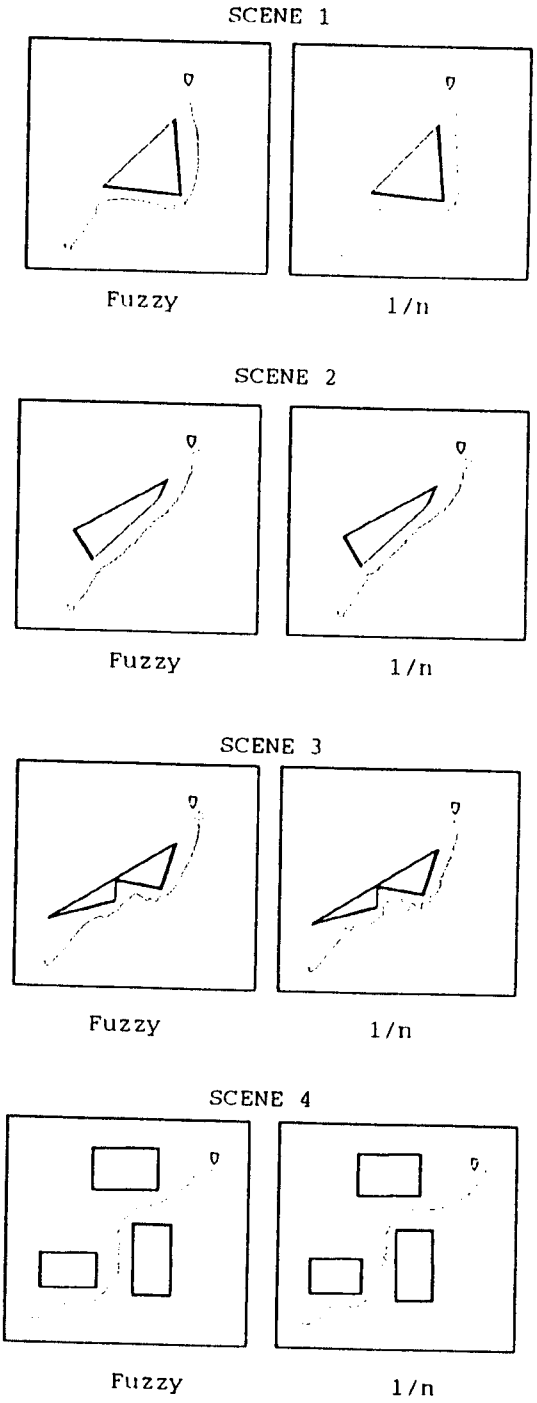


Figura 5: Comparación de dos sistemas

mentados previamente en el simulador, el mejor rendimiento del sistema borroso ha quedado probado, como muestra en la tabla 1.

6 Conclusiones y trabajos futuros

Se ha presentado una metodología basada en los conceptos de lógica borrosa y en las arquitecturas reactivas, para integrar distintos comportamientos para la navegación de un robot móvil. Partiendo del uso de variables cuantitativas se ha pasado al uso de descripciones linguísticas similares a las utilizadas en el lenguaje natural. De esta forma se facilita el tratamiento de un problema matemáticamente muy complejo.

Este trabajo presenta una arquitectura teórica, con n niveles y una implementación en un robot real con dos niveles. La evaluación del comportamiento global del robot muestra que el sistema con fusión difusa tiene un comportamiento superior al de aquellos que realizan una fusión tradicional.

Entre los trabajos futuros se puede incluir una primera tarea de prueba de la arquitectura con mayor número de sensores. Con una percepción mayor del entorno, el robot tendrá más información y se espera que muestre un comportamiento más inteligente. Este tipo de extensión sería inabordable en forma de un único controlador fuzzy al incrementarse exponencialmente su complejidad. Con el tipo de arquitectura propuesta sería posible obtener un comportamiento complejo sin un gran incremento de la complejidad del sistema.

El siguiente tipo de prueba consistirá en aumentar el número de comportamientos. En ese sentido se pueden añadir comportamientos como bordear un obstáculo o crear mapas. Este segundo grupo de pruebas demostrarán la habilidad del sistema para abordar tareas más complejas.

Por último, indicar que el interés básico de investigación del Grupo de Agentes Inteligentes de la Universidad Carlos III es el desarrollo de modelos cognitivos que permitan la cooperación de agentes, tanto software (asistentes personales, sistemas expertos, etc.) como hardware (robots), en la resolución de problemas complejos. Así, la arquitectura de comportamientos fuzzy presentada es el nivel reactivo de un modelo multinivel para la cooperación de robots en tareas simples (traslado de objetos, limpieza de suelos, etc), en que se está trabajando dentro del grupo.

Referencias

[1] Badreddin, E., *Fuzzy Relations for Behavior-Fusion of Mobile Robots.* Proceedings of IEEE Int. Conference on Robotics and Automation, 3278-3283, 1994.

[2] Belmonte-Serrano, M., Sierra, C., Lopez de Mántaras, R., *RENOIR: An Expert System Using Fuzzy Logic for Rheumatology Diagnosis.* International Journal of Intelligent Systems, Vol. 9, 985-1000, 1994.

[3] Braitenberg, V., *Vehicles. Experiments in Synthetic Psycology.* The MIT Press, Cambridge, Massachusetts, 1984.

[4] Brooks, R.A., *A Roboust Layered Control System for a Mobile Robot.* IEEE Journal of Robotics and Automation RA-2, 14-23, 1986.

[5] Brooks, R.A., *Intelligence without representation.* Artificial Intelligence, 47, 139-159, 1991.

[6] Goodridge, S. G., Luo, R.C. *Fuzzy Behavior Fusion for Reactive Control of an Autonomous Mobile Robot: MARGE* Proceedings of IEEE Int. Conference on Robotics and Automation, 1622-1627, 1994.

[7] Ishikawa, S. *A method of autonomous mobile robot navigation by using fuzzy control.* Advanced Robotics, Vol. 9, No. 1, pp. 29.52, 1995.

[8] Mondada, F., Franzi, P.I., *Mobile Robot miniaturisation: A tool for investigation in control algorithms.* Proceedings of the third International Symposium on Experimental Robotics. Kyoto, Japan, 1993.

[9] Nilsson, N.J. (ed.), *Shakey the robot.* SRI A.I. Center Technical Note 323, April, 1984.

[10] Pin, F.G., Watanabe, Y, *Driving a Car using Reflexive Fuzzy Behaviors.* Proceedings of the Second International Conference on Fuzzy Systems, pag. 1425-1430, San Francisco, 1993.

[11] Reignier, P., *Fuzzy logic techinques for mobile robot obstacle avoidance.* Robotics and Autonomous Systems, 12, pag. 143-153, 1994.

[12] Zadeh, L.A., *Outline of a New Approach to the Analysis of Complex Systems and Decision Processes.* Trans. on SMC, vol 3, 1, January 1973.

[13] Zimmerman, H.J., *Fuzzy Sets Theory and Applications.* Kluwer, Boston, 1992.

Control Fuzzy de la Velocidad en un Sistema Ferroviario a Escala*

Gerardo Bandera Burgueño
Dpto. Arquitectura de Computadores
Facultad de Informática
Universidad de Málaga
e-mail: bandera@atc.ctima.uma.es

Jesús María Vicente del Rey
Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingenieros Industriales
Universidad de Málaga
Plaza El Ejido, s/n. 29013 Málaga

Resumen

El trabajo que aquí se presenta ha consistido en el estudio, diseño e implantación de un sistema de control de la posición, la velocidad y la aceleración de un conjunto de trenes a escala reducida que circulan por un pequeño sistema ferroviario. Se han utilizado para ello tanto técnicas clásicas de control, basadas en los tradicionales reguladores PID, como estrategias basadas en el control borroso directo con objeto de realizar una comparación de los resultados obtenidos con ambas estrategias. Asimismo, se ha pretendido que dicho sistema ferroviario constituya una plataforma real de trabajo que haga posible una futura experimentación de diferentes técnicas avanzadas de control.

Palabras Clave: Control de procesos, lógica borrosa, control borroso, razonamiento aproximado.

1 Introducción

En los últimos años, una buena parte del esfuerzo investigador en control de procesos se ha dedicado al control de sistemas que presentan no linealidades en su comportamiento. Esta clase de sistemas son difícilmente modelables y controlables por los métodos tradicionales; tal es el caso de del empleo de controladores P, PD, PI, PID y sus variantes. Uno de los resultados del esfuerzo investigador ya referido es la aplicación en ese tipo de procesos de **técnicas de control borroso** o difuso. Estas técnicas se apoyan en el empleo de un protocolo de control basado en reglas lingüísticas (o expresables en palabras) muy simples, construidas en función del conocimiento heurístico que se posee sobre el comportamiento del proceso que se debe controlar. Estas nuevas estrategias están teniendo gran auge en los últimos años, aún habiendo sido introducida la lógica borrosa en 1965 por Lofti Zadeh.

En este trabajo se presenta una aplicación de estas técnicas a un sistema ferroviario a escala en el que están fuertemente presentes los citados problemas de no linealidades que dificultan obtener buenos resultados de control mediante técnicas clásicas. Del mismo modo, se ha pretendido realizar una comparación de los resultados de la aplicación de ambas estrategias a este problema particular.

Además, a través de este trabajo, se ha intentado crear una pequeña plataforma real de experimentación en diferentes estrategias de control borroso así como en otras técnicas de control inteligente.

2 Descripción general del sistema

La plataforma de experimentación construida consiste en un sistema ferroviario a escala 2N constituido por tres bucles de vías enlazados entre sí por una serie de ramales comunes, sistema por el que circula un par de trenes que en su recorrido comparten un conjunto de tramos del mismo. Un esquema de este sistema se muestra en la figura 1.

Los recorridos de ambos trenes en el circuito ferroviario son los siguientes: el tren 0 circula por las zonas 0, 1, 2, 3, 4, 1, 0, ... según el sentido mostrado en la figura 1, mientras que el tren 1 recorre las zonas 5, 3, 2, 1, 4, 3, 5, ...

En dicho sistema por un lado se ha realizado un seguimiento de la posición de los trenes y una asignación de tramos o zonas de vía y cruces evitando que pueda existir una colisión entre los móviles. Para ello se ha diseñado un automatismo basado en redes de Petri [10] puesto que esta técnica de diseño está especialmente indicada en sistemas que, como éste, funcionan de forma concurrente y comparten recursos.

Por otra parte se ha realizado el control de la velocidad de circulación de los trenes, que se ha fijado a

* Este trabajo ha sido subvencionado por el proyecto TAP93-0608 de la CICYT.

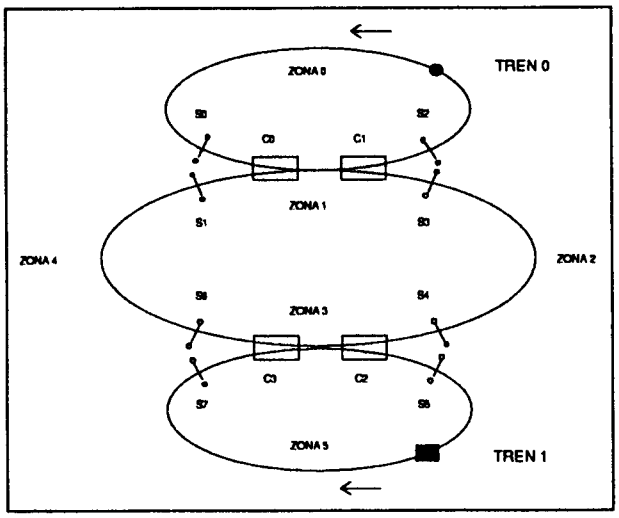


Figura1: Estructura del sistema ferroviario.

valores diferentes en distintos tramos de sus recorridos (así, por ejemplo, se desean velocidades menores mientras los trenes circulan por las zonas de cruces con objeto de evitar descarrilamientos). Además se ha llevado a cabo un control de las aceleraciones y deceleraciones necesarias para hacer que los cambios de velocidades sean lo menos bruscos posible. De este modo, cuando los trenes se aproximan a las zonas de cruce se inicia una deceleración progresiva de los mismos que continua hasta su frenado total si la zona a la que éste ha de acceder está ocupada o hasta la velocidad de paso por el cruce en caso de estar disponible para el tren.

Para controlar el acceso de los trenes a los tramos de vías compartidos se han incorporado al sistema una serie de sensores de presencia por infrarrojos que aparecen referenciados en la figura anterior como 'Si' (i = 0, 7).

Por otro lado, para hacer posible un seguimiento de los trenes a lo largo de su recorrido, se decidió llevar a cabo una cuenta del número de traviesas recorridas por los mismos con objeto de poder determinar en cada momento la posición de cada uno de los móviles que intervienen en el sistema, así como su velocidad a través del conocimiento del número de traviesas recorridas en un periodo de tiempo determinado.

La totalidad del sistema de control del proceso ha sido implantado en un computador compatible PC, dotado de una serie de tarjetas de entrada/salida.

3 Estrategias de control de la velocidad de los trenes

3.1 Regulador tipo PID

Como se indicó anteriormente, una de las técnicas de control empleadas para la regulación de la velocidad de los trenes que circulan por el sistema ferroviario ha consistido en el empleo de sendos reguladores digitales tipo PID para ambos móviles [8].

Diferentes reguladores fueron sintonizados empleando la técnica de Ziegler - Nichols en bucle abierto [7, 9], considerando el comportamiento dinámico de los trenes en las proximidades de los diferentes puntos de trabajo de los mismos.

Una primera opción empleada consistió en el uso de un único controlador PID para cada tren en todo el espacio de trabajo del mismo. Los resultados poco satisfactorios obtenidos mediante el empleo de esta técnica llevaron a proponer una serie de alternativas de diseño del regulador. Por un lado, en la línea de lo propuesto por Astrom y Witternmark en [1], se consideró una disminución lineal de los parámetros derivativo e inverso del integral en las zonas próximas a la saturación de los controladores. Con ello se consiguió una mejora de estabilidad del sistema de control en dichas zonas. Y por otra parte se contempló el empleo de diferentes controladores PID para un mismo tren, previamente sintonizados para un funcionamiento óptimo en una determinada zona de trabajo, considerando tanto saltos graduales como bruscos de los parámetros de los controladores en los momentos de tránsito de una zona de trabajo a otra.

3.2 Regulador borroso

Como ya se ha referido, los métodos de control borroso están basados en las técnicas de razonamiento aproximado propuestas por Zadeh.

La idea básica de todos estos reguladores es la de incorporar la experiencia del operador o técnico del proceso expresada en forma lingüística mediante sentencias del tipo: "si aumenta la temperatura la presión también aumenta y viceversa". Mediante dichas sentencias se expresan las relaciones cruzadas existentes entre las variables de control y las variables de medida de un proceso.

3.2.1 Variables lingüísticas

La estrategia de diseño que se decidió emplear en los controladores borrosos de la velocidad de los trenes fue la utilización de sendos reguladores PI borrosos [3, 4]. Para ellos se aplicaron estructuras incrementales del tipo:

$$\Delta u = F(\Delta e, e)$$

donde e representa el error de velocidad de los trenes, Δe el incremento de dicho error entre periodos consecutivos de muestreo y Δu el incremento de la señal

de control de los trenes, que en este caso es la tensión aplicada a los mismos.

En función de la utilización de esta estrategia de control las variables lingüísticas empleadas en la base de reglas fueron el error de velocidad, el incremento del error de velocidad y el incremento de la tensión aplicada a los móviles y, por tanto, tales reglas tiene la forma siguiente:

IF (e = propiedad_de_e **AND** Δe = propiedad_de_Δe)
THEN (Δu = propiedad_de_Δu).

3.2.2 Modificadores lingüísticos

Vitecek, en [10] recomienda que el número de términos ligüísticos asociados a cada una de las variables del protocolo debe estar comprendido entre 3 y 10. Conforme a ello se decidió emplear en principio tres términos para cada una de las variables, sin embargo las numerosas pruebas realizadas llevaron a concluir que era necesaria una mayor precisión en relación con la señal de control por lo que hubo que ampliar a 7 el número de términos lingüísticos para dicha variable. Los modificadores finalmente utilizados fueron los siguientes:

- Para las variables de error y de incremento del error: Positivo (P), Cero (Z) y Negativo (N).
- Para la variable incremento de tensión: Positivo Grande (PG), Positivo Medio (PM), Positivo Pequeño (PP), Cero (Z), Negativo Pequeño (NP), Negativo Medio (NM) y Negativo Grande (NG).

3.2.3 Funciones de pertenencia

La elección de las funciones de pertenencia de los modificadores lingüísticos considerados en un protocolo es una tarea de extrema importancia, ya que de ello depende la correcta caracterización del proceso a controlar.

En este trabajo se han utilizado funciones de pertenencia tipo **triangular** y **trapezoidal**. porque su simplicidad asegura una gran eficiencia computacional y un sensible ahorro de memoria.

Es importante también una elección adecuada del intervalo de discretización empleado para cada variable del sistema. En dicha elección se han contemplado aspectos tales como la velocidad de respuesta, la estabilidad y el carácter despreciable de ciertas variaciones demasiado pequeñas.

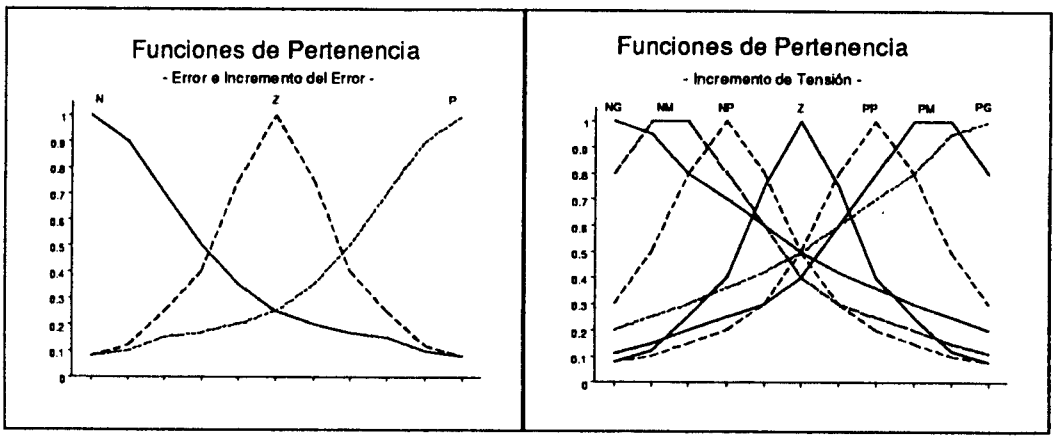


Figura 2: Funciones de Pertenencia de los modificadores de las variables de entrada y salida del controlador.

Regla 1: IF e = P AND Δe = P THEN Δu = PG
Regla 2: IF e = P AND Δe = Z THEN Δu = PM
Regla 3: IF e = P AND Δe = N THEN Δu = PP
Regla 4: IF e = Z AND Δe = P THEN Δu = PP
Regla 5: IF e = Z AND Δe = Z THEN Δu = Z
Regla 6: IF e = Z AND Δe = N THEN Δu = NP
Regla 7: IF e = N AND Δe = P THEN Δu = NP
Regla 8: IF e = N AND Δe = Z THEN Δu = NM
Regla 9: IF e = N AND Δe = N THEN Δu = NG

Tabla 1: Base de reglas del sistema de control borroso.

En la figura 2 se muestran las funciones de pertenencia elegidas para los diferentes modificadores lingüísticos de las variables de entrada y salida de los controladores.

3.2.4 Bases de reglas

El conjunto de reglas que caracteriza el comportamiento del sistema de control que se ha utilizado en este trabajo ha sido el que se muestra en la tabla 1.

Esta base de reglas se justifica directamente en función del conocimiento heurístico que se posee sobre el funcionamiento del proceso a controlar. Así, por ejemplo, la regla 1 indica que cuando la velocidad actual del tren está por debajo de su velocidad deseada

(error=Positivo) y existe más error que en el intervalo de control anterior (incr. error=Positivo), la tensión de salida debe ser mucho mayor que la anterior (incr. tension=Positivo Grande) para que el tren se aproxime a su velocidad deseada.

3.2.5 Mecanismo de inferencia

En la bibliografía [3] se describen dos tipos fundamentales de mecanismos de inferencia: los basados en la composición de reglas [5, 6] y los basados en reglas individuales. Para este trabajo se ha optado por el primero de ellos porque, aunque presenta un coste computacional algo superior al otro, este tipo de inferencia tiene en cuenta simultáneamente el estado de activación de todas las reglas existentes en el protocolo, con lo que en principio se obtienen a través de él unos mejores resultados.

A la hora de la implantación del controlador se ha optado por la utilización del método de la **Matriz de Inferencia**. Esto se ha debido a que esta técnica, aunque precisa un número elevado de operaciones para la generación del algoritmo de control, éstas se realizan "off-line", es decir, en una etapa previa a la implantación del controlador. Sin embargo, en la etapa "on-line" el procedimiento de decisión se reduce a consultar una matriz denominada matriz de inferencia que resultó obtenida en el procedimiento "off-line" antes referido, así como realizar ciertas interpolaciones. Como alternativa podría haberse utilizado la estrategia consistente en llevar a cabo inferencias sobre la base de reglas en cada periodo de muestreo. Frente a esta idea, la técnica de la Matriz de Inferencia ofrece como ventaja establecer una relación directa entre las distintas variables del sistema y por ello, la obtención de la salida del controlador, dada una entrada determinada es una operación de ejecución casi inmediata.

Simultáneamente a la utilización de la técnica anterior, se ha empleado el procedimiento de interpolación denominado de **Aproximación Trapezoidal** [4], pues con él se reducen los errores propios de la discretización llevada a cabo para la obtención de la matriz de inferencia.

3.2.6 Fuzzificación

Para la generación de la Matriz de Inferencia es necesario llevar a cabo una fuzzificación de los posibles valores escalares alcanzables por las señales de entrada del controlador, o sea, asociarles un conjunto borroso con el que sea posible acceder a la base de reglas borrosas. De todas las técnicas descritas en la bibliografía, se ha optado en este trabajo por utilizar el **método de Mamdani** [5].

3.2.7 Defuzzificación

De una forma similar, durante el procedimiento de generación de la Matriz de Inferencia es necesario defuzzificar los conjuntos borrosos inferidos a partir de la base de reglas, o sea, calcular un valor escalar de la señal de control que sea representativo de los mismos. De los distintos métodos existentes, se ha utilizado en este trabajo la técnica del **Centro de Masas** o **Area Mitad** [3].

4 Resultados

Una vez realizadas múltiples pruebas de funcionamiento de los diferentes sistemas de control empleados durante el desarrollo del presente trabajo se obtuvieron los resultados que se describen a continuación.

Como idea general en relación con los controladores tipo PID puede indicarse que ha resultado muy laborioso ajustar sus parámetros de forma que el comportamiento de los sistemas de control resultantes fuera aceptable en

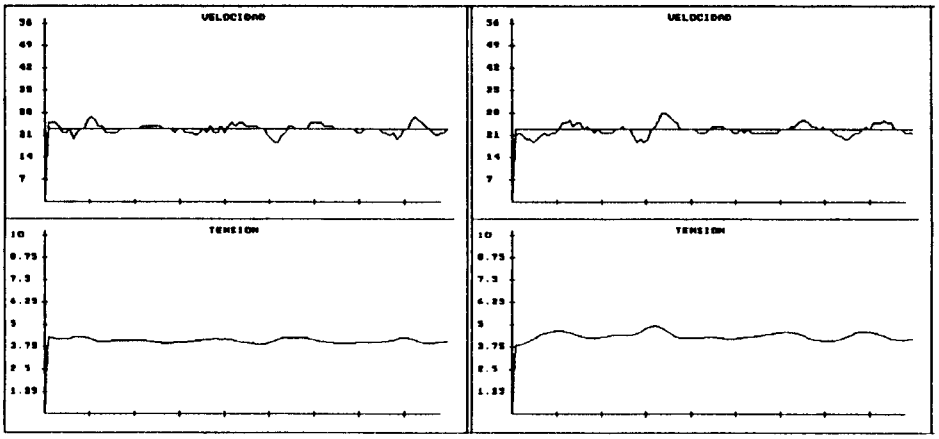


Figura 3: Comportamiento del sistema regulado con un PID único y con un PID por zonas.

todo el espacio de trabajo de los trenes. Por otra parte, la utilización de controladores PID diferentes, según la zona del espacio de trabajo recorrida por los trenes en cada momento, no ha redundado en términos generales en una mejora del comportamiento de los controladores.

En la figura 3 se muestra el comportamiento de uno de los trenes cuya velocidad ha sido regulada por sendos controladores PID, el primero fijo para todo el espacio de trabajo y el segundo con diferentes controladores para distintas zonas de trabajo realizando cambio gradual de los parámetros del controlador al pasar de una zona a otra.

Los mejores resultados se han alcanzado sin lugar a dudas con el empleo de controladores borrosos. En la figura 4 se presenta una comparación del comportamiento del regulador PID clásico y el PI borroso ante un determinado escalón en la consigna de velocidad de uno de los trenes, en ella puede observarse cómo el primero de ellos resulta ser mucho más lento con una señal de control más oscilante.

Entre las distintas cuestiones analizadas con objeto de obtener un controlador borroso óptimo, junto al número de modificadores lingüísticos empleados, una de las que han resultado de vital importancia es la elección del rango de trabajo para la variable de salida del regulador (incremento de tensión). En las figuras 4 (derecha) y 5 se muestra la respuesta del sistema controlado utilizando, por un lado un dominio de la variable de salida comprendido entre ± 1.25 voltios, y por otro, sendos dominios comprendidos entre ± 5 voltios en un caso y ± 2.5 voltios en el otro. En ellas se puede observar que la mejor respuesta se consigue con el segundo de estos rangos, ya que aunque el sistema resulta ser un poco más lento, el esfuerzo de control es mucho menor y no se producen sobreoscilaciones.

Por otra parte, ha sido realizado un estudio comparativo de la respuesta de los dos tipos de reguladores analizados (clásicos y difusos) en los momentos en que se comandan cambios de velocidad de

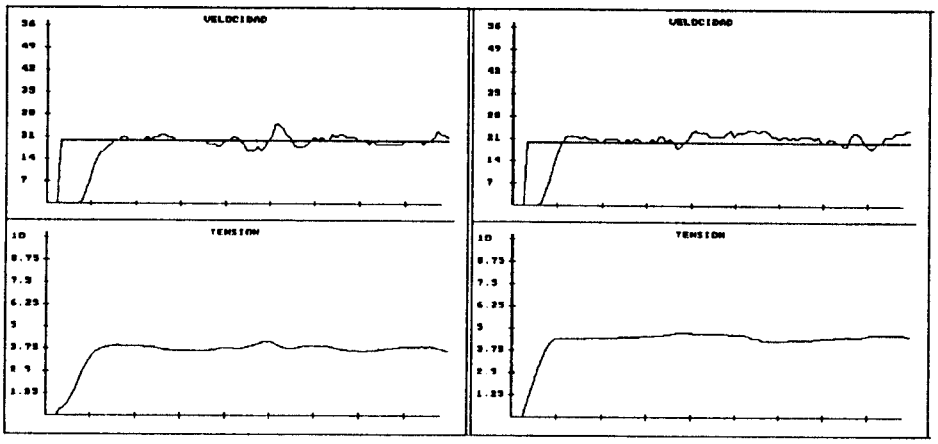


Figura 4: Respuesta del sistema controlado mediante un regulador PID y mediante un regulador borroso.

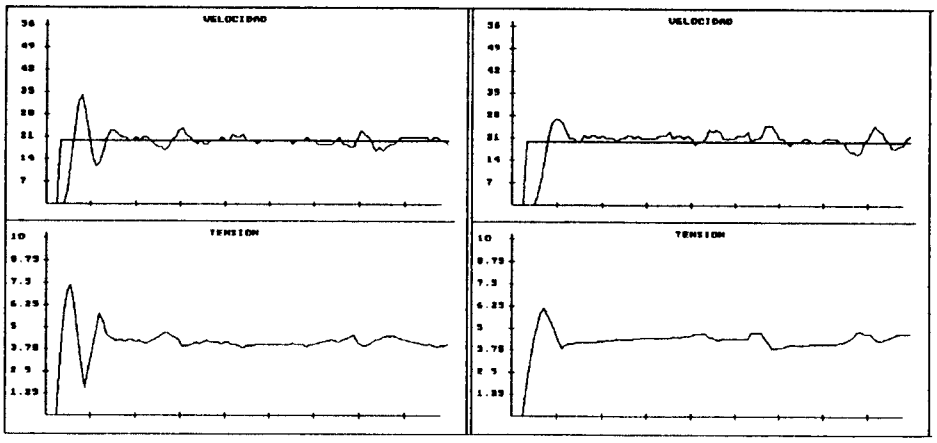


Figura 5: Respuesta del sistema con diferentes reguladores borrosos.

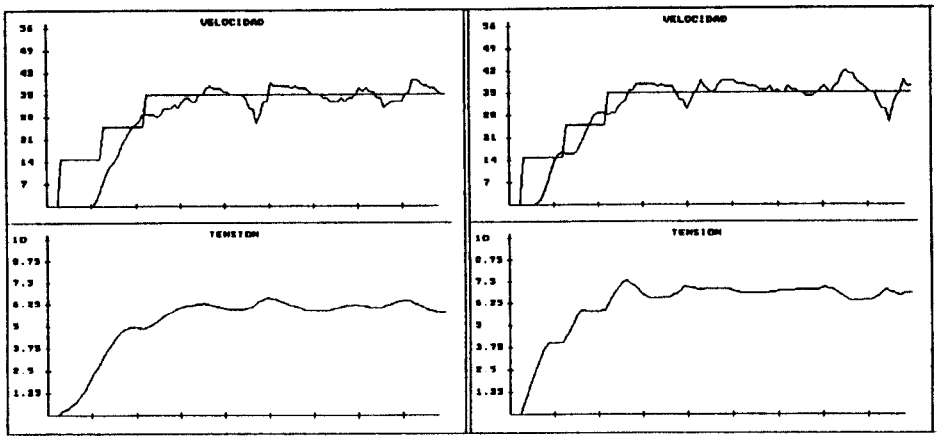


Figura 6: Aceleración progresiva del sistema controlado con un PID tradicional y con un PI borroso.

los trenes. En la figura 6 aparece un ejemplo que puede servir como ilustración de los diferentes casos analizados a lo largo del presente trabajo. El mismo recoge la respuesta de ambos sistemas de control en una situación de aceleración. En términos generales puede decirse a partir del estudio realizado que el regulador borroso normalmente responde antes y menos bruscamente ante los cambios de velocidad deseada y que realiza un mejor seguimiento de la consigna.

5 Conclusiones y futuras mejoras

A partir de todos los resultados obtenidos, así como de la experiencia adquirida en la elaboración de este trabajo, se puede concluir que en esta aplicación las prestaciones del controlador borroso son superiores a las del controlador PID tradicional.

Por otra parte el proceso de diseño de los controladores borrosos ha resultado ser mucho menos laborioso pues no ha sido necesario llevar a cabo un estudio exhaustivo del comportamiento dinámico de los trenes previo a la obtención del controlador. Asimismo, en el caso borroso la obtención de un controlador considerable óptimo en función de la técnica de diseño empleada hizo necesario un número de refinamientos de los primeros controladores alcanzados mucho menor que en el caso de los reguladores PID.

Finalmente es de reseñar que el trabajo aquí presentado es susceptible de ser modificado y ampliado fácilmente, para así poder adaptarlo a nuevos objetivos. Algunas ideas al respecto pueden ser: a) considerar en la aplicación otras estrategias de control borroso y realizar un estudio comparativo global de las mismas; b) analizar cuestiones relativas a la estabilidad de los controladores en la línea de lo propuesto en [4]; c) incluir la aceleración de los trenes, además de su velocidad, como variables de control del sistema, podría redundar en un mejor comportamiento de éstos ante cambios de velocidad de referencia.

Referencias

[1] Astrom, J.K., Wittenmark, B. (1990) "Computer Controlled Systems", Ed. Prentice-Hall.

[2] Bandera, G. (1994), "Técnicas de Control Inteligente Aplicadas a un Sistema Ferroviario a Escala", Proyecto Fin de Carrera, F. Informática de Málaga.

[3] Driankov, D., Hellendoorn, H., Reinfrank, M. (1993), "An Introduction to Fuzzy Control", Ed. Springer-Verlag.

[4] García Cerezo, A.J. (1987), "Aplicaciones del Razonamiento Aproximado en el Control y Supervisión de Procesos", Tesis Doctoral, Servicio de Publicaciones de la Universidad de Santiago de Compostela.

[5] García Cerezo, A.J. (1991), "Sistema de Control Borroso. Panorámica Actual. Nuevas Aportaciones", Universidad de Vigo.

[6] Lee, C.C. (1990), "Fuzzy Logic in Control Systems: Fuzzy Logic Controller (Parts I & II)", IEEE Transactions on Systems Man and Cybernetics, Vol 20, No. 2, March / April, pp 404-435.

[7] Ogata, K. (1993), "Ingeniería de Control Moderna", Ed. Prentice-Hall.

[8] Ogata, K. (1987), "Discrete-Time Control Systems", Ed. Prentice-Hall.

[9] Quevedo, F. (1988), "Reguladores Industriales", Notas del VIII Curso "Automática en la Industria", S'Agaro (Gerona), Univ. Politécnica de Barcelona.

[10] Silva, M. (1985), "Las Redes de Petri: En la Automática y en la Informática", Ed. AC.

[11] Vicetek, A, Smutny, L., Kusyn, J. (1984), "Fuzzy Approach to the Design of the Simple Control Algorithms". IFAC World Congress. Budapest, Vol. 2, 4/B-1, pp 234-239.

Hardware

Una Filosofía Novedosa para la Realización de Circuitos Integrados Analógicos para Lógica Fuzzy

Francisco Castillo
Dpto. Ing. Electrónica
E.U.P. Vilanova i la Geltrú
Universitat Politecnica de Catalunya
e-mail: castillo@eel.upc.es

Ignacio Puyo
E.U.P. Vilanova i la Geltrú
Universitat Politecnica de Catalunya

Resumen

En este artículo se presenta una idea novedosa para la realización de circuitos integrados aplicados a la lógica fuzzy. El principio de funcionamiento se basa en la emulación de la función global entrada-salida que realiza un sistema de lógica fuzzy cualquiera, en contra de la realización de cada uno de los diferentes bloques que la componen. Este tipo de realizaciones está pensado para aplicaciones en donde la lógica fuzzy no es adaptativa. La forma de aproximar la función entrada-salida consiste en la aproximación lineal a tramos y para ello se presentan los circuitos básicos que se han de interconectar para formar la función deseada. Estos circuitos funcionan en modo corriente y son digitalmente programables. También se presentan algunos resultados de la simulación de los circuitos.

Palabras clave: hardware, modo corriente, microelectrónica, circuitos analógicos.

1 Introducción

La lógica fuzzy en este momento es una tecnología que está madurando rápidamente, y se empiezan a ver productos comerciales e industriales que la incorporan como parte de un control inteligente general. Esto es especialmente significativo en Japón [1] en donde han aparecido electrodomésticos, cámaras, fotocopiadoras, etc. que incorporan la lógica fuzzy en su control. Es muy posible que en los próximos años se extienda mas aún esta tecnología hasta alcanzar el punto en donde su utilización sea de lo mas habitual.

El hardware es una parte importante de este campo, ya que ésta habilita su utilización en aplicaciones en donde hasta ahora no ha sido posible debido al espacio físico que podría suponer la incorporación de un emulador/simulador fuzzy, o bien por requerimientos de velocidad, o bien por consumo. Se ha visto como gradualmente la lógica fuzzy se ha ido incorporando más y más profundamente dentro del hardware, primero como un conjunto de instrucciones en C o lenguaje máquina que se pueden incorporar en microcontroladores o microprocesadores [2], y en autómatas programables [3]. El segundo paso ha sido el desarrollo de microprocesadores especializados para la lógica fuzzy, tanto a nivel comercial [4], así como a nivel de investigación [5][6].

Los circuitos integrados digitales tienen la ventaja de ser muy flexibles, ya que estos se pueden programar, y generalmente, es posible realizar cualquier función fuzzy. En cambio, existe también un ámbito de aplicaciones en donde la utilización de CIs analógicos parece mas conveniente, son aplicaciones en donde se exige una realización compacta, rápida o bien de bajo consumo. Una realización analógica es solo razonable si la función fuzzy no es adaptativa (fija tras programarse). Un ejemplo de aplicación en donde existen tales requerimientos es en sistemas integrados ("embedded systems"), en donde el espacio físico que ocupa el circuito es determinante.

2 Filosofía de Diseño

Gran parte de las realizaciones que se proponen (ver por ejemplo [7] [8]) intentan la realización directa de cada uno de los bloques que forman parte del sistema de lógica difusa: el normalizador, fuzificador, máquina de inferencia, defuzificador, desnormalizador, representado esquemáticamente en la figura 1.

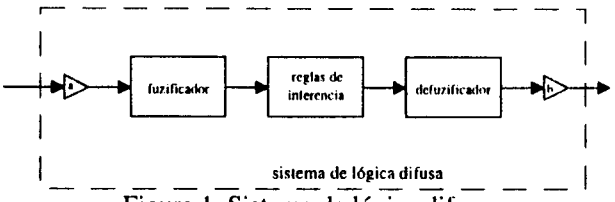


Figura 1. Sistema de lógica difusa con sus diferentes componentes.

En cambio, y especialmente cuando se habla de una realización analógica, en donde las diferentes partes no son adaptativas, todo el bloque de lógica difusa se puede pensar en un solo proceso cuya relación entrada-salida (E/S) es altamente no-lineal . Esto tiene el conveniente de que el diseñador del circuito únicamente se ha de preocupar en una reproducción fehaciente de la función no-lineal entrada-salida, y no en la precisión de cada uno de los bloques internos. Además, el ahorro de espa-

un solo bloque a realizar, también porque la realización de varios de los bloques de la figura 1 necesitan circuitos complejos tales como divisores para el defuzificador, y multiplicadores para el fuzificador y defuzificador.

El uso de esta técnica no implica que la relación E/S sea fija. En la realización que se explica mas adelante, se verá que ésta la programa externamente el usuario. Normalmente se dispondrá de un compilador encargado de traducir las especificaciones difusas a un equivalente para el circuito integrado.

Esta filosofía es además ampliable a cualquier tipo de controlador o bloque de E/S que se desee, y no únicamente para lógica difusa.

3 Descripción de los circuitos

Ya que en general la relación entrada-salida es altamente no-lineal, una posible forma de realizar el circuito es mediante aproximaciones a tramos lineales de esta función (figura 2). Dependiendo de la precisión necesaria, serán necesarias mas o menos rectas. Esta solución está comprendida de dos funciones básicas: una parte de ganancia programable con un offset también programable, y una discontinuidad en donde entra en funcionamiento la siguiente recta.

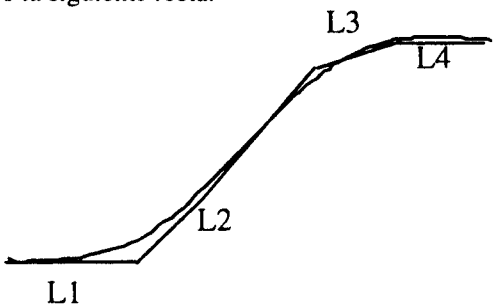


Figura 2. Aproximación de una curva por tramos lineales.

En modo tensión, se puede apreciar que las topologías que actualmente se están utilizando para realizar ganancias programables son circuitos basados en capacidades conmutadas, o bien basados en OTAs. Si comparamos éstos con las soluciones en modo corriente, se puede apreciar que resulta mucho mas sencillo trabajar en este último modo para CMOS, pues los circuitos resultan mas compactos.

En [9] se describe una célula que puede ser utilizada como base para la construcción de los circuitos en modo corriente (fig. 3). Con una sola variable de entrada, cada una de estas células realiza una de las rectas que aproximan la función E/S. Para la realización de la función: $I_{out} = A \cdot I_{in} - C$ por ejemplo, se utiliza el circuito de la figura 3(a), en donde C=B I_{ref}. Para rectas de pendiente negativa, el circuito de la figura 3(b) se utiliza.

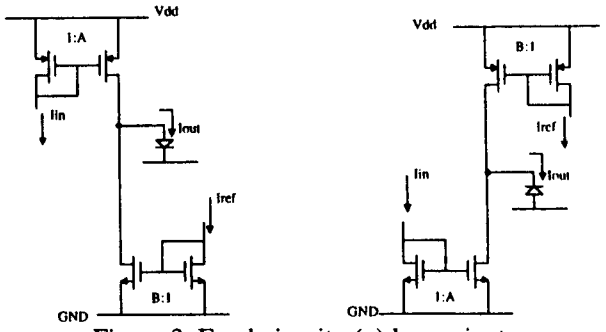


Figura 3. En el circuito (a) la corriente I_{out}=A I_{in} - B I_{ref}, y en (b) I_{out} = -A I_{in} + B I_{ref}.

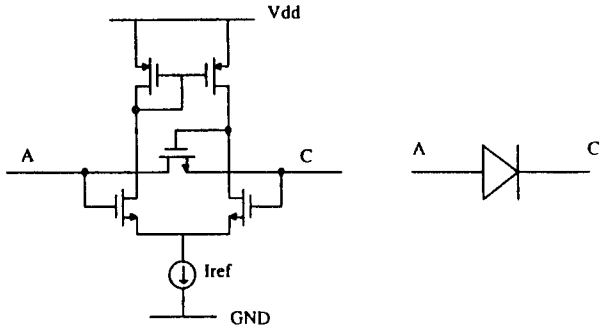


Figura 4. Realización diferencial de un diodo.

La discontinuidad necesaria para activar cada bloque de ganancia lineal la introduce el diodo, que en CMOS se puede realizar de diversas formas. En nuestro caso, se ha escogido un circuito diferencial con conmutador (figura 4) que tiene la ventaja en que tiene una caída de tensión A-C mínima. La etapa diferencial detecta cambios de tensión entre A y C, cuando esta tensión supera unos cuantos mV, la tensión en la puerta del conmutador rápidamente asciende, permitiendo el flujo de corriente de A a C.

Para poder juntar todo y realizar funciones no lineales, se utilizarán los circuitos de la figura 3 para realizar cada una de las rectas que formarán la función E-S. Para cada tramo lineal se necesitará uno de estos circuitos. Para formar la función global, la salida de cada uno de estos bloques se conectará a una salida común. En tal caso, la corriente de salida resultante será entonces la suma de cada una de las contribuciones individuales de cada bloque. Por ejemplo, para realizar la función L3 de la figura 5, se necesitarán 2 bloques debido a los dos tramos lineales. El primer bloque, del tipo fig. 3(a) realizará la recta L1. A partir de la discontinuidad entrará en funcionamiento el 2° bloque del tipo fig. 3(b) que le restará a la corriente del primer bloque, una cantidad representada por la recta L2. La corriente neta y por tanto resultante será entonces L3.

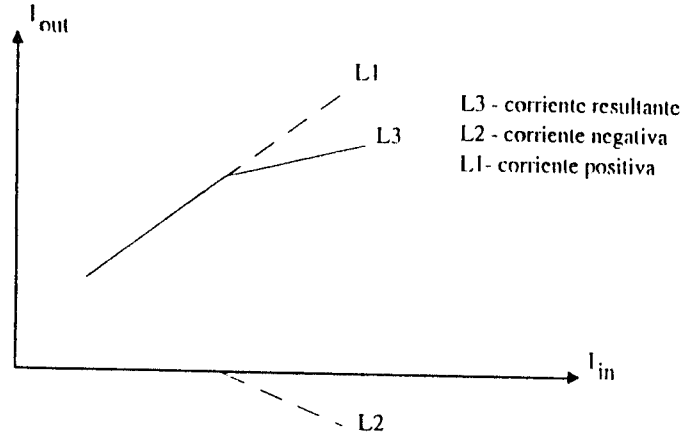


Figura 5. Corriente resultante a partir de generadores de corriente positiva y negativa

En la práctica, los circuitos de las figuras 3(a) y (b) son insuficientes, ya que su impedancia de salida es baja. Para mejorar la impedancia de salida, espejos cascodo son utilizados. Por ejemplo, para implementar una función triangular, el circuito de la figura 6 es utilizado, generando la señal mostrada en la figura 7. Este circuito está compuesto por 4 bloques básicos, cada uno produciendo las corrientes denominadas I1..I4. La corriente I1 es la corriente constante de 20 μ A que aparece en la base del triángulo para corrientes de entrada menores de 20 μ A. Cuando la corriente de entrada supera los 20 μ A, entonces entra en funcionamiento un segundo bloque que añade una rampa positiva, indicada como I2 en la figura. Una vez alcanzados los 50 μ A a la entrada, entonces entra en funcionamiento I3, que es una corriente negativa que resta sobre las otras, produciendo la pendiente negativa. Finalmente, para corrientes mayores a 80 μ A, se añade una corriente adicional, I4, produciendo la corriente constante de 20 μ A a la salida.

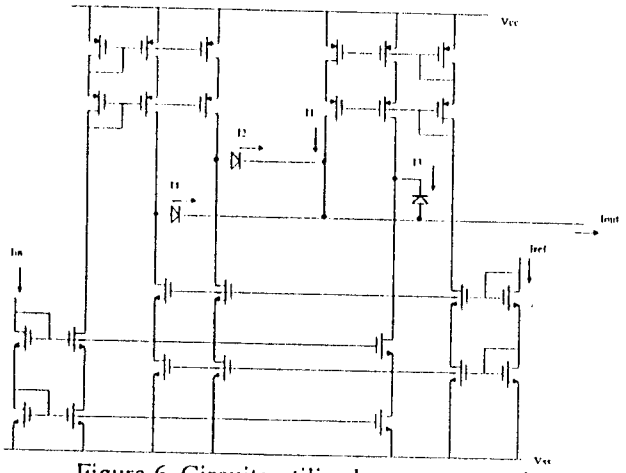


Figura 6. Circuito utilizado para generar la función triangular.

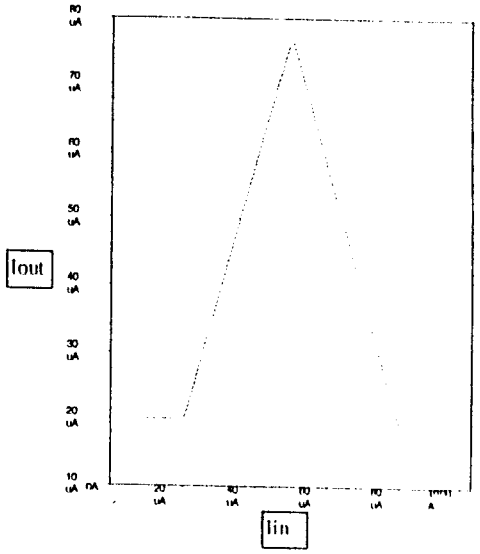


Figura 7. Simulación spice del circuito de la figura 6.

Para realizaciones de mas entradas, cada bloque contribuirá con un hiperplano en el hiperespacio de la gráfica E/S. Por ejemplo, para 2 entradas, cada bloque realiza la función de un plano (ver figura 8). Para aproximar una superficie arbitraria, se utilizarán tramos de estos planos para construir la función total, P1..P5, cada plano de éstos siendo el resultado neto de las sumas y restas de corriente que cada bloque contribuye. Para construir un circuito de estas características, es únicamente necesario añadir etapas tal y como se hizo en el caso unidimensional, pero a su vez teniendo bloques con Iin1 como entrada, y otros con Iin2 como entrada. La suma se vuelve hacer de forma implícita, uniendo todas las salidas de los bloques.

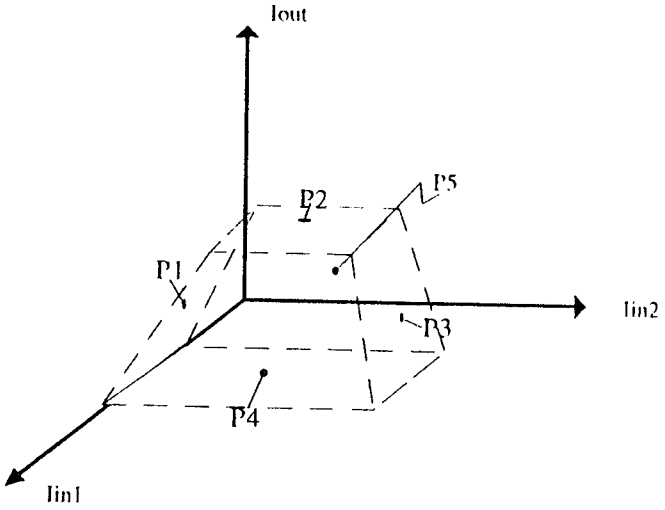


Figura 8. Ejemplo para una función de 2 entradas.

Por otro lado, los espejos de corriente de cada bloque se pueden programar digitalmente mediante el método mostrado en la figura 9. a1...a4 son en este caso, 4 bits que provienen de un registro de desplazamiento digital

que se programa desde el exterior. Estos bits definen si los conmutadores que aparecen en el espejo de corriente permanecen abiertos o cerrados. Cada transistor del espejo de corriente se conecta o desconecta para variar la relación 1:A entre el tamaño efectivo de la parte de entrada y la parte de salida, a su vez definiendo la ganancia de corriente, A. Para definir todos estos bits de programación para cada etapa, se dispone de un registro que contiene la configuración. Como el registro ha de ser escrito solo al comienzo, es posible utilizar un solo registro de desplazamiento que se puede escribir secuencialmente al inicio de la utilización del CI, de esta forma solo serán necesarias 3 patillas adicionales de entrada: la de entrada de datos, el reloj, y la de habilitación del desplazamiento.

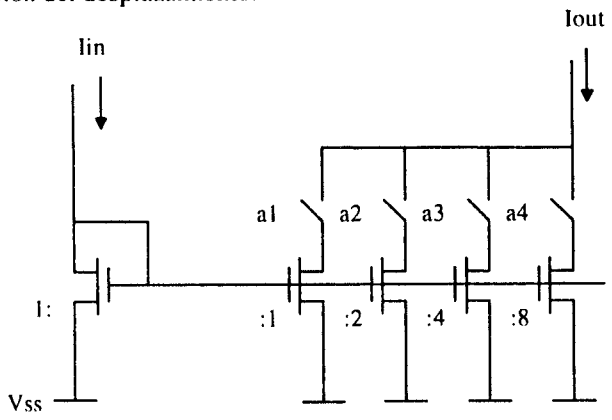


Figura 9. Espejo de corriente programable mediante palabra digital a1..a5.

4 Conclusiones

Se ha presentado una nueva forma de sintetizar circuitos integrados analógicos para la emulación de lógica fuzzy, que consiste en la aproximación mediante tramos lineales de la función entrada-salida del sistema. En este sentido, esta técnica de construcción no se limita únicamente a lógica fuzzy, sino que se puede extender a otros diversos tipos de bloques de procesado no-lineal.

La estructura resultante para la síntesis de circuitos fuzzy es relativamente sencilla. Además, es compacta al estar basada en espejos de corriente que se programan mediante el estado de unos conmutadores, que definen la relación entre la corriente de entrada y de salida. Por último, la programación se hace muy sencillamente utilizando un registro de desplazamiento.

Referencias

[1] C. Chien Lee, *Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I*, IEEE Trans. on Systems, Man and Cybernetics, Vol. 20, No. 2, Abril de 1990.

[2] fuzzyTech 3.1 Precompiler Edition, Inform GmbH, 1994.

[3] Manual de referencia: SUCOcontrol PS4 Fuzzy Logic, Klockner-Moeller.

[4] Product Bulletin: Fuzzy Logic Products, VY86570 12-bit Fuzzy coprocessor, VLSI Technology Inc., 1993.

[5] E.Gandolfi, M.Masetti, et. al., *Architecture of a 50 MFIPS Fuzzy Processor and the related 1 μm VLSI CMOS Digital Circuits*, Proc. of the 4th Intl. Conference on Microelectronics for Neural Networks and Fuzzy Sys., IEEE Computer Soc. Press, Sept., 1994.

[6] H.Eichfeld, M.Klimke, et. al., *A General-Purpose Fuzzy Inference Processor*, Proc. of the 4th Intl. Conference on Microelectronics for Neural Networks and Fuzzy Sys., IEEE Computer Soc. Press, Sept., 1994.

[7] 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, No.3, Mayo 1993.

[8] V.Catania, A.Puliafito, M.Russo, L.Vita, *A VLSI Inference Processor Based on a Discrete Analog Approach*, IEEE Trans. on Fuzzy Systems, Vol.2, No.2, Mayo 1994.

[9] T.Kettner, C.Heite, K.Schumacher, *Analog CMOS Realization of Fuzzy Logic Membership Functions*, IEEE Journal of Solid-State Circuits, Vol.28, No.7, Julio 993.

Aprendizaje Supervisado de Reglas en un Controlador Difuso Analógico

Fernando Vidal-Verdú
Dpto. de Electrónica
Universidad de Málaga
e-mail: vidal@ctima.uma.es

Angel Rodríguez-Vázquez
Dpto. de Diseño Analógico
Centro Nacional de Microelectrónica
e-mail: angel@cnm.us.es

Resumen

En este artículo se presenta un controlador difuso en el cual la información inicial se reduce a las variables de entrada y de salida, siendo las reglas aprendidas a partir de la operación de un humano o un controlador software. El aprendizaje debe llevarse a cabo bajo restricciones hardware en linealidad, rango y resolución. Por ello, se realiza un modelado de los bloques de diseño en una arquitectura neuro-difusa y se discuten aspectos relativos al aprendizaje on-chip. Las simulaciones por ordenador muestran que el aprendizaje es posible para resoluciones de hasta 7 bits, abordables incluso con las tecnologías VLSI más baratas.

Palabras Clave: *Controladores neuro-fuzzy, aprendizaje on-chip, implementación analógica.*

1 Introducción

Un sistema de inferencia difuso convencional implementa una serie de valores *fijos*, con funciones de pertenencia de forma y localización *predefinidas*. Por tanto:

- Los productos difusos basados en esta aproximación convencional son incapaces de adaptar su operación ante ambientes cambiantes. Debido a esto, se suelen diseñar para trabajar con condiciones ambiente promedio, por lo que su rendimiento no será el óptimo en la mayoría de los casos prácticos.
- Incluso si la falta de adaptatividad es tolerable, la formulación de una serie apropiada de reglas difusas capaces de hacer frente a estas condiciones promedio es aún una tarea difícil que require un conocimiento previo de la operación del sistema.

Los sistemas neuro-difusos intentan dar solución a estos problemas a través de la incorporación de aprendizaje al algoritmo convencional difuso. Por tanto, no se requiere conocimiento previo (es decir reglas difusas) de la operación del sistema, que será capaz de aprender de la operación sobre el terreno de otro controlador software o humano. Asimismo, ya que los sistemas neuro-difusos son intrínsecamente adaptativos, son capaces de evolucionar ante cambios del entorno.

La figura 1(a) muestra una arquitectura en capas para la implementación hardware de sistemas neuro-fuzzy. Los circuitos asociados a los bloques funcionales han sido propuestos previamente por los autores en [1]. Un diseño adecuado requiere que el comportamiento de estos circuitos sea expresado por medio de ecuaciones, de forma que el aprendizaje se pueda simular bajo condiciones realistas. El desarrollo de estos modelos, y su utilización para demostrar el aprendizaje de reglas por parte de un controlador neuro-difuso es nuestro principal objetivo en esta comunicación.

2 Modelado de los Bloques de Diseño

Las *funciones de pertenencia* se implementan en [1] utilizando dos pares diferenciales, como muestra la figura 1(b). Esta configuración posibilita el ajuste independiente de la anchura y la localización, que se relacionan linealmente con las tensiones E_{1ij} y E_{2ij}. El ajuste independiente de la pendiente se consigue sustituyendo el transistor simple por los transistores compuestos de las figuras 1(g)-(h). El parámetro de ajuste B es una tensión para los transistores compuestos de las figuras 1(g) y 1(h), representando el número decimal codificado en la palabra digital en las puertas de las llaves de corriente del transistor compuesto de la figura 1(f). La forma de campana de la figura 1(b) se obtiene sumando las salidas de los dos pares diferenciales, que en el caso de transistores simples son:

$$i_o \approx \begin{cases} \sqrt{2\beta I_Q} x_d \sqrt{1 - x_d^2 \beta / (2I_Q)} & |x_d| \le \sqrt{I_Q/\beta} \\ I_Q \mathrm{sgn} x_d & |x_d| \ge \sqrt{I_Q/\beta} \end{cases} \quad (1)$$

Donde x_d es la tensión de entrada diferencial, I_Q es la intensidad de polarización y β es el factor de transconductancia del transistor MOS. La ecuación (1) es válida para los transistores simple y digital de la figura 1(f). El par diferencial con el transistor paralelo de la figura 1(h) obedece a la ecuación en la figura 2(a). El par con el transistor serie de la figura 1(g) conlleva un cálculo más duro, dado que el transistor de abajo trabaja en zona óhmica. Sin embargo, un análisis de su comportamiento muestra que la sustitución de β en (1) por β_{eff} en la figura 2(b) da errores por debajo

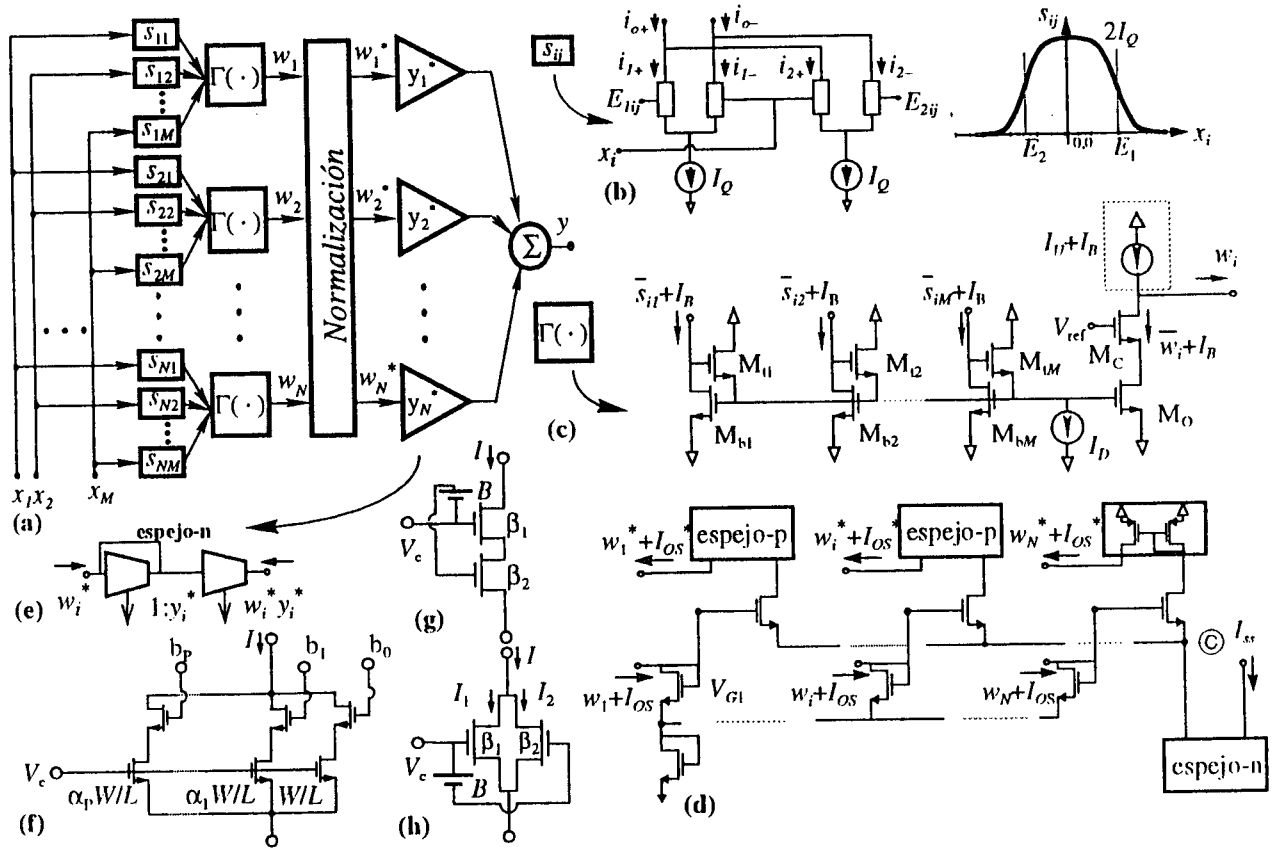


Figura 1 : Arquitectura neuro-fuzzy (a), circuito función de pertenencia (b), t-norma (c), normalización (d), ponderado singleton (e), transistor de control digital (f), transistor serie(g) y transistor pralelo (h).

del 1.5%, permitidos en las aplicaciones típicas.

Los espejos de corriente que implementan el *ponderado singleton*, utilizan también transistores compuestos con el fin de obtener programabilidad eléctrica. Las ecuaciones en las figuras 2(c), 2(d) y 2(e) expresan el comportamiento de los espejos con transistores serie, paralelo y digital respectivamente.

Finalmente, la figura 2(f) muestra la ecuación asociada al circuito de normalización. A diferencia de nuestra propuesta en [1], aquí no se consideraun offset de corriente a la salida por las contrapartidas que supone al introducir aprendizaje. Para las simulaciones de la sección 4, el offset se eliminó dimensionando con distintos tamaños los transistores de entrada y de salida del circuito, es decir, haciendo $\beta_t > \beta_b$ en la figura 2(f).

3 Aspectos de aprendizaje

3.1 Algoritmos de aprendizaje

En [2] se propone un algoritmo híbrido para implementaciones VLSI. Este algoritmo actualiza los parámetros asociados a la función de pertenencia con el algoritmo *weight perturbation* [3] , y los valores singleton mediante la regla *outstar* [4]. Weight perturbation sustituye las derivadas por diferencias finitas y calcula la influencia de cada parámetro sobre el error global, evitando los caminos de realimentación. Si ω es el parámetro a actualizar y $\zeta(\bullet)$ el error global a la salida, el cambio en el valor de ω es,

$$\Delta\omega = \frac{-\eta\,[\zeta(\omega) - \zeta(\omega + pert)]}{pert} = \\ = G(pert)\,[\zeta(\omega) - \zeta(\omega + pert)] \qquad (2)$$

donde *pert* es una pequeña perturbación y η es la razón de aprendizaje, siendo ambos constantes. Nótese que el hardware de actualización de los parámetros implica la evaluación del error con el parámetro perturbado y sin perturbar y la multiplicación por una constante.

Nosotros usamos esta estrategia para actualizar las funciones de pertenencia. En cuanto a los singletons, se explotan las similitudes entre el algoritmo de inferencia difuso y la red *counterpropagation*. Esto resulta evidente cuando se asocian conjuntos tradicionales (*crisp*) en lugar de difusos a las etiquetas de entrada. En base a este hecho, nuestro algoritmo utiliza la regla *outstar*,

$$y^*_{i_{new}} = y^*_{i_{old}} + \mu\,[T - y(x)] \qquad (3)$$

donde T es la salida a aprender y μ es la razón de aprendi-

$$i_{1,2} = \begin{cases} \sqrt{2(\beta_1+\beta_2)\left(I_Q - \frac{2\beta_1\beta_2}{\beta_1+\beta_2}B^2\right)}x_i \sqrt{1 - x_i^2 \frac{(\beta_1+\beta_2)}{2\left(I_Q - \frac{2\beta_1\beta_2}{\beta_1+\beta_2}B^2\right)}} & |x_i| \le I_1 \quad c = \frac{2\beta_1(\beta_1+\beta_2)}{2\beta_1+\beta_2} \quad \dot{x}_i = x_i - \frac{\beta_2}{\beta_1+\beta_2}B \\ I_Q - \frac{c}{\beta_1+\beta_2}\left(I_Q - \frac{\beta_1\beta_2}{\beta_1+\beta_2}B^2\right) + c\left(1 - \frac{c}{\beta_1}\right)\dot{x}_i + & |x_i| \ge I_1 \qquad I_1 = \sqrt{\frac{\left(I_Q - \frac{2\beta_1\beta_2}{\beta_1+\beta_2}B^2\right)}{(\beta_1+\beta_2)}} \\ + \frac{c}{\beta_1}\dot{x}_i\sqrt{c(c-2\beta_1)\dot{x}_i^2 + 2\frac{c\beta_1}{\beta_1+\beta_2}\left(I_Q - \frac{\beta_1\beta_2}{\beta_1+\beta_2}B^2\right)} & |x_i| \le I_2 \\ I_Q \operatorname{sgn} x_i & |x_i| \ge I_2 \quad I_2 = \frac{\beta_2 B}{\beta_1+\beta_2} + \sqrt{\frac{I_Q - \frac{\beta_1\beta_2}{\beta_1+\beta_2}B^2}{\beta_1+\beta_2}} \end{cases}$$

(a)

$$\beta_{eff} = \frac{g_m^2}{2I_Q} = \frac{2}{I_Q}\left(\frac{\beta_1\beta_2}{\beta_1+\beta_2}\sqrt{\frac{I_Q}{2\beta_2}} + \left(\sqrt{\frac{I_Q}{2\beta_1}} + B\right)^2\left(1 - \frac{B\left(\sqrt{\frac{\beta_2}{\beta_1+\beta_2}}\right)}{\sqrt{\frac{I_Q}{2\beta_2} + \frac{I_Q}{2\beta_1} + 2B\sqrt{\frac{I_Q}{2\beta_1}} + B^2\frac{\beta_2}{\beta_1+\beta_2}}}\right)\right)^2$$

(b)

$$i_o = \left[\sqrt{\left(\frac{1}{\beta_{1o}} + \frac{2}{\beta_{2o}}\right)f^2 - 2Bf} - \frac{f}{\sqrt{\beta_{1o}}}\right]^2$$

$$f = \frac{\beta_{2o}}{\beta_{2i}}\frac{\sqrt{i_i}}{\left[\sqrt{\left(\frac{1}{\beta_{1i}} + \frac{2}{\beta_{2i}}\right)} - \sqrt{\frac{1}{\beta_{1i}}}\right]} + \beta_{2o}B$$

(c)

$$i_o = \begin{cases} \frac{\beta_{o1}+\beta_{o2}}{\beta_i}i_i - 2\beta_{o2}\sqrt{\frac{i_i}{\beta_i}}B + \beta_{o2}B^2 & \text{if} \quad B \le \sqrt{\frac{i_i}{\beta_i}} \\ \frac{\beta_{o1}}{\beta_i}I_i & \text{if} \quad B \ge \sqrt{\frac{i_i}{\beta_i}} \end{cases}$$

(d)

$$i_o = i_i \sum_{i=1,P} b_i \alpha_i \frac{W}{L}$$

(e)

$$w_i^* = \frac{\beta_t}{\beta_b} w_i \left[1 + \frac{\sum_{i=1,N}\sqrt{w_i}}{N\sqrt{w_i}}\left(\sqrt{1 + \frac{N\left(\left(\frac{\beta_b}{\beta_t}I_{SS}\right) - \sum_{i=1,N} w_i\right)}{\left(\sum_{i=1,N}\sqrt{w_i}\right)^2}} - 1\right)\right]^2$$

(f)

Figura 2: Conjunto de ecuaciones de modelado.

zaje; y_i^* es el singleton cuyo antecedente de regla es máximo, es decir $w_i^*(\mathbf{x}) = max\{w_1^*(\mathbf{x}), w_2^*(\mathbf{x}), ...w_N^*(\mathbf{x})\}$.

3.2 Criterios para la elección de los parámetros del aprendizaje

Como todos los algoritmos basados en la disminución del gradiente, el weight perturbation es bastante sensible a los parámetros de aprendizaje. Valores grandes de la perturbación y de la razón de aprendizaje elevan la velocidad de aprendizaje, pero dificultan la convergencia. Por otra parte, valores pequeños de estos parámetros hacen que el sistema converja lentamente [5]. La regla outstar presenta un comportamiento similar, lo que se debe a que esta regla se puede interpretar como un algoritmo basado en disminución del gradiente. Aparte de estas consideraciones, serían deseables unos criterios más concretos a la hora de escoger los parámetros de aprendizaje. En las implementaciones hardware, esta elección viene condicionada por la resolución [5]. Los criterios en la sección 4 han sido la elección de un valor bastante pequeño de la perturbación, cuyo valor mínimo es el mínimo incremento posible para el parámetro a actualizar. Por otra parte, ya que weight perturbation implementa una derivada aproximada, la estabilidad en el punto de mínimo error se evita de forma intrínseca en este algoritmo, que fuerza pequeñas oscilaciones en torno a este mínimo cuando la convergencia tiene éxito, excepto en el caso en que una resolución limitada paralice la evolución del error. Una vez que se han escogido una perturbación y una máxima variación del error a la salida, la razón de aprendizaje viene determinada por el hecho de que el cambio mínimo del parámetro está limitado por la resolución, es decir:

$$\eta \ge \frac{pert \times \Delta\omega_{min}}{||\zeta(\omega) - \zeta(\omega + pert)||} \tag{4}$$

La razón de aprendizaje en la regla outstar se determina por el mismo procedimiento,

$$\mu \geq \frac{\Delta\omega_{min}}{|[T - y(\mathbf{x})]|} \qquad (5)$$

donde el máximo error deseado a la salida es el otro parámetro necesario para calcular μ.

4 Resultados

En esta sección se presentan algunos resultados para un sistema neuro-difuso con la arquitectura de la figura 1(a). La función muestra de la que se obtuvieron los datos pares de entrenamiento es 2.5+sin(πx)sin(πy). La figura 3 muestra esta función en el intervalo [0, 2]x[0,2], así como la superficie resultante del entrenamiento de un sistema con 36 reglas y funciones de pertenencia trapezoidales ideales, tras haber sido entrenada con 111 pares de datos. El sistema fue inicializado con las funciones de pertenencia uniformemente distribuidas a lo largo del universo de discurso, mientras que los singleton se igualaron al valor medio de la función. El error final RMSE está por debajo del 2.5%.

Dado que los algoritmos de la sección 3 tienen un tiempo de ejecución de varias horas en una estación de trabajo SPARC, y que son necesarias muchas iteraciones para obtener suficiente información del comportamiento del sistema, posteriores simulaciones se realizaron en el intervalo [0,1]x[0,1], lo que acelera las simulaciones sin pérdida de generalidad, gracias a la simetría de la superficie.

Los modelos de las secciones precedentes pueden introducirse en la arquitectura de la figura 1(a). Muchas combinaciones son posibles, si consideramos las tres posibilidades para los circuitos función de pertenencia y singleton. Consideraremos cinco posibilidades, una con ambos circuitos implementados con el transistor digital de la figura 1(f), y cuatro posibles combinaciones entre los dos restantes, es decir el transistor paralelo en el circuito función de pertenencia y el serie en el circuito singleton (lo llamaremos *parser*), el transistor serie en el circuito función de pertenencia y el paralelo en el circuito singleton (*serpar*), y ambos, paralelo y serie en los circuitos función de pertenencia y singleton (*parpar* y *serser* respectivamente).

Se realizaron muchas simulaciones para las cinco posibilidades del párrafo anterior, para diferentes valores de la resolución y tamaño del paso de aprendizaje, así como para los algoritmos híbrido y weight perturbation. Los bloques funcionales fueron diseñados para cubrir el mismo rango, que es de uno a cuatro para los valores singleton, de uno a dos para las pendientes de las funciones de pertenencia, y de -0.5 a 1.5 para los parámetros E_{kij}. Las figuras 4 y 5 muestran las curvas RMSE para estas simulaciones, donde los parámetros de aprendizaje se escogieron según los criterios de la sección 3. Para el algoritmo weight perturbation, estos parámetros fueron *pert* = 0.03, η/*pert*= 0.26 (ver ecuación (3)) para una resolución de 7bits, para todas las combinaciones excepto para la digital, donde los parámetros fueron *pert* = 0.12 y η/*pert*= 0.5 para la misma resolución. Para el algoritmo híbrido, el parámetror μ, además de los anteriores para weight perturbation, se fijó a 0.01 para las combinaciones "analógicas", excepto para la combinación con el transistor serie en las dos capas adaptativas, donde su valor fue 0.03. En el caso "digital", su valor se fijó a 0.1.

Las curvas RMSE en las figuras 4 y 5 muestran un mejor comportamiento del algoritmo híbrido comparado con el algoritmo weight perturbation, lo que se puede observar en la Tabla I, donde dos parámetros de rendimiento se dan para ambos algoritmos. Los parámetros de rendimiento son [5]:

- *el valor asintótico del error RMSE* $E_\infty = \lim_{n\to\infty} E_{salida}(n)$, donde n es el número de épocas, por lo que este parámetro es el valor del error de salida RMSE después de un entrenamiento suficientemente largo y
- *el tiempo de convergencia* T_C, expresado en número de épocas desde el principio del entrenamiento hasta el momento en que E_{salida} se reduce por debajo de un determinado umbral. Una época se define como la aplicación de toda la serie de pares de entrenamiento al sistema. El umbral escogido en nuestro caso fue RMSE=0.1.

	error RMSE asintótico	tiempo de convergencia TC
parpar_h	0.099	17
parpar_wp	-	-
parser_h	0.097	3
parser_wp	0.085	3
serpar_h	0.084	9
serpar_wp	0.090	77
serser_h	0.096	17
serser_wp	0.063	68
digital_h	0.087	7
digital_wp	0.096	30

Tabla 1: Parámetros de rendimiento del aprendizaje.

Al margen de datos concretos, que pueden ser mejora-

dos por ajuste de los parámetros de aprendizaje, podemos decir que, en general, el algoritmo híbrido mejora el tiempo de convergencia. La razón podría encontrarse en que este algoritmo refuenza la localidad [6], ya que refuerza preferentemente las reglas que tienen máxima infuencia en cada dato de entrenamiento. Además de esta ventaja, la localidad es un aspecto importante en los sistemas difusos, ya que nos permite establecer una correspondencia entre el valor de un singleton de una regla y los valores de la función a aproximar en el área del universo de discurso que es "ocupada" por la regla. Esta "transparencia" nos permite introducir conocimiento en los sistemas difusos, lo que es su característica principal. Por tanto, un sistema neuro-difuso entrenado con el algoritmo híbrido se inicializará mucho mejor, y se disminuirá el riesgo de caer en mínimos locales.

Finalmente, podemos destacar la baja resolución requerida para los propósitos de arendizaje, 7 bits en nuestras simulaciones, lo que se puede conseguir con las tecnologías VLSI más baratas [7]. El incremento de la resolución no aporta necesariamente una gran mejora en la capacidad de aproximación, como ilustra la figura 6 para la configuración *parpar*.

Referencias

[1] F. Vidal-Verdú , A. Rodríguez-Vázquez, B. Linares-Barranco and E. Sánchez-Sinencio: "A Basic Building Block Approach to CMOS Design of Analog Neuro/ Fuzzy Systems", *Proc. of the Third IEEE International Conference on Fuzzy Systems*, June 1994.

[2] F. Vidal-Verdú, B. Linares-Barranco and A. Rodríguez-Vázquez: "Circuits and Algorithms for Adaptive Neuro-Fuzzy Analog Chips", *Proc. of the Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems*, (Turin, Italy), September 1994.

[3] M. Jabri and B. Flower: "Weight Perturbation: An Optimal Architecture and Learning Technique for Analog VLSI Feedforward and Recurrent Multilayer Networks" *IEEE Transactions on Neural Networks*, Vol. 3, No. 1, pp. 154-157, January 1992.

[4] J. M. Zurada. "*Introduction to Artificial Neural Systems*". West Publishing 1992.

[5] L. M. Reyneri and Enrica Filippi: "An Analysis on the Performance of Silicon Implementations of Backpropagation Algorithms for Artificial Neural Networks". *IEEE Transactions on Computers*, Vol. 40, No. 12, December 1991.

[6] J. Nie and D. A. Linkens: "Learning Control Using Fuzzified Self-Organizing Radial Basis Function Network", *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 4, pp. 280-287, November 1993.

[7] M.J.M. Pelgrom et al.: "Matching Properties of MOS Transistors". *IEEE Journal of Solid-State Circuits*, Vol. 39, pp. 1433-1440, June 1990.

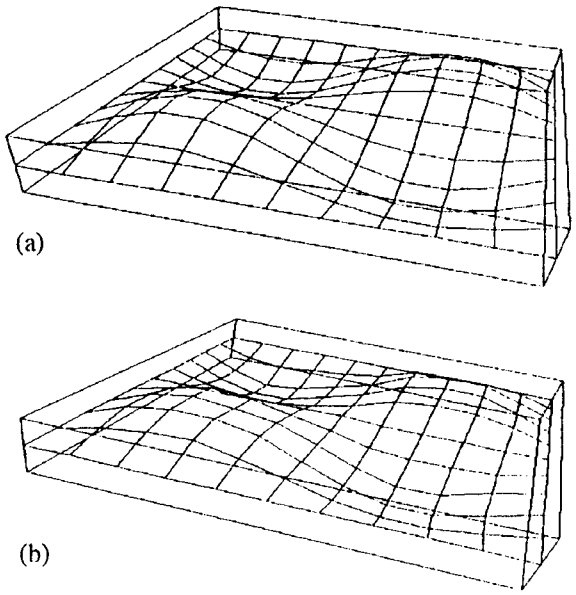


Figura 3: Superficie original (a) y superficie aprendida (b) en el intervalo [0,2]x[0,2].

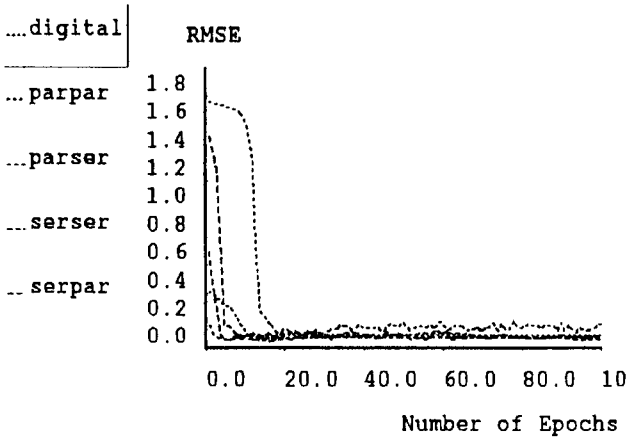


Figura 4: Curvas de error RMSE para distintas configuraciones y para el algoritmo weight perturbation.

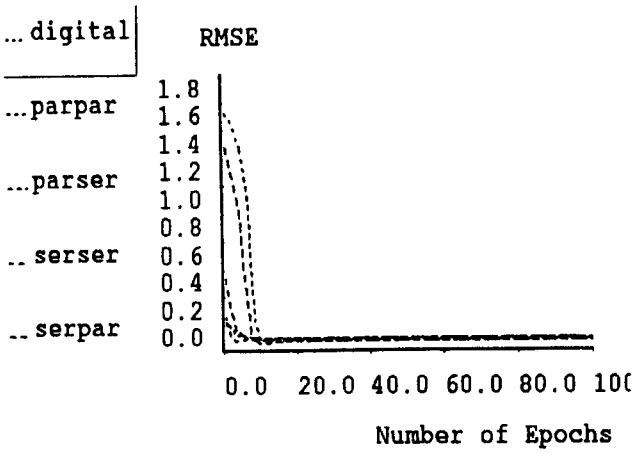


Figura 5: Curvas de error RMSE para distintas configuraciones y para el algoritmo híbrido.

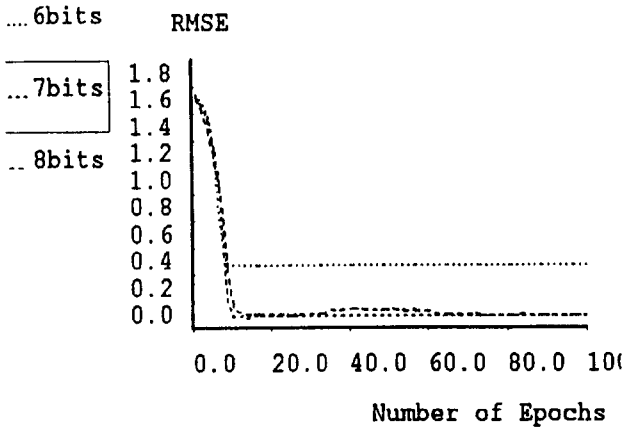


Figura 6: Ejemplo de influencia de la resolución en el aprendizaje.

Memoria Asociativa Difusa para reconocimiento de modelos

Mario Reyes de los Mozos
Dpto. Informàtica
Universitat Autònoma de Barcelona
Centre Nacional de Microelectrònica (CSIC)
e-mail: mario@cnm.es

Elena Valderrama
Dpto. Informàtica
Universitat Autònoma de Barcelona
Centre Nacional de Microelectrònica (CSIC)
e-mail: elena@cnm.es

Santiago Ullastre
Dpto. Informàtica
Universitat Autònoma de Barcelona

Resumen

En este trabajo se presenta una F.A.M. (Fuzzy Associative Memory), diseñada e implementada mediante FPGA's (Field Programmable Gate Arrays).

Esta memoria asociativa se compone de tres bloques básicos: módulo de cálculo de similitud, módulo de ordenación y módulo de control, así como un número detreminado de memorias internas. Este tipo de memorias son muy útiles en aplicaciones donde se presenta el problema de reconocer modelos.

Palabras Clave:*Fuzzy Associative Memory, modelos, atributos, Field Programmable Gate Arrays, reconocimiento.*

1 Introducción

En los sistemas donde se presenta el concepto de memoria dos son las fases que se requieren. En una primera fase de adquisición se memorizan ciertos aspectos de la situación actual en la que se halla el sistema, acontecimientos u objetos que la caracterizan. En una segunda fase de actualización se identifican o restituyen los datos memorizados en el transcurso de la fase de adquisición. Diversos son los comportamientos que pueden presentar estos sistemas: reconocimiento, reconstrucción, evocación o reaprendizaje. En todos ellos la información se almacena en la memoria mediante una representación interna, pudiendo ser completamente diferente de la información adquirida. En el modelo de memoria tanto o más importante es cómo se recupera la información almacenada que cómo se codifica ésta. Podemos clasificar los dispositivos de memoria según diversos aspectos, pudiendo clasificarlos según el tipo de acceso (*Accesibles por Dirección, Accesibles por Contenido*), o según se almacene la información (*Localizadas o Distributivas*).

De entre todos los tipos de memoria que se pueden presentar nosotros nos centramos en las **memorias asociativas**. En las memorias asociativas la información se estructura en *modelos*, representado cada uno de ellos por su vector propio, donde cada componente de éste proporciona información sobre una determinada *característica o atributo* del modelo. Este tipo de memorias aplican una medida de similitud (*distancia*) entre el dato de entrada y los modelos que forman parte de la memoria, - la información a recuperar -, proporcionando como resultado la información almacenada que mayor grado de similitud tiene referencia al dato de entrada. Dentro de las memorias asociativas existen dos grandes grupos: las *memorias distributivas* y las *memorias direccionables por contenido*, siendo de este último tipo la memoria que aquí se presenta.

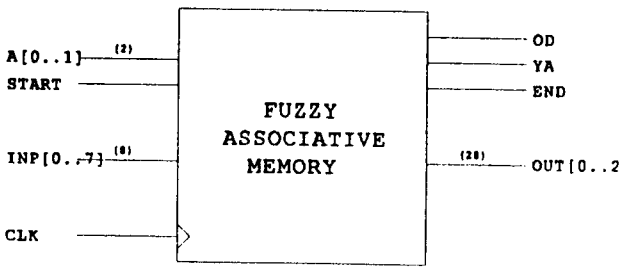


Figura 1: Fuzzy Associative Memory

En este trabajo se presenta una *memoria asociativa difusa* (*F.A.M. -> Fuzzy Associative Memory*), o más concretamente una *memoria difusa direccionable por contenido* (*F.C.A.M. -> Fuzzy Content Addressable Memory*). Como ya se ha comentado en líneas anteriores, cada modelo es representado por su vector de característica, donde para el caso que se presenta cada componente almacena un valor del intervalo [0,1], correspondiendo con el grado de presencia de la característica asociada a dicho componente. De esta forma se reduce la importancia vital que tienen las desviaciones de bits en las memorias clásicas direccionables por contenido. Reduciendo a su vez el número de modelos necesarios para llegar a unos resultados aceptables de recuperación de la información.

En este trabajo se presenta el diseño de una FAM implementada mediante FPGA's (*Field Programmable Gat Array*), concretamente se han necesitado 3 componentes, una para cada uno de los bloques básicos de la FAM: *módulo de cálculo de similitud, módulo de ordenación y unidad de control*. La FAM se presenta bajo el control de un ordenador personal, incluyendo por tanto una circuitería adicional para realizar la interfase de la memoria y el sistema.

2 Características de la FAM

La memoria que se presenta es programable, de forma que además de la función básica de lectura de datos, presenta otras funciones de consulta del estado interno de la FAM. En esta primera versión se implementan cuatro funciones: **i)** inicialización de la FAM; **ii)** lectura básica de la FAM; **iii)** consulta del vector difuso de atributos de un determinado modelo; y **iv)** consulta de un modelo y su grado de similitud asociado respecto a la última entrada dada.

La FAM presenta una estructura modular fácilmente modificable según las restricciones que planteen las diferentes aplicaciones (cambio de función de similitud, cambio del algoritmo de ordenación, suma de nuevas funciones sobre la memoria, como podrían ser: añadir nuevos modelos, modificar modelos ya almacenados,...).

No todo son maravillas, la FAM también presenta una serie de restricciones en cuanto al número de bits por atributo, y al número de modelos y atributos. Cada atributo se representa con un grupo de cuatro bits, de forma que el grado de presencia de una determinada característica tiene 16 posibles valores. La otra restricción que presenta hace referencia al número de modelos y atributos posibles. Para ambos casos el número mínimo es 16 y el número máximo es 255, debiendo cumplir la siguiente expresión:

$$(n^{o}\textit{de atributos})\ (n^{o}\textit{de modelos}) \leq 4096 \qquad (1)$$

3 Memorias internas

Tal como muestra la figura 2 se requieren cuatro módulos de memoria. En la ***memoria de modelos*** se almacenan los vectores difusos correspondientes a los modelos que componen la base de datos de la FAM. Es una memoria de atributos, es decir, cada celda de memoria corresponde al grado de presencia de una cierta característica en un determinado modelo.

En la ***memoria de trabajo*** se almacena la dirección de memoria de modelos en la cual se encuentra el atributo cero de cada uno de los modelos. Gracias a esta memoria podemos realizar diversas consultas sobre la FAM.

Al realizar una lectura de la FAM se calcula el grado de similitud de la entrada con cada uno de los modelos, siendo necesario almacenar en la ***memoria de entrada*** el vector difuso que representa la entrada dada.

Por último en la **memoria de resultados** se guardan los resultados obtenidos en el cálculo del grado de similitud de la entrada con cada uno de los modelos. Posteriormente se ordena esta memoria de tal forma que el modelo con mayor grado de similitud se sitúe en la posición cero y el de menor grado en la posición **m**, siendo **m** el número de modelos que componen la FAM.

4 Descripción de los módulos

4.1 Cálculo de similitud

Para el cálculo de la medida de similitud muchas son las funciones candidatas, siendo la aplicación a la que van enfocadas la que determina cual de ellas es la finalmente escogida. En este caso se ha escogido la propuesta por B. Kosko, debido principalmente a su versatilidad con respecto a su uso:

$$M.S. \quad \frac{\sum_{i}^{n} \mu_{A\,B}}{\sum_{i}^{n} \mu_{A\,B}} \qquad (2)$$

En el diseño de la FAM nos marcamos dos objetivos básicos: por un lado el conseguir una FAM con un tiempo de lectura breve; y por otro conseguir un hardware compacto y reducido. Con el fin de obtener el mejor

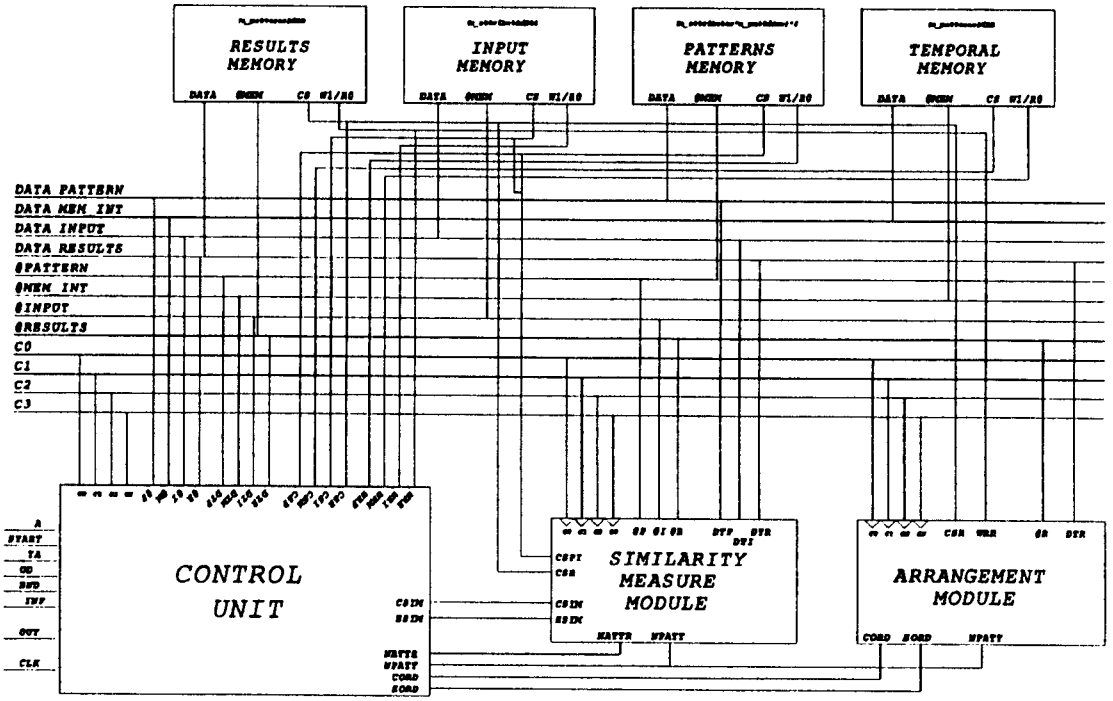


Figura 2: Estructura modular de la memoria.

equilibrio entre los dos objetivos se ha diseñado la arquitectura de la FAM utilizando técnicas pipeline. Por otro lado el módulo es totalmente independiente del resto de módulos que componen la FAM, comunicándose con ellos de forma asíncrona. Cuando se presenta un pulso alto sobre la señal CSIM el módulo toma el control, iniciándose el proceso de cálculo de la función de similitud para todos los modelos que componen la FAM, enviando un pulso alto sobre la señal ESIM una vez ha finalizado.

4.2 Módulo de ordenación

Para la ordenación de la memoria de resultados varios son los métodos posibles. De entre el amplio abanico de posibilidades se ha optado por implimentar una variante del método de ordenación por selección. (Actualmente se está diseñando un nuevo algoritmo con el fin de mejorar el tiempo de respuesta de éste). Por otro lado, y de igual forma que el módulo anterior, el módulo de ordenación es totalmente independiente del resto de módulos, comunicándose de forma asíncrona. Cuando se presenta un pulso sobre la señal CORD el módulo toma el control, iniciándose el proceso de ordenación de la memoria de resultados, enviando un pulso alto sobre la señal EORD una vez ha finalizado.

4.3 Unidad de control

La unidad de control se encarga de sincronicar el funcionamiento del conjunto de los módulos que componen la FAM para todas las funciones posibles que se presenta. Además controla la comunicación de la FAM con el exterior.

La comunicación interna con cada uno de los módulos de la FAM es asíncrona, así como la comunicación exterior de la FAM con el PC (en este caso). El proceso es el siguiente: **i)** primero determina la función que debe realizar la FAM, para ello se utilizan las señales A[0..1] (Tabla 1); **ii)** a continuación se coloca un pulso alto sobre la señal START, dando comienzo la función indicada; **iii)** se realiza la función especificada; **iv)** por último la FAM coloca un pulso alto sobre la línea END una vez ha finalizado el proceso de la función especificada.

La FAM presenta cuatro funciones. La **función 0** se encarga de inicializar la FAM (fase de adquisición), lo que implica almacenar la información necesaria en la FAM (*vectores difusos de cada uno de los modelos*).

La **función 1** corresponde al proceso de lectura de la memoria (FAM), lo que implica calcular el grado de similitud de la entrada con cada uno de los modelos que

forman la FAM, así como ordenar la memoria de resultados, de forma que se proporcione como resultado el modelo que mayor grado de similitud posee, el más parecido a la entrada.

A[0..1]	Función
00	0: Inicialización de la FAM
01	1: Lectura de la FAM
10	2: Consulta del vector difuso de un modelo determinado
11	3: Consulta de los modelos según su grado de similitud con respecto a la última entrada.

Tabla 1: Funciones de la unidad de control

La **función 2** proporciona el vector difuso de un modelo determinado, atributo a atributo. El modelo se determinaa través del identificador correspondiente.

La **función 3** proporciona el contenido de la posición indicada de la memoria de resultados, permitiendo conocer el orden de similitud entre la entrada y los diversos modelos que componen la FAM.

5 Implementación física

Se ha diseñado una placa donde se halla la FAM y la circuitería necesaria para comunicación de ésta con el ordenador al que está conectado.

Cada uno de los módulos que se ha descrito anteriormente se implementa en una FPGA's de la casa Xilinx. Para la implementación del módulo de cálculo de similitud se ha utilizado una 3090 de 175 pins. Para el módulo de ordenación se ha utilizado una 3064 de 84 pins, y para la unidad de control una 3090 de 175 pins.

A las anteriores FPGA's se ha de sumar una adicional, necesaria para la implementación del interfase entre la FAM y el ordenador.

Se han realizado una serie de simulaciones mediatne le simulador lógico SILOS con el objetivo de determinar el tiempo de respuesta de la FAM para cada una de las funciones que implementa. El tiempo de respuesta viene expresado en número de ciclos pipeline. El tiempo de respuesta para la *función cero* se obtiene mediante la expresión :

$$2+(n^{o}\ atributos\ n^{o}\ modelos)+2 \qquad (3)$$

El tiempo de respuesta de la función uno, es decir el tiempo de lectura de la FAM viene dado por la siguiente expresión:

$$1+n^{o}\ atributos+ t_{SM}+ t_{OR}+1 \qquad (4)$$

donde t_{SM} es el tiempo de respuesta del módulo de cálculo de la medida de similitud, y el término t_{OR} es el tiempo de respuesta del proceso de ordenación. El tiempo de respuesta del cálculo de similitud viene dado por la expresión:

$$1 +(n^{o}\ atributos\ n^{o}\ modelos) + t_{DIV} + 1 \qquad (5)$$

donde t_{DIV} es el tiempo de respuesta del divisor (doce ciclos pipeline). Y el tiempo de respuesta del proceso de ordenación es:

$$2(n\text{-}1) + 3\sum_{i=0}^{n} i \qquad (6)$$

para el mejor caso, es decir cuando la memoria de resultados está totalmente desordenada. Y:

$$5(n\text{-}1) + 3\sum_{i=0}^{n-2} i + 1 \qquad (7)$$

para el peor caso, es decir cuando la memoria está totalmente desordenada.

El tiempo de respuesta de la *función dos* viene determinado por la expresión:

$$3 + n^{o}\ atributos + 1 \qquad (8)$$

Y por último, el tiempo de respuesta de la *función tres* es de 3 ciclos pipeline.

6 Conclusiones

En este artículo hemos presentado una FAM (*Fuzzy Associative Memory*) implementada mediante FPGA's. La FAM presenta una estructura modular con el fin de facilitar

las posibles modificaciones que se requieran (cambio de función de similitud, cambio del algoritmo de ordenación, introducción de nuevas funciones, ...). Se ha diseñado una placa impresa donde se encuentra la FAM, así como la interfase de ésta con el ordenador con la que se comunica. Para la realización de la placa se han utilizado FPGA's de la casa Xilinx, así como diferentes memorias y componentes discretos.

Este tipo de memorias son muy útiles en aplicaciones de reconocimiento de modelos (como podría ser el reconocimiento de matrículas -se ha programado un software de simulación de esta aplicación específica-), así como para sistemas multisensor, o determinadas aplicaciones de control.

7 Agradecimientos

Este trabajo se inició durante la estancia de Mario Reyes de los Mozos en el FLSI (*Fuzzy Logic System Institute*), sito en la ciudad de Iizuka (Japón). Damos las gracias al Institut Català de Tecnologia y al Departament de Treball de la Generalitat de Catalunya por el interés mostrado en la realización de dicha visita.

Referencias

[1] Kosko, B.: *Neural Networks and Fuzzy Systems; a dynamical approach to machine intelligence*, New Jersey, Prentice Hall, 1992.

[2] Fujimoto J., Yasuda S. et al.: Speaker-independent word recognition using fuzzy theory, en *Preprints of Second IFSA Congress*, Tokyo, July 20-25, 1987.

[3] Kosko, B.: Fuzzy Associative Memories, en *Fuzzy Expert Systems*, A. Kandel, Ed. Addison-Wesley, 1987.

[4] Lisa Mingo, F.: *Clasificación de caracteres mediante redes neuronales artificiales. Aplicación al reconocimiento de matriculas de coches*, Memoria del trabajo experimental dentro del programa de tercer ciclo de Microelectrónica. Universitat Autònoma de Barcelona, Bellaterra, Julio de 1994.

Metodología para la Materialización de Algoritmos Borrosos: de una Especificación de Alto Nivel a un Asic*

Jean-Pierre Deschamps
Dpto. Ingeniería Informática
Universidad Autónoma de Madrid
e-mail: tallerm@ccuam3.sdi.uam.es

José Ignacio Martínez Torre
Dpto. Informática y Automática
Facultad de Ciencias Físicas
Universidad Complutense de Madrid
e-mail: jimt@eucmvx.sim.ucm.es

Resumen

En este documento se pone de manifiesto el hecho de que los algoritmos clásicos de control borroso se prestan muy bien a la aplicación de métodos de síntesis de alto nivel (SAN). De este modo, el diseñador puede elegir entre distintas alternativas de materialización en función de los requisitos impuestos.

Para ello se ha propuesto una posible especificación de un algoritmo borroso que sea útil para SAN, se han obtenido dos planificaciones y dos rutas de datos apropiadas para la materialización mediante Standard Cells y FPGAs y se ha diseñado una celda de biblioteca adecuada para las operaciones requeridas.

Palabras clave: *Síntesis de alto nivel, algoritmo borroso, planificación, ruta de datos, hardware.*

1 Introducción

Los controladores borrosos pueden ser materializados de múltiples maneras con *hardware* estándar como ordenadores personales con tarjetas de adquisición de datos, ordenadores industriales basados en un bus estándar, etc. Para aplicaciones en las que se han de ejecutar un número elevado de reglas en muy poco tiempo, se han diseñado microcontroladores dedicados (coprocesadores [1] - [3]).

Este artículo pone de manifiesto el que los algoritmos de control borroso pueden ser integrados en un ciclo de diseño general, con una cierta especialización, ya que se prestan bien a la aplicación de métodos clásicos de síntesis de alto nivel basados en la planificación de las operaciones (*scheduling*) y en la asignación de recursos y memorias (*memory and resource allocation*). Esto sugiere una materialización en la que en vez de utilizar un microcontrolador dedicado, se usaría un circuito integrado específico (*Standard Cell (SC), FPGA*, etc.) para la aplicación concreta.

Las herramientas clásicas de síntesis permiten explorar el espacio de las soluciones del diseño en función de los requisitos de la aplicación, de modo que el diseñador dispone de una metodología que le permite obtener rápidamente una familia de materializaciones de un algoritmo borroso con un grado de flexibilidad mayor que si utilizara un microcontrolador dedicado. Además, las herramientas de síntesis permiten generar de forma automática la lista de conexiones (*netlist*) del circuito necesaria para la materialización.

En el segundo apartado, se presenta el ciclo de diseño completo. A continuación en el tercer apartado, se describe la especificación de un algoritmo borroso y los cálculos que conlleva la obtención del resultado. Como cuarto apartado, se especifica el algoritmo para la síntesis de alto nivel. Seguidamente en el apartado quinto, se obtiene una planificación y una ruta de datos apropiada para *Standard Cells* , una nueva planificación y ruta de datos adecuada para la materialización con *FPGAs*, y además se describe una celda de biblioteca diseñada al efecto. Por último, se presentan las conclusiones y algunos futuros desarrollos previstos.

2 Ciclo de diseño de un algoritmo borroso

Se propone un posible ciclo de diseño completo capaz de desarrollar un algoritmo borroso, o más genéricamente de desarrollar una aplicación para controlar algún proceso que necesite materializar algún algoritmo borroso, tal y como se muestra en la figura 1.

Por tanto, se puede partir de una especificación de un algoritmo borroso, o bien de una especificación de un proceso.

En el primer caso, si la especificación ha sido previamente validada, sólo tendremos que convertir el algoritmo para que sirva como entrada para la síntesis; si no ha sido validada, tendremos que utilizar algún simulador, especializado o no, para comprobar que funciona y poder pasar a síntesis.

* Este trabajo ha sido subvencionado por la CICYT mediante el proyecto TIC 92/0088.

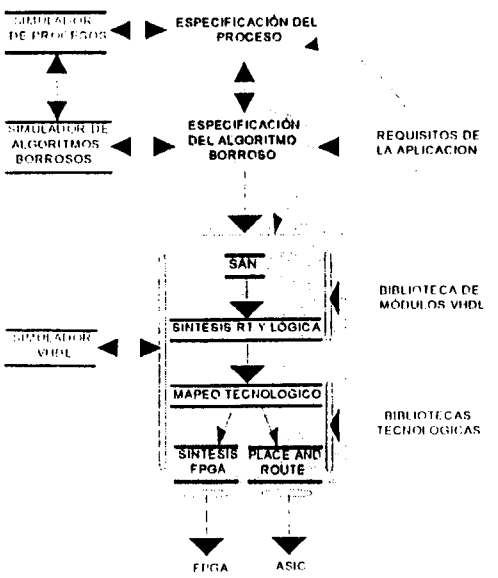


Figura 1: Ciclo de diseño propuesto.

En el segundo caso, de nuevo si la especificación del proceso ya está validada tendremos que especificar y simular solo el algoritmo borroso e integrarlo en el proceso global para comprobar su bondad antes de pasar a síntesis; si no ha sido validada, tendremos que especificar y simular el proceso completo además del propio algoritmo borroso para poder pasar a síntesis.

En la fase final es donde se realiza la síntesis de alto nivel, la síntesis RT y lógica y el mapeo tecnológico, en forma de *FPGA*, *ASIC*, etc, de la ruta de datos y del controlador generados. Durante toda esta fase se hacen simulaciones VHDL para comprobar la consistencia de los resultados en las diferentes etapas intermedias.

3 Algoritmo borroso

Un algoritmo borroso se puede especificar de la siguiente manera:

Lista de las señales de entrada:

x_1, x_2, ... , x_{ult}, donde x_i pertenece al conjunto $\{0, 1, 2, ..., 2^{mi}-1\}$.

Lista de las señales de salida:

z_1, z_2, ... , z_{ult}, donde z_j pertenece al conjunto $\{0, 1, 2, ..., 2^{mj}-1\}$.

Funciones de pertenencia de las entradas:

$A_1, A_2, ..., A_{ult}$, con A_k: $\{0, 1, 2, ..., 2^{ak}-1\} \rightarrow [\,0, 1]$.

Funciones de pertenencia de las salidas:

$B_1, B_2, ..., B_{ult}$, con B_l: $\{0, 1, 2, ..., 2^{bl}-1\} \rightarrow [\,0, 1\,]$.

Reglas de inferencia:

IF x_1 is A_i and ... and x_{ult} is A_{ult}
THEN z_1 is B_1 and ... and z_{ult} is B_{ult}

Con esta especificación, el cálculo del valor de una salida, por ejemplo z_1, se realizaría del siguiente modo:

1.- Se escribe la lista de todas las reglas R_i que definen el grado de pertenencia de z_1 a B_1:

IF x_1 is A_i and ... and x_{ult} is A_k THEN z_1 is B_1
IF x_1 is A_j and ... and x_{ult} is A_l THEN z_1 is B_1
...

2.- Se calculan los valores r_i:

$r_1 = \min (A_i(x_1), ..., A_k(x_{ult}))$
$r_2 = \min (A_j(x_1), ..., A_l(x_{ult}))$
...

donde r_i representa el grado de veracidad mínimo que las entradas integrantes de la regla R_i aportan a la salida z_1.

3.- Se calcula el valor w_{11}:

$w_{11} = \max (r_1, r_2, ...)$

w_{11} representa que si bien cada regla aporta un grado de veracidad a la salida z1, al activarse varias reglas sobre la salida, el grado de certeza final debe ser el máximo de los valores aportados por todas las reglas activadas.

4.- Se repiten los pasos 1 a 3, generando los grados de veracidad de la salida z_1 respecto al resto de sus etiquetas B_2, B_3, ... B_{ult}, mediante el calculo de w_{12}, w_{13}, ..., w_{1ult}.

5.- Hecho esto, calculamos la acción concreta a llevar a cabo como el centro de gravedad de la función resultante , es decir:

$z_1(y) = \max(\min(w_{11}, B_1(y)), \min(w_{12}, B_2(y)), ..., \min(w_{1ult}, B_{ult}(y)))$

Por tanto:

$$z_1 = \frac{\sum_{y=0}^{2^{m_1}-1} y \cdot z_1(y)}{\sum_{y=0}^{2^{m_1}-1} z_1(y)}$$

La figura 2 muestra gráficamente el modo de cálculo.

Se ha supuesto en todo el desarrollo que las especificaciones son coherentes, es decir que las entradas o salidas respecto a sus funciones de pertenencia cubren el mismo dominio.

Como es bien conocido, existen otras alternativas para conseguir que la salida sea útil en el mundo real, no borroso. Por ejemplo, la llamada media ponderada que consiste en calcular los centros de gravedad (z_{ij}) de cada una de las funciones B_j ponderadas por el grado de certeza de que z_i sea B_j.

Para z_1 calcularíamos los centros de gravedad de las funciones:

$z_{11}(y) = \min (w_{11}, B_1(y))$
$z_{12}(y) = \min (w_{12}, B_2(y))$
...
$z_{1ult}(y) = \min (w_{1ult}, B_{ult}(y))$

y luego la media ponderada con el grado de certeza de cada afirmación:

$$z_1 = \frac{w_{11} \cdot z_{11} + w_{12} \cdot z_{12} + \cdots}{w_{11} + w_{12} + \cdots}$$

de otro modo:

$$z_1 = \frac{C_1(w_{11}) + C_2(w_{12}) + \cdots}{w_{11} + w_{12} + \cdots}$$

Donde los C_j se generan de la siguiente manera:

$$C_j(w_{ij}) = \frac{w_{ij} \cdot \sum_{y=0}^{2^{mj}-1} y \cdot \min(w_{ij}, B_j(y))}{\sum_{y=0}^{2^{mj}-1} \min(w_{ij}, B_j(y))}$$

En lugar de utilizar las funciones de pertenencia B_1, B_2,....B_{ult} utilizaremos las funciones C_1, C_2, ... C_{ult}, donde: C_k: [0, 1] → [0, 1, 2, ..., 2^{bk}-1].

Una buena opción es calcular antes de la materialización del algoritmo todos esos valores y dejarlos almacenados en memorias.

4 Síntesis de alto nivel: especificación y algoritmo

A continuación se muestra que una especificación tipo de un algoritmo borroso clásico se adecua muy bien a los problemas que resuelven las técnicas de SAN.

Trataremos de sintetizar un sistema, con dos entradas (x_1, x_2), y una salida (z), especificado como sigue:

1.- x_1, x_2: números de N bits.

2.- z: número de M bits.

3.- A_1, A_2, A_3: funciones de pertenencia comunes para las entradas. A_i: {0, 1, 2, ..., 2^N-1} → [0, 1].

4.- B_1, B_2, B_3: funciones de pertenencia para la salida, B_j: {0, 1, 2, ..., 2^M-1} → [0, 1].

5.- Reglas:

If x_1 is A_1 and x_2 is A_1 then z is B_1
If x_1 is A_1 and x_2 is A_2 then z is B_2
If x_1 is A_1 and x_2 is A_3 then z is B_3
If x_1 is A_2 and x_2 is A_1 then z is B_1
If x_1 is A_2 and x_2 is A_2 then z is B_2
If x_1 is A_2 and x_2 is A_3 then z is B_3
If x_1 is A_3 and x_2 is A_1 then z is B_1
If x_1 is A_3 and x_2 is A_2 then z is B_2
If x_1 is A_3 and x_2 is A_3 then z is B_3

Desde la especificación se puede obtener de forma automática el siguiente algoritmo para el cálculo de z:

$p_1 = A_1(x_1)$
$p_2 = A_2(x_1)$
$p_3 = A_3(x_1)$
$q_1 = A_1(x_2)$
$q_2 = A_2(x_2)$
$q_3 = A_3(x_2)$
$r_1 = \min(p_1, q_1)$
$r_2 = \min(p_1, q_2)$
$r_3 = \min(p_1, q_3)$
$r_4 = \min(p_2, q_1)$
$r_5 = \min(p_2, q_2)$
$r_6 = \min(p_2, q_3)$
$r_7 = \min(p_3, q_1)$
$r_8 = \min(p_3, q_2)$
$r_9 = \min(p_3, q_3)$
$r_{14} = \max(r_1, r_4)$
$w_1 = \max(r_{14}, r_7)$ $(= \max(r_1, r_4, r_7))$
$r_{25} = \max(r_2, r_5)$
$w_2 = \max(r_{25}, r_8)$ $(= \max(r_2, r_5, r_8))$
$r_{36} = \max(r_3, r_6)$
$w_3 = \max(r_{36}, r_9)$ $(= \max(r_3, r_6, r_9))$
$v_1 = C_1(w_1)$
$v_2 = C_2(w_2)$
$v_3 = C_3(w_3)$
$s_1 = v_1 + v_2$
$N = s_1 + v_3$ $(= v_1 + v_2 + v_3)$
$s_2 = w_1 + w_2$
$D = s_2 + w_3$ $(= w_1 + w_2 + w_3)$
$z = N / D$

donde todos los identificadores que aparecen a la izquierda del igual son variables intermedias salvo la salida. Esta descripción es ya sintetizable.

5 Síntesis de alto nivel

Para llevar a cabo el ejemplo se ha hecho uso de una herramienta de síntesis de alto nivel diseñada por

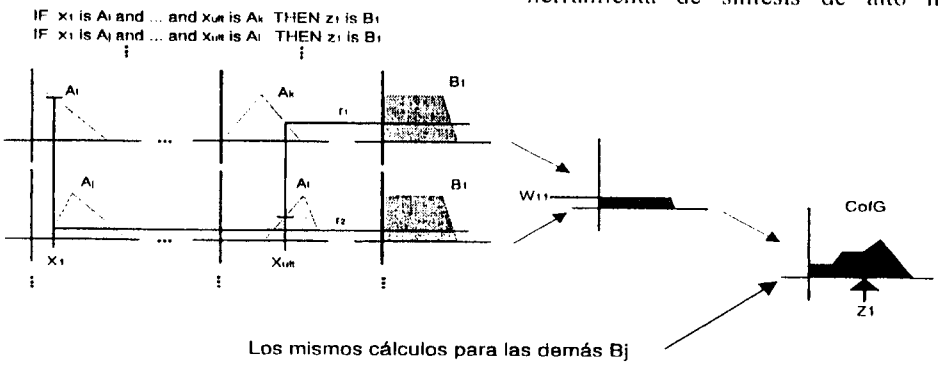


Figura 2: Explicación gráfica de los cálculos.

el grupo de Arquitectura de Computadoras del departamento de Informática y Automática de la UCM, llamada FIDIAS [4].

5.1 Planificación y asignación para SC

La primera tarea dentro de la síntesis de alto nivel consiste en planificar cómo y cuándo se van a realizar cada una de las operaciones del algoritmo. Dependiendo de las restricciones de la aplicación (tiempo de ejecución, recursos disponibles, etc) las soluciones para materializar el algoritmo pueden ser muy diferentes. Encontrar todas esas soluciones es la tarea del planificador integrado en la herramienta [5].

Si suponemos que tenemos disponibles todas las operaciones y que todas ellas son monociclo, necesitaríamos 10 ciclos de reloj para el cálculo del algoritmo, contando como ciclos la lectura y escritura de los datos desde y hacia los puertos del circuito.

En la planificación ejemplo suponemos como restricciones *hardware* el que solo disponemos de un recurso para cada una de las operaciones A_1, A_2, A_3, C_1, C_2, y C_3 (Memorias), de 1 divisor, y de 3 recursos multioperacionales con min, max y + como operaciones permitidas.

En la figura 3 se muestra el grafo de flujo de datos generado tras la planificación, donde los nodos ASN no son operacionales.

La planificación obtenida procesa los datos en 12 ciclos de reloj del siguiente modo:

M1: Lectura de los puertos x_1 y x_2;
M2: $p_1 = A_1(x_1)$,$q_2 = A_2(x_2)$, $q_3 = A_3(x_2)$;
M3: $p_2 = A_2(x_1)$, $p_3 = A_3(x_1)$, $q_1 = A_1(x_2)$, $r_2 = \min(p_1, q_2)$, $r_3 = \min(p_1, q_3)$;
M4: $r_1 = \min(p_1, q_1)$, $r_5 = \min(p_2, q_2)$, $r_8 = \min(p_3, q_2)$;
M5: $r_4 = \min(p_2, q_1)$, $r_6 = \min(p_2, q_3)$, $r_{25} = \max(r_2, r_5)$;
M6: $r_7 = \min(p_3, q_1)$, $r_{14} = \max(r_1, r_4)$, $r_{36} = \max(r_3, r_6)$;
M7: $r_9 = \min(p_3, q_3)$, $w_1 = \max(r_{14}, r_7)$, $w_2 = \max(r_{25}, r_8)$;
M8: $w_3 = \max(r_{36}, r_9)$, $v_1 = C_1(w_1)$, $v_2 = C_2(w_2)$;
M9: $s_1 = v_1 + v_2$, $s_2 = w_1 + w_2$, $v_3 = C_3(w_3)$;
M10: $N = s_1 + v_3$, $D = s_2 + w_3$;
M11: $z = N / D$;
M12: Escritura en el puerto z;

Una vez hemos decidido una planificación de las operaciones, nos queda seleccionar con qué unidades operacionales concretas y con qué interconexionado vamos a materializar el circuito. Esta tarea la lleva a cabo el asignador de *hardware* integrado en la herramienta.

Un posible diseño de la ruta de datos aparece en la figura 4, donde las funciones A_1, A_2, A_3 y C_1, C_2 y C_3 han sido realizadas con Memorias (*ROM*).

Este tipo de síntesis es útil para materializaciones con *Standard Cells* o *Sea of Gates*, ya que sus sistemas de desarrollo suelen incluir compiladores de Memorias.

En el caso del ejemplo, si N = M = 8 (256 elementos como universo de discurso para entradas y salidas), si todos los grados de pertenencia se expresan con una precisión de 8 bits y si los valores de las funciones C_i se almacenan con 16 bits de precisión, se necesitarán tres bloques de memoria de 256 *bytes* para A_1, A_2, A_3 y otros tres bloques de 512 *bytes* para C_1, C_2, y C_3.

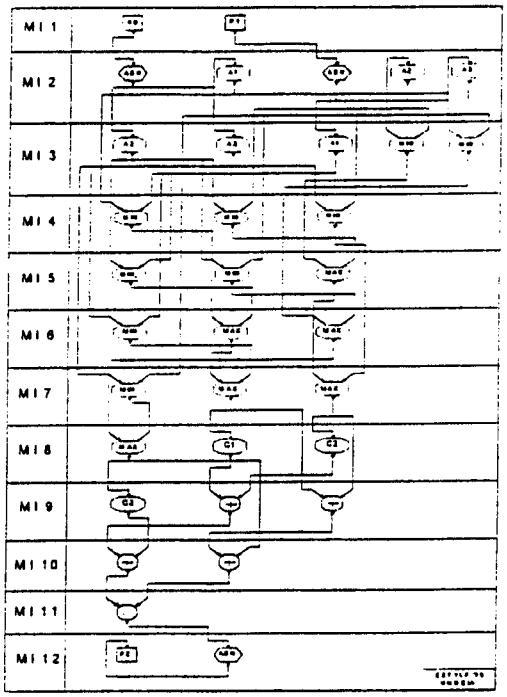


Figura 3: Planificación de las operaciones.

La frecuencia máxima de muestreo de las señales de entrada depende del número de ciclos y del tiempo de cálculo del recurso más lento (el divisor).

5.2 Planificación y asignación para FPGA

Una solución diferente consistiría en descomponer la división en operaciones elementales que pudiesen ser ejecutadas por alguno de los operadores ya disponibles en la ruta de datos, de modo que si bien el número de ciclos aumenta, también cada uno de ellos sería de menor duración. Por ejemplo, podríamos realizar la división mediante restas sucesivas.

Supongamos que disponemos de 1 recurso que materializa las operaciones A_1, A_2, A_3, C_1, C_2 y C_3 y de 3 recursos que materializan las operaciones min, max, suma y división.

Como se muestra en el ciclo de diseño es muy importante contar con celdas de biblioteca útiles para materializar algoritmos borrosos con prestaciones muy estrictas, de ahí que, como ejemplo, planteemos una celda adecuada para nuestro propósito.

5.2.1 Unidad aritmético-lógica

Vamos a describir una *ALU* capaz de realizar las operaciones min, max, suma y división,

Los dos operandos de la división son:

$N = C_1(w_1) + C_2(w_2) + ... + C_p(w_p)$

$D = w_1 + w_2 + ... + w_p$

donde $w_i \in [0,1]$ y $C_i: [0,1] \rightarrow [0, 2^m-1]$ siendo 2^m-1 el valor máximo de z.

Supongamos que existe un k tal que $p < 2^k$, entonces $N \in [0,2^{m+k}-1]$ y $D \in [0,2^k-1]$.Por otra parte, $z = N / D < 2^m$ con lo que $N < 2^m \cdot D \in [0, 2^{m+k}-1]$.

Si se expresan N y D con una precisión de r bits, entonces:

$D = d_{k-1} ... d_1 d_0 , d_{-1} d_{-2} ... d_{-(r-k)}$

$N = n_{m+k+1} ... n_1 n_0 , n_{-1} n_{-2} ... n_{-(r-m-k)}$

con lo que:

$2^m \cdot D = d_{k-1} ... d_1 d_0 , d_{-1} ... d_{-m} d_{-(m+1)} ... d_{-(r-k)}$

El divisor tiene que calcular el cociente de dos números de r bits (N y $2^m \cdot D$), tales que $N < 2^m \cdot D$, con una precisión de m bits. El cociente de dos números x e y, tales que $x < y$, puede ser calculado mediante una serie de divisiones sucesivas (*restoring algorithm*) con el siguiente algoritmo:

$r_0 := N$; $y := 2^m \cdot D$;
for i in 0 to m-1 loop
if $2 \cdot r_i \geq y$ then $q_i+1 := 1$; $r_i+1 := 2 \cdot r_i - y$;
else $q_i+1 := 0$; $r_i+1 := 2 \cdot r_i$;
end if;
end loop;
$Q := q_1 q_2 ... q_m$;

Las distintas operaciones son:

min(x, y): calcular x-y, y seleccionar x ó y en función del signo del resultado.

max(x, y): lo mismo.

x + y: operación básica de la unidad.

div(x, y): realización de una etapa del algoritmo de división, es decir, calcular $2 \cdot x - y$, seleccionar $2 \cdot x$ ó $2 \cdot x - y$ en función del signo del resultado y generar q también en función del signo.

La unidad aritmético-lógica se materializa con un sumador y una serie de multiplexores como se muestra en la figura 5.

Con este nuevo recurso de cálculo se puede modificar el algoritmo del apartado anterior, de tal modo que se sustituya la última microinstrucción :

$z := N / D \leftrightarrow Z := Reg_1 / Reg_0$

por la secuencia de microinstrucciones :

$Reg_2 := div(Reg_1, Reg_0)$, $Reg_1(M-1) := q$; $Reg_2 := div(Reg_2, Reg_0)$, $Reg_1(M-2) := q$;
...
$Reg_2 := div(Reg_2, Reg_0)$, $Reg_1(0) := q$;
$z := Reg_1$;

donde las operaciones han sido hechas por la nueva*ALU* y donde se supone que el registro Reg_1 puede trabajar como registro de desplazamiento donde se almacenen los *bits* del cociente z.

La planificación obtenida se muestra en la figura 6.

Este tipo de síntesis es conveniente para materializaciones por medio de *Gate Arrays*, *FPGA* o *PLD*, es decir, mediante dispositivos que, usualmente, no permiten integrar bloques de memoria compilados. Si se utilizara una *FPGA* (por ejemplo, una *LCA*) el controlador borroso se podría materializar con una *PROM* de configuración de la *FPGA* (función de las reglas de inferencia) y otra *PROM* que materialice las funciones de pertenencia A_1, A_2, A_3, C_1, C_2 y C_3.

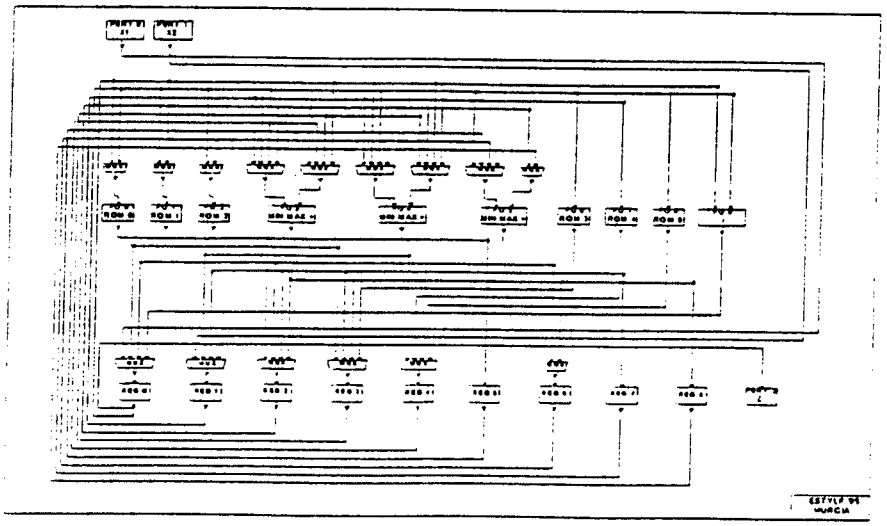


Figura 4: Ruta de datos para la primera planificación

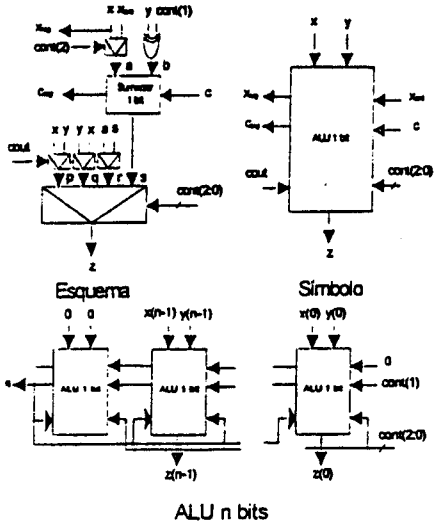


Figura 5: Materialización de una celda de biblioteca útil.

6 Conclusiones y futuros desarrollos

La principal conclusión de este artículo es que los algoritmos de control borroso se prestan bien a la aplicación de los métodos clásicos de síntesis de alto nivel. Así, el diseñador puede elegir en función de los requisitos del sistema la solución más conveniente. Si se necesitan frecuencias altas de muestreo, puede añadir recursos en paralelo disminuyendo el número de ciclos; en cambio, si no se necesita una frecuencia alta e importa el área total ocupada, se podrán reutilizar recursos aumentando el número de ciclos. Las herramientas de síntesis de alto nivel permiten explorar de forma sencilla y rápida el espacio de las soluciones y elegir la mejor configuración que cumpla con las expectativas. Además se propone incluir el desarrollo de algoritmos de control borroso en un ciclo de diseño más general.Para llevar a la práctica las ideas expresadas, conviene desarrollar celdas de biblioteca útiles tales como unidades aritmético-lógicas incluyendo min y max, divisores, desplazadores, etc. Cada una de ellas con diferentes versiones en cuanto a velocidad de cálculo, área consumida, consumo, etc y parametrizables en cuanto a precisiones , retardos, etc.
En el futuro, está previsto el desarrollo de un puesto de trabajo para el diseño de controladores borrosos que incluirá: un software de desarrollo y simulación de algoritmos borrosos, un simulador de procesos (herramientas capaces de concebir y simular las funciones de pertenencia y las reglas de inferencia del sistema), una herramienta de síntesis de alto nivel especializada en capturar la especificación de algoritmos borrosos (clásicos o no) y convertirla en una descripción VHDL simulable y sintetizable del sistema lógico que lo materialice (ruta de datos y unidad de control), de modo que una vez comprobado que se cumple con las expectativas planteadas, se pueda materializar en

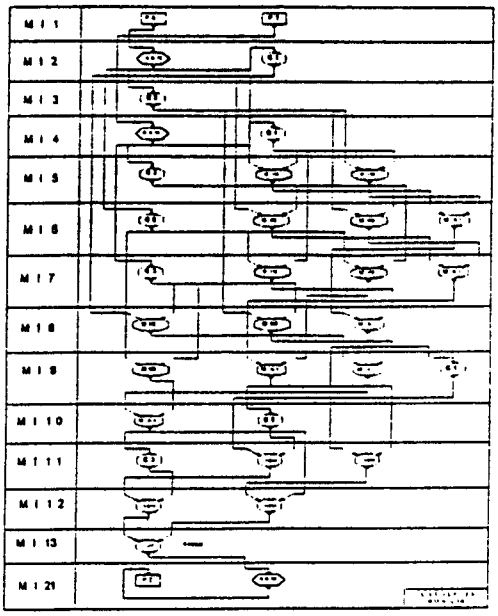


Figura 6: Planificación de las operaciones.

forma de *FPGA*, de *ASIC*, de *PCB*, etc.

7 Agradecimientos

A Román Hermida, Rafael Moreno y José Manuel Mendias (Dpto. de Informática y Automática UCM) por su ayuda en el aprendizaje de fidias y por su asesoramiento en algunas de las ideas expuestas.

Referencias

[1] Togai, M., Watanabe, H., *Expert System on a Chip: An Engine for Real-Time Approximate Reasoning*, IEEE EXPERT, vol. 1, nº 3, pp. 55-62, August 1986.

[2] H. Watanabe, H., Dettlof, W.D., Yount, E., *A VLSI Fuzzy Logic Controller with Reconfigurable, Cascadable Architecture*, IEEE Journal on Solid-State Circuits, vol. 25, nº 2, April 1990.

[3] Ungering, A.P., Thuener, K., Goser, K., *architecture of a PDM Fuzzy Logic Controller with Pipelining and Optimized Chip Area*, Proceedings of IEEE International Conference on Fuzzy Systems, pp. 447-452, 1993.

[4] 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.

[5] Moreno, R., Hermida, R., Fernández, M., Mendías, J.M., Septien, J., *Global Analysis for Integrated Scheduling and Module Allocation*, Proceedings of APCHDL'94, pp 103-109, 1994.

Sobre las Prestaciones de los Procesadores de Información Borrosa

Julio Gutiérrez, Luis de Salvador
Depto. de Tecnología Fotónica de la Universidad Politécnica de Madrid y
Grupo de Trabajo de Hardware y Control Avanzado del Instituto Nacional de Técnica Aeroespacial

Resumen

Se presenta un análisis de las prestaciones que un procesador borroso de propósito general debería incluir con el fin de cubrir las necesidades de proceso de información borrosa. Para ello se estudia la naturaleza de la información y las características de los conjuntos básicos de etiquetas lingüísticas, para posteriormente obtener conclusiones acerca de la implementación de distintos tipos de operaciones y procesos que han de ser factibles. Se propone el uso de modificadores como solución a muchos problemas de expresión y caracterización de la información borrosa, concluyéndose en determinadas pautas para el diseño de arquitecturas de procesador.

Introducción

La naturaleza de la información borrosa plantea diferencias sustanciales en cuanto a los medios de proceso de la información, de forma que los procesadores convencionales actuales, aunque capaces de llevar a cabo este tipo de procesos como procesadores universales que son, no dan un rendimiento adecuado en relación con los recursos que emplean.

A medida que se han ido desarrollando sistemas basados en lógica borrosa han ido surgiendo distintos medios para llevar a cabo su proceso. Sin embargo, todavía no se puede hablar de procesadores borrosos universales, sino de sistemas que cubren de alguna manera un determinado tipo de proceso. Pero, realmente, tampoco se puede decir que esto sea consecuencia de la problemática de la actual tecnología del hardware por sus dificultades de adaptación a la información borrosa. En efecto, aunque ésta es una faceta que no se puede dejar a un lado, puesto que obliga a un replanteamiento de la tecnología desde el nivel físico al nivel de arquitectura, existen otras razones importantes que son de índole teórica.

Por poner un ejemplo, en la lógica borrosa se han definido muchos conjuntos distintos de operadores para la unión, la intersección y la complementación. Igualmente están bién definidos los axiomas que deben respetar estos conjuntos de operadores, así como las funciones de agregación y se estudian otras propiedades de cada uno de estos conjuntos de operadores. Sin embargo, rara vez se especifican las aplicaciones de estos conjuntos de operadores a la realización de sistemas borrosos. Así pues, dado que el máximo y el mínimo son los únicos idempotentes y, además, cumplen la ley de De Morgan con la complementación según $1 - \mu(x)$, resulta que casi siempre se adoptan estos operadores para la realización de sistemas, cuando en la práctica una elección de este tipo puede acarrear problemas.

Pero lo que se observa en una buena generalidad de casos es algo curioso: los diseñadores de aplicaciones eligen el proceso max-min porque creen que lo avala la teoría, mientras que los teóricos lo dan por bueno porque piensan que lo requieren las aplicaciones.

Puesto que no se dispone de conclusiones a nivel teórico sobre el particular, el objetivo del presente artículo es someter a discusión las prestaciones que deberían contemplar los sistemas de proceso de información borrosa sin entrar en principio en la incidencia que ésto tendría en el hardware, para más adelante exponer las soluciones a nivel de procesador que se pueden dar al respecto.

Características de la Información Borrosa a procesar y su Representación

Los conjuntos borrosos son en esencia una función uniforme sobre el Universo de discurso. En consecuencia, se trata de información de naturaleza no numérica, aunque, al igual que sucede con las señales analógicas, puedan representarse mediante un conjunto de muestras numéricas.

La primera pregunta que cabe hacerse al respecto es si son admisibles o no ciertas restricciones en la forma y características de los conjuntos borrosos que han de manejarse. Dichas restricciones se han utilizado mucho y de muy diversas formas en la concepción de procesadores borrosos, tanto a nivel hardware como software [Zim 93], debido fundamentalmente al elevado volumen de información necesario para representar un conjunto borroso cualquiera. Por ejemplo, en muchas ocasiones se establece la restricción de que los conjuntos tengan forma trapezoidal o incluso triangular, otras veces se obliga a que las funciones de pertenencia sean

simétricas. En otras ocasiones se restringe el grado de incertidumbre de los flancos de los conjuntos y en ciertas aplicaciones se toma como función genérica de los conjuntos borrosos la campana de Gauss u otras... Nuestra conclusión es que tales restricciones no son nunca deseables puesto que coartan la capacidad de expresión. Dentro de ello, no obstante, la representación mediante trapecios o tramos rectos [Rui 94] resulta admisible en la gran mayoría de los casos puesto que consigue aproximarse razonablemente a una forma cualquiera, mientras que la representación de dicha información se resume en una serie de números representando los vértices. La simetría por lo contrario no es aplicable más que a una parte de los casos, dado que no es algo ni mucho menos natural en los conjuntos borrosos. Pongase por caso la velocidad de un coche: nunca podrá considerarse el mismo concepto de *velocidad alta* para la marcha adelante que para la marcha atrás.

Ahora bien, sí hay una restricción que pertenece a la propia naturaleza de los conjuntos borrosos y, por tanto, es perfectamente factible: es lo que podríamos llamar la *frecuencia* de la función (frecuencia en el sentido de Fourier). En efecto, una función de *alta frecuencia* tendría muchas variaciones seguidas alternativamente ascendentes y descendentes en tramos cortos del Universo de Discurso, lo cual destruye cualquier posibilidad de darle contenido conceptual a dicho conjunto o, lo que es lo mismo, su contenido conceptual sería el mismo tras filtrar las altas frecuencias. La consecuencia que se obtiene de ello es que la representación mediante muestreo con una frecuencia de muestreo moderada es suficiente, lo mismo que sucede con las señales analógicas al aplicarle el criterio de Nyquist. Solamente aparecerán altas frecuencias con significado real cuando se opere con conjuntos nítidos o con un grado de incertidumbre muy pequeño, de forma que los flancos de dichos conjuntos serían muy abruptos. Continuando con el símil de las funciones de pertenencia con las señales analógicas, la presencia de conjuntos nítidos sería un caso análogo a la presencia de señales digitales.

Un aspecto importante es la distinción que existe entre la representación teórica de un conjunto borroso y el manejo de los mismos en casos prácticos. Teóricamente, un conjunto borroso se encuentra definido sobre un universo de discurso continuo, y representa la variación, tambien continua, de los valores de pertenencia a ese conjunto de cada uno de los elementos del universo del discurso. Esto quiere decir que entre dos valores del universo de discurso siempre existe otro valor distinto, y que entre dos valores de pertenencia distintos siempre se encuentra otro valor de pertenencia distinto de ellos.

Por otro lado, en cualquier sistema de control de una aplicación industrial, los valores de las variables de entrada siempre se encuentran afectadas por ruido (sobre todo si el universo de discurso se considera a priori discreto). Este ruido es introducido por el sistema de adquisición de datos. Así mismo, la salida del sistema se encuentra afectada por un ruido, suma del anterior, de la operación del sistema y de los sistemas transductores. Si este ruido se encuentra por debajo de los niveles considerados aceptables por la aplicación, el sistema funciona correctamente. Es más, un sistema más preciso será desechado si resulta económicamente más caro. Por ejemplo: construir un aparato que emita 100 imágenes por segundo es inútil para una persona cuyo cerebro no asimila más de 25 imágenes.

Por lo tanto, es factible sustituir el universo de discurso continuo por un muestreo del mismo siempre que el ruido introducido se encuentre por debajo de los niveles aceptables por la aplicación. Así mismo, es posible cuantizar los valores de pertenencia de un conjunto difuso siempre que el ruido introducido por la cuantización se encuentre por debajo de los límites de ruido aceptables en la aplicación. Es importante notar que cuantizar y muestrear no significa utilizar lógica multivaluada, sino que simplemente supone la aplicación práctica y dentro de los límites de coste aceptables de la lógica difusa.

Se puede establece un ejemplo paralelo con la transformada de Fourier, en la que se basa la teoría de proceso de señal. Actualmente ningún sistema emplea transformadas de Fourier, sino que implementan series de Fourier. Esto no implica que toda la teoría desarrollada con la primera se utilice en aplicaciones prácticas realmente implementadas en la segunda.

Otra pregunta importante es cuál es el medio idóneo de representación. Aunque éste es muy dependiente de la arquitectura, sí se puede decir en términos generales que los medios de representación basados en la expresión de la función o en la codificación de la misma consiguen ventajas en cuanto a ocupación de memoria respecto al muestreo, pero están en clara desventaja en cuanto a la capacidad operativa, ya que con las muestras se puede operar directamente mientras que en los otros casos es necesario primero extraer los valores y luego operar. Si se tiene en cuenta que el uso de expresiones o de codificación tienden a restringir la forma de los conjuntos borrosos, parece que el muestreo reúne las mejores posibilidades.

Etiquetas Lingüísticas

Hemos preferido tratar el tema de las etiquetas lingüísticas aparte de los medios de representación de la información puesto que consideramos que el uso de etiquetas linguísticas es más un aspecto semántico que una técnica de representación de conjuntos borrosos. Dicho de otra forma, las etiquetas lingüísticas permiten la utilización de conceptos bajo su denominación en lenguaje natural, lo cual es esencialmente distinto de la codificación de conjuntos para su representación interna en una máquina.

Sin embargo, es importante señalar que el conjunto de etiquetas lingüísticas debe ser libre en el sentido de la forma que deben adoptar los conjuntos representados por etiquetas y su distribución a lo largo del universo de

discurso. Pero, por otra parte, también es importante preservar el objetivo ya mencionado del conjunto de etiquetas: proporcionarnos el modelo de los conceptos primitivos. Dado que cada variable debe tener su conjunto de etiquetas propio, lo que sí se puede considerar fuera de lugar en un conjunto de etiquetas es que éstas se solapen en exceso, ya que ello daría lugar a que un mismo elemento perteneciera a muchos conceptos, lo que redundaría en un grado alto de confusión [Klir, 88]. Nosotros hemos considerado que no deberían existir en un mismo conjunto más de tres etiquetas no disjuntas entre sí.

Conviene recalcar, no obstante, que lo dicho respecto al conjunto de etiquetas se refiere al conjunto de *primitivas* no a la totalidad de conceptos aplicables a una determinada variable, cosa que dejaremos resuelta mediante el uso de *modificadores*, como se verá má adelante.

Velocidad de proceso

Dentro de las prestaciones con las que se debe dotar a un procesador borroso se encuentra la velocidad de proceso. Resulta obvio que el análisis de prestaciones debe orientarse a aplicaciones complejas pues, de lo contrario, no haría falta perseguir el diseño de procesadores, dado que bastaría con un computador convencional sin que los defectos de rendimiento pudieran constituir un problema de consideración.

Así pues, no cabe duda de que la velocidad de proceso es algo que hay que elevar todo lo posible siempre y cuando no se sacrifique la aplicabilidad genérica. Igual que sucede con los procesadores crisp, la velocidad es un problema compartido entre la tecnología de base y la arquitectura del procesador. Los avances tecnológicos benefician paralelamente a todo tipo de procesador, sea o no borroso, de forma que el problema que se debe abordar es fundamentalmente estructural.

Al decir esto hemos admitido implícitamente que nos referimos a los procesadores digitales. Por tanto, es necesario hacer la salvedad de que el proceso de infomación borrosa es perfectamente factible a nivel analógico [Rui 94], cosa que no se puede decir de los procesadores crisp. Además, creemos que merece la pena la investigación relativa a los procesadores analógicos, pues no se ha explorado aún en toda su amplitud y profundidad. Pero también se puede afirmar que hoy por hoy las ventajas y grado de desarrollo de la tecnología digital predisponen a esta opción para el desarrollo de procesadores borrosos a medio plazo.

Hay que señalar además, que una vía muy potente para el desarrollo de aplicaciones es la de desarrollar procesadores específicos. Igual que sucede con los sistemas convencionales, muchas aplicaciones de tiempo real crítico requieren el diseño de una máquina dedicada con diseño específico para el caso, so pena de tener que emplear una máquina de propósito general de desproporcionadas prestaciones, cosa que, además, sólo sería posible en según qué casos.

Pero la idea y las arquitecturas de los procesadores no destruyen en absoluto esta posibilidad sino todo lo contrario: una arquitectura de propósito general debidamnete dimensionada y configurada puede perfectamente ser la solución idónea para un sistema dedicado, sobre todo cuando el marco de aplicación concuerda. Por consiguiente, es importante distinguir entre lo que es una arquitectura de propósito general y un procesador de propósito general. Lo que realmente merece la pena es lo primero puesto que permite orientarse tanto hacia la aplicación concreta como al desarrollo en general de aplicaciones.

Implementación de T-Normas y T-Conormas

Ya en la Introducción habíamos mencionado el problema de las T-Normas y T-Conormas y la extendida tendencia a acogerse al modo max-min.

Es evidente que la propiedad de idempotencia y la ley de De Morgan proporcionan ventajas al uso de max-min, pero en ningún caso se podría demostrar que esto es determinante para a su adopción. No hay que olvidar que, por lo contrario, estas operaciones no cumplen con principios que podrían considerarse importantes, como son los de *No Contradicción* y del *Tercero Excluido*. Es más, estos principios son incompatibles con la idempotencia y la distributividad, de forma que siempre se violan algunas leyes del Álgebra de Boole. Por otra parte, no hay que olvidar que *min* es la mayor de todas las intersecciones posibles, mientras que *max* es la menor de las uniones, dentro de que ambas han de cumplir los axiomar básicos respecto a las condiciones de contorno, continuidad, monotonicidad y asociatividad. Por tanto, se puede considerar como una solución extrema.

Como ejemplo de las consecuencias que esto tiene en el significado de las deducciones lógicas, considérese un conjunto de etiquetas lingüísticas como el representado en la figura 1.a).

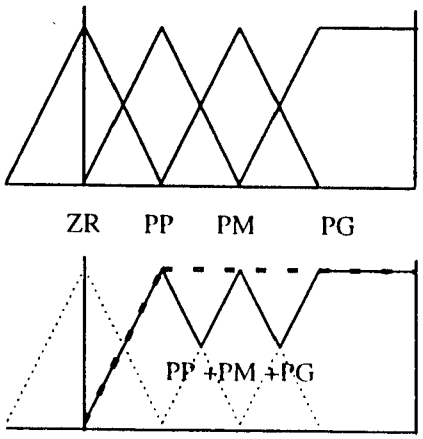


Fig. 1: Unión de PP, PM y PG según Normas

Supóngase que se requiere la obtención del conjunto de los *valores positivos* mediante la unión de los tres conjuntos en uso dentro del conjunto de etiquetas que contienen números positivos, es decir, el *Positivo Pequeño* (PP), el *Positivo Medio* (PM) y el *Positivo Grande* (PG). Si se utiliza max-min, el resultado sería el conjunto representado mediante trazo contínuo en la figura 1.b). Es evidente que ésta no es la forma esperada para el conjunto de *los valores positivos*, puesto que existen tramos en los que la *positividad* decrece cuando el valor positivo aumenta

Sin embargo, si esta misma operación de unión se lleva a cabo mediante:

$$V(a,b) = \min(1, a+b)$$

$$\Lambda(a,b) = \max(0, a+b-1)$$

$$\neg a = 1-a$$

el resultado es el que se representa mediante línea gruesa de puntos en la figura 1.b), la cual se ajusta mejor sin duda al concepto que se pretendía obtener. No obstante, se podría dar la circustancia de que estos operadores tampoco fueran del todo válidos si el conjunto de etiquetas fuera más disperso y en la intersección entre dos etiquetas la suma de ambos valores de pertenencia fuera inferior a la unidad. Esto, sin embargo, podría ser al mismo tiempo un indicio de falta de buen planteamiento en el conjunto de etiquetas, en lugar de ser un problema del operador.

La conclusión que se obtiene de ello es que un procesador borroso no debe encontrarse limitado a un solo modo en el tratamiento de la unión y la intersección, aunque ciertamente no merece la pena que el procesador sea capaz de operar según todos los modos posibles. En consecuencia, parece razonable que una de las formas de operar sea según el max-min, por cumplir la idempotencia y otra, por el contrario, que cumpla con los principios del tercero excluido y no contradicción, como es la señalada más arriba. No obstante, según se verá en el apartado siguiente, el uso de modificadores puede llegar a obviar esta circunstancia.

Uso de Modificadores

Aunque en la literatura especializada no resulta habitual el uso de modificadores en la inferencia borrosa, consideramos que este es un punto esencial, pues de otro modo el lenguaje que nos permite expresar reglas queda excesivamente limitado. Además es el procedimiento más potente que hemos encontrado para simplificar el conjunto de reglas.

En la sección dedicada a las Etiquetas Lingüísticas, enunciamos que dichas etiquetas no son otra cosa más que *Primitivas*. Pero eso no quiere decir que sean los únicos conceptos que se pueden emplear, tal como sucede en el lenguaje natural. En efecto, si se dispone de una primitiva cuyo significado es *Grande*, el uso de modificadores nos permitiría hablar de *muy grande*, de *más bien grande*, de *moderadamente grande*, de *bastante grande*, etc..., de forma que sea posible emplear un número muy superior de conjuntos borrosos, todos ellos dependientes del conjunto de etiquetas y función del modificador concreto.

Ahora bien, contemplar cualquier tipo de modificador podría derivar en excesiva redundancia y complejidad en los procesadores, entre otras cosas porque el significado de muchos modificadores de uso en lenguaje natural es borroso y no muy bien definido. Por ejemplo los modificadores *más bien grande* y *moderadamente grande* vienen a significar lo mismo, aunque tendrían diferentes interpretaciones según personas.

Analizando este problema, nos encontramos con una controversia similar a la mantenida entre los procesadores clásicos CISC y RISC: por una parte se puede ahorrar instrucciones incorporando otras más complejas, pero por otro se aumenta considerablemente la complejidad de los procesadores. Ahora bien, con todo ello, es fundamental que el conjunto de instrucciones con mayor o menor granularidad sea capaz de cubrir el Universo de posibilidades. Con esta misma filosofía, hemos definido un conjunto de lo que denominaremos *Modificadores Esenciales*, con el fin de cubrir todas las necesidades de expresión y aprovechar convenientemente las ventajas del uso de modificadores sin incrementar peligrosamente la complejidad necesaria en el hardware.

Atendiendo a la simplificación de reglas, véase el mapa de reglas de la figura 2.

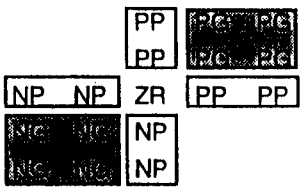


Fig.2: Mapa de Reglas con reglas adyacentes agrupables según el consecuente

Este mapa tiene una configuración muy frecuente, especialmente en lo que respecta a la coincidencia de un mismo consecuente en posiciones adyacentes, circunstancia que se ha señalado en la figura mediante rectángulos de diferente sombreado. En consecuencia, si los modificadores permitiesen agrupar los antecedentes de cada rectángulo, cada uno de ellos constituiría una sola regla, sustituyendo a todas las reglas que engloba. Pero en lo que respecta a la capacidad expresiva, estos mismos modificadores cumplen un importante cometido, pues realmente no significa lo mismo la composición de las reglas englobadas que la regla única que lo sustituye.

En efecto, los modificadores que consideraremos como *esenciales* son los siguientes:

Más que:

$$\mu_{\text{más que } A}(x) = \begin{cases} 0 \text{ para } x < x_0 \\ 1 - \mu_A(x) \text{ para } x \geq x_0 \end{cases}$$

Menos que:

$$\mu_{\text{menos que } A}(x) = \begin{cases} 0 \text{ para } x > x_0 \\ 1 - \mu_A(x) \text{ para } x \leq x_0 \end{cases}$$

donde x_0 es el punto máximo o un punto cualquiera de la meseta del conjunto.

Así como la combinación de ambos, *más que A y Menos que B*. De la misma forma, aunque en este caso no se trate de un modificador propiamente dicho, los antecedentes de las reglas deben admitir la negación. Por ejemplo:

si x es A e y no es B, entonces ...

cosa que es muy importante en la línea de lo expuesto, y que sin embargo no se contempla en los procesadores actuales, a no ser que se incluyan las negaciones dentro del conjunto de etiquetas, con lo que se pierde el sentido real de las etiquetas lingüísticas.

Puede comprobarse fácilmente que el uso de modificadores elimina los inconvenientes de utilizar max-min en el sentido expuesto en la figura 1. En efecto, si se aplica el modificador *más que Cero* (ZR) para obtener el conjunto de los valores positivos, se obtiene el conjunto en línea de trazos gruesos de la fig.1.b).

Proceso de información Borrosa

Muchas de las necesidades de proceso en lógica borrosa se orientan hacia la ejecución de inferencias en una base de reglas. En este caso, el proceso necesario se divide en tres etapas: fuzzificación de las variables externas, realización de la inferencia borrosa resolviendo cada una de las reglas definidas en el sistema borroso, y finalmente la defuzzificación de los valores inferidos en valores aplicables al universo exterior.

Tanto la primera como la última etapa podrían quedar eliminadas, bien por que así lo exige la aplicación o bien por que se solape con el proceso de evaluación de la inferencia. En cualquier caso, es necesaria la existencia de una base de reglas que sea evaluada de forma sistemática cada vez que cambien las condiciones del conjunto de variables antecedentes.

Las aplicaciones que precisan este tipo de proceso son diversas: aplicaciones de control [Ped 92], sistemas expertos, bases de datos borrosas, etc. Existe otros tipos de aplicaciones: como las redes neuronales borrosas, filtros o los sistemas expertos con evaluación hacia delante y hacia atrás. Pero, en cualquier caso, estas están compuestas por un conjunto de unidades de inferencia, de dos hasta infinito, que trabajan de forma interconectada [Kos 88], [Kan 92].

La evaluación de inferencias puede tener distintos grados de complejidad: pueden incluirse combinaciones de variables borrosas y no borrosas, pueden incluirse variables comunes a la parte antecedente y consecuente, o se pueden realizar interacciones con sistemas no borrosos, como controles PID.

Aunque la evaluación de inferencias es utilizada en gran número de aplicaciones que utilizan la lógica difusa, no es la única operación que se emplea cuando se utilizan técnicas borrosas de manipulación de la información [Kli 88]. Existen aplicaciones que en vez de emplear una base de reglas utilizan medidas borrosas y medidas de borrosidad.

La utilización de medidas borrosas y medidas de borrosidad requiere unas capacidades de proceso completamente diferentes de aquellas aplicaciones que realizan la inferencia de una base de reglas. Si bien estas últimas no se adecuan con perfección a los procesadores convencionales, las aplicaciones que utilizan cualquier tipo de medida borrosa sí se adecuan mejor a estos.

En consecuencia, los procesos de inferencia son los que verdaderamente requieren procesadores específicos si se pretende llegar a un proceso eficiente de la información borrosa.

Arquitecturas de Procesador Borroso

Existen muchos tipos de arquitecturas de procesador borroso, bien implementadas, bien tan solo definidas sobre el papel. Sin embargo, la definición de arquitecturas óptimas es un tema que todavía no se encuentra cerrado, sino al contrario, se encuentra en plena evolución [Man 87] [Yam 88] [Man 90] [Man 90] [Bug 92] [Fer 92] [Zim 93] [Ung 93] [Gut 93] [Sas 93][Rui 94] [Til 94] [Cat 94].

De todo lo expuesto en los apartados anteriores se deduce un criterio de desarrollo que se considera muy importante. Antes de buscar soluciones globales y absolutas, es necesario definir una arquitectura que cumpla las siguientes características: óptima, flexible, adaptable a las necesidades reales del usuario y configurable siguiendo criterios económicos e intereses industriales.

Este objetivo se puede alcanzar en dos fases: la primera mediante la definición de la arquitectura del ordenador borroso; la segunda mediante la definición de la arquitectura del procesador borroso.

La arquitectura del ordenador borroso ha de ser, siguiendo los criterios anteriormente señalados, un sistema compuesto por una arquitectura convencional al que se le ha añadido un coprocesador borroso capaz de ejecutar inferencias de forma altamente eficiente [Gut 93].

El procesador borroso ha de ser un sistema capaz de ejecutar inferencias, tanto en forma de coprocesador dentro de una arquitectura convencional, como en configuración *stand-alone* para aplicaciones embarcadas, ocupando el puesto de los microcontroladores industriales. Pero lo más importante, es que saque partido de las ventajas de modelización de la lógica borrosa realizando la inferencia a gran velocidad.

Todas las aplicaciones no van a tener la misma complejidad ya que cada una de ellas va a tener unas necesidades concretas en número de antecedentes y consecuentes, precisión, muestreo o número de reglas o forma de los conjuntos. Por lo tanto, es preciso que la arquitectura del procesador borroso sea lo suficientemente flexible para poder desarrollar una familia de micro-chips capaces de satisfacer las necesidades de una gran variedad de usuarios [Sal 95].

Esta es la filosofía con la que se han desarrollado las arquitecturas sistólicas multinivel [Sal 93] [Sal 94] [Sal 94b]. Este tipo de arquitecturas se encuentran orientadas al proceso masivo de inferencias borrosas y han sido desarrolladas empleando técnicas de descripción HDL que les permiten una gran flexibilidad y adaptabilidad a las necesidades de cada usuario concreto.

Bibliografía

[Man 87] M. A. Manzoul, "Fuzzy Inference on a Systolic Array", *Proceedings of 18th Modelling and Simulation Conference*, 1987, pp. 1103-1108.

[Kli 88] Klir, T.A. Folger, *Fuzzy sets, uncertainty and information*, Prentice-Hall, 1988

[Kos 88] B. Kosko, *Neural Networks and Fuzzy Systems*, Prentice-Hall, 1988

[Yam 88] T. Yamakawa, "High Speed Fuzzy Controller Hardware System: The Mega-FLIPS Machine", *Information Sciences 45*, pp 113-128, Elsevier Science Publishing, 1988

[Man 90] M.A. Manzoul, S. Tayal, "Systolic VLSI Array for Multi-Variable Fuzzy Control System", *Cybernetics and Systems: An International Journal* 21:27-42, 1990

[Bug 92] A. Bugarín, S. Barro, R. Ruiz, "Soluciones Sistólicas en Control Borroso", *Proc. II Spanish Association on Technology and Fuzzy Logic Congress* 1992

[Fer 92] F. Fernandez, A. Ruiz, J. Gutiérrez, "Diseño Sistemático de un Procesador Sistólico de Inferencias Difusas", *Proc. II Spanish Association on Technology and Fuzzy Logic Congress* 1992

[Gut 92] J. Gutierrez, "Hardware Fuzzy", *Aplicaciones de la Lógica Borrosa - Nuevas tendencias - CSIC*, 1992

[Kan 92] Fuzzy Expert Systems, Editor A. Kandel, CRC Press, 1992

[Man 92] M.A. Manzoul, D. Jayacharathi, "Fuzzy Controller on FPGA chip", *Proc. 1st IEEE Conf. on Fuzzy Systems*, 1309-1316, 1992

[Ped 92] W. Pedrycz, Fuzzy Control and Fuzzy Systems, Research Studies Press Ltd, 1992

[Gut 93] J. Gutierrez, "Procesadores de Información Borrosa", ARBOR CXLI, 573,-574, 1993

[Ung 93] A.P. Ungering, K. Thuener, K. Goser, "Architecture of a PDM VLSI Fuzzy Logic Controller with Pipelining and Optimized Chip Area", IEEE - 1993

[Zim 93] H.J. Zimmermann, *Fuzzy Technologien. Prinzipien, Werkzeuge, Potentiale*. VDI Verlag 1993

[Sal 93] L. de Salvador, M. García, J. Gutiérrez, "Implementaciones Digitales VLSI de Controladores Borrosos", Lecture in the Hw. and Advanced Ctrl. Laboratory Exhibit, 1993

[Sas 93] M. Sasaki, F. Ueno, T. Inoue, "7.5 MFlips Fuzzy Microprocessors using SIMD and Logic-in-Memory Structure", IEEE 1993

[Cat 94] V. Catania, A. Puliafito, M. Russo, L. Vita, "A VLSI Fuzzy Inference Processor Based on a Discrete Analog Approach" *IEEE Transactions on Fuzzy Systems*, vol. 2, n. 2, may 1994

[Rui 94] A. Ruiz, J. Gutierrez, *Hardware Básico de Motores de Inferencia Borrosa Utilizando Trapecios Codificados*, Ph.D. Thesis, Politechnic University of Madrid, 1994

[Sal 94] L. de Salvador, M. García, J. Gutiérrez, "Arquitecturas Sistólicas de Procesamiento Difuso y su Simulación", *Proc. IV Spanish Association on Technology and Fuzzy Logic Congress*, pg. 171- 176, 1994

[Sal 94b] L. de Salvador, J. Gutierrez, "High Perfomance Multilevel Systolic Fuzzy Inference Unit", *Proc. IEEE NAFIS/IFIS/NASA*-1994.

[Til 94] T.A.W. Tilli, "High-Speed Fuzzy Inference Methods with Low Memory Requeriments", ELITE-Foundation, EUFIT-1994

[Zha 94] J. Zhang, J. Raczkowsky, M. Webler, "VLSI-Design, Specification and Simulation of a Fuzzy Co-Processor", ELITE-Foundation, EUFIT-1994

[Sal 95] L. de Salvador, J. Gutiérrez, "Impact of the Migration from VHDL'87 to VHDL'92 on a Design", *Proc. User's Forum European Design and Test Conference*, 1995

SÍNTESIS AUTOMÁTICA DE SISTEMAS DE CONTROL BASADOS EN LÓGICA DIFUSA

C.J. Jiménez, D. Gálan, A. Barriga y S. Sánchez-Solano

Centro Nacional de Microelectrónica. Dpto. de Diseño Analógico.
Avda. Reina Mercedes s/n Edif. CICA. 41012-Sevilla
Tlf: (95) 4623811
Fax: (95) 4624506
e-mail: cjesus@cnm.us.es

Resumen

Se describe un entorno de diseño para la realización *hardware* de sistemas de control basados en lógica difusa. Dicho entorno combina una serie de herramientas de CAD que facilitan la definición de las especificaciones del controlador, la descripción de su arquitectura y la síntesis automática del circuito. Entre sus principales características cabe destacar la utilización de un lenguaje de descripción de *hardware* estándar (VHDL) y su compatibilidad con otras herramientas de diseño de circuitos integrados.

1. Introducción.

La realización física de sistemas electrónicos requiere la utilización de herramientas de CAD que facilitan los diferentes pasos del proceso de diseño. Las características y objetivos de dichas herramientas varían en función de las diferentes actividades de diseño (descripción, síntesis, verificación) y del tipo de sistema a desarrollar. En el caso de los sistemas de control basados en lógica difusa aparecen algunas particularidades. Por un lado, suelen contemplarse dos niveles de diseño claramente diferenciados: el nivel algorítmico y el nivel de circuitos. Por otra parte, las simulaciones correspondientes a ambos niveles se realizan con herramientas distintas. En las simulaciones a nivel algorítmico se utilizan entornos específicos de simulación fuzzy, mientras que en las simulaciones a nivel de circuito se emplean entornos de diseño convencionales.

En esta comunicación se describe un conjunto de herramientas de CAD enfocadas a la realización *hardware* de controladores difusos. La combinación de estas herramientas constituye un entorno de diseño, verificación y síntesis automática para este tipo de sistemas, facilitando su desarrollo a nivel algorítmico (elección de los mecanismos de inferencia y construcción de la base de datos de conocimiento), arquitectural (selección de las estructuras de circuito que constituyen el controlador) y de implementación (obtención del circuito integrado).

2. Descripción del entorno.

El esquema general del entorno de diseño se ilustra en la Figura 1. La descripción algorítmica del controlador puede introducirse textualmente, utilizando un lenguaje

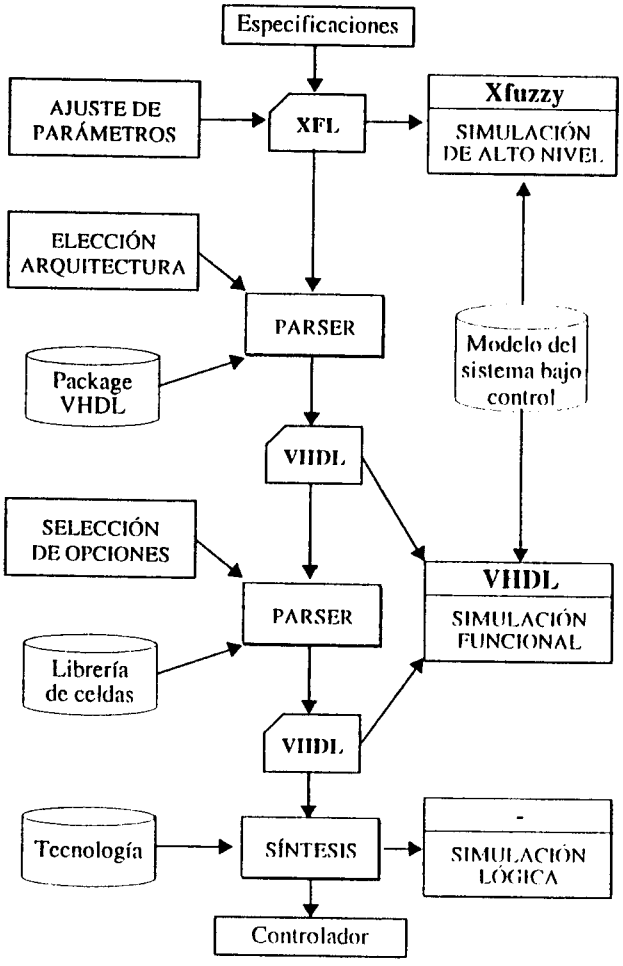


Fig. 1: Esquema general del entorno.

formal de descripción al que denominamos "*Xfuzzy Description Language*" (XFL), o bien a través de las facilidades gráficas proporcionadas por Xfuzzy [1]. Las actividades de diseño asociadas a esta etapa de desarrollo del controlador tienen como objetivo elegir el mecanismo de inferencia a utilizar y definir y ajustar la base de conocimientos (reglas, funciones de pertenencia de antecedentes y consecuentes, etc.).

Una vez concluida esta etapa de diseño, la descripción XFL del controlador es traducida a otro lenguaje de descripción más próximo a la realización *hardware*. En este caso hemos optado por utilizar VHDL, ya que se trata de un lenguaje estándar que, además, proporciona un mecanismo de entrada unificado hacia numerosas herramientas de verificación y síntesis de circuitos integrados de distintos fabricantes. El proceso de traslación genera la descripción de la estructura de circuito del controlador de acuerdo con una arquitectura preestablecida desarrollada previamente por los autores [2] [3].

Una vez fijadas las diversas alternativas arquitecturales, la descripción VHDL de alto nivel es trasladada a una descripción VHDL de bajo nivel que utiliza los bloques parametrizados contenidos en una librería de componentes. La nueva descripción es sintetizable mediante diferentes herramientas de síntesis de circuitos integrados. Dependiendo de la aplicación a que se destine el controlador, éste puede fabricarse siguiendo diferentes técnicas de implementación de circuitos integrados (ASICs, FPGAs, etc.).

Como puede apreciarse en la Figura 1, los diferentes niveles de descripción incluyen etapas de simulación que permiten verificar la corrección del proceso de diseño. Con objeto de facilitar la simulación del conjunto formado por el controlador y el sistema bajo control, en las simulaciones a nivel algorítmico el modelo del sistema bajo control es descrito mediante un programa en lenguaje C. A nivel arquitectural se utiliza el propio lenguaje VHDL, siguiendo el esquema de "*test-bench*" habitual en este tipo de herramientas.

3. Nivel Algorítmico.

La Figura 2 ilustra de una forma simbólica los componentes del entorno de simulación para el nivel algorítmico. El núcleo de la herramienta Xfuzzy consiste en la implementación software de un motor de inferencia difuso. Los módulos que definen el comportamiento del controlador (conjunto de reglas y funciones de pertenencia) y el sistema bajo control (modelo) interactúan con este motor de inferencia. El entorno se completa con una interfaz de usuario que, a través de un grupo de menús, facilita las tareas de edición de las funciones de pertenencia, las reglas, y el modelo del sistema, así como la representación gráfica de los resultados de simulación. La aplicación ha sido

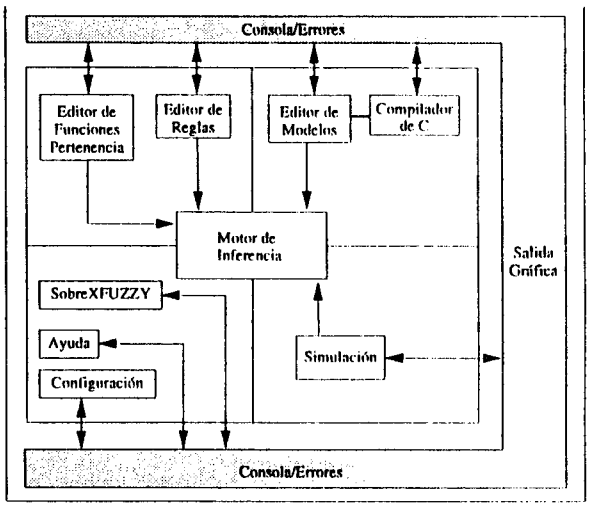


Fig. 2: Componentes de Xfuzzy.

desarrollada en lenguaje C sobre entorno de ventanas *X-Windows*.

El ciclo típico de utilización de Xfuzzy se muestra en la Figura 3. Dicho ciclo comienza con la elección, por parte del usuario, de los distintos componentes que definen el mecanismo de inferencia a utilizar. Nos referimos a cuestiones tales como los conectivos de unión de antecedentes y de agregación de reglas, la función de implicación y el método de defuzzificación utilizado. A continuación se definen las funciones de pertenencia asociadas con cada uno de los antecedentes y consecuentes y la base de reglas que gobierna el comportamiento del sistema. En la etapa de simulación, para cerrar el lazo de realimentación entre el controlador y el sistema bajo control, se debe determinar la relación entre las salidas y las entradas del controlador. Esta relación se define mediante un programa escrito por el usuario en el lenguaje de programación C de acuerdo con las directivas establecidas por un patrón que genera el entorno. El hecho de que sea un programa C lo que interactúa con el entorno da al sistema mayor flexibilidad, ya que permite generar información de salida que puede ser usada por las facilidades gráficas del entorno o bien por otras utilidades (ajuste de reglas, ajuste de factores de escala, etc.). Una vez se ha escrito el modelo, éste se compila. A continuación el *linker* es invocado con una opción específica que permite cargar el modelo en tiempo de simulación. Esta técnica permite al usuario modificar el modelo interactivamente con la ejecución del entorno facilitando las tareas de depuración y realimentación en el proceso de diseño.

A partir de este punto, y en función del tipo de estudio que se pretenda realizar, el usuario puede iterar a lo largo de las diferente etapas del ciclo de operación. Algunas de las trayectorias posibles incluirían volver al editor de reglas para ver la influencia de añadir, quitar o modificar al-

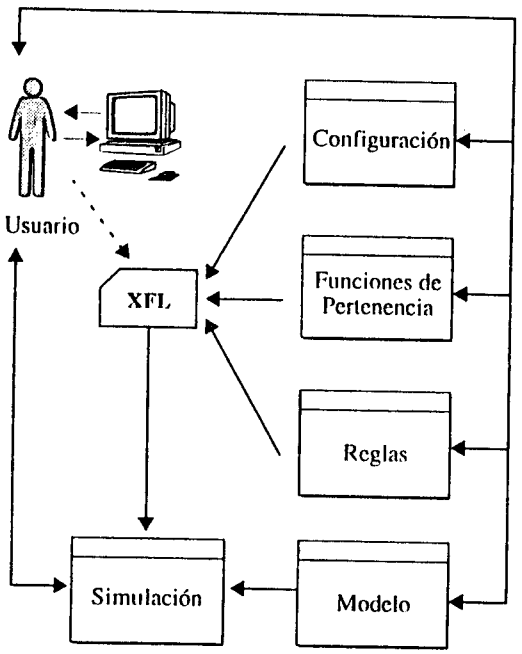


Fig. 3: Diseño a nivel algorítmico.

guna de las reglas de comportamiento, o volver al editor de funciones de pertenencia para evaluar los efectos producidos al variar la definición de alguno de los subconjunto difusos.

A lo largo del ciclo de diseño la descripción XFL del controlador es mantenida en una base de datos. Una vez que el usuario está satisfecho con la simulación a nivel algorítmico del sistema, dicha descripción es traducida a una descripción VHDL que actúa como entrada al nivel de arquitectura que describiremos a continuación.

4. Nivel de Arquitectura.

La realización eficiente en *hardware* de controladores difusos requiere la elección de una estructura de circuito que minimice el área de silicio consumida y maximice su velocidad de operación. El tipo de arquitecturas utilizadas en el entorno que estamos describiendo corresponde a las presentadas por los autores en [2] y [3]. Sus principales características son la limitación a priori del grado de solapamiento de las funciones de pertenencia de los antecedentes de las reglas y la utilización de métodos de defuzzificación simplificados. El uso de un procedimiento de inferencia gobernado por las reglas activas y la introducción de un adecuado esquema de temporización a nivel de circuito permiten obtener una arquitectura de complejidad reducida y excelente comportamiento dinámico. La Figura 4 muestra el diagrama de bloques de dicha arquitectura.

Por otra parte, algunos autores han investigado la posibilidad de usar VHDL como un lenguaje que soporta las estructuras de datos y las funciones requeridas para la simulación de sistemas difusos [4]. Además de las ventajas apuntadas en el apartado 2, la utilización de este lenguaje de descripción de *hardware* permite utilizar el mismo entorno de simulación en los niveles arquitectural y de implementación, por lo que los resultados obtenidos en ambos niveles pueden ser fácilmente comparados.

Para realizar la descripción del controlador hemos desarrollado un *package* VHDL que contiene las definiciones de las estructuras de datos que almacenan la información fuzzy y el conjunto de funciones que facilitan la aplicación de los diferentes algoritmos de inferencia [5]. Este *package* contiene una serie de tipos básicos que serán utilizados para almacenar los valores de las variables de entrada y salida, las descripciones de las funciones de pertenencia y los grados de activación de las diferentes reglas. Utilizando estos tipos básicos, se incluye otra serie de definiciones de tipos más complejos que permiten agrupar los datos correspondientes a las diferentes variables lingüísticas que aparecen en las reglas.

Haciendo uso de las definiciones de esos tipos de datos se han implementado un conjunto de funciones para ma-

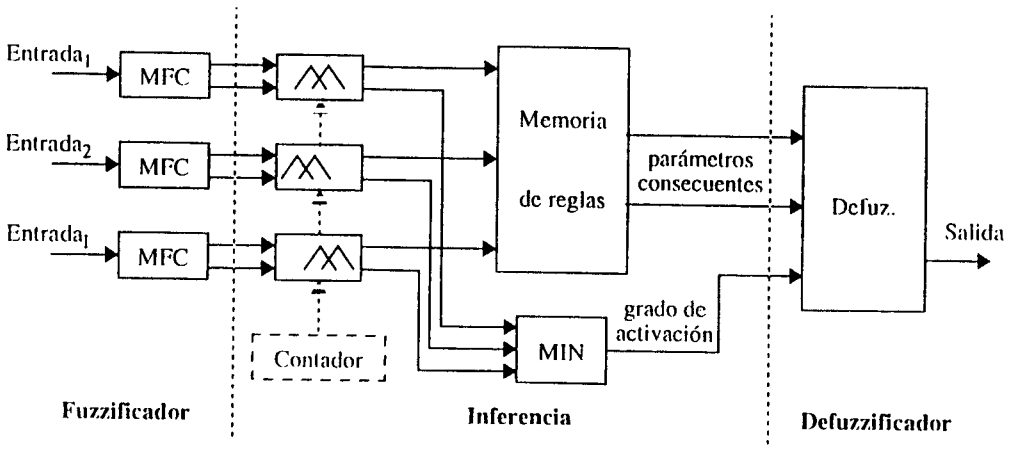


Fig. 4: Diagrama de bloques de la arquitectura del controlador difuso.

nejarlos. Entre las más significativas, la función "fzis" realiza la fuzzificación, es decir, devuelve el grado de pertenencia de la variable de entrada a una etiqueta lingüística. "fzand" y "fzor" son los conectivos de las diferentes reglas. "fzthen" realiza la función de implicación fuzzy y la agregación de las conclusiones de las diferentes reglas. Finalmente, la función "defuzzifier" proporciona el valor representativo de cada variable de salida de acuerdo con el método de defuzzificación seleccionado [6] [7]. Las estrategias de defuzzificación consideradas en el *package* corresponden tanto a métodos simplificados (Fuzzy Mean, Weighty Fuzzy Mean, etc.) como a métodos convencionales (Center of Area, Mean of Maximum, etc.). Estos últimos sólo con propósito de comparación.

El diseño a nivel arquitectural tiene como objetivo verificar que las especificaciones introducidas a nivel algorítmico pueden obtenerse con los recursos de *hardware* disponibles. Para ello, el usuario puede actuar sobre los diferentes parámetros de diseño y simular el comportamiento funcional del controlador utilizando cualquier herramienta de simulación VHDL. Posteriormente, la descripción funcional del controlador debe ser convertida en una descripción sintetizable como paso previo a su implementación en un circuito integrado.

5. Nivel de Implementación.

En el proceso de translación de la descripción funcional del controlador a una descripción sintetizable es posible elegir entre diferentes alternativas. En concreto, la estrategia utilizada para almacenar las funciones de pertenencia de los antecedentes (mediante elementos de memoria o circuitos aritméticos), la definición de la base de reglas (en memoria RAM/ROM o implementándola como un bloque lógico combinacional) y el método de defuzzificación utilizado (actualmente es posible elegir entre cuatro estructuras que permiten implementar los métodos descritos anteriormente, así como utilizar un defuzzificador programable que engloba a todos ellos).

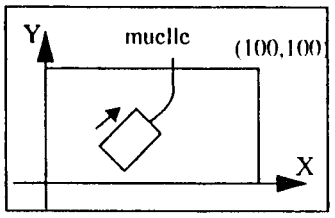
En el proceso de síntesis del controlador se utiliza una librería de celdas parametrizadas que contiene las descripciones VHDL de los bloques involucrados en las distintas opciones de arquitectura. Dichos bloques corresponden tanto a bloques de operación (que implementan el algoritmo de inferencia) como a bloques de control (encargados de realizar la carga de las memorias y de generar las señales de temporización). La descripción VHDL sintetizable se obtiene a partir de las celdas de la librería mediante un programa de traslación o *parser* que coloca e interconecta los elementos necesarios. Este programa produce como salida tres ficheros que contienen las definiciones de todas las constantes necesarias para los bloques parametrizados, la declaración de la entidad que coloca y conecta todos los bloques y una plantilla para la simulación del circuito.

La técnica de implementación que utilizaremos, así como la elección de las posibles opciones de diseño, van a depender, fundamentalmente, del dominio de aplicación a que se dedique el controlador fuzzy y del número de circuitos que se pretende fabricar. Así, aunque existe un amplio abanico de soluciones intermedias, las situaciones más lógicas son: 1) realización de sistemas de propósito general con funciones de pertenencia y base de reglas programables mediante memoria RAM y uso del defuzzificador programable; 2) desarrollo de prototipos basados en FPGAs con la base de conocimientos y el método de defuzzificación preestablecidos; y 3) fabricación de circuitos de aplicaciones específicas (ASICs) con la base de conocimientos almacenada en memoria ROM o generada mediante bloques lógicos combinacionales y elección del método de defuzzificación que mejor se adapte a la aplicación.

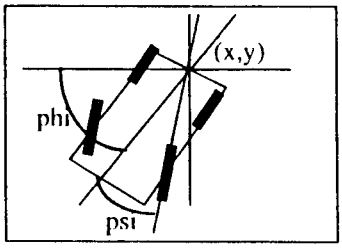
6. Ejemplo de diseño.

En esta sección se muestra la utilidad de las herramientas desarrolladas, aplicándolas a un ejemplo clásico de control fuzzy descrito en la literatura [8]. Se trata de controlar la trayectoria de un camión que se dirige marcha atrás hacia un muelle de carga (Figura 5). Si suponemos que existe una distancia suficiente en el eje Y entre la situación inicial y el punto de destino, el movimiento del camión depende exclusivamente de su coordenada X y del ángulo (phi) que forma el eje longitudinal del camión con el eje X. Estas dos variables son las entradas al controlador difuso que suministra como salida la dirección que deben tomar las ruedas del camión (psi).

El modelo del sistema considerado en este ejemplo es muy simple. Suponiendo que el camión se mueve una distancia d en cada ciclo de inferencia, las nuevas coordenadas se calculan en función de las coordenadas previas y del ángulo proporcionado por el controlador fuzzy de



Situación



Significado de las variables

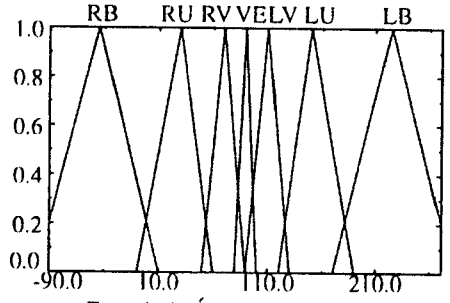
Fig. 5: Descripción gráfica del ejemplo.

acuerdo con las siguientes ecuaciones:

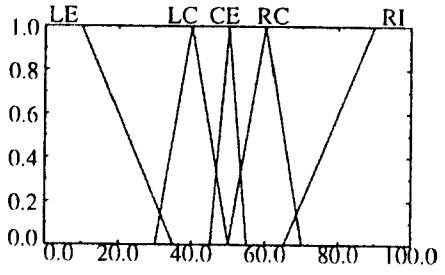
$$phi_t = phi_{t-1} + psi$$
$$x_t = x_{t-1} + d * \cos(phi_t)$$
$$y_t = x_{t-1} + d * \sin(phi_t)$$

Las Figuras 6 y 7 muestran, respectivamente, las funciones de pertenencia y la base de reglas utilizadas. Dada la simplicidad del ejemplo esta información es obtenida, utilizando Xfuzzy, mediante un sencillo procedimiento de prueba y error. Para problemas de mayor complejidad está prevista la inclusión de nuevas facilidades en el entorno que automaticen el proceso de aprendizaje de reglas y ajuste de funciones de pertenencia.

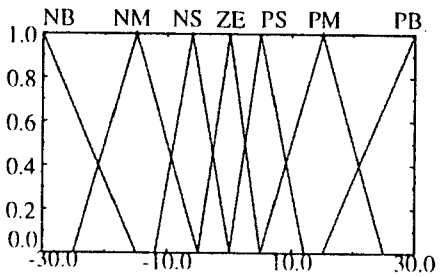
Existe una correspondencia directa entre la descripción algorítmica (XFL) del controlador y la descripción VHDL de su arquitectura. Ambas descripciones incluyen las siguientes etapas: 1) inicialización de las funciones de pertenencia y las variables de entrada y salida; 2) descripción de la base de reglas y ejecución del proceso de inferencia; y 3) defuzzificación y asignamiento de los resultados a la salida.



Entrada 1: Ángulo entre el camión y la horizontal (phi).



Entrada 2: Posición horizontal.



Salida: Ángulo de las ruedas (psi).

Fig. 6: Definición de las funciones de pertenencia.

PHI \ X	LE	LC	CE	RC	RI
RB	PS	PM	PM	PB	PB
RU	NS	PS	PM	PB	PB
RV	NM	NS	PS	PM	PB
VE	NM	NM	ZE	PM	PM
LV	NB	NM	NS	PS	PM
LU	NB	NB	NM	NS	PS
LB	NB	NB	NM	NM	NS

PSI

Fig. 7: Base de reglas.

Tanto en la simulación con Xfuzzy como en la verificación VHDL, el código que modela al sistema bajo control incluye una serie de sentencias que generan los datos necesarios para representar gráficamente la trayectoria seguida por el camión para distintos puntos de partida. Los resultados obtenidos en ambos casos (Figura 8) indican que, para el ejemplo considerado, el objetivo es siempre alcanzado, independientemente del método de defuzzificación empleado.

Por último, la Figura 9 muestra el *layout* de un controlador de propósito general que puede aplicarse al ejemplo que venimos describiendo. Este circuito utiliza la opción de cálculo aritmético de los antecedentes e incluye un defuzzificador programable. La base de reglas y las funciones de pertenencia de los antecedentes son almacenadas en memorias RAM, lo que permite que el controlador pueda ser fácilmente adaptado a diferentes aplicaciones. El diseño se ha realizado con una tecnología CMOS de 1.0 µm. El área de silicio consumida es de 17,40 mm^2.

Referencias

[1] Masa, J., Barriga, A. y Sánchez-Solano, S.: "Un entorno de simulación para sistemas de control basados en lógica difusa". Proc. VII Congreso de Diseño de Circuitos Integrados (DCIS'92), pp. 547-548, Toledo, Nov. 1992

[2] Jimenez, C.J., Sánchez-Solano, S., Barriga, A. y Huertas, J.L.: "Implementación digital de un controlador difuso SISC". X Congreso de Diseño de Circuitos Integrados (DCIS'94), Las Palmas de Gran Canaria, Nov. 1994.

[3] Jiménez, C.J. , Sánchez-Solano, S. y Barriga, A.: "Hardware Implementation of a General Purpose Fuzzy Controller". Proc. 6th IFSA World Congress, (pendiente de publicación) Sao Paulo, Jul. 1995.

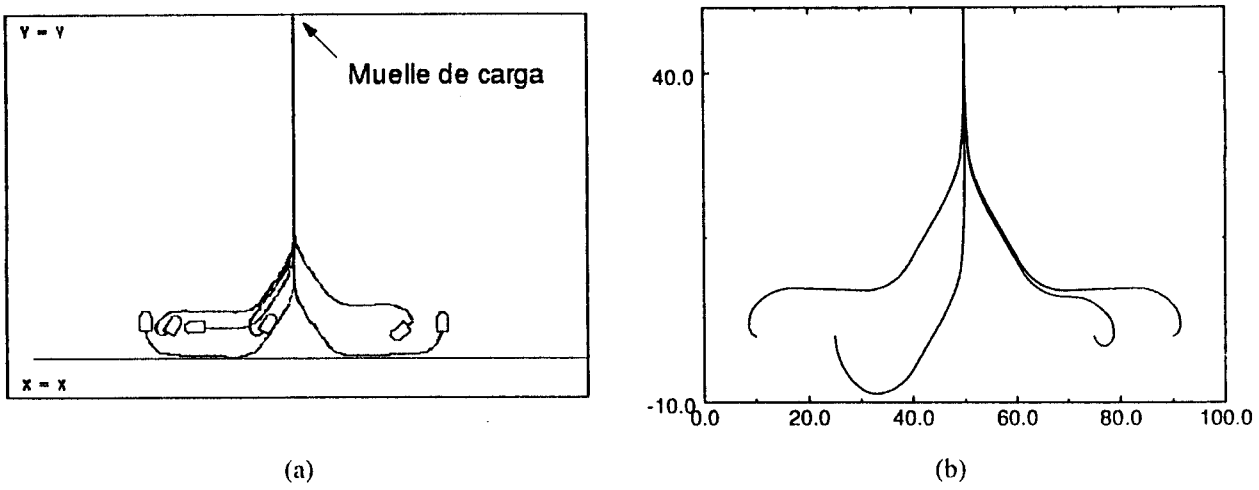


Fig. 8: Resultados de simulación en diferentes niveles: a) Xfuzzy; b) VHDL

[4] A. Zamfirescu, "Logic and Arithmetic in Hardware Description Languages", "Fundamentals and Standards in Hardware Description Languages" Edited by Jean P. Mermet, NATO ASI Series, pp. 109-151, 1993.

[5] D. Galán, C.J. Jimenez, A. Barriga y S. Sánchez-Solano: "VHDL Package for Description of Fuzzy Logic Controllers". EURO-VHDL'95 (pendiente de publicación) Brighton, Sept. 1995.

[6] H. Hellendoorn y C. Thomas, "Defuzzification in fuzzy controllers", *Jour. of Intelligent and Fuzzy Systems*, vol. 1, pp. 109-123, 1993.

[7] R. Jager, H. B. Verbruggen, y P. M. Bruhx, "The role of defuzzification methods in the application of fuzzy logic", in *Proc. Symp. on Intelligent Components and Instruments for Control Applications*, pp. 111-116, Málaga, 1992.

[8] B. Kosko, "Neural Network and Fuzzy Systems", Prentice Hall, 1992.

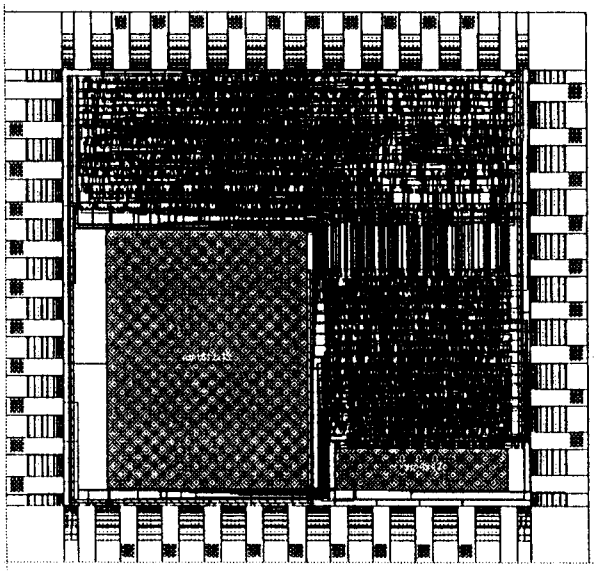


Fig. 9: Layout del controlador fuzzy.

Metodología

Synthesis of information in KBS Validation

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

Abstract

In this paper we study a synthesis function for a set of diagnosis given by experts. The results are applied to a sample case used to validate the expert system Pneumon-ia developed over the shell Milord.

Keywords: Synthesis of information, Consensus, KBS Validation, Fuzzy relations.

1. Introduction

The validation of a knowledge based system encompasses three different analysis [8],[6]: verification, evaluation and test. On one hand, verification and evaluation consist of checking whether the system satisfies, respectively, completely formalizable requirements and partially formalizable ones (inconsistencies, circularity rules,... e.g., [7], [10]). On the other hand, a test is an exam of the behaviour of the system with respect to a given sample case set (e.g., in [4]).

To test a KBS together with a set of cases (see [10] about the process of case selection) we need the optimum solution, the golden standard. However, as the golden standard is often not known and, instead, it is available the diagnosis given by a set of experts, the golden standard is approximated by a synthesised diagnosis obtained from the diagnosis of the experts through a consensus function.

However, these synthesis functions are usually defined in an adhoc way. The results in [12],[13] were oriented, instead, to define these functions axiomatically (i.e., to state some natural conditions that the functions should satisfy and derive from them the functions). The results were obtained through functional equations [1].

In [12],[13] instead of synthesising diagnosis (i.e., mappings from possible solutions to linguistic labels) we studied synthesis of fuzzy relations (fuzzy relations between pairs of possible solutions). This was due to the fact that, in a diagnosis, a linguistic label in isolation has no clear meaning. In fact, the meaning of a label depends on the whole set of linguistic labels used and also on the expert that uses it. Therefore, we consider more meaningful, and more comparable, the relation among pairs of values than the values themselves. In this case, we approximate the relation corresponding to the golden standard as the synthesis of the relations built from expert's diagnosis.

In a first approximation, in [12], we introduced a consensus interval where the golden standard relation should belong. We introduced both crisp and fuzzy consensus interval, stated some properties of both intervals and analysed the results of the expert system Pneumon-ia in relation to them (by means of a Turing Test).

In [13] we built a function for fuzzy relations when it is supposed to be a quasi-arithmetic mean [1]. The synthesised function was used to evaluate again the behaviour of the expert system Pneumon-ia.

In this work, instead, we study if once the synthesised fuzzy relation has been calculated it is possible to obtain a diagnosis (i.e., a mapping from possible solutions to linguistic labels) that generates the synthesised fuzzy relation. This diagnosis would correspond to the golden standard.

Here, although the diagnosis of the experts are defined as mappings from possible solutions to linguistic labels, we work in the unit interval. We assume that there is a valuation function (see definition 3) that returns a numeric value for each linguistic label. An alternative approach within the framework of linguistic labels is possible. This approach would use the operations and synthesis functions defined over linguistic labels introduced in [3],[5]. Some work is intended to be done in this direction.

2. Preliminaries

In this section we review some definitions given in [12],[13] used later on in this paper. First, we define a family $\rho(x,y)$ that evaluates in the fuzzy

setting the inequality x>y for x and y in the unit interval. Then, we introduce a valuation function, Val(I), that given a linguistic label in a predefined set L returns a value in the unit interval. Lately, given a diagnosis function V from a set of possible solutions, M, to L (i.e., a function V that attaches a linguistic label l in L to each m in M) we define a fuzzy relation between pairs of elements in M. This fuzzy relation is defined by means of V together with Val(x) and $\rho(x,y)$.

Definition 1. A fuzzy relation definition function $\rho:[0,1]*[0,1]\rightarrow[0,1]$ of the form:

$$\rho(x,y)=\varphi(\phi(\varphi^{-1}(x)-\varphi^{-1}(y)))$$

with φ and ϕ invertible functions, is a ***fuzzy valuation*** of x>y.

Definition 2. Let L be a set of ordered linguistic labels, then a function $Val:L\rightarrow[0,1]$ is a ***valuation function*** over L if it is monotonic increasing in relation to the ordering in L. (i.e., for all x,y in L, x>y implies Val(x)>Val(y)).

Definition 3. Let L be a set of ordered linguistic labels, let Val be a valuation function over L, let $\rho(x,y)$ be a fuzzy valuation of x>y, and let V be a diagnosis defined over a set of solutions M, i.e., a mapping V from M to L. In these conditions we define the fuzzy relation $P:M*M\rightarrow[0,1]$ as:

$$P(x,y) = \rho(Val(V(x)), Val(V(y))) \quad \text{for each } x,y \in M$$

Definitions 4. Let E={E1, E2, E3, E4, E5, SE} be a set of experts, let L={Not-possible (NOT-P), not-very-possible (NV-P), slightly-possible (SLI-P), moderately-possible (MOD-P), possible (POSSI), quite-possible (QUI-P), very-possible (VER-P), sure (SURE)} be a set of ordered linguistic labels, let L be the set defined as L={Not-considered (NON-C)} $\cup$ L', and let M be the set of possible solutions. Then a case is a set of valuations V_i: M -> L, where there is one valuation for each expert E_i of E.

Definitions 5. Let Val be the valuation function defined as zero if the label is `non-considered` and Pos(x)/14 + .5 if x is a label in L'. Pos(x) refers to the position where the linguistic label is placed in L', i.e., Pos(Not-P)=0, Pos(NV-P)=1, ... Let ρ be the fuzzy relation: $\rho(x,y)=(x-y+1)/2$.

3. Synthesis of diagnosis

We begin showing that given a synthesised relation (see [12]) there is not a unique set of labels that generates it. Next we define a method to fix the linguistic labels of the golden standard.

Example 1. Let E, L, Val, ρ be functions according to definitions 4 and 5. Let $Val^{-1}(x)$ be the inverse function of Val defined as:

$$Val_i^{-1}(v) = \begin{cases} \text{non-considered} & \text{if } v \leq .45 \\ Pos^{-1}(14*(v - 0.5)) & \text{if } v > .45 \end{cases}$$

where Pos^{-1}, the inverse function of Pos, is a function that given a value j in the interval [0,Card(L')-1], where Card(L') is the cardinality of the set of linguistic labels L', returns a label in L'. It returns the label in the i-th position for all $j \in (i - 0.5, i + 0.5]$. Notice that $Val^{-1}(Val(x))=x$ for all x in L but it is not true $Val(Val^{-1}(x))=x$ for $x\in[0,1]$.

As we suppose that the synthesised relation, ρ_{GS}, is known for all x,y we can isolate the fuzzy valuation of y from the one of x and the value of $\rho_{GS}(x,y)$:

$$Val(V_{GS}(y)) = \rho_D(Val(V_{GS}(x)),\rho_{GS}(x,y))$$

where $\rho_D(\alpha,\gamma)=\varphi(- \phi^{-1}(\varphi^{-1}(\gamma))+\varphi^{-1}(\alpha))$.

In the same way, we can obtain the linguistic label attached to y in the golden standard

$$V_{GS}(y)=Val^{-1}(\rho_D(Val(V_{GS}(x)),\rho_{GS}(x,y)))$$

However, as it can be seen, the label of each solution depends on the value attached to $Val(V_{GS}(x))$. When this value changes, the linguistic label attached to each possible solution also changes.

In the following we give the linguistic label attached to each aetiology in {T, L, M, H, A, E, P} when the fuzzy relation $\rho(x,y)$ is the one in table 1, and the fuzzy valuation attached to H, $V_{GS}(H)$ changes from 0 to 1 by steps of 0.1. The fuzzy relation of table 1 is defined in [13] and from a case of the sample case set used in [13], [8], [9] to evaluate the behaviour of the expert system Pneumon-ia [14] developed over the shell Milord [11].

with $\rho_{GS}(H,y)$=(.44 .42 .36 .5 .44 .32 .06)

$Val(V_{GS}(H))$, Aetiologies:

	T	L	M	H	A	E	P
0	(NON-C	NON-C	NON-C	NON-C	NON-C	NON-C	QUI-P)
0.1	(NON-C	NON-C	NON-C	NON-C	NON-C	NON-C	SURE)
0.2	(NON-C	NON-C	NOT-P	NON-C	NON-C	NV-P	SURE)
0.3	(NON-C	NON-C	SLI-P	NON-C	NON-C	MOD-P	SURE)
0.4	(NV-P	NV-P	MOD-P	NON-C	NV-P	POSSI	SURE)
0.5	(SLI-P	MOD-P	QUI-P	NOT-P	SLI-P	VER-P	SURE)
0.6	(POSSI	POSSI	VER-P	SLI-P	POSSI	SURE	SURE)
0.7	(QUI-P	VER-P	SURE	MOD-P	QUI-P	SURE	SURE)
0.8	(SURE	SURE	SURE	QUI-P	VER-P	SURE	SURE)
0.9	(SURE	SURE	SURE	VER-P	SURE	SURE	SURE)
1	(SURE	SURE	SURE	SURE	SURE	SURE	SURE)

In this example we have attached values to $Val(V_{GS}(H))$ without taking into account the value that the experts attached to it (i.e., the values $Val(V_i(H))$. We now consider that this value can be obtained through a synthesis function from the values given by the experts. That is,

$$Val(V_{GS}(H)) = F(Val(V_1(H)), Val(V_2(H)), ..., Val(V_n(H)))$$

We introduce a proposition that considers the synthesised relation and requires that this relation is built from the synthesised fuzzy valuation. We establish and proof the proposition when two relations are synthesised and then we establish the general case with n relations.

Let F_1 be a fuzzy relation synthesis function, and let F_2 be a synthesis function of experts values. In this case we require that the synthesised relation when two diagnosis V_1 and V_2 are considered:

$$F_1(\rho(Val(V_1(x)),Val(V_1(y))), \rho(Val(V_2(x)),Val(V_2(y))))$$

equals the relation ρ build by means of the synthesis of the fuzzy values $Val(V_1(x))$, $Val(V_2(x))$ and $Val(V_1(y))$, $Val(V_2(y))$, i.e.,

$$\rho(F_2(Val(V_1(x)),Val(V_2(x))), F_2(Val_1(V_1(y)),Val_2(V_2(y))))$$

Defining $t_i=Val(V_i(t))$ we can express this condition as:

$$F_1(\rho(x_1,y_1),\rho(x_2,y_2)) = \rho(F_2(x_1,x_2),F_2(y_1,y_2))$$

	T	L	M	H	A	E	P
T	.50	.49	.42	.56	.51	.39	.13
L	.51	.50	.44	.58	.52	.40	.14
M	.58	.56	.50	.64	.59	.46	.21
H	.44	.42	.36	.50	.44	.32	.06
A	.49	.47	.41	.55	.50	.38	.12
E	.61	.60	.53	.68	.62	.50	.24
P	.87	.86	.79	.94	.88	.76	.50

Table 1: Synthesised fuzzy relation.

Proposition 1. Let ρ be a fuzzy relation definition function from [0,1]*[0,1] to [0,1] of the form of definition 1. Let ϕ and φ be its generators, both invertible, continuous and derivable functions. Let F_1 be a fuzzy relation combination function of the form of a quasi-arithmetic mean [1] and let g be its generator, this is

$$F_1(x,y) = g\left(\frac{g^{-1}(x)+g^{-1}(y)}{2}\right) \tag{1.1}$$

Let F_2 be a fuzzy valuation combination function also of the form of a quasi-arithmetic mean, and let f be its generator, so:

$$F_2(x,y) = f\left(\frac{f^{-1}(x)+f^{-1}(y)}{2}\right) \tag{1.2}$$

In this conditions

$$F_1(\rho(x_1,y_1),\rho(x_2,y_2)) = \rho(F_2(x_1,x_2),F_2(y_1,y_2)) \tag{1.3}$$

if and only if

$$f^{-1}(\varphi(x))=k_1x+k_2$$
$$g^{-1}(\varphi(\phi(x)))=k_3x+k_4 \tag{1.4}$$

Proof. We begin proving that (1.3) leads to (1.4), and then the other implication. First, in (1.3) we substitute ρ by its definition and F_1 and F_2 by their definition in (1.1) and (1.2) obtaining:

$$g\left(\frac{g^{-1}\left(\varphi(\phi(\varphi^{-1}(x_1)-\varphi^{-1}(y_1)))\right)+g^{-1}\left(\varphi(\phi(\varphi^{-1}(x_2)-\varphi^{-1}(y_2)))\right)}{2}\right)$$
$$= \varphi(\phi(\varphi^{-1}\left(f\left(\frac{f^{-1}(x_1)+f^{-1}(x_2)}{2}\right)\right) - \varphi^{-1}\left(f\left(\frac{f^{-1}(y_1)+f^{-1}(y_2)}{2}\right)\right))$$

If we define h^{-1} as:

$$h^{-1}(x) = g^{-1}(\varphi(\phi(x))$$

we can express the previous equality as:

$$h\left(\frac{h^{-1}(\varphi^{-1}(x_1)-\varphi^{-1}(y_1))+h^{-1}(\varphi^{-1}(x_2)-\varphi^{-1}(y_2))}{2}\right) = \varphi^{-1}\left(f\left(\frac{f^{-1}(x_1)+f^{-1}(x_2)}{2}\right)\right) - \varphi^{-1}\left(f\left(\frac{f^{-1}(y_1)+f^{-1}(y_2)}{2}\right)\right)$$

that with the changes

$$x_i=\varphi(a_i) \qquad \text{and} \qquad y_i=\varphi(b_i)$$

is rewritten as:

$$h\left(\frac{h^{-1}(a_1-b_1)+h^{-1}(a_2-b_2)}{2}\right) = \varphi^{-1}\left(f\left(\frac{f^{-1}(\varphi(a_1))+f^{-1}(\varphi(a_2))}{2}\right)\right) - \varphi^{-1}\left(f\left(\frac{f^{-1}(\varphi(b_1))+f^{-1}(\varphi(b_2))}{2}\right)\right)$$

If we define

$$m^{-1}(x)=f^{-1}(\varphi(x))$$

then we get that

$$h\left(\frac{h^{-1}(a_1-b_1)+h^{-1}(a_2-b_2)}{2}\right) = m\left(\frac{m^{-1}(a_1)+m^{-1}(a_2)}{2}\right) - m\left(\frac{m^{-1}(b_1)+m^{-1}(b_2)}{2}\right) \tag{1.5}$$

If we suppose in this equation that $b_1=b_2=0$, we get

$$h\left(\frac{h^{-1}(a_1)+h^{-1}(a_2)}{2}\right) = m\left(\frac{m^{-1}(a_1)+m^{-1}(a_2)}{2}\right)$$

that is the Jensen equation [1]. This functional equation has as solution

$$h^{-1}(a_1)=\alpha m^{-1}(a_1)+\beta \tag{1.6}$$

If we substitute h(x) and $h^{-1}(x)$ in (1.5) we get

$$m\left(\frac{m^{-1}(a_1-b_1)+m^{-1}(a_2-b_2)}{2}\right) = m\left(\frac{m^{-1}(a_1)+m^{-1}(a_2)}{2}\right) - m\left(\frac{m^{-1}(b_1)+m^{-1}(b_2)}{2}\right)$$

that is equivalent to:

$$m\left(\frac{m^{-1}(a_1-b_1)+m^{-1}(a_2-b_2)}{2}\right) + m\left(\frac{m^{-1}(b_1)+m^{-1}(b_2)}{2}\right) = m\left(\frac{m^{-1}(a_1)+m^{-1}(a_2)}{2}\right)$$

Defining

$$a_i = \alpha_i + \beta_i \qquad \text{and} \qquad b_i=\beta_i$$

we get

$$m\left(\frac{m^{-1}(\alpha_1)+m^{-1}(\alpha_2)}{2}\right) + m\left(\frac{m^{-1}(\beta_1)+m^{-1}(\beta_2)}{2}\right) = m\left(\frac{m^{-1}(\alpha_1+\beta_1)+m^{-1}(\alpha_2+\beta_2)}{2}\right)$$

This equation leads to:

$$m^{-1}(x) = k_1x+k_2$$

According to the definition of m this is equivalent to

$$f^{-1}(\varphi(x)) = k_1x + k_2 \tag{1.7}$$

In the same way, according to the definition of h and equality (1.6), we get

$$g^{-1}(\varphi(\phi(x)))=k_3x+k_4 \tag{1.8}$$

With (1.7) and (1.8) the first implication is proven. If we substitute this expressions in (1.5), equation equivalent to (1.3), then, the second implication is proven, and therefore the proposition. ❑

This proposition can be generalised to the case of n experts and n relations.

Proposition 2. Let ρ be a fuzzy relation definition function from [0,1]*[0,1] to [0,1] of the form of definition 1. Let ϕ and φ be its generators, both invertible, continuous and derivable functions. Let F_1 be a fuzzy relation combination function and let g be its generator, this is:

$$F(x_1,\ldots,x_n) = g\left(\frac{\sum_{i=1}^{n} g^{-1}(x_i)}{n}\right)$$

Let F_2 be a fuzzy valuation combination function also of the form of a quasi-arithmetic mean, and let f be its generator.

In these conditions:

$$F(\rho(x_1,z_1),\rho(x_2,z_2),....,\rho(x_n,z_n))= \rho(F_2(x_1,x_2,....,x_n),F_2(y_1,y_2,....,y_n))$$

if and only if

$$f^{-1}(\varphi(x))=K_1x + K_2$$
$$g^{-1}(\varphi(\phi(x)))=k_3x+k_4$$

Propositions 1 and 2 state that when a fuzzy relation definition function is given, the combination functions are completely determined. In fact, condition (1.4) implies a g^{-1} of the form:

$$g^{-1}(x)=K\phi^{-1}(\varphi^{-1}(x))+K'$$

and a $f^{-1}(x)$ of the form:

$$f^{-1}(x)=K_1\,\varphi^{-1}(x)+K_2$$

Example 2. The results achieved in proposition 2 are used here to determine the diagnosis corresponding to the golden standard. We define ρ, Val and Val^{-1} as in example 1. We define the fuzzy valuation of each expert according to table 2. These valuations are the ones used in example 1 to calculate the synthesised fuzzy relation of table 1.

In these conditions, we can apply proposition 2 obtaining the combination function F_2 with a generator f(x) of the form:

$$f^{-1}(x) = k_1\,x + k_2$$

A generator of this form implies that F_2 is the arithmetic mean.

In table 2, together with the valuations of the experts, we give the fuzzy valuation corresponding to the golden standard. That is, the arithmetic mean of the valuations of the experts. The fuzzy valuation corresponding to the golden standard, through function ρ and proposition 2, generates the synthesised fuzzy relation given in table 1.

In table 3, we give the linguistic label corresponding to the golden standard. It has been calculated through the inverse function Val^{-1} defined in example 3.

	T	L	M	H	A	E	P
E1	9/14	11/14	0	0	0	9/14	12/14
E2	0	0	0	0	0	8/14	12/14
E3	0	0	9/14	0	8/14	8/14	12/14
E4	0	0	0	0	0	0	13/14
E5	0	0	13/14	0	0	0	12/14
SE	0	0	0	9/14	10/14	10/14	11/14
EC	.12	.15	.31	0	.11	.35	.87

Table 2

GS	-	-	-	-	-	-	QUI-F

Table 3

4. Conclusions

In this paper we have studied a synthesis function for diagnosis of experts. We have applied the results obtained to a case of a sample case set used in [13], [8], [9] to examine the behaviour of the expert system Pneumon-ia [14] developed over the shell Milord [11].

Acknowledgements

The help of Professor Claudi Alsina is gratefully acknowledged. I would also thank the help of the Institut d'Investigació en Intel.ligència Artificial (IIIA, CSIC).

References

[1] Aczél, J., *A short course on functional equations*, (D. Reidel Publishing Company, 1987).

[2] Ayel, M., Vignollet, L., SYCOJET and SACCO, Two Tools for Verifying Expert Systems, *International Journal of Expert Systems* 6:3 (1993) 357-382.

[3] Delgado, M., Verdegay, J.L., Vila, M.A., On Aggregation Operations of Linguistic Labels, *Int. J. of Intelligent Systems*, Vol. 8, (1993) 351-370.

[4] Hernandez, C., Sancho, J.J., Belmonte, M.A., Sierra, C., Sanz, F., Validation of the Medical Expert System RENOIR, *Computers and Biomedical Research* 27 (1995) 456-471.

[5] Herrera, F., Herrera-Viedma, E., Verdegay, J.L., A Sequential Selection Process in Group Decision Making with a Linguistic Assessment Approach, *Int. J. Information Science*, Vol. 80, (1995) 1-17.

[6] Hoppe, T., Meseguer, P., VVT Terminology: A Proposal, *IEEE Expert*, June (1993) 48-55.

[7] Kinkielélé, D., Ayel, M., On Discovering Potential Inconsistencies in Validating Fuzzy Knowledge Bases, in *Proceedings of the Fifth International Conference on Information*

Processing and Management of Uncertainty in Knowledge-Based Systems, Paris, July 4-8, (1994) 533-539.

[8] Meseguer, P., *Validation of Multi-Level Rule-Based Expert Systems*, Thesis, Universitat Politècnica de Catalunya (1992).

[9] Meseguer, P., Verdaguer, A., Verification of Multi-Level Rule-Based Expert Systems: Theory and Practice, *International Journal of Expert Systems* 6:2 (1993) 163-192.

[10] O'Keefe, R.M., O'Leary, D.E., Expert System Verification and Validation a Survey and Tutorial, *Artificial Intelligence Review* 7 (1993) 3-42.

[11] Sierra, C., *MILORD: Arquitectura multinivell per a sistemes experts en classificació*, Thesis, Universitat Politècnica de Catalunya (1989).

[12] Torra, V., *Contribució a l'estudi de funcions de síntesi per a la intel.ligencia artificial*, Thesis, Universitat Politècnica de Catalunya (1994).

[13] Torra, V., Synthesis of fuzzy relations for knowledge based systems, Submitted to *Fuzzy Sets and Systems*. (1995).

[14] Verdaguer, A., *PNEUMON-IA: Desenvolupament i validació d'un sistema expert d'ajuda al diagnòstic mèdic*, Thesis, Universitat Autònoma de Barcelona, (1989).

Revising and Updating a Knowlegde Base by means of Fuzzy Similarity Relations

Ricardo Rodríguez, Pere Garcia and Lluís Godo

Institut d'Investigació en Intel.ligència Artificial (IIIA)
Consejo Superior de Investigaciones Científicas (CSIC)
{ricardo,pere,godo}@iiia.csic.es

Abstract

Similarity-based models were first used by Ruspini to give semantics for fuzzy logic ([7]). In these models, incomplete information is represented by an evidential set, *i.e. a set of possible worlds that are compatible with the evidence, together with a* fuzzy similarity relation *on the set of possible worlds that allows to describe the resemblance of arbitrary subsets of worlds to those belonging to the evidencial set. On the other hand, the question addressed by theory change formalisms is which kind of modifications have to be performed to a knowledge base when adding (retracting) a new proposition to (from) that knowledge base. Revision and update are the two theory change operations which have been paid more attention in the literature. In this paper we study a connection between similarity-based models and theory change, coming from the fact that revision and updating operators have been characterized by Katsuno and Mendelson in terms of pre-order relations in a set of possible worlds, which in turn naturally arise from some similarity measures definable in the above mentioned similarity-based models.*

Keywords: *Fuzzy Similarity Relation, Implication and Consistency Measures, Updating and Revision Operators.*

1 Introduction

The aim of this paper is to explore some relationships between similarity-based models and theory change operators. Similarity-based models were first used by Ruspini to give semantics for fuzzy logic ([7]). These models, which combine logic and metric concepts, are given by an *evidential set*, i.e. a set of possible worlds that are compatible with evidence, together with a fuzzy similarity relation on the set of possible worlds that allows to describe the resemblance of arbitrary subsets of worlds to those belonging to the evidencial set. On the other hand, theory change formalisms deal with mechanisms that permit to give amount of the tasks of adding (retracting) a proposition to (from) an existing knowledge base. The natural question addresed by these formalisms is what should the resulting theory be. In particular, one of the basic problems is when the new information to be added is inconsistent with the given knowledge base. Concerning this problem, a growing body of work takes as a departure point the postulates proposed by Alchourrón, Gärdenfors and Makinson ([1]) for the so-called revision operators and the Katsuno and Mendelson's postulates ([5]) for the so-called updating operators. Both kind of operators have been also characterized by Katsuno and Mendelson in terms of pre-order relations among the set of interpretations (worlds) of the logical language considered.

There is a straightforward way of relating theory change (revision and update) and similarity-based models. Namely, some measures of proximity definable in similarity models naturally induce pre-ordering relations between worlds that, in turn, are required by Katsuno and Mendelson ([4]and [5]) to characterize theory change operators that satisfy the postulates of revision and updating operations. In this paper, we investigate this relationship and study which are the properties of the theory change operators arising from such connection.

This paper is organized as follows. Section 2 and Section 3 contain some basic background knowledge about the notion of similarity-based models and about revision and updating operators including the set of corresponding postulates and representation theorems. Section 4 is devoted to present our approach to connect theory change and similarity-based models together. We end with some remarks and conclusions.

2 Similarity-based Models

As already noticed, Ruspini's approach to possibilistic reasoning is based on the introduction of a fuzzy binary relation $\mathbf{S}$, called similarity relation, that maps pairs of possible worlds into numbers between 0 and 1 ($\mathbf{S} : \mathcal{W} \times \mathcal{W} \longmapsto [0,1]$). This function captures a notion of proximity between possible worlds, with a value of 1 corresponding to the identity of possible worlds and a value 0 indicating that knowledge of propositions that are true in one possible world does not provide any indication about propositions that are true in the other. Some requirements must be satisfied by similarity functions. For example, the similarity degree of any world with itself has the highest value (i.e. 1), it is symmetric and satisfies a relaxed form of transitivity. This is expresed as: $S(w,w'') \geq S(w,w') \otimes S(w',w'')$. In general, $\otimes$

is considered to be a triangular norm (or T-norm). Its quasi-inverse is represented by $\otimes\longrightarrow$ defined as:

$$a \otimes\longrightarrow b = \text{SUP}\ \{c \in [0,1] \mid a \otimes c \leq b\}.$$

We say that a similarity function is discriminating if, and only if, it holds:

$$\forall w_1 w_2\ S(w_1, w_2) = 1 \Longleftrightarrow w_1 = w_2.$$

From now on let us consider a propositional logical language L and let $\mathcal{W}$ be the set of its interpretations. Having defined a similarity function **S** on $\mathcal{W}$ and an evidence set $\mathbf{E} \subseteq \mathcal{W}$, Ruspini introduces a pair of unconditoned implication and consistecy measures on L defined respectively as:

$$I_{S,E}(p) = \underset{w_1 \in E}{INF}\ \underset{w_2 \models p}{SUP}\ S(w1, w2)$$
$$C_{S,E}(p) = \underset{w_1 \in E}{SUP}\ \underset{w_2 \models p}{SUP}\ S(w1, w2).$$

These measures, as Ruspini points out, are lower and upper bounds of the resemblance degrees between p-worlds and those in **E**, from the point of view of **E**. The value of $I_{S,E}(p)$ provides the measure of what extent p can be considered certain with respect to the incomplete knowledge represented by **S** and **E**. In particular, $I_{S,E}(p) = 1$ iff $\mathbf{E} \models p$. On the other hand, the value of $C_{S,E}(p)$ provides the measure of what extent p can be considered compatible with the available knowledge. In particular, in the finite case, $C_{S,E}(p) = 1$ iff $\mathbf{E} \not\models \neg p$. Observe that when the evidencial set is a singleton w both measures coincide, i.e. $I_{S,w}(p) = C_{S,w}(p)$ for any world w and proposition p.

Since $C_{S,E}$ is a possibility measure, it holds that $C_{S,E}(p \vee q) = max(C_{S,E}(p), C_{S,E}(q))$ for all propositions p and q. Therefore we also have $C_{S,E}(p) = \max\{C_{S,E}(p \wedge q), C_{S,E}(p \wedge \neg q)\}$. In particular, when $C_{S,E}(p \wedge q) > C_{S,E}(p \wedge \neg q)$, it results that $C_{S,E}(p) = C_{S,E}(p \wedge q)$. This can be interpreted as saying that the $p \wedge q$-worlds are closer (consistent) to the evidence set **E** than the $p \wedge \neg q$-worlds are. In this context, the term "closer" is used in the sense of "more similar".

Furthermore, Ruspini generalizes the semantical entailment relationship between propositions in terms of a measure of their neighborhood to the evidence set **E**, by defining the so-called degree of conditioned implication and consistency measures as follows:

$$I_{S,E}(q \mid p) = \underset{w \in E}{INF}\,(I_{S,w}(p) \otimes\longrightarrow I_{S,w}(q)).$$
$$C_{S,E}(q \mid p) = \underset{w \in E}{SUP}\,(C_{S,w}(p) \otimes\longrightarrow C_{S,w}(q)).$$

Notice that in particular, when $\mathbf{E} \models p$ then we recover the the previous unconditioned measures. Moreover, if **S** reduces to the clasical equality, we have that $I_{S,E}(q \mid p) = 1$ whenever $\mathbf{E} \wedge p \models q$ and $C_{S,E}(q \mid p) = 0$ whenever $\mathbf{E} \models p$ and $\mathbf{E} \models \not\perp$.

However, these conditioned measures may have some difficulties in a general setting since, for instance, it may happen that $I_{S,E}(q|p) > 0$ or $C_{S,E}(q|p) = 1$ while beging p and q mutually exclusive. In [3] a modified version of these conditional measures are proposed by defining:

$$I^2_{S,E}(q \mid p) = \underset{w \in E}{INF}\,(I_{S,w}(p) \otimes\longrightarrow I_{S,w}(q \wedge p)).$$
$$C^2_{S,E}(q \mid p) = \underset{w \in E}{SUP}\,(C_{S,w}(p) \otimes\longrightarrow C_{S,w}(q \wedge p)).$$

Observe that when p and q are mutually exclusive, $I^2_{S,E}(q \mid p) = 0$ and $C^2_{S,E}(q \mid p) = 0$. Note, moreover, that $C^2_{S,E}(q \mid p) = 1$ only if there exists one world w in **E** such that $C_{S,w}(p) = C_{S,w}(p \wedge q)$. This is interpreted as saying that the $p \wedge q$-worlds are almost as close (consistent) to the evidence set **E** as the $p \wedge \neg q$-worlds are. Reciprocally, if there does not exist such a world w, then $C^2_{S,E}(\neg q \mid p) < 1$ and meaning that the $p \wedge q$-worlds are closer (consistent) to the evidence set **E** than the $p \wedge \neg q$-worlds are.

3 Theory Change

Given a knowledge base KB (represented by a set of sentences in a language L) and a sentence ϕ, $KB^\star_\phi$ denotes the revision of KB by ϕ, that is, the new knowledge base resulting from the addition of ϕ to the old knowledge base KB.

In [1], Alchourrón, Gärdenfors and Makinson proposed eight postulates which, they argue, must be satisfied by any reasonable revision operator. These postulates are [1]:

⋆**1.** $KB^\star_\phi = Cn(KB^\star_\phi)$.

⋆**2.** $\phi \in KB^\star_\phi$.

⋆**3.** $KB^\star_\phi \subseteq Cn(KB \cup \{\phi\})$.

⋆**4.** if $\neg\phi \notin KB$ then $Cn(KB \cup \{\phi\}) \subseteq KB^\star_\phi$.

⋆**5.** if $Cn(\phi) \neq \mathrm{L}$ then $KB^\star_\phi \neq \mathrm{L}$.

⋆**6.** if $\phi \equiv \nu$ then $KB^\star_\phi \equiv KB^\star_\nu$.

⋆**7.** $KB^\star_{\phi \wedge \nu} \subseteq Cn((KB^\star_\phi) \cup \{\nu\}))$.

⋆**8.** if $\neg\phi \notin KB^\star_\nu$ then $Cn(KB^\star_\phi \cup \{\nu\}) \subseteq KB^\star_{\phi \wedge \nu}$.

In [4] Katsuno and Meldelzon gave a model theoretic characterization of these postulates by means of orderings among interpretations when the KB is finite. Let I be the set of all the interpretations of a finitary propositional language L. A pre-order $\leq$ over I is a reflexive and transitive relation on it. A pre-order is total if for all I and J interpretations, either $I \leq J$ or $J \leq I$. They consider a function that assigns to each knowledge base KB a pre-order $\leq_{KB}$ over I. They say this assignment is *faithful* [2] if the following three conditions hold:

1. if $I, I' \in \mathrm{Mod}(KB)$ then $I <_{KB} I'$ does not hold,
2. if $I \in \mathrm{Mod}(KB)$ and $I' \notin \mathrm{Mod}(KB)$ then $I <_{KB} I'$ holds, and
3. if $KB \equiv \phi$ then $\leq_{KB} \equiv \leq_\phi$,

[1] Cn is any consequence operator which includes classical propositional logic, is compact and satisfies the deduction theorem.

[2] The term *persistent* was used in [4],but it was replaced by *faithful* in [5]

where Mod(KB) stands for the set of models of the knowledge base.

Let M be a subset of I. An interpretation I is minimal in M with respect to $\leq_{KB}$ if $I \in$ M and there is no $I' \in$ M such that $I' <_{KB} I$. Let Min(M,$\leq_{KB}$) be the set of all $I \in$ M such that I is minimal in M with respect to $\leq_{KB}$. Katsuno and Mendelzon established the following characterization of all revision operators.

Theorem 1: A revision operator $\star$ satisfies the conditions ($\star$1) $\sim$ ($\star$8) if and only if there exists a *faithful* assignment that maps each knowledge base KB to a total pre-order $\leq_{KB}$ such that Mod($KB^{\star}_{\phi}$) = Min(Mod(ϕ),$\leq_{KB}$).

Moreover, they proposed a new kind of modifications for a knowledge base, the so-called *update operators*. They consist of bringing the knowledge base up to date when the world described by it, changes. They showed that the postulates ($\star$1) $\sim$ ($\star$8) only apply to revision, not to update. They provided a new set of postulates that apply to update and characterize all operators satisfying these postulates in terms of a set of partial orders defined among possible worlds. They use $KB^{\diamond}_{\phi}$ to denote the result of updating a knowledge base KB with the sentence ϕ. The postulates for update are:

$\diamond$ **1.** $KB^{\diamond}_{\phi} \models \phi$.

$\diamond$ **2.** If $KB \models \phi$ then $KB^{\diamond}_{\phi} \equiv KB$.

$\diamond$ **3.** If both KB and ϕ are satisfiable then: $KB^{\diamond}_{\phi}$ is also satisfiable.

$\diamond$ **4.** If $KB_1 \equiv KB_2$ and $\phi_1 \equiv \phi_2$ then: $(KB_1)^{\diamond}_{\phi_1} \equiv (KB_2)^{\diamond}_{\phi_2}$.

$\diamond$ **5.** $KB^{\diamond}_{\phi} \bigcup \{\nu\} \models KB^{\diamond}_{\phi \wedge \nu}$.

$\diamond$ **6.** if $KB^{\diamond}_{\phi_1} \models \phi_2$ and $KB^{\diamond}_{\phi_2} \models \phi_1$ then: $KB^{\diamond}_{\phi_1} \equiv KB^{\diamond}_{\phi_2}$.

$\diamond$ **7.** If KB is complete then: $KB^{\diamond}_{\phi_1} \bigcup KB^{\diamond}_{\phi_2} \models KB^{\diamond}_{\phi_1 \vee \phi_2}$.

$\diamond$ **8.** $(KB_1 \bigcap KB_2)^{\diamond}_{\phi} \equiv ((KB_1)^{\diamond}_{\phi} \bigcap (KB_2)^{\diamond}_{\phi})$.

The following representation theorem is stablished for update operators by Katsuno and Mendelzon in [5] for finite knowledge bases.

Theorem 2: An update operator $\diamond$ satisfies conditions ($\diamond$1) $\sim$ ($\diamond$8) if and only if there exists a *faithful* assignment that maps each interpretation I to a partial pre-order $\leq_I$ such that

$$\mathrm{Mod}(KB^{\diamond}_{\phi}) = \bigcup_{I \in Mod(KB)} Min(Mod(\phi), \leq_I).$$

There are two basic differences between revision and update from a model theoretic point of view. One is that revision is characterized by total pre-orders while update is defined from a family of partial pre-orders. However, it is also possible to build a class of update operators based on a family of total pre-orders, namely, when the postulates ($\diamond$6) and ($\diamond$7) are replaced by a new postulate:

$\diamond$9. If KB is complete and $KB^{\diamond}_{\phi} \bigcup \{\nu\}$ is satisfiable then $KB^{\diamond}_{\phi \wedge \nu} \models KB^{\diamond}_{\phi} \bigcup \{\nu\}$.

As for the seconf difference, in the case of update a different ordering is induced by each model of KB, while for revision only one ordering is induced by the whole KB. This "local" behaviour of update contrasts with the "global" behaviour of revision.

4 Similarities and Revision/Update Operators

Now, we will show a relationship between the similarity models and the two considered kinds of modifications to be performed to a knowledge base when new information arrives. First, we will propose a definition for revision and update, and then we will prove that they verify some of the correspoding eight postulates.

Definition 1: $\psi \in KB^{\star}_{\phi} \stackrel{\text{def}}{\Longleftrightarrow}$ Mod($\phi \& \neg\psi$) = $\emptyset$ or $C_{S,\mathcal{KB}}(\neg\psi \& \phi) < C_{S,\mathcal{KB}}(\psi \& \phi)$ [3].

Definition 2: $\psi \in KB^{\diamond}_{\phi} \stackrel{\text{def}}{\Longleftrightarrow}$ Mod($\phi \& \neg\psi$) = $\emptyset$ or $C^2_{S,\mathcal{KB}}(\neg\psi \mid \phi) < 1$.

In the first definition, a formula ψ belongs to the revision of KB by ϕ if, and only if, it is either a logical consequence of ϕ or the $\phi \wedge \psi$-worlds are closer (consistent) to the set of KB-worlds than the $\phi \wedge \neg\psi$-worlds are. In the second definition, a formula ψ belongs to the update of KB by ϕ if, and only if, it is either a logical consequence of ϕ or the $p \wedge q$-worlds are closer (consistent) to each KB-world than the $p \wedge \neg q$-worlds are. This interpretation is exploted in [6] to relate counterfactuals and theory change via similarity functions.

The theory change operators above defined verify most of postulates for revision and update operators, as it is proved by the following two theorems.

Theorem 3: The operator $\star$ defined in Definition 1 verifies the postulates ($\star$1) $\sim$ ($\star$3) and ($\star$5) $\sim$ ($\star$8), but not ($\star$4).

Theorem 4: The operator $\diamond$ defined in Definition 2 verifies the postulates ($\diamond$1), ($\diamond$2'), and ($\diamond$3) $\sim$ ($\diamond$8), where the new postulate ($\diamond$ 2') is:

$\diamond$ **2'.** If $KB \models \phi$ then $KB^{\diamond}_{\phi} \subseteq KB$.

These results are not surprizing. Postulates ($\star$4) and ($\diamond$2) are not verified only in the infinite case. In the finite case these properties are recovered, as we see below. For this, we will use the consistency measures introduced in section 2 to define the following pre-orders.

[3] $\mathcal{KB}$ notes the set of worlds that satisfy the knowledge base KB

Definition 3: Given a similarity function S, we define for each knowledge base KB a total pre-order $\leq_{KB}$ on the set of possible worlds, as follows:

$$w_1 \leq_{KB} w_2 \stackrel{\text{def}}{\Longleftrightarrow} C_{S,\mathcal{KB}}(w_1) \geq C_{S,\mathcal{KB}}(w_2).$$

Definition 4: Given a similarity function S, for each world w we define the total pre-order $\leq_w$ on the set of possible worlds as follows:

$$w_1 \leq_w w_2 \stackrel{\text{def}}{\Longleftrightarrow} C_{S,w}(w_1) \geq C_{S,w}(w_2).$$

In the particular case of KB being finite, theorems 1 and 2 allow us to define a revision and an update operators, $\star_f$ and $\diamond_f$ respectively, using the above pre-orders. Namely, the set of models of the revised and updated knowledge base KB by ϕ are respectively:

$$\text{Mod}(KB^{\star_f}_{\phi}) =$$

$$\{w' \models \phi \mid \exists w \models KB \text{ and } S(w,w') = C_{S,\mathcal{KB}}(\phi)\},$$

$$\text{Mod}(KB^{\diamond_f}_{\phi}) =$$

$$\{w' \models \phi \mid \exists w \models KB \text{ and } S(w,w') = C_{S,w}(\phi)\}$$

Now we can present the main result in this paper: in the finite case, the operators $\star_f$ and $\diamond_f$, obtained via the Katsuno Mendelzon's representation theorem from the pre-orders of definitions 3 and 4 , are exactly those given by the above definitions 1 and 2, respectively. This is formally expressed in the next two theorems.

Theorem 5: $\psi \in KB^{\star_f}_{\phi} \Longleftrightarrow \text{Mod}(\phi \& \neg\psi) = \emptyset$ or $C_{S,\mathcal{KB}}(\neg\psi \& \phi) < C_{S,\mathcal{KB}}(\psi \& \phi)$.

Theorem 6: $\psi \in KB^{\diamond_f}_{\phi} \Longleftrightarrow \text{Mod}(\phi \& \neg\psi) = \emptyset$ or $C^2_{S,\mathcal{KB}}(\neg\psi \mid \phi) < 1$.

Proofs can be found in [6].

5 Related and Future Work

We have redefined the concepts of update and revision via the similarity-based models for possibilistic and fuzzy reasoning. The revision operator is similar to that given by Dubois and Prade ([2]) to characterize revision in Possibilistic Logic. However, there are two significative differences. The first one is that our revision operator is not commutative, this is, the ordering of change is significative. The second one is about the worlds which are contradictory with the new information. In the Dubois and Prade's approach, those worlds are considered as impossible in the new state of affairs. However, this does not occur in our approach, because we do not change the similarity relation.

The update operator is something different from the one obtained by Dubois and Prade via possibilistic imaging [2], since in our approach, we do not need any external "closest function". Moreover, the characterizetion result of theorem 6 has not an analogous counterpart in Possibilistic Logic.

As a further relationship to be studied between the possibilistic and similarity based approach is the following one. In [3] Esteva, Garcia-Calves and Godo define possibility distributions via similarity functions and subsets of worlds. On the other hand, Dubois and Prade propose an update operator based on a possibility distribution on the set of possible worlds ([2]). An idea is to relate the update operation of our proposal with that of Dubois and Prade defined from the possibility distribution resulting from the mapping proposed by Esteva et al.

As a final remark, let us notice that the symmetry and transitivity properties of the fuzzy similarity relations are not needed to generate the pre-orders for defining revision and update operators. Therefore it seems interesting to generalize our approach allowing more general models than those based on fuzzy similarity relations. Namely we could allow models with a fuzzy binary relation, representing some general notion of "closeness", for which only the reflexivity and discriminant properties would be required.

References

[1] "On the logic of theory change: partial meet contraction and revision function". Alchourrón C., Gärdenfors P. and Makinson D., Journal Symbolic Logic, 50:510-530,1985.

[2] "Belief Revision and Update in numerical formalisms". Dubois D. and Prade H. Proc. of the IJCAI'93, 620-625.

[3] "Relating and Extending Semantical Approaches to Possibilitic Reasoning". F. Esteva, P. Garcia-Calvés and L. Godo. International Journal of Approximate Reasoning 10:311-344, 1994.

[4] "Propositional knowledge base revision and minimal change". H. Katsuno and A. Meldelzon. Artificial Intelligence 52, 263-294. 1991.

[5] "On The difference between update a knowledge base and revising it". Katsuno H. and Mendelzon A., KR'91,387-394.

[6] "Relating Similarity-based models, Counterfactual Conditionals and Theory Change". R. Rodriguez. Research Report IIIA 95/6, Institut d' Investigació en Intel.ligencia Artificial, CSIC. 1995.

[7] "Possibity as similarity: the semantics of fuzzy logic". E. Ruspini. Uncertainty in Artificial Intelligence 6, P.P. Bonissone, M.Henrion, L.N.Kanal, J.F.Lemmer (editors). Elsevier Science Publishers B.V. 271-280,1991.

Diseño de un ATMS Genérico: Casos Multivaluado y Difuso

Juan Luis Castro y Jose Manuel Zurita
Dpto. de Ciencias de la Computación e Inteligencia Artificial
E.T.S de Ingeniería Informatica
Universidad de Granada, 18071 - Granada
e-mail:castro, zurita@robinson.ugr.es

Resumen

El objetivo principal que se persigue en este articulo, es la elaboración de un Sistema de Mantenimiento de la Verdad General basado en el algoritmo de De Kleer [5]. Este tipo de sistema pretende ser diseñado de forma que pueda ser utilizado en distintos modelos lógicos monótonos proposicionales de sistemas de razonamiento. Se describe el sistema de base de conocimiento que interactuará con él y se estudia la eficiencia que presupone el trasladar el ATMS a una lógica con múltiples valores de verdad. Se dan las definiciones y propiedades del ATMS genérico y los algoritmos que éste realiza. Posteriormente, se particulariza el ATMS genérico para que interactue con un sistema de razonamiento basado en lógica multivaluada, concretamente para el caso de las lógicas [0,1]-valuadas, así como con sistemas de razonamiento basados en lógica difusa. Este último caso, se diseñará para razonar con valores de verdad difusos, aunque un tratamiento paralelo podría seguirse utilizando directamente etiquetas linguísticas.

Palabras clave: *Sistema de Mantenimiento de la Verdad, Sistema Basado en el Conocimiento, Resolvedor de Problemas, Lógica Multivaluada, Lógica Difusa.*

1 Introducción

Fue Doyle en 1979 [4] quien introdujo la primera visión formal de un TMS. Las funciones principales que realiza un TMS se pueden resumir en tres. Mantenimiento de las Inferencias no Monótonas: permite el razonamiento no monótono. Resolución de Contradicciones. Módulo de Explicaciones. Algunas de las limitaciones de este sistema, tales como la necesidad de que la base de conocimiento siempre tenga que ser mantenida consistente, o la dificultad que supone el cambio de contexto, lo cual nos impede comprobar dos posibles soluciones del problema, así como la no existencia de una conexión directa con los resultados obtenidos en lógica monótona, fueron resueltos en su mayor parte por De Kleer [5] con el diseño de un nuevo sistema de mantenimiento de la verdad, en este caso, basado en suposiciones (ATMS).

El resolvedor de problemas realiza inferencias e introduce hipótesis en la resolución del problema. Estas inferencias son almacenadas por el ATMS en una estructura interna de nodos en forma de justificaciones. Un entorno es un conjunto de suposiciones. Una suposición puede aparecer en mas de un entorno, por tanto, los entornos pueden subsumir unos en otros. El conjunto de todas las suposiciones definen un retículo de entornos. El contenido de un contexto lo forman las suposiciones de un entorno y todos los hechos deducibles a partir de ellas. Cada nodo ATMS es un triple de la forma (N, L, J). El primer elemento es el nombre del dato utilizado por el r.p. El segundo elemento L es la etiqueta. Por último, el tercer elemento es el conjunto de justificaciones, J, que tienen N como consecuente. La etiqueta de cada nodo ATMS es un conjunto de entornos. El ATMS debe mantener la etiqueta: Consistente: Ningún entorno de la etiqueta genera una inconsistencia. Sólida: Cada entorno de la etiqueta deduce el nodo. Completa: Cualquier entorno que deduzca el nodo debe ser un superconjunto de algún entorno de la etiqueta. Mínima: Ningún entorno es un subconjunto de ningún otro de la etiqueta.

Otros sistemas de mantenimiento de la verdad se han desarrollado con posterioridad, basados en los anteriores: McAllester [10], diseña un TMS basado en justificaciones para lógica proposicional. Falkenhainer [7], aplica la teoría de la evidencia de Dempster-Shafer en el TMS de Doyle. D'Ambrosio [3], utiliza un ATMS para calcular creencias en la teoría de Dempster-Shafer. De Kleer y Williams [6], asignan probabilidades a suposiciones y utilizan un ATMS para detectar múltiples fallos simultaneamente en un sistema. Fringuelli et al. [8], diseña un ATMS para trabajar con factores de certeza, el método de inferencia que utiliza para el resolvedor de problemas está basado en el principio de resolución difuso propuesto por Lee. Meseguer [11], [12], asocia factores de certeza a entornos, si bien no define un sistema de mantenimiento de la verdad, sino que su objetivo es aplicar esta información a la validación de bases de conocimiento multivaluadas. Parte de los mecanismos aquí utilizados para lógica multivaluada coinciden con los que él desarrolla.

Haremos un estudio acerca de los problemas que supone para el ATMS la utilización, por parte del sistema de razonamiento, de una lógica que emplea diferentes valores de verdad en las proposiciones, a la vez que incertidumbre en la deducción de la información; y se definirá un ATMS de propósito general para trabajar con cualquier tipo de lógica monótona proposi-

cional. En [15] se hace un desarrollo más extenso al respecto.

2 Retículo de entornos

Una de las características fundamentales del ATMS, es la posibilidad de establecer un retículo entre todas las agrupaciones de suposiciones. La creación del retículo favorece la eliminación, o la no consideración, de la mayor parte de los entornos, por quedar recogida la información que proporcionan éstos, en el conjunto de entornos que constituyen la etiqueta del nodo.

Si se desea guardar toda la información resultante de las distintas inferencias en el sistema de razonamiento, puede suceder que el espacio de búsqueda que se presente sea prácticamente inmanejable. Por ejemplo, en el caso de tener 5 suposiciones y una lógica que utiliza 10 valores de verdad, el número de entornos a considerar sería de 161.050, frente a los 32 que componen el retículo en la lógica clásica. Sin embargo, si reducimos a dos el número de valores de verdad de la lógica (distintos de los de presencia o ausencia), para el mismo conjunto de suposiciones, el espacio de búsqueda queda drásticamente reducido a 242 entornos. En la Figura 1, podemos observar el reticulo resultante para el caso de dos suposiciones {A, B} y tres estados lógicos distintos $\{v_1, v_2, v_3\}$. Se nos plantea, por tanto, *evaluar las caracteristicas del sistema de razonamiento y las necesidades de funcionamiento del mismo, así como la base de conocimiento específica que se va a utilizar.* Una vez hecho ésto, se presentan dos alternativas:

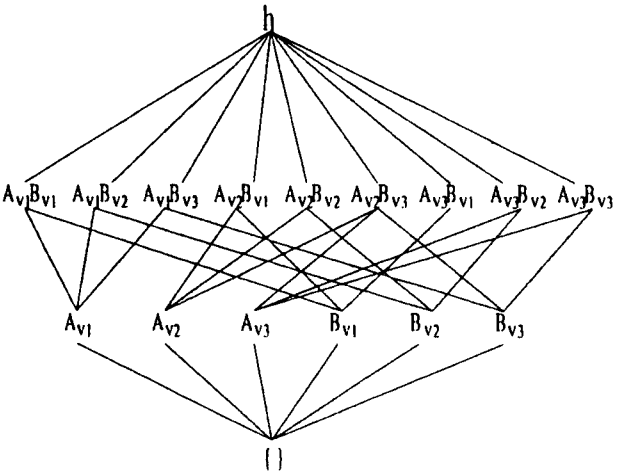


Figura 1: Retículo de entornos para {A,B} y $\{v_1, v_2, v_3\}$.

- Si el análisis efectuado se considera viable, podríamos disponer del conjunto global de entornos en los cuales aparezcan las suposiciones conjuntamente con sus valores de verdad. Por ejemplo, podríamos tener nodos del tipo

$$N_h : ((\ldots), (\{A_{v_1}, B_{v_2}\}, \ldots, \{A_{v_3}, B_{v_3}, D_{v_3}\}), (\ldots)).$$

- O bien, si debido a las dimensiones que adquiere el problema, hace computacionalmente poco recomendable la solución anterior, trabajaríamos siempre a partir del estado actual del sistema, con lo cual, el valor de verdad para cada suposición es el que tiene asignado en ese momento en la base. Por ejemplo, consideraríamos nodos de la forma

$$N_h : ((\ldots), (\{A, B\}, \ldots, \{A, B, D\}), (\ldots)),$$

donde los valores lógicos v_i asignados a cada suposición quedan implícitos por el estado actual de la base de conocimiento.

2.1 Eficiencia

Planteamos por tanto, la posibilidad de trabajar con dos clases de retículos, aquellos en los que cada elemento es identificado por dos parámetros (nombre de las suposiciones y valor de verdad de cada una de ellas), y aquellos en los que se prescinde del valor de verdad de la suposición, entendiéndose por tal, el que tiene actualmente asignado en la base de conocimiento. Calculemos la eficiencia en número de componentes de cada retículo, a los que denominaremos Total y Actual.

Retículo Total

Dadas n suposiciones de partida y una lógica m-valuada, el número de componentes por nivel:

- **En el nivel 1**. Se tienen n suposiciones y m valores de verdad, luego el total de hechos supuestos diferentes es de $n \times m$.
- **En el nivel 2**. Donde se constituyen los entornos que tienen dos suposiciones. Habrá que saber cuantas posibles formas hay de combinar en grupos de dos, las n suposiciones que existen, independientemente de los grados de verdad, es decir, habría C_n^2 entornos con dos suposiciones. Ahora bien, como cada suposición puede tomar m valores lógicos distintos, cada grupo anterior constituye $m \times m$, es decir, m^2 entornos distintos, en ninguno de los cuales aparece una misma suposición con dos valores de verdad.
- **En el nivel 3**. El razonamiento sería exactamente el mismo que en el caso anterior, sólo que las combinaciones serán de tres suposiciones, C_n^3. Siendo $m \times m \times m = m^3$ el total de entornos con tres suposiciones diferentes a que da lugar cada combinación.
- **En el nivel n**. Cuando se tienen entornos con todas las suposiciones, el número C_n^n obviamente vale 1, y la cantidad de entornos que se pueden generar es de m^n.

Siendo por tanto el tamaño total, del orden que describe la expresión, $O(\sum_{i=1}^{n} C_n^i * m^i)$.

Retículo Actual

En este caso, las dos alternativas lógicas que se tienen para una suposición son, la ausencia, y la presencia con el valor de verdad del estado actual. Por tanto, el retículo queda reducido al de la lógica clásica, en cuanto a tamaño. Quedando éste descrito por la

expresión $O(\mathcal{P}(\mathcal{A}))$, siendo $\mathcal{P}$ y $\mathcal{A}$ la función partes de un conjunto y el conjunto de suposiciones, respectivamente. Expresión que se traduce en $O(2^n)$ cuando $|\mathcal{A}| = n$.

3 Definiciones y propiedades

Definición 3.1. Dado un item de información $q \in KBS$, se define el **Nodo ATMS** asociado a q, que notaremos por $\mathbf{N_q}$, al triple de la forma, N_q : ((valores de verdad)(Etiqueta)(Justificaciones)), o $N_q : ((\mathcal{V})(\mathcal{E})(\mathcal{J}))$, donde *valores* son los distintos valores lógicos calculados a partir de los distintos entornos que componen la *etiqueta* por medio de las *justificaciones*. Más especificamente, podemos estructurar un nodo como la expresión

$$N_q : ((v_i, \ldots, v_n)(E_1, \ldots, E_n)(J_1, \ldots, J_p)), \qquad (1)$$

o lo que es lo mismo, $N_q : ((v_i, \ldots, v_n)(P_1, \ldots, P_p)(J_1, \ldots, J_p))$, donde los distintos P serán subconjuntos de entornos para los cuales se verifica $J_s \cup E_i \vdash_k q$ con $v_i(q)$; $\forall E_i \in P_s$, que expresa la deducción de q con una certeza v_i para un estado k del KBS, explicitando de esta forma, a través de P_i, los entornos que subyacen en cada justificación J_i.

El término **Inconsistencia** debe ser especificado en el desarrollo de los casos particulares según el tipo de lógica utilizada.

La etiqueta $\mathcal{E}$ de un nodo N_q se desea **consistente, sólida, completa** y **mínima**, definidas dichas propiedades de la siguiente forma

Consistencia Una etiqueta $\mathcal{E}$ se dice consistente, si

- *Todos sus entornos son consistentes*, es decir $\forall E_i \in \mathcal{E}$; $E_i \cup \mathcal{P} \not\vdash_k \bot$.
- *No existe inconsistencia entre los distintos valores de verdad calculados mediante los entornos que aparecen en la etiqueta*. Es decir $\nexists v_i \ldots v_k \in \mathcal{V}$ t.q. $v_i \wedge \ldots \wedge v_k = \bot$.

Solidez Se verifica esta propiedad, si cada uno de los entornos E_i a través de las justificaciones descritas en la etiqueta, reflejan un camino de inferencias del nodo q con los valores de verdad que aparecen en $\mathcal{V}$, $\forall P_s \forall E_i \in P_s$, $J_s \cup E_i \vdash_k q$ con $v_i(q)$; $v_i(q) \in \mathcal{V}$.

Completitud Con ella se indica que todas las pruebas del dato se hallan presentes en el nodo

$$\forall E \text{ t.q. } E \cup \mathcal{P} \vdash q \text{ con } v_i(q) \Rightarrow$$
$$\exists E_j \in \mathcal{E} \text{ t.q. } Subsume(E, E_j),$$

donde *Subsume* es una relación binaria que se aplica entre conjuntos de entornos.

Definición 3.2. Diremos que el entorno E_i **subsume en el entorno** E_j, que representamos por **Subsume**(E_i, E_j), cuando

$$\mathbf{Subsume}(E_i, E_j) \Leftrightarrow \begin{cases} E_j \subseteq E_i \\ \mathbf{mas_restrictivo}(v_j, v_i) \end{cases}$$

mas_restrictivo será una relación binaria entre el conjunto de los valores de verdad $\mathcal{T}$, es decir, mediante la completitud, aseguramos que en cualquier deducción del dato a partir de un entorno, existirá otro entorno en la etiqueta del nodo que es un subconjunto del anterior, y proporcionará una certeza sobre el dato deducido más precisa (restrictiva) de lo que lo hace el otro entorno. O con otras palabras, con esta propiedad, nos estamos asegurando que no hay *pérdida de información, o información no reflejada, respecto al proceso de deducción del nodo*, ya que toda demostración quedaría reflejada en la etiqueta.

Minimalidad Establece que los caminos de inferencias utilizados en cada una de las pruebas formales son mínimos respecto a la deducción lógica del dato, $\forall E_i, E_j \in \mathcal{E}$, **Not** $Subsume(E_i, E_j)$; $i \neq j$

La eficacia del TMS subyace en el mantenimiento de la consistencia, solidez, completitud y minimalidad de la etiqueta de cada nodo, tras cada inferencia realizada por el solutor del problemas. El algoritmo principal, que activa el ATMS cuando una inferencia es realizada por el r.p., será similar al descrito originariamente por De Kleer en [5], si bien, la estructura de nodo descrita en (1), provoca dos matizes fundamentales que deberán reflejarse:

La dependencia del resolvedor de problemas en la combinación de entornos.

Ejemplo 3.1. Supongamos que en un determinado momento, el PS proporciona al ATMS la siguiente justificación $N_i, N_j, N_k \Rightarrow N_h$, siendo los nodos

$$N_i : ((v_{i1}, v_{i2})(\{A_1, A_2\}\{A_3\})(\ldots))$$
$$N_j : ((v_{j1})(\{A_3, A_4\})(\ldots))$$
$$N_k : ((v_{k1})(\{A_1, A_5\})(\ldots)),$$

el nodo correspondiente al hecho h se creará ó actualizará como

$$N_h : ((v_{h1}, v_{h2})(\{A_1, A_2, A_3, A_4, A_5\}\{A_1, A_3, A_4, A_5\}) (\{N_i, N_j, N_k\}))$$

$$v_{h1} = I(v_{i1}, v_{j1}, v_{k1})$$
$$v_{h2} = I(v_{i2}, v_{j1}, v_{k1}),$$

donde I es el resultado que produce el r.p. al realizar la inferencia que describe la justificación original, pero con los valores de certeza que aparecen entre paréntesis.

El posible caracter gradual que puede presentar la contradicción. Definiremos para ello, $Incon(v_i, v_j)$. Cuando el grado de inconsistencia de un conjunto de suposiciones es $\geq \alpha$; retractar el valor de certeza de alguna de las suposiciones, o eliminar la inconsistencia de las etiquetas de los nodos.

4 Casos Particulares: Lógica [0,1]-valuada y Lógica Difusa

4.1 Razonamiento con Lógica [0,1]-valuda

En [2] [8] [11] [12], se hace un estudio al respecto. Las funciones originarias de los conectivos $\wedge$, $\vee$, son desarrolladas ahora por las llamadas **t-normas (T)** y **t-conormas (S)**, que son operaciones binarias extedidas al intervalo [0,1] con un conjunto de propiedades exigibles. **La relación entre t-normas y t-conormas** viene expresado, como en el caso de la

lógica clásica, a través del **conectivo de negación**. El **conectivo de implicación** es, junto con la regla de inferencia del Modus Ponens, la cuestión fundamental del razonamiento lógico. Trillas y Valverde [13], dan una axiomática de los conectivos de implicación para una lógica multivaludada en [0,1]. **La función generadora de Modus Ponens, m, (respecto a I)**, proporciona una cota inferior para el valor de verdad del consecuente, $m(x, I(x,y)) \leq y$.

La base de conocimiento KB estará constituida por un conjunto de reglas de la forma $f_1 \wedge \ldots \wedge f_p \longrightarrow h$ con $cf = \gamma$; $f_i \in \mathcal{F}$, $h \in \mathcal{H}$, y la base de hechos FB por, f_1 con $cf = \alpha_1, \ldots, f_q$ con $cf = \alpha_q$, siendo cf los factores de certeza pertenecientes al intervalo [0,1], con los que son creidos las reglas y los hechos. Siendo $\mathcal{H} \subseteq \mathcal{F}$ los hechos de salida y $\mathcal{F}$ el conjunto de hechos.

Sea $\mathcal{V}$ el conjunto de todas las valuaciones sobre $\mathcal{F}$, $\mathcal{V} = \{v : \mathcal{F} \longrightarrow [0,1]\}$, se observa que, $|\mathcal{V}| = \infty$, a diferencia del valor $2^{|\mathcal{F}|}$ calculado en la lógica {0,1}. Cada función de evaluación $v \in \mathcal{V}$ hace corresponder a todo $f \in \mathcal{F}$ el valor $cf(f)$ en un determinado momento. Este valor nos indica que la creeencia que poseemos en el hecho f está comprendida en el intervalo $[cf(f), 1]$. Podemos expresar, por tanto, más generalmente, que $v : \mathcal{F} \longrightarrow [0,1] \times [0,1]$, donde $v(f) = \{a,b\}$ significa que $v(f)$ pertenece al intervalo $[a,b] \subseteq [0,1]$. La ignorancia total para f quedará reflejada por $v(f) = \{0,1\}$. Inicialmente,

$$v(h) = \{0,1\}; \quad \forall h \in \mathcal{H}$$
$$v(e) = \{cf(e), 1\}; \quad \forall e \in \mathcal{I}.$$

Recordamos que $cf(f) = a$ iff $cf(\neg f) \leq 1 - a$.

Se generará una inconsistencia cuando se deduzca un $h \in \mathcal{H}$ tal que $v_k(h) = \{a,b\}$ y $a > b$.

La operación *Unión*, aplicada a varios entornos, se define como, $E = \mathit{Unión}\ (E_1, \ldots, E_n) \Leftrightarrow E = E_1 \cup \ldots \cup E_n$, dicha operación queda, por tanto, reducida a la unión definida en la teoría de conjuntos clásicos. La *minimalidad* en la etiqueta se consigue a partir de la operación de *subsumisión*. Diremos que el entorno E_i subsume en el entorno E_j para la etiqueta del nodo N_h, cuando

$$\mathit{Subsume}(E_i, E_j) \Leftrightarrow \begin{cases} E_j \subseteq E_i \\ \textbf{mas_restrictivo}(cf_{E_i}, cf_{E_j}) \end{cases}$$

mas_restrictivo será una relación binaria en el conjunto $\mathcal{T} = [0,1]$, y será definida como, **mas_restrictivo**$(cf_{E_i}, cf_{E_j}) \Leftrightarrow cf_{E_i} \leq cf_{E_j}$.

4.2 Razonamiento con Lógica Fuzzy

En lógica difusa, existen dos grandes alternativas de razonamiento, el uso de etiquetas linguísticas y el uso de valores lógicos de verdad. En el primero de los casos, el ATMS generaría nodos de la forma, $(N_z$ es $H', (\{v\}), (\ldots))$, $(N_z$ es $H'', (\{u,v\}), (\ldots))$, necesitándose por tanto, un retículo de entornos por cada etiqueta diferente de una misma variable de salida. Sin embargo, si utilizamos la equivalencia, z es $H' \equiv z$ es H es τ, tendremos nodos de la forma N_z es $H \quad : \quad ((\tau_a, \tau_b, \tau_c)(\{u\}, \{u,w\}, \{u,v,w\})(\ldots)$, siendo τ_i valores lógicos difusos, en el intervalo $[0,1]$. Baldwin [1], definió un conjunto de funciones para

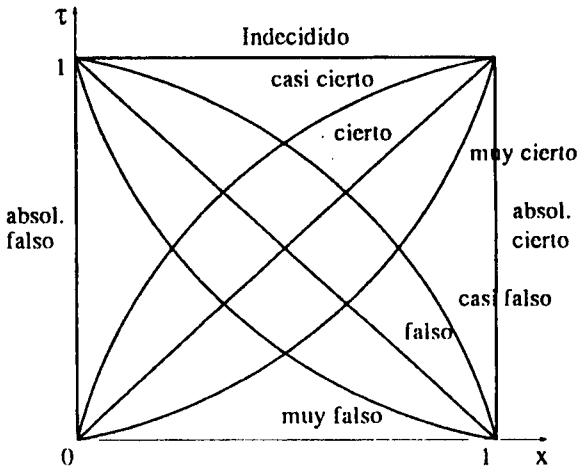


Figura 2: Funciones de Verdad Difusas.

nueve valores de verdad difusos, como se observa en la Figura 2.

Dados τ_i y $\tau_j \in \mathcal{T}$, definiremos la *inconsistencia* entre ambos, que notaremos por $\mathit{Incon}(\tau_i, \tau_j)$ como $1 - \sup_x\{\tau_i(x) \wedge \tau_j(x)\}$. Esta definición detecta los puntos de intersección de las funciones τ_i y τ_j. En la Figura 3 se muestran cuales son éstos, y se enumeran del 1 al 7. Estos siete valores, indican una

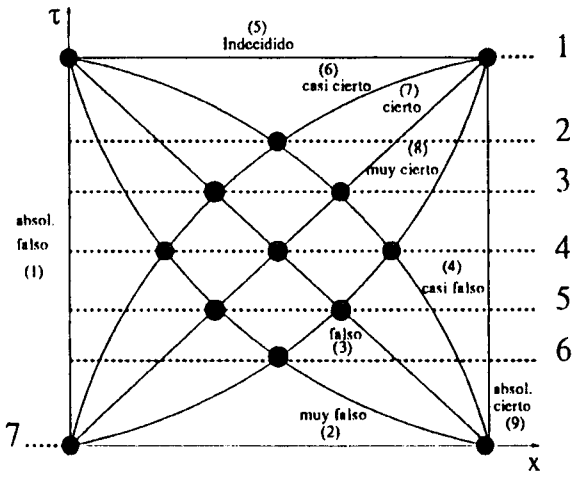


Figura 3: Representación gráfica de $\mathit{Incon}(\tau_i, \tau_j)$.

granularidad en la definición de inconsistencia. Las intersecciones etiquetadas con el número 7, establecen una inconsistencia total entre los pares que ahí concurren. Las intersecciones etiquetadas con el número 1, indican por el contrario, que los valores de verdad que intersectan son totalmente compatibles y consistentes entre sí.

La operación *Unión*, aplicada a varios entornos, se define como, $E = \mathit{Unión}\ (E_{1_\tau}, \ldots, E_{n_\tau}) \Leftrightarrow E = E_{1_\tau} \cup \ldots \cup E_{n_\tau}$, dicha operación queda reducida también a la operación unión definida en la teoría de conjuntos clásicos. La *minimalidad* en la etiqueta se consigue a partir de la operación de *subsumisión*. Diremos que el entorno E_i subsume en el entorno E_j para la etiqueta

del nodo N_h, cuando

$$Subsume(E_i, E_j) \Leftrightarrow \begin{cases} E_j \subseteq E_i \\ \textbf{mas_restrictivo}(\tau_{E_j}, \tau_{E_i}) \end{cases}$$

mas_restrictivo será una relación binaria entre el conjunto de las etiquetas linguísticas de verdad $\mathcal{T}$ y será definida como, $\textbf{mas_restrictivo}(\tau_j, \tau_i) \Leftrightarrow \mu_{\tau_j}(x) \leq \mu_{\tau_i}(x); x \in [0,1]$, ó lo que es lo mismo

$$\textbf{mas_restrictivo}(\tau_j, \tau_i) \Leftrightarrow \begin{cases} j \leq 5, i \leq 5, j \leq i \\ j \geq 5, i \geq 5, j \geq i \end{cases}$$

siendo j y i dos valores enteros que identifican a los nueve valores de verdad difusos representados. Se diferencian dos grupos: *Los valores etiquetados desde el 1 al 4*. Constituido por: *absolutamente falso*, *muy falso*, *falso*, *casi falso*, son valores que denotan cierta falsedad del dato, y *el grupo etiquetado desde el 6 al 9*. Constituido por: *casi cierto*, *cierto*, *muy cierto*, *absolutamente cierto*, son valores que acreditan creencia. En cada uno, es posible que unos valores sean más restrictivos que otros. Sin embargo, no hay restricción posible entre ambos grupos, es decir, no existe relación alguna entre sus valores de verdad. *Esto fortalece la definición dada de ser más restrictivo, ya que no tendría sentido que eliminásemos un entorno por verificar esta relación cuando se aplica a dos valores tan contradictorios como los que aparecen en ambas situaciones.*

5 Conclusiones

En este artículo se ha diseñado un ATMS genérico. Posibilitando, de esta forma, un marco de desarrollo de ATMS, capaz de interactuar con sistemas que utilizan cualquier tipo de lógica monótona proposicional en su razonamiento. Se ha hecho, para ello, un análisis exhaustivo de la eficiencia que requiere el abordar otro tipo de lógica distinta de la $\{0,1\}$ en el diseño del ATMS. Se ha particularizado el ATMS genérico para el caso de la lógica multivaluada con factores de certeza, más concretamente, para la lógica [0,1]-valuada. Se ha dado un criterio para establecer cuando podemos prescindir de un entorno que involucra la deducción de un dato, sin perder información al respecto. También se ha particularizado el ATMS para el caso de la lógica difusa con valores de verdad difusos. Se ha introducido la noción de entorno parcialmente inconsistente, definiéndose la inconsistencia y la subsumisión sobre un conjunto de nueve valores de verdad lógicos.

Referencias

[1] Baldwin J.F. A new approach to approximate reasoning using a fuzzy logic. *Fuzzy Sets and Systems*, **2**, 1979 309-325.

[2] Castro, J.L., Zurita, J.M. A Multivalued Logic ATMS. Submitted to *International Journal of Intelligent Systems*,1995.

[3] D'Ambrosio, B. A hybrid approach to reasoning under uncertainty. *International Journal of Approximate Reasoning*, **2**, (1), 1988, 29-47.

[4] Doyle, J. A truth maintenance system. *Artificial Intelligence*, 12, 1979, 231-272.

[5] De Kleer, J. An assumption-based TMS. *Artificial Intelligence*, 28, 1986, 127-162.

[6] De Kleer, J., Williams, B.C. Diagnosing multiple faults. *Artificial Intelligence*, 32, 1986, 97-130.

[7] Falkenhainer, B. Towards a general-purpose belief maintenance system. En: *Uncertainty in Artificial Inteligence*, **2**, (J.F. Lemmer, L.N. Kanal, eds.) North-Holland, Amsterdam, 1988.

[8] Fringuelli, B., Marcugini, S., Milani, A., Rivoira, S. A reason maintenance system dealing with vague data. *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence* (D. D'Ambrosio, Ph. Smets, P. Bonissone, eds.) Los Angeles, 1991, 111-117.

[9] Laskey, K.B., Lehner, P.E. Assumptions, beliefs and probabilities. *Artificial Intelligence*, **35**, 1988, 25-79.

[10] McAllester, D.A. An outlook on truth maintenance. *Massachusetts Institute of Technology. Artificial Intelligence Laboratory*, AI Memo No. 551, 1980.

[11] Meseguer, P. A new method to checking rule bases for inconsistency: a Petri net approach. *Proceedings of the 9th European Conference on AI (ECAI-90)*, Stockholm, 1990, 437-442.

[12] Meseguer, P. Incremental verification of rule-based expert systems. *Proceedings of the 10th European Conference on Artificial Intelligence*, Vienna, 1992, 840-844.

[13] Trillas, E., Valverde, L. On Implication and Indistinguishability in the setting of Fuzzy Logic. *Management decision support using fuzzy sets and possibility theory* (J. Kacprzyk, R.R. Yager, eds.), Verlag TUV, 1985, 198-212.

[14] L. Zadeh (1983) *A Computational Approach to Fuzzy Quantifiers in Natural Language*, Comps & Maths. with Appls. 9, 149-184.

[15] Zurita, J.M. Diseño de un ATMS Genérico. Aplicaciones al Aprendizaje y a la Validación de Bases de Conocimiento. *PhD. diss.*, 1994, Universidad de Granada. Spain.

Representación de Bases de Conocimiento Borroso sobre Redes de Petri: Operación en el dominio de la verdad

A. Bugarín y S. Barro *
Departamento de Electrónica e Computación.
Facultade de Física.
Universidade de Santiago de Compostela.
Email: elbugari@usc.es

Resumen

En este trabajo se analiza el proceso de realización de inferencias en sistemas basados en reglas borrosas con encadenamiento. La ejecución de una Base de Conocimiento Borroso se entiende como la propagación de los valores lingüísticos de verdad de las proposiciones que la forman, lo cual es posible mediante la realización de las inferencias en el dominio de la verdad y mediante la descripción de un proceso de compactación de las reglas encadenadas. De esta forma se consiguen ventajas importantes desde el punto de vista computacional, como son el tratamiento homogéneo de todas las variables (independencia de los respectivos universos de discurso) y la realización a priori de parte de las operaciones involucradas en el proceso (aquellas relacionadas con los encadenamientos entre proposiciones). La eficiencia computacional se incrementa notablemente mediante una descripción paramétrica del proceso basada en los valores de verdad lingüísticos descritos por Baldwin [1], que permite realizar la ejecución de una Base de Conocimiento Borroso a través de operaciones entre valores numéricos simples, eludiendo en todo momento la manipulación analítica directa de distribuciones de posibilidad. Ello es posible mediante una restricción en cuanto a las posibilidades de encadenamiento, que evita, además, la aparición de situaciones de incertidumbre o indecisión total en cuanto a la inferencia.

La descripción paramétrica obtenida permite la plasmación del proceso sobre un formalismo de representación basado en Redes de Petri que permite describir de forma sencilla diferentes procesos dinámicos que se realizan sobre la Base de Conocimiento Borroso, tales como la ejecución de la misma según distintas estrategias. El modelo de representación propuesto permite describir de forma sencilla los algoritmos que llevan a cabo éste y otros procesos, de modo que resulte más fácil su implementación software y/o hardware.

Palabras clave: *Sistemas Expertos Borrosos, Valores lingüísticos de Verdad, Razonamiento borroso en el dominio de la verdad, Redes de Petri Borrosas.*

*Este trabajo ha sido financiado por la Xunta de Galicia y por la CICyT a través de los proyectos de investigación XUGA20603B94 y TIC95-0604.

1 Introducción

El elemento básico sobre el que se fundamenta la representación borrosa del conocimiento es la variable lingüística [2] o variable cuyos valores son palabras o frases del lenguaje natural. En un nivel superior, las variables lingüísticas se estructuran en proposiciones y, finalmente, en reglas que agrupadas en Bases de Conocimiento Borroso (BCBs) expresan el saber acerca de un dominio de aplicación concreto. En este trabajo abordaremos el problema de la ejecución de BCB compuestas por reglas que presentan encadenamiento entre las variables que las conforman, persiguiendo que este proceso se realice con la mayor eficiencia posible desde el punto de vista de su coste computacional. Para ello analizaremos las expresiones que describen el proceso y, manteniendo en todo momento la mayor flexibilidad posible en cuanto a los operadores involucrados, describiremos un proceso de compactación del conjunto de reglas que permitirá la realización a priori de parte de las operaciones que, de otro modo, recaerían en la etapa de ejecución de la BCB. Este proceso presentará características diferentes y particulares en cada uno de los ámbitos de aplicación más frecuentes de los sistemas basados en reglas borrosas: los sistemas de control borroso y los sistemas expertos borrosos. Centraremos este trabajo en describir un mecanismo de representación basado en el formalismo de Redes de Petri (RP) para el ámbito de los sistemas expertos borrosos, que permitirá la representación de la propia BCB como de diferentes procesos dinámicos que se pueden llevar a cabo sobre ella (por ejemplo, la ejecución de la misma siguiendo distintas estrategias).

2 Esquema de Razonamiento basado en la Regla Composicional de Inferencia "Sup-*".

La formulación de la regla r-ésima R^r de una BCB puede ser como sigue:

$$\begin{array}{c} SI\ X_1^r\ ES\ A_1^r\ y\ \cdots\ y\ X_{M_r}^r\ ES\ A_{M_r}^r \\ ENTONCES \\ X_{M_r+1}^r\ ES\ B_1^r\ y\ \cdots\ y\ X_{M_r+N_r}^r\ ES\ B_{N_r}^r \end{array} \quad (1)$$

donde $X_{m_r}^r$, $m_r=1,\ldots,M_r$ y $X_{M_r+n_r}^r$, $n_r=1,\ldots,N_r$ son variables lingüísticas y $A_{m_r}^r$, m_r, $m_r=1,\ldots,M_r$ y $B_{n_r}^r$,

$n_r=1,...,N_r$ son valores lingüísticos de las variables correspondientes y llevaran asociada una distribución de posibilidad que exprese su significado.

Denotaremos por $\mathrm{A}^r_{m_r}(i)$, $m_r=1,...,M_r$, $r=1,...,R$, $i=1,..,I$, la distribución de posibilidad asociada al valor lingüístico $A^r_{m_r}$ de la variable $X^r_{m_r}$ en la proposición "$X^r_{m_r}$ *ES* $A^r_{m_r}$" y, análogamente, por $\mathrm{B}^r_{n_r}(i)$, $n_r=1,...,N_r$, $r=1,...,R$, $i=1,...,I$, la distribución de posibilidad asociada al valor lingüístico de la variable $X^r_{M_r+n_r}$ en la proposición "$X^r_{M_r+n_r}$ *ES* $B^r_{n_r}$".

Mediante $\tilde{\mathrm{A}}^r_{m_r}(i)$, $m_r=1,...,M_r$, $i=1,...,I$, nos referiremos a cada una de las distribuciones de posibilidad asociadas a los valores lingüísticos observados para las correspondientes variables $X^r_{m_r}$ de la parte antecedente de cada regla y por $\tilde{\mathrm{B}}^r_{n_r}(i), n_r=1,...,N_r$, $i=1,...,I$, cada una de las distribuciones de posibilidad inferidas para las correspondientes variables $X^r_{M_r+n_r}$ de la parte consecuente de cada regla.

Este proceso de ejecución de las reglas se lleva a cabo mediante la Regla Composicional de Inferencia (RCI) "Sup-*", que lo interpreta como la composición de una relación de conjunción que liga los valores observados para las variables de la parte antecedente de la regla y una relación de implicación inducida por la existencia de la regla. Para evitar que el procedimiento de inferencia sea estable para razonamientos encadenados [3], [4] se ha propuesto sustituir el operador t-norma "*" genérico de la RCI por la **Función Generadora de Modus Ponens** [5] correspondiente a la relación de implicación escogida.

2.1 Realización de inferencias en el dominio de la verdad

El proceso de inferencia en una regla R^r de una BCB puede realizarse de forma totalmente independiente de los universos de discurso en que se definen las variables si se lleva a cabo en el **dominio de la verdad** [1] ,[6]. Desde un punto de vista cuantitativo, en esta aproximación (que se deriva directamente de la RCI "Sup-*") se calcula un Grado de Verificación Lingüístico (GDVL) para cada proposición en la parte antecedente de R^r, definido como

$$\tau_{A^r_{m_r}}(a) = \bigvee_{i/A^r_{m_r}(i)=a} \tilde{\mathrm{A}}^r_{m_r}(i),\ a \in [0,1] \qquad (2)$$

y cuyo significado es un Valor Lingüístico de Verdad (VLV) que cuantifica el grado en que se cumple la proposición correspondiente. En el caso de reglas del tipo Múltiple-Antecedente es posible combinar en un único GDVL global para la parte antecedente de la regla los GDVL de cada proposición de dicha parte antecedente (proceso de conjunción lingüística). Si denotamos este GDVL por τ^r_σ, su definición puede expresarse [6] como

$$\tau^r_\sigma(a) = \bigvee_{a=\sigma(a_1,...,a_{M_r})} \sigma\left(\tau_{A^r_1}(a_1),...,\tau_{A^r_{M_r}}(a_{M_r})\right)$$
$$a_{m_r} \in [0,1],\ \forall m_r = 1,...,M_r,\ a \in [0,1] \quad (3)$$

donde "σ" es el operador de conjunción.

El proceso de inferencia se describe en el dominio de la verdad a través de la Función de Razonamiento Hacia Adelante (FRHA) τ_{B^r}, definida como

$$\tau_{B^r}(b) = \bigvee_{a\in[0,1]} \tau^r_\sigma(a) * \rho(a,b),\ b \in [0,1] \qquad (4)$$

donde "ρ" es la relación de implicación y "*" el operador descrito en la RCI "Sup-*" (t-norma o función generadora de modus ponens).

La FRHA se compone con las distribuciones de posibilidad $B^r_{n_r}(i), n_r=1,...,N_r$, $i=1,...,I$, asociadas a los valores lingüísticos de igual nombre en la parte consecuente de las reglas, para dar lugar a la distribución de posibilidad inferida

$$\tilde{\mathrm{B}}^r_{n_r}(i) = \tau_{B^r}\left(B^r_{n_r}(i)\right) \qquad (5)$$

En caso de que la regla se encuentre cualificada por un VLV τ^r, tendremos

$$\tilde{\mathrm{B}}^r_{n_r}(i) = \tau^r\left(\tau_{B^r}(B^r_{n_r}(i))\right) \qquad (6)$$

De esta forma se modifica la distribución de posibilidad inferida, dando lugar a valores más específicos cuando la cualificación sea más restrictiva que CIERTO (MUY CIERTO, ABSOLUTAMENTE CIERTO, ...) y valores menos específicos cuando la cualificación sea menos restrictiva que CIERTO (MÁS O MENOS CIERTO, ...). Tomaremos como definición de VLV la de Baldwin [1], que utiliza funciones polinómicas

$$\begin{aligned} \tau:\ [0,1] &\longrightarrow [0,1] \\ x &\longrightarrow \tau(x) = x^q,\ q \in \Re^+ \end{aligned} \qquad (7)$$

que pueden ser descritas de forma paramétrica, ordenada y unívoca a través del exponente q. Además puede definirse siempre el VLV inverso de un VLV dado mediante el exponente $1/q$.

Vamos a pasar a considerar seguidamente la situación de encadenamiento entre reglas, con objeto de describir analíticamente el proceso de compactación que hemos citado inicialmente.

3 Sistemas Expertos Borrosos con Encadenamiento entre Reglas

Consideraremos un ejemplo que nos permitirá discutir la situación con total generalidad. Trataremos con una BCB constituida por un conjunto de S reglas que realizan inferencias sobre una misma variable intermedia y una regla R^T que la incluye en su parte antecedente:

$$\begin{aligned} &SI\ X^1_1\ ES\ A^1_1\ Y...\ ENTONCES\ \mathbf{X^1_{M_1+1}}\ \mathbf{ES}\ \mathbf{B^1_1}\ Y... \\ &\qquad\qquad \vdots \\ &SI\ X^S_1\ ES\ A^S_1\ Y...\ ENTONCES\ \mathbf{X^S_{M_S+1}}\ \mathbf{ES}\ \mathbf{B^S_1}\ Y... \\ &SI\ \mathbf{X^T_1}\ \mathbf{ES}\ \mathbf{A^T_1}\ Y...\ ENTONCES\ X^T_{M_T+1}\ ES\ B^T_1\ Y... \end{aligned} \qquad (8)$$

con

$$X^S_{M_S+1} = X^1_1 = X^T_1 \triangleq X \tag{9}$$

Caracterizaremos esta situación como de encadenamiento, puesto que la distribución de posibilidad asociada a la variable X^T_1 se obtendrá mediante la agregación de las inferencias realizadas con anterioridad en el conjunto de S reglas, y no por observación externa.

A diferencia de los sistemas de control [8], [7] en SEB el operador de agregación deberá seleccionar la distribución de posibilidad más específica, de modo que se esté considerando en todo momento aquella de las inferencias que proporcione más información. Esto se logra, por ejemplo, utilizando como operador de agregación la t-norma "mínimo" ("$\wedge$"). Teniendo en cuenta que la distribución de posibilidad asociada a la variable X^T_1 va a ser obtenida como la agregación de las distribuciones de posibilidad inferidas en las reglas R^s, s=1,...,S, el GDVL de la proposición "X^T_1 ES A^T_1" puede obtenerse como

$$\tau_{A^T_1}(a) = \bigvee_{i/A^T_1(i)=a} \bigwedge_{s=1}^{S} \tau^s\left(\tau_{B^s}\left(B^s_1(i)\right)\right),\ a \in [0,1] \tag{10}$$

Sin embargo, cuando el proceso de inferencia en una BCB involucra operadores de implicación formales pueden producirse situaciones de total incertidumbre [3]. Este inconveniente puede evitarse si se impone una restricción que exija que coincidan núcleo y soporte de todas las distribuciones de posibilidad referidas a una misma variable. Analíticamente, dado que B^s_1, $s = 1,...,S$ y A^T_1 son valores referidos a la misma variable (expresión 9) exigiremos que

$$\begin{aligned} B^s_1(i) &= \tau^s_{M_s+1}(A(i)),\ s = 1,...,S \\ A^T_1(i) &= \tau^T_1(A(i)) \end{aligned} \tag{11}$$

esto es, que sean modificaciones lingüísticas de un mismo valor lingüístico A asociado a la variable. La Figura 1 muestra un ejemplo gráfico de familias de valores lingüísticos que cumplen la condición exigida.

Por tanto, teniendo en cuenta esta restricción podemos poner que

$$\tau_{A^T_1}(a) = \bigwedge_{s=1}^{S} \tau^s\left(\tau_{B^s}\left(\mu_{B^s_1,A^T_1}(a)\right)\right),\ a \in [0,1] \tag{12}$$

donde se definen los **Valores de Compactación Lingüística** $\mu_{B^s_1,A^T_1}(a)$, $a \in [0,1]$ como

$$\mu_{B^s_1,A^T_1}(a) = \tau^s_{M_s+1}((\tau^T_1)^{-1}(a)),\ a \in [0,1] \tag{13}$$

y pueden ser calculados en el momento en que se define la BCB.

La Expresión 12 demuestra que el proceso de ejecución de una BCB con encadenamiento entre reglas en el ámbito de los SEB puede realizarse sin salirse en ningún momento del dominio de la verdad y entenderlo como propagación de los GDVL de las proposiciones que forman las reglas. Demuestra igualmente cómo el proceso puede realizarse utilizando un método de compactación que traslada parte de la carga computacional desde la etapa de ejecución al momento de definición de la base de conocimiento. En la Figura 2 pueden apreciarse estas circunstancias sobre un ejemplo gráfico.

Puede verse [7] que el cálculo de los GDVL de las proposiciones encadenadas (expresión 12), y de los valores de compactación lingüística (expresión 13) son siempre operaciones cerradas para la parametrización de Baldwin, mientras que el cálculo de la conjunción lingüística (expresión 3) y de la FRIIA (expresión 4) lo es bajo ciertas condiciones, tal y como se expone en las Tablas 1 y 2. Adicionalmente, el proceso de agregación final (asignación de una distribución de posibilidad a cada variable) es también un proceso cerrado para la parametrización considerada. Por tanto, tenemos que para un amplio número de combinaciones de operadores es posible llevar a cabo el proceso de inferencia realizando únicamente operaciones simples entre las representaciones paramétricas de los VLV.

Seguidamente describiremos en el formalismo de Redes de Petri el proceso de ejecución de una BCB que acabamos de caracterizar. El modelo dinámico que propondremos redefine algunos de los elementos descritos en el modelo de sistemas de control borroso [8] y facilita, al igual que en aquel caso, la descripción de distintos algoritmos que llevan a cabo diferentes procesos sobre la BCB.

4 Formalismo basado en Redes de Petri para la Representación de BCB con encadenamiento

El formalismo de representación que utilizaremos se basa en la definición clásica de Red de Petri (RP) como la terna

$$RP = (L, T, A)$$

donde $T = t^j$, $j = 1,...,J$ es un conjunto finito de **transiciones**, $L = l_k$, $k = 1,...,K$, es un conjunto finito de **lugares** y $A \subseteq \{T \times L\} \cup \{L \times T\}$ es un conjunto de **arcos dirigidos**.

El significado de los lugares se describe mediante una aplicación biyectiva α, que asigna a cada proposición de la BCB un lugar, y sólo uno, que constituirá su representación gráfica.

Se define el conjunto de transiciones T como

$$T = T^R \cup T^C = \{t^1,...,t^R,t^{R+1},...,t^{R+C}\}$$

donde el subconjunto $T^R = \{t^1,...,t^R\}$ agrupa a las transiciones asociadas a cada una de las reglas que conforman la BCB, mientras que el subconjunto $T^C = \{t^{R+1},...,t^{R+C}\}$ agrupa a las transiciones que en nuestra representación se asocian a la existencia de encadenamiento entre reglas. Sobre el conjunto T se definen la **aplicación de entrada** $E: T \longrightarrow \wp(L)$ y la **aplicación de salida** $S: T \longrightarrow \wp(L)$, cuyo significado difiere según el subconjunto de T en el que se consideren. Así,

- Si $t^j \in T^R$, entonces, $\forall l_i \in L$,

$$l_i \in E(t^j) \Leftrightarrow \alpha(l_i) \subset ANTECEDENTE\ DE\ R^j$$

y, de forma análoga, $\forall l_i \in L$,

$$l_i \in S(t^j) \Leftrightarrow \alpha(l_i) \subset CONSECUENTE\ DE\ R^j$$

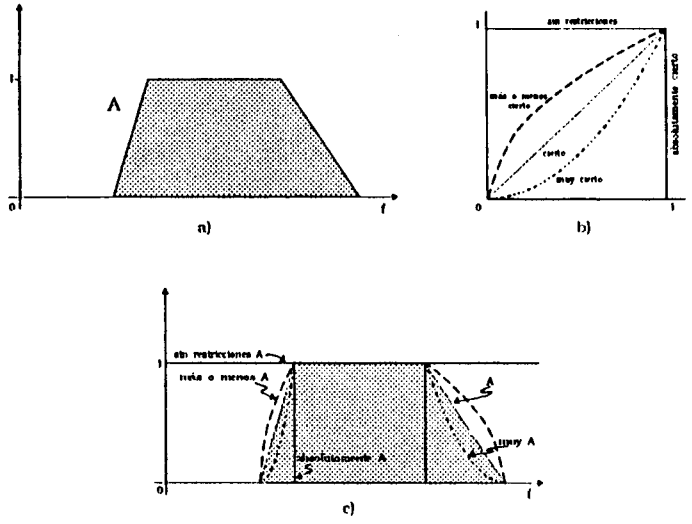


Figura 1: Restricción en encadenamiento. Cálculo de distintos valores lingüísticos (c) a partir de la composición de valores de verdad (b) y una distribución de posibilidad A (a).

IMPLICACIÓN	DEFINICIÓN	q_{R^σ}
Lukasiewicz	$R_a(a,b) = 1 \wedge (1-a+b)$	q_σ *si* $q_\sigma \geq 1$ *no cerrada si* $q_\sigma < 1$
Mizumoto	$R_*(a,b) = 1 - a + a \cdot b$	
Booleana	$R_b(a,b) = (1-a) \vee b$	
Goguen	$R_\Delta(a,b) = \begin{cases} 1 & si\ a \leq b \\ b/a & si\ a > b \end{cases}$	$1 \wedge q_\sigma$
Gödel	$R_g(a,b) = \begin{cases} 1 & si\ a \leq b \\ b & si\ a > b \end{cases}$	

Tabla 1: Cierre del cálculo de la FRHA para algunas R y S-implicaciones. Se ha supuesto como operador "*" la función Generadora de Modus Ponens correspondiente y el GDVL global de la regla descrito como q_σ.

- Si $t^j \in T^C$,

$$l_i \in E(t^j),\ l_k \in S(t^j) \Leftrightarrow$$
$$\Leftrightarrow \alpha(l_i)\ ENCADENADA\ CON\ \alpha(l_k)$$

Dado que cada transición $t^r \in T^R$ se asocia a la existencia de una regla en la BC, y, dado que asumimos que las reglas se encuentran cualificadas por un valor lingüístico de la variable verdad τ^r, resultará de utilidad disponer, como veremos, de una función que asigne a cada transición de este tipo el valor de verdad que tenga asociada la regla en cuestión. Esta función, que denominaremos **función de verdad**, denotada por f, se define como

$$\begin{aligned} f:\ T^R &\longrightarrow V \\ t^r &\longrightarrow f(t^r) = \tau^r \end{aligned}$$

donde V representa el conjunto de los valores lingüísticos de la variable verdad. Además resulta conveniente, por claridad a la hora de describir procesos dinámicos sobre la BCB realizar una clasificación de los lugares asociados a las proposiciones que la forman: Un lugar $l_i \in L$ se clasifica como **lugar de entrada** si $\forall t^j \in T,\ l_i \notin S(t^j)$, como **lugar de salida** si $\forall t^j \in T,\ l_i \notin E(t^j)$. Cualquier otro lugar se denominará **lugar intermedio**. Igualmente, clasificaremos las variables descritas en la BCB como **variables de entrada, intermedias** y de **salida**, dependiendo del tipo de lugar al que estén asociadas las proposiciones a las que aparecen vinculadas.

Mediante el modelo y las definiciones que acabamos de presentar es posible describir cualquier BCB con encadenamiento a nivel de variables entre las reglas que la componen. La representación de procesos dinámicos como la ejecución de la BCB requiere asignar un significado a elementos clásicos como la **función de marcación** y definir adicionalmente nuevos elementos. En el siguiente apartado plantearemos el modelo de RP dinámica correspondiente al ámbito de los SEB, que utiliza como soporte básico el modelo estático que acabamos de definir.

OPERACIÓN CERRADA: $q_\sigma = q_1 \wedge q_2$		OPERACIÓN NO CERRADA	
T-NORMA	DEFINICIÓN	T-NORMA	DEFINICIÓN
Mínimo	$t(a,b) = a \wedge b$	Yager	$t(a,b) = 0 \vee \left[(1-((1-a)^\lambda + (1-b)^\lambda)^{1/\lambda}\right]$
Producto	$t(a,b) = a \cdot b$	Yu Yandong	$t(a,b) = 0 \vee [(1+\lambda)\cdot(a+b-1) - \lambda \cdot a \cdot b]$
t_5	$t(a,b) = \frac{a\cdot b}{a+b-a\cdot b}$	Weber	$t(a,b) = 0 \vee \frac{a+b-1+\lambda\cdot a\cdot b}{1+\lambda}$
Producto Drástico	$t(a,b) = \begin{cases} a & si\ b=1 \\ b & si\ a=1 \\ 0 & en\ otro\ caso \end{cases}$	Producto Acotado	$t(a,b) = 0 \vee (a+b-1)$
Dombi	$t(a,b) = \frac{1}{1+\left(\left(\frac{1}{a}-1\right)^\lambda + \left(\frac{1}{b}-1\right)^\lambda\right)^{1/\lambda}}$		
Dubois	$t(a,b) = \frac{a\cdot b}{a\vee(b\vee\lambda)}$		
Hamacher	$t(a,b) = \frac{\lambda\cdot a\cdot b}{1-(1-\lambda)\cdot(a+b-a\cdot b)}$		

Tabla 2: Cierre del proceso de conjunción lingüística para los operadores t-norma más frecuentemente descritos en la bibliografía. Se ha considerado el problema elemental de conjunción de dos proposiciones cuyos GDVL se describen paramétricamente por q_1 y q_2.

4.1 Representación dinámica en el formalismo RP

Únicamente describiremos de forma explícita en este apartado aquellos elementos que presentan significado distinto o que son adicionales al formalismo clásico de RP. Realizaremos una primera descripción cualitativa de los mismos para pasar a continuación a analizar el proceso de ejecución de BCB bajo el prisma de la herramienta formal que acabamos de definir.

Una marca asociada a un lugar mediante la Marcación indicará que es conocido el GDVL de la proposición asociada a dicho lugar y que, por tanto, dicha proposición puede ser utilizada en la realización de nuevas inferencias. Dado que la marcación es modificada con la evolución dinámica de la red y resulta útil disponer en todo momento de los GDVL de las distintas proposiciones, definimos la aplicación de compatibilidad g

$$g : L \longrightarrow \Re^+$$

que asigna a cada lugar un valor real positivo que cuantifica el parámetro que describe el GDVL de la proposición asociada a dicho lugar. Por su parte, el disparo de una transición activa $t^j \in T$ tiene en este formalismo la siguiente equivalencia con el proceso de ejecución de la BCB:

•Si $t^j \in T^R$, el disparo de t^j equivale a la ejecución de la regla R^j, lo que permitirá calcular el GDVL global para la parte antecedente de R^j.

•Si $t^j \in T^C$, el hecho de que t^j se encuentre activada equivale a que se han realizado con anterioridad inferencias sobre la variable intermedia correspondiente y que es posible, por tanto, aplicar el mecanismo de compactación parametrizada que se ha descrito anteriormente. Ello nos permitirá conocer directamente los GDVL de las proposiciones descritas en la parte antecedente de otras reglas y que involucren a la misma variable intermedia.

La obtención de los GDVL correspondientes a las proposiciones asociadas a lugares de entrada puede describirse de la siguiente manera: cada variable de entrada X establece una clase de equivalencia L_X sobre el conjunto de lugares de entrada LE, definida como

$$L_X = \{l^r_{m_r} \in LE \ / \ a(l^r_{m_r}) = \text{"}X\ ES\ \tau^r_{m_r}(A)\text{"}\}$$

que agrupa a los lugares asociados a proposiciones que involucran a X. Por tanto, en el cálculo de los GDVL de las proposiciones asociadas a los lugares de la clase L_X intervendrá una única distribución de posibilidad observada, que supondremos descrita por el parámetro q_X. Teniendo en cuenta la restricción de encadenamiento, tendremos que

$$g(l^r_{m_r}) = \frac{q_X}{q^r_{m_r}},\ \forall l^r_{m_r} \in L_X$$

donde se ha supuesto que $A^r_{m_r} = \tau^r_{m_r}(A)$ viene descrito por el parámetro $q^r_{m_r} \in \Re^+$.

R^1: SI X_1^1 ES A_1^1 ENTONCES X ES B_1^1 (τ^1)
R^2: SI X_1^2 ES A_1^2 ENTONCES X ES B_1^2 (τ^2)
R^3: SI X ES A_1^3 ENTONCES X_2^3 ES B_1^3 (τ^3)

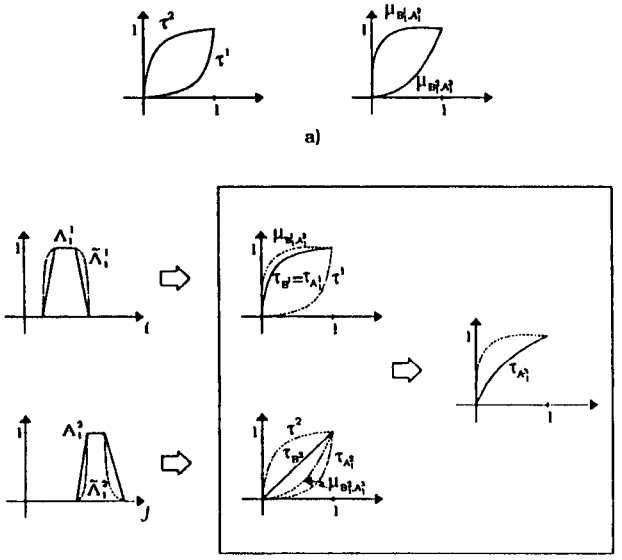


Figura 2: Proceso de inferencia en una BCB en el dominio de la verdad utilizando la implicación de Gödel. a) Información disponible a priori. b) Inferencia en el dominio de la verdad y cálculo del GDVL asociado a la proposición "X ES A_1^3".

Por su parte, el proceso de conjunción lingüística puede describirse directamente mediante la expresión

$$g(l_{n_s}^s) = \bigwedge_{m_s=1}^{M_s} g(l_{m_s}^s) \qquad (14)$$
$$n_s = M_s + 1, ..., M_s + N_s,\ s = 1, ..., S$$

Si suponemos que la FRIIA τ_{B^*} se describe mediante el paramétro $q_{B^*} \in \Re^+$, el GDVL de la proposición "X_1^T ES A_1^T" (expresión 12) puede describirse en la notación introducida por el formalismo de RP como

$$g(l_1^T) = \bigvee_{s=1}^{S} q_{B^*} \cdot q_{\mu_{l^*_{M_s+1}, l_1^T}}$$

donde hemos redefinido la notación del valor lingüístico de compactación de la siguiente manera:

$$\mu_{l^*_{M_s+1}, l_1^T} = \mu_{\tau^*(B_1^s), A_1^T}$$

y lo hemos supuesto parametrizado a través de $q_{\mu_{l^*_{M_s+1}, l_1^T}} \in \Re^+$.

Finalmente, el proceso de agregación para cada variable de salida puede describirse de la siguiente manera: Cada variable de salida X establece una clase de equivalencia L_X sobre el conjunto de lugares de salida LS definida como

$$L_X = \{l_{n_r}^r \in LS \ / \ \alpha(l_{n_r}^r) = "X\ ES\ \tau_{n_r}^r(A)"\}$$

que agrupa a los lugares asociados a las proposiciones que involucran a X y donde $\tau_{n_r}^r(A) = B_{n_r - M_r}^r$, $n_r \in M_r + 1, ..., M_r + N_r$. Si denotamos por q_X el parámetro que describe la distribución final correspondiente a la clase L_X, podemos poner

$$q_X = \bigvee_{l_{n_r}^r \in L_X} q_r \cdot q_{B^r} \cdot q_{n_r}^r$$

donde hemos utilizado la notación habitual para describir de forma paramétrica los VLV de cualificación de las reglas, la FRIIA y los VLV asociados a las proposiciones involucradas en el proceso.

Estas expresiones proporcionan la base teórica necesaria para poder describir los algoritmos de ejecución de una BCB siguiendo distintas estrategias, pues permiten interpretar el proceso de ejecución como la propagación de los parámetros que simbolizan los GDVL de las proposiciones que componen las reglas. En el siguiente apartado nos centraremos en la estrategia de ejecución Dirigida por Datos (DD), aunque la riqueza representativa y la flexibilidad del modelo propuesto permiten también la implementación de otro tipo de estrategias (considerando distintas situacione en cuanto a la información disponible [8], estrategia dirigida por metas [9]) e incluso otro tipo de procesos distintos de la ejecución de la BCB (detección de ciclos y lazos [10]).

4.2 Algoritmo de ejecución según la estrategia DD

La Figura 3 presenta un diagrama del proceso de ejecución de una BCB siguiendo la estrategia Dirigida por Datos. Las flechas sencillas que unen las diferentes etapas simbolizan que las operaciones involucradas se llevan a cabo entre valores numéricos simples parametrizados, a diferencia de las etapas inicial y final (flecha gruesa), en las que intervienen distribuciones de posibilidad

Supondremos conocida la representación RP de la BCB, que servirá como soporte representativo del algoritmo de ejecución.

PASO 1: Cálculo de los GDVL para las proposiciones de entrada.

Sea $l_{m_r}^r \in LE$. Entonces $X_{m_r}^r$ es la variable de entrada tal que $\alpha(l_{m_r}^r) = "X_{m_r}^r\ ES\ \tau_{m_r}^r(A)"$, $L_{X_{m_r}^r}$ es su clase de equivalencia asociada y $\tilde{A}_{X_{m_r}^r} = \tau_{X_{m_r}^r}(A)$ el valor observado para la variable de la clase. Entonces repetir $\forall l_{m_r}^r \in LE$, $\forall\ l_k^r \in L_{X_{m_r}^r}$

$$g(l_k^r) = \frac{q_{X_{m_r}^r}}{q_k^r}$$
$$M(l_k^r) = 1$$

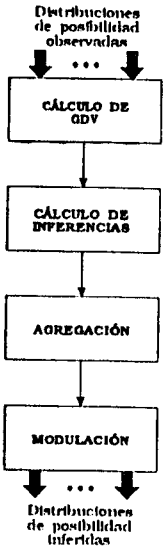


Figura 3: Diagrama de bloques de la ejecución de una BCB siguiendo la estrategia Dirigida por Datos.

PASO 2: Se realiza el disparo de las transiciones activas.

Seleccionar una transición activa cualquiera t^j; es decir,

$$t^j \in T \;/\; \forall l_k \in E(t^j),\ M(l_k) = 1$$

La nueva marcación M' se obtiene a partir de la actual M mediante

$$M' = ft(M, t^j)$$

donde la función de transición ft se define como

$$\begin{aligned} ft : M \times T &\longrightarrow M \\ (M,t) &\longrightarrow ft(M,t) = M' \end{aligned}$$

con

$$M'(l_i) = \begin{cases} 0 & si\ l_i \in E(t) \\ 1 & si\ l_i \in S(t) \\ M(l_i) & en\ otro\ caso \end{cases}$$

Se obtienen los GDVL correspondientes:

- Si $t^j \in T^R$,

$$g(l^j_{n_j}) = \bigwedge_{l^j_{m_j} \in E(t^j)} g(l^j_{m_j})\ \forall l^j_{n_j} \in S(t^j)$$

- Si $t^j \in T^C$,

$$g(l^r_{m_r}) = \bigvee_{l^r_{m_r} \in E(t^j)} \left[q_{B^r} \cdot q_{\mu_{l^r_{m_r}, l^r_{m_r}}} \right],\ \forall l^r_{m_r} \in S(t^j)$$

PASO 3:

Hacer $M = M'$.

Si

$$\exists t^j \in T \;/\; M(l_k) = 1\ \forall l_k \in E(t^j)$$

es decir, si existen transiciones activas, volver al PASO 2

PASO 4: Agregación.

Sea $l^r_{n_r} \in LS$. Entonces $X^r_{n_r}$ es la variable de salida tal que $\alpha(l^r_{n_r})$ = "$X^r_{n_r}$ ES $\tau^r_{n_r}(A)$", $L_{X^r_{n_r}}$ es su clase de equivalencia asociada y $\tilde{B}_{X^r_{n_r}} = \tau_{X^r_{n_r}}(A)$ la distribución inferida para la variable de la clase. Entonces repetir $\forall l^r_{n_r} \in LS$

$$q_{X^r_{n_r}} = \bigvee_{l^r_k \in L_{X^r_{n_r}}} q_r \cdot q_{B^r} \cdot q_{k_r}$$

Ejemplo 1: Supongamos que una BCB está formada por el conjunto de reglas siguiente:

$SI\ X_1\ ES\ A^1_1\ y\ X_6\ ES\ A^1_2\ ENTONCES\ X_2\ ES\ B^1_1\ (\tau^1)$
$SI\ X_1\ ES\ A^2_1\ y\ X_3\ ES\ A^2_2\ ENTONCES\ X_2\ ES\ B^2_1\ (\tau^2)$
$SI\ X_2\ ES\ A^3_1\ y\ X_5\ ES\ A^3_2\ ENTONCES\ X_4\ ES\ B^3_1\ (\tau^3)$
$SI\ X_2\ ES\ A^4_1\ y\ X_5\ ES\ A^4_2\ ENTONCES\ X_4\ ES\ B^4_1\ (\tau^4)$

Supondremos, además, que $\tau^1(x) = x^2$, $\tau^2(x) = x^{1/2}$, y $\tau^3(x) = \tau^4(x) = x$.

Tomaremos como conjuntos de lugares L y de transiciones T los siguientes:

$$\begin{aligned} L &= \{l^1_1, l^1_2, l^1_3, l^2_1, l^2_2, l^3_3, l^3_1, l^3_2, l^3_3, l^4_1, l^4_2, l^4_3\} \\ T &= T^R \cup T^C = \{t^1, t^2, t^3, t^4\} \cup \{t^5\} \end{aligned}$$

La aplicación α, descrita en notación compacta, será

$$\begin{aligned} \alpha :\ L &\longrightarrow P \\ l^r_k &\longrightarrow \alpha(l^r_k) = \begin{cases} X^r_k\ ES\ A^r_k\ si\ k = 1,2 \\ X^r_3\ ES\ B^r_1\ si\ k = 3 \end{cases} \end{aligned}$$

donde X^r_k, k=1,2,3, r=1,...,4, será, en cada caso, la variable de valor lingüístico A^r_k. Supondremos una marcación inicial donde son conocidos los GDV: $g(l^1_1)$=0.5, $g(l^1_2)$=2, $g(l^2_1)$=4, $g(l^2_2)$=2, $g(l^3_2)$=2 y $g(l^4_2)$=1, correspondientes a todas las variables de entrada X_1, X_3 y X_5, tal y como se muestra en la Figura 4. Además, suponiendo que $\mu_{l^1_3, l^1_1}$=1, $\mu_{l^1_3, l^4_1}$=0.9, $\mu_{l^2_3, l^2_1}$=2 y $\mu_{l^2_3, l^4_1}$=0.2, la ejecución de la BCB discurre como muestra la Tabla 3 para la relación de implicación de Gödel y la t-norma de conjunción "mínimo".

5 Conclusiones

En este trabajo se ha analizado el proceso de realización de inferencias en sistemas basados en reglas borrosas con encadenamiento. La ejecución de una Base de Conocimiento Borroso se entiende como la propagación de los valores lingüísticos de verdad de las proposiciones que la forman, lo cual es posible mediante la realización de las inferencias en el dominio de la verdad y mediante la descripción de un proceso de compactación de las reglas encadenadas. De esta forma se consiguen ventajas importantes desde el punto de vista computacional, como son el tratamiento homogéneo de todas las variables (independencia de los respectivos universos de discurso) y la

MARCACIÓN $l_1^1 l_2^1 l_3^1\ l_1^2 l_2^2 l_3^2\ l_1^3 l_2^3 l_3^3\ l_1^4 l_2^4 l_3^4$	TRANSICIÓN ACTIVA	TRANSICIÓN DISPARADA	GDVL
110 110 010 010	t^1, t^2	t^1, t^2	$g(l_3^1) = 0.5$ $g(l_3^2) = 2$
001 001 010 010	t^5	t^5	$g(l_1^3) = 8$ $g(l_1^4) = 0.8$
000 000 110 110	t^3, t^4	t^3, t^4	$g(l_3^3) = 2$ $g(l_3^4) = 0.8$
000 000 001 001			

Tabla 3: Aplicacion del algoritmo de ejecución siguiendo la estrategia dirigida por datos a la BCB del Ejemplo 1

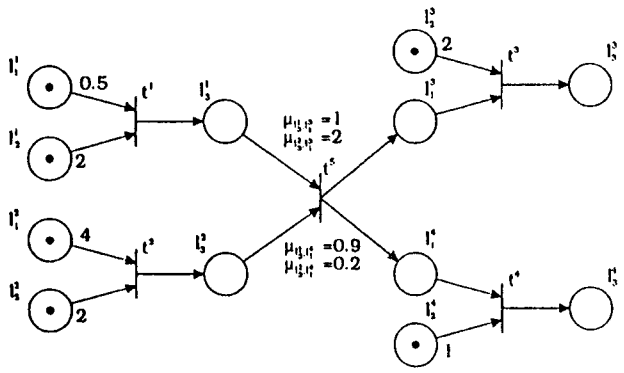


Figura 4: Representación gráfica de la BCB del Ejemplo 1 en el formalismo RP. Las etiquetas numéricas indican el valor parametrizado del GDVL correspondiente.

realización a priori de parte de las operaciones involucradas en el proceso (aquellas relacionadas con los encadenamientos entre proposiciones). La eficiencia computacional se incrementa notablemente mediante una descripción paramétrica del proceso, que permite realizar la ejecución de una Base de Conocimiento Borroso a través de operaciones entre valores numéricos simples, eludiendo en todo momento la manipulación analítica directa de distribuciones de posibilidad. La descripción paramétrica obtenida permite la plasmación del proceso sobre un formalismo de representación basado en Redes de Petri que permite describir de forma sencilla diferentes procesos dinámicos que se realizan sobre la Base de Conocimiento Borroso, tales como la ejecución de la misma según distintas estrategias, derivándose soluciones atractivas incluso para una posible implementación *hardware* del proceso.

Referencias

[1] Baldwin, J.F. A New Approach to Approximate Reasoning using a Fuzzy Logic, *Fuzzy Sets and Syst.*, vol. 2, no. 2, pp. 309-325, 1979.

[2] Zadeh, L.A.,Outline of a New Approach to the Analysis of Complex Systems and Decission Processes, *IEEE Trans. on Syst., Man and Cybernetics*, vol. 3, no. 1, pp. 28-44, 1973.

[3] López de Mántaras, R., *Approximate Reasoning Models*. Ellis Horwood Series in Artificial Intelligence, Chichester, England, 1990.

[4] Valverde, L., García, P., Godó, L., Técnicas de Razonamiento Aproximado. En *Inteligencia Artificial: Conceptos, Técnicas y Aplicaciones*, Marcombo Boixareu Eds., Barcelona, 1987.

[5] Trillas, E., Valverde, L., On mode and implication in approximate reasoning. En M.M. Gupta, A. Kandel, W. Bandler y J.B. Kiszka (Eds.) *Approximate Reasoning in Expert Systems*, North-Holland, Amsterdam, 1985, pp. 157-166.

[6] Godó, L., Jacas, J., Valverde, L., Fuzzy Values in Fuzzy Logic. *International Journal of Intelligent Systems*, vol. 6, pp. 199-212, 1991.

[7] Bugarín, A., Barro, S., Reasoning with Truth values on Compacted Fuzzy Chained Rules. *Reporte Interno USC/DEC/GSI* 3-95, 1995.

[8] Bugarín, A., Barro, S. Fuzzy Reasoning Supported by Petri Nets, *IEEE Trans. Fuzzy Systems*, vol. 2, no. 2, pp. 135-150, 1994.

[9] Bugarín, A., Barro, S., Goal-Driven Reasoning for Fuzzy Knowledge-Based Systems using a Petri Net Formalism, *3rd IEEE Int. Conf. Fuzzy Systems FUZZ-IEEE '94*, 2061-2066, 1994.

[10] Bugarín, A. *Soluciones para la computación eficiente en sistemas basados en lógica borrosa*. Tesis Doctoral, Universidade de Santiago de Compostela, 1994. I.S.B.N. 84-8121148-6.

Relaciones de igualdad borrosas: aplicación a un sistema "fuzzy" para CAGD

J. Jacas
Sec. Matemàtiques i Informàtica.
E.T.S. Arquitectura de Barcelona
Univ. Politècnica de Catalunya
e-mail: jacas@ea.upc.es

A. Monreal
Sec. Matemàtiques i Informàtica.
E.T.S. Arquitectura de Barcelona
Univ. Politècnica de Catalunya
e-mail: monreal@ea.upc.es

J. Recasens
Sec. Matemàtiques i Informàtica.
E.T.S. Arquitectura del Vallés
Univ. Politècnica de Catalunya
e-mail: recasens@ea.upc.es

Resumen [1]

Este trabajo es una primera aproximación de un sistema "fuzzy" para el diseño de curvas y superficies en CAGD (Computer Aided Geometric Design). Su diseño esta basado en la generación de números borrosos por medio de relaciones borrosas admisibles sobre la recta real. Utiliza ideas de Takagi-Sugeno para generar, mediante razonamiento aproximado basado en lógica "fuzzy", curvas y superficies que interpolan, o están controladas (según el tipo de relación borrosa definida sobre **R***) por un conjunto de puntos.*

Palabras clave: Igualdad borrosa, número borroso, Razonamiento Aproximado, CAGD.

1 Introducción

Este trabajo pretende elaborar un sistema "fuzzy" para el diseño de curvas y superficies en CAD mediante el uso de razonamiento aproximado basado en reglas.

La idea fundamental consiste en construir una función cuya gráfica pase (o esté controlada) por un conjunto de $n+1$ puntos de control P_i del espacio. Estos datos son interpretados como una aproximación de los puntos sobre los cuales se desea construir la curva o superficie.

En el caso del trazado de la gráfica de una función $y = f(x)$, esta idea se traduce en un conjunto de reglas borrosas del tipo:

R_i: Si x es próximo a a_i, entonces $f(x)$ ha de ser próximo a b_i.

Reglas similares pueden definirse para curvas parametrizadas en el plano y en el espacio (sección 3) y para superficies paramétricas o para gráficas de funciones de dos variables (sección 4).

El modelo es un caso particular de sistema difuso basado en reglas y utiliza el punto de vista de Takagi-Sugeno [5,6] para control difuso puesto que el "output" del sistema no necesita de "defuzzyficado" debido a que la ponderación de los puntos borrosos asociados a los datos conduce directamente la curva o superficie solución.

El grado de continuidad de la curva o superficie trazada es del mismo orden que el de los números difusos μ_i que modelizan "x es próximo a a_i"

A partir de los mismos datos, la elección de los números borrosos μ_i y la modelización de las reglas de inferencia puede conducir a curvas o superficies diferentes.

El efecto de la elección de los números difusos puede verse en distintos ejemplos. Sin embargo cabe destacar los siguientes rasgos fundamentales:

- Si se desea que una curva o superficie tenga un cierto orden de continuidad, éste debe imponerse a los números difusos μ_i que modelizan los antecedentes de las reglas. Por ello, si sobre la recta real definimos una T-igualdad borrosa admisible [3] (T una t-norma continua) cuya familia asociada esté formada por números triangulares o trapezoidales, nuestro sistema solo conseguirá interpolación o control lineales.
- La relación de igualdad borrosa admisible que define el conjunto de números borrosos asociados a los datos debe de ser tal que dicho conjunto sea un recubrimiento borroso del dominio de la función. En caso contrario, se obtienen curvas discontínuas.
- Supongamos que la familia asociada a los antecendentes de las reglas es una recubrimiento borroso

[1] Research partially supported by DGICYT p.n. PB91-0334-CO3

del dominio del dominio de la función. Consideremos un punto $P_i = (a_i, b_i)$ (para este punto $\mu_i(a_i) = 1$)), si existe otro número μ_j de la familia tal que $\mu_j(a_i) \neq 0$ $(i \neq j)$ entonces la imagen de a_i se verá influida por la regla R_j y la curva no pasará por P_i que se comportará como un punto de control de la gráfica. Por ello el sistema unifica los dos tipos de construcción de curvas y superficies que se resumen en la próxima sección.

Finalmente, cabe puntualizar que diversos resultados [1,4,7] permiten asegurar que cualquier curva o superficie (definida en un compacto) puede aproximarse con una precisión deseada mediante un sistema difuso. Por tanto, el presente sistema tiene carácter universal en el sentido de poder obtener cualquier forma tomando los puntos y las reglas adecuadas.

2 Relación con otras técnicas de CAGD.

En el área del diseño geométrico de curvas existen básicamente dos tipos de métodos [2], basados ambos en utilizar como datos unos puntos de control y en usar funciones polinónicas para modelizar las soluciones:

a) **Los métodos interpoladores,** en los que la curva resultante pasa por los puntos de control. Entre ellos podemos citar: modelo de interpolación de Hermite y de Bessel que utiliza polinomios de grado $2n+1$ para conseguir continuidad C^n en los que cada punto afecta solamente un entorno de la curva, interpolación por "splines" que utiliza polinomios de menor grado para obtener la misma contuididad. Este método requiere, sin embargo la resolución de sistemas de ecuaciones lineales con matrices tridiagonales y del mismo orden que el número de puntos utilizado. En este método cada punto tiene un efecto global sobre la curva.

b) **Métodos de forma libre,** en los que los puntos dados solamente controlan a distancia la forma de la curva. Entre estos métodos cabe destacar la curvas de Bézier (simples o compuestas) y los B-splines que usan una base de curvas de soporte local. Los B-splines racionales, del cual los NURBS son un caso particular, usan funciones racionales para mejorar el control sobre la curva.

   La solución aportada por cada método goza de una serie de propiedades que justifican su utilización frente a otros. Entre ellas podemos mencionar:

   - Solución acotada por la envolvente convexa de los datos.
   - Precisión lineal.
   - Control local.
   - Acotación del número de oscilaciones.
   - Sencillez de cómputo

El método que se propone en esta comunicación permite cumplir cualquiera de las condiciones anteriores ajustando convenientemente el tipo de funciones que representan los datos.

Por lo que respecta a las superficies, hay también familias de métodos, unos basados en puntos de control y otros basados en familias de curvas que la superficie solución debe interpolar.

Entre los primeros se encuentran las superficies de Bézier (simples o compuestas), los B-splines en versiones para mallas cuadriculares o triangulares,etc... Entre los segundos podemos citar los "Coons patches" que parten de curvas-frontera y sus generalizaciones, las superficies de Gordon que interpolan una malla de curvas.

Un problema especialmente complejo es el de interpolar un conjunto disperso de puntos que aparece cuando hay que ajustar una superficie a una familia de puntos obtenidos experimentalmente (tal es el caso de mediciones topográficas). En este caso, hay que triangular el dominio de los parámetros. El método propuesto simplifica este problema de forma importante puesto que la forma en que están distribuidos los puntos no influye en el proceso de cálculo de la solución.

Cuando aplicamos al diseño cualquiera de los casos descritos anteriormente, siempre hay un proceso iterativo cuyo coste computacional depende del método usado y que puede resumirse en:

a) El usuario proporciona una colección de puntos o curvas de control iniciales.

b) El sistema produce una curva o superficie solución

c) El usuario, a la vista de la solución, rectifica algunos puntos o curvas y reinicia el cálculo.

Este proceso continua hasta que el usuario consigue la forma deseada. Esta descripción del proceso de diseño pone de manifiesto el carácter intrínsecamente borroso de los datos de control que justifica, desde el punto de vista aplicado, el paso inicial del método propuesto.

3 Modelado de curvas.

Se aborda en primer lugar el caso de una curva del plano que represente la gráfica de una función $y = f(x)$ y que pase o esté controlada por $n+1$ puntos de control $P_i = (a_i, b_i)$ $i = 0, 1..n$ con $a_i \neq a_j$ si $i \neq j$. Para ello definimos, por ejemplo mediante una relación de igualdad borrosa en $\mathbf{R}$, $n+1$ números borrosos μ_i que modelizan la propiedad "x es próximo a a_i". Suponemos que $\mu_i(a_i) = 1$ y que para todo $x \in [a_0, a_n]$ existe μ_k tal que $\mu_k(x) \neq 0$, es decir, la familia $\{\mu_j\}_{j \in \{0,1,\ldots n\}}$ es un recubrimiento borroso del intervalo $[a_0, a_n]$. La imagen de un punto $x \in [a_0, a_n]$ se define, siguiendo las ideas de Takagi-Sugeno [6] por

$$y = f(x) = \frac{\sum b_i \mu_i(x)}{\sum \mu_i(x)}$$

Ejemplo 1 *Sea el conjunto de puntos* $P_0 = \{1,5\}$, $P_1 = \{2,4\}$, $P_2 = \{3,7\}$, $P_3 = \{4,6\}$, $P_4 = \{5,9\}$, $P_5 = \{6,8\}$, $P_6 = \{7,2\}$, $P_7 = \{8,4\}$, $P_8 = \{9,3\}$. *Mediante la relación de igualdad borrosa admisible*

$$E_f(x,y) = Max\,(1 - |f(x) - f(y)|, 0),$$

que está definida por la escala $f(x) = x+2$, *se obtienen los números borrosos* μ_i, *asociados a los datos, que son una familia de números triangulares. En efecto para cada* $x = a_i$, *el número borroso asociado es*

$$\mu_i(x) = E_f(x, a_i) = Max\,(1 - |x - a_i|, 0)$$

y la curva solución está representada en la figura 1. Cambiando la función escala por $f(x) = \frac{1}{2}x+2$, *obtenemos la curva representada en la figura 2.*

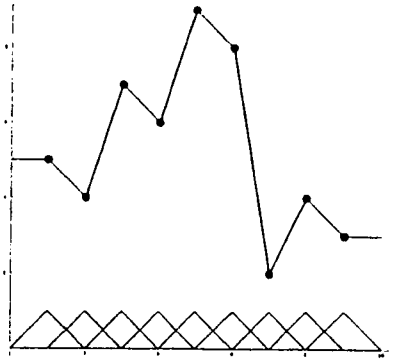


Figura 1

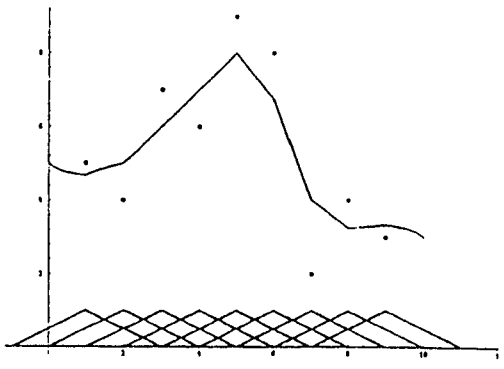


Figura 2

Los casos más generales de una curva parametrizada en el plano o el espacio se tratan de forma análoga determinando cada coordenada por separado (figuras 3 y 4)

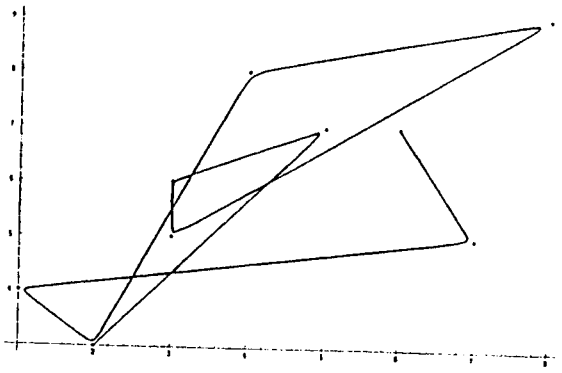


Figura 3

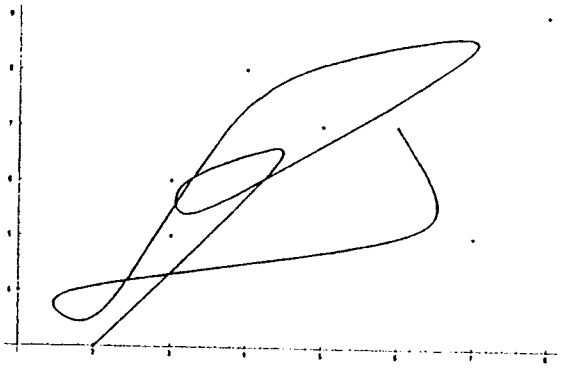


Figura 4

4 Modelado de superficies

Se desea trazar la gráfica de una función $z = f(x,y)$ que pase (o esté controlada) por n puntos de control $P_i = (a_i, b_i, c_i)$.

Las reglas son las siguientes:

R_i: Si x es próxima a a_i y y es próxima a b_i, entonces $z = f(x,y)$ ha de ser próxima a c_i.

Para asegurar el orden de continuidad n de f es necesario que la función que modelice la conjunción en las reglas tenga este orden. Por este motivo se ha escogido el producto para modelizarla.

Si μ_i es el número difuso "x es próximo a a_i" y ν_i el número difuso "y es próximo a b_i", entonces la imagen de un punto (x,y) se define por

$$f(x,y) = \frac{\sum c_i \mu_i(x)\nu_i(y)}{\sum \mu_i(x)\nu_i(y)}.$$

Ejemplo 2 *El conjunto de datos está formado por los puntos* $\{0,0,1\}$, $\{4,0,3\}$, $\{1,1,0\}$, $\{2,1,1.5\}$, $\{3,2,-1\}$, $\{0,3,5\}$, $\{2,3,1\}$, $\{4,3,2\}$). *La familia de borrosos que*

representan los antecedentes de las reglas y la superficie solución están representadas en las figuras 5 y 6.

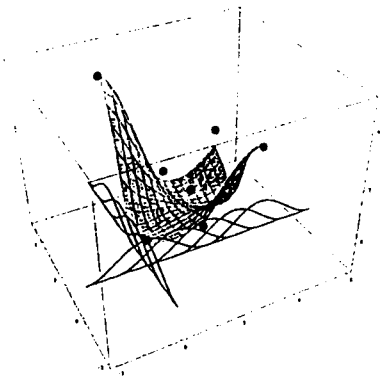


Figura 5

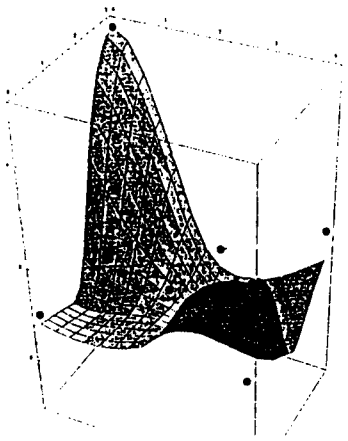


Figura 6

El caso de una superficie parametrizada se trata de forma análoga determinando cada coordenada por separado.

Referencias

[1] Castro, J.L., Delgado, M., Fuzzy Systems with defuzzification are universal approximators, *Fuzzy Sets & Systems* (1993).

[2] Farin, G., *Curves and Surfaces for CAGD.* Academic Press, 1990.

[3] Jacas, J., Recasens, J., Números borrosos y relaciones borrosas de igualdad. *Actas del IV Congreso Español sobre Tecnologías y Lógica Fuzzy*, Santiago de Compostela, 1994.

[4] Kosko, B. (1992), Fuzzy Systems as Universal Approximators, *FUZZ'IEEE-92.* San Diego, 1153-1162.

[5] Kruse, R., Gebhard, J., Klawoon, F. (1994), *Foundations of Fuzzy Systems*, John Wiley & Sons, New York.

[6] Takagi, H., Sugeno, M. Fuzzy identification of Systems and its applications to modelling and control, *IEEE Trans. on Systems, Man and Cybernetics*, 15, 116-132, (1985).

[7] Wang, Li-Xin, 1992, Fuzzy Systems Are Universal Approximators, *FUZZ'IEEE-92*, San Diego (1992), 1163-1170.

Asignación Básica de Probabilidad Difusa *

José A. Herencia
Dpto. de Matemáticas
Facultad de Ciencias
Universidad de Córdoba
e-mail: malhegoj@lucano.uco.es

M. Teresa Lamata
Dpto. de Ciencias de la Computación e I.A.
E.T.S. Ingeniería Informática
Universidad de Granada
e-mail: mtl@robinson.ugr.es

Resumen

En el presente trabajo se define el concepto de "Asignación Básica de Probabilidad Difusa" , similar a la asignación básica de probabilidad definida en la teoría de la evidencia, pero donde se incorpora la posibilidad de que los conjuntos focales sean difusos.

Palabras clave:: *Incertidumbre, conjuntos difusos, asignación básica de probabilidad, creencia, plausibilidad.*

1 Introducción

Con el término "incertidumbre" nos podemos referir a varios conceptos. Desde el punto de vista de los modelos matemáticos elaborados para su tratamiento, podemos seguir la clasificación establecida por Klir [9], quien distingue los dos aspectos siguientes de la incertidumbre:

1. Vaguedad (o borrosidad o "fuzziness"), recogida en los conjuntos difusos [17].

2. Ambigüedad, recogida en la Teoría de la Evidencia [15]. Dentro de este aspecto podemos considerar los conceptos de "imprecisión" (o no-especificidad) y de aleatoriedad (o "randomness"). Este último, a su vez, comprende la discordancia (o disonancia o conflicto) y la confusión.

Podemos ilustrar los distintos tipos de incertidumbre con un ejemplo sencillo como es el resultado de tirar un dado. Se presenta un resultado vago o difuso cuando nos dicen que ha aparecido "un número bajo". El resultado es impreciso si se trata de "un número par" (sin especificar cual de los tres posibles) y es aleatorio si sólo nos dicen que "los 6 números son equiprobables" apareciendo entonces confusión y discordancia al poderse obtener resultados totalmente diferentes.

Los distintos tipos de ambigüedad pueden aparecer recogidos dentro de una evidencia sobre el resultado de tirar el dado. Por ejemplo, mediante la Asignación Básica de Probabilidad (ABP) siguiente: pr(3)=pr(5)=pr(número par)=1/3.

En el apartado tercero del presente trabajo introducimos la noción de "Asignación Básica de Probabilidad Difusa" (ABPD) como una ABP cuyos elementos focales pueden ser subconjuntos difusos.

De esta forma se engloban la vaguedad y la ambigüedad, apareciendo todos los tipos de incertidumbre antes mencionados. Continuando con el ejemplo anterior, podemos considerar la ABPD siguiente: pr(1)=pr(número alto)=1/2.

Previamente a la definición de ABPD, consideraremos en el siguiente apartado las nociones preliminares básicas que usaremos sobre Teoría de Conjuntos Difusos y Teoría de la Evidencia.

2 Preliminares

En todo lo que sigue supondremos fijado un universo de discurso finito $X = \{x_1, x_2, ..., x_n\}$. Notaremos por $P(X)$ el conjunto de partes (o subconjuntos) de X, por $|C|$ el cardinal de cualquier conjunto C y por I el intervalo unidad $[0,1]$.

2.1 Conjuntos difusos

Se llama **subconjunto difuso** de X (o simplemente conjunto difuso) a toda aplicación de X en I. Notaremos por $F(X)$ al **conjunto de partes difusas** de X y consideraremos que $P(X) \subseteq F(X)$, identificando cada subconjunto de X con su función característica. A los subconjuntos de $P(X)$ les denominados "crisp" por oposición a "difuso", aunque resulten ser un caso particular.

Sea μ un subconjunto difuso cualquiera de X. Se dice que μ es **normal** cuando $\exists x \in X$ tal que $\mu(x) = 1$. El **soporte** de μ es el subconjunto crisp siguiente: $Sop(\mu) = \{x \in X : \mu(x) > 0\}$.

Se llama **punto difuso** (o "singleton" difuso) de **valor** $\lambda \in (0,1]$ y soporte $x \in X$, al conjunto difuso λ_x definido por:

$$\lambda_x(x) = \lambda, \ \lambda_x(y) = 0, \ \forall y \neq x.$$

El conjunto $F(X)$ es un **retículo distributivo completo** con las operaciones unión e intersección y el orden parcial inclusión definidos como sigue:

$$(\bigcup_{j \in J} \mu_j)(x) := sup\{\mu_j(x) : j \in J\}$$

*Este trabajo ha sido financiado por la DGICYT bajo el proyecto PB92-0959

$$(\bigcap_{j\in J}\mu_j)(x) := inf\{\mu_j(x) : j \in J\}$$

$$\mu \subseteq \nu :\Leftrightarrow \mu \leq \nu$$

Dado $\mu \in F(X)$, el conjunto difuso $\mu'(x) := 1 - \mu(x), \forall x \in X$, se denomina **pseudocomplementario** de μ o simplemente complementario de μ, aunque no es un complemento para el retículo $F(X)$. Y el número real $|\mu| := \sum_{x\in X}\mu(x)$ se denomina **cardinal** de μ.

Se llama **medida difusa** sobre $P(X)$ a toda aplicación $g : P(X) \to I$ que verifique las tres condiciones siguientes:

- $g(\emptyset) = 0$.
- $A \subseteq B \Rightarrow g(A) \leq g(B)$.
- $g(X) = 1$.

La **medida dual** de g se define como $g'(A) := 1 - g(X \setminus A)$ y es también una medida difusa.

2.2 Teoría de la Evidencia

Una **asignación básica de probabilidad** (ABP) sobre X es una aplicación $m : P(X) \to I$ que verifica:

- $m(\emptyset) = 0$.
- $\sum_{A\subseteq X} m(A) = 1$.

Elemento focal de m es todo subconjunto A de X al que se asigna una probabilidad positiva: $m(A) > 0$. Asociadas a cada ABP sobre X pueden construirse dos medidas difusas duales sobre $P(X)$, denominadas **creencia** (o "belief") y **plausibilidad** (o "plausibility") y definidas respectivamente de la forma siguiente ($\forall A \subseteq X$):

$$Bel(A) := \sum_{B\subseteq A} m(B)$$

$$Pl(A) := \sum_{B\cap A\neq\emptyset} m(B)$$

También es medida difusa, que denominaremos **plausibilidad ponderada**, la determinada por la fórmula siguiente (establecida en la medida de discordancia definida por Klir y Ramer [10] que consideraremos más adelante):

$$Plp(A) = \sum_{m(B)>0} m(B)\frac{|B\cap A|}{|B|}, \quad \forall A \subseteq X$$

Cualquiera de las tres funciones, m, Bel o Pl, determina unívocamente las otras dos y la información que aporta es denominada **evidencia** sobre X.

Dos tipos de evidencias son especialmente importantes:

1. Cuando los elementos focales de m forman una cadena (respecto a la inclusión de conjuntos), se trata de una **evidencia consonante**. Esta viene determinada por la distribución de posibilidad $\pi : X \to I$ dada por $\pi(x) = \sum_{x\in B} m(B), \forall x \in X$ (que es un conjunto difuso normal). Entonces Bel y Pl se denominan **necesidad** y **posibilidad** respectivamente.
   Como caso particular tenemos la **ignorancia**, presente cuando no poseemos ningún dato sobre subconjuntos concretos, sino que la evidencia se reparte en todo $X : m(X) = 1$.

2. Cuando todos los elementos focales de m son unitarios, se trata de una **evidencia probabilística**. Entonces $p(x) = m(\{x\}), \forall x \in X$, es una distribución de **probabilidad** y $Bel(A) = Pl(A) = \sum_{x\in A} p(x)$.
   Como caso particular tenemos la **certeza**, presente cuando toda la evidencia se concentra en un solo elemento: $\exists x \in X \quad t.q. \quad m(\{x\}) = 1$.

3 Asignaciones básicas de probabilidad difusas

Definición 1.
Llamaremos **medida difusa** sobre $F(X)$ a toda aplicación $g : F(X) \to I$ que verifique las tres condiciones siguientes:

- $g(\emptyset) = 0$.
- $\mu \subseteq \nu \Rightarrow g(\mu) \leq g(\nu)$.
- $g(X) = 1$.

Notaremos por G al conjunto de todas las medidas difusas sobre $F(X)$ y definimos la **medida dual** de $g \in G$ como $g'(\mu) = 1 - g(\mu')$, siendo μ' el complementario de μ.

Teniendo en cuenta que $\forall \mu, \nu \in F(X)$ es cierta la implicación $[\mu \subseteq \nu \Rightarrow \nu' \subseteq \mu']$, y la igualdad $\mu'' = \mu$; de las definiciones se deduce inmediatamente la proposición siguiente:

Proposición 1.
$\forall g, f \in G$, se verifica:

1. $g' \in G$.
2. $g \leq f \Rightarrow f' \leq g'$.
3. $g'' = g$.

Definición 2.
Llamaremos **asignación básica de probabilidad difusa** con soporte finito (abreviadamente, ABPD) sobre X a toda aplicación $m : F(X) \to I$ que verifique:

- $F := \{\mu \in F(X) : m(\mu) > 0\}$ es finito.
- $m(\emptyset) = 0$.

- $\sum_{\mu \in F} m(\mu) = 1$.

Denominamos **elementos focales** de m a los elementos de F y notamos por M al conjunto de todas las ABPD sobre X.

Definición 3.
Asociadas a cada ABPD sobre X definimos las aplicaciones de $F(X)$ en I denominadas **creencia** (o "belief"), **plausibilidad** y **plausibilidad ponderada** de igual forma que se definen para una ABP. Es decir, que vienen dadas respectivamente, $\forall \mu \in F(X)$, por:

$$Bel(\mu) := \sum_{\nu \subseteq \mu} m(\nu)$$

$$Pl(\mu) := \sum_{\nu \cap \mu \neq \emptyset} m(\nu)$$

$$Plp(\mu) := \sum_{m(\nu)>0} m(\nu) \frac{|\nu \cap \mu|}{|\nu|}$$

Proposición 2.
$\forall m \in M$, se verifica:

1. *Las aplicaciones Bel, Pl y Plp son medidas difusas sobre $F(X)$.*
2. *$Bel \leq Plp \leq Pl$.*

La proposición anterior se obtiene sin dificultad porque la inclusión y la intersección poseen propiedades análogas en $F(X)$ y en $P(X)$.
En cambio, no ocurre lo mismo con la complementación. Por este motivo, Bel y Pl no son medidas duales sobre $F(X)$; sino que, mediante la dualidad, originan las nuevas medidas siguientes:

$$Bel'(\mu) = \sum_{\nu \not\subseteq \mu'} m(\nu) \quad \forall \mu \in F(X)$$

$$Pl'(\mu) = \sum_{\nu \cap \mu' = \emptyset} m(\nu) \quad \forall \mu \in F(X)$$

La relación entre estas medidas y sus duales aparece en la proposición siguiente:

Proposición 3.
Las medidas difusas asociadas a cualquier $m \in M$ cumplen:

1. *$Pl' \leq Bel' \leq Pl$.*
2. *$Pl' \leq Bel \leq Pl$.*
3. *Bel' y Bel no son comparables.*

Demostración:

1. De los apartados (2) de las dos proposiciones anteriores se deduce que $Pl' \leq Bel'$.
   Para demostrar que $Bel' \leq Pl$ basta comprobar que $\forall \mu, \nu \in F(X)$ se verifica la implicación: $\nu \not\subseteq \mu' \Rightarrow \nu \cap \mu \neq \emptyset$. En efecto, veamos que se cumple su contrarrecíproco: supuesto que $\nu \cap \mu = \emptyset$ y dado cualquier $x \in X$ puede ser $\nu(x) = 0$ o bien $\nu(x) > 0$. En el primer caso es evidente que $\nu(x) \leq \mu'(x)$. En el segundo, por ser $\mu \cap \nu = 0$, se tiene que $\mu(x) = 0$ con lo que $\mu'(x) = 1$ y también resulta ser $\nu(x) \leq \mu'(x)$.

2. Por lo demostrado en el apartado anterior se deduce que, $\forall \mu, \nu \in F(X)$, si $\nu \cap \mu' = \emptyset$ entonces $\nu \subseteq \mu$. De ahí resulta que $Pl' \leq Bel$.

3. Consideremos el ejemplo siguiente para el caso n=2: sean los puntos difusos ν_1 de valor 1 y soporte x_1 , ν_2 de valor 1/2 y soporte x_2 y la ABPD m determinada por las igualdades $m(\nu_1) = m(\nu_2) = 1/2$. Entonces se comprueba fácilmente que $Bel(\mu_1) = 1$, $Bel'(\mu_1) = 1/2$, $Bel(\mu_2) = 1/2$ y $Bel'(\mu_2) = 1$ para los subconjuntos difusos siguientes: $\mu_1(x_1) = 1$, $\mu_1(x_2) = 1/2$, $\mu_2(x_1) = 1/2$, $\mu_2(x_2) = 1$.

Nota: A diferencia de lo que ocurre con una ABP, aquí no puede asegurarse la igualdad entre Bel' y Pl (ni entre Pl' y Bel) porque no es siempre cierto el recíproco de la implicación anteriormente demostrada:

$$\mu \cap \nu = 0 \Rightarrow \nu \subseteq \mu'$$

. Esto puede comprobarse con el simple contraejemplo siguiente: sea μ el punto difuso de soporte x_1 y valor 1/2 y sea $\nu = 1/4$ constante.

4 Conclusiones

La definición de "Asignación Básica de Probabilidad Difusa con soporte finito" (ABPD) sobre un conjunto $X = \{x_1, x_2, ..., x_n\}$ permite modelizar situaciones en las que aparecen simultáneamente los dos tipos fundamentales de incertidumbre:

1. La vaguedad presente en los elementos focales, ya que éstos son conjuntos difusos.

2. La ambigüedad propia de toda Asignación Básica de Probabilidad (ABP), que puede existir como imprecisión y como aleatoriedad.

El segundo apartado puede expresarse mediante las medidas difusas "creencia" (Bel) y "plausibilidad" (Pl) extendidas al retículo $F(X)$ de partes difusas de X. En este caso ambas medidas no son duales, como sí ocurre con cualquier ABP, donde dichas medidas se definen sobre el conjunto $P(X)$ de partes crisp de X.
Una línea de investigación, que están desarrollando los autores, consiste en definir medidas de incertidumbre aplicables a una ABPD. Interesará generalizar, de esta forma, diversas medidas de entropía ya definidas independientemente para medir el grado de vaguedad [1,11,14] o de ambigüedad [6,12,16].

Referencias

[1] A. De Luca and S. Termini, A definition of a non-probabilistic entropy in the setting of fuzzy sets theory, Inform. and Control 20 (1972) 301-312.

[2] M. Delgado and S. Moral, "A definition of inclusion for evidences." Fuzzy Mathematics, 7 (1987) 81-87.

[3] D. Dubois and H. Prade, A note on measures of specificity for fuzzy sets, BUSEFAL 19 (1984) 83-89.

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

[5] D. Dubois and H. Prade, "On several representations of an uncertain body of evidence. En Fuzzy Information and Decision Processes, (M.M.Gupta, E.Sanchez eds.) North-Holland, (1982) 167-181.

[6] M. Higashi and G.J. Klir, Measures of uncertainty and information based on possibility distributions, Int. J. Gen. Syst. 9 (1983) 43-58.

[7] U. Höhle, Fuzzy plausibility measures, in: E.P. Klement, Ed., Proc. Third Intern. Seminar on Fuzzy Set Theory, Johannes Kepler Univ. (Linz, 1981) 7-30.

[8] A. Kaufmann, Introduction to the Theory of Fuzzy Subsets. Academic Press, New York, (1975).

[9] G.J. Klir, Where do we stand on measures of uncertainty, ambiguity, fuzziness, and the like?, Fuzzy Sets and Systems 24 (1987) 141-160.

[10] G.J. Klir and A. Ramer, Uncertainty in Dempster-Shafer theory: A critical re-examination, Int. J. Gen. Syst. 18 (1990) 155-166.

[11] J. Knopfmacher, On measures of fuzziness, J. Math. Anal. Appl. 49 (1975) 529-534.

[12] M. Lamata y S. Moral, Measures of entropy in the theory of evidence, Int. J. Gen. Syst. 14 (1987) 297-305.

[13] M. Lamata y S. Moral, Classification of fuzzy measures, Fuzzy Sets and Systems 33 (1989) 243-253.

[14] S.G. Loo, Measures of fuzziness, Cybernetica 20 (1977) 201-210.

[15] G. Shafer, "A mathematical theory of Evidence". Princenton University Press, Princenton, New Jersey, (1976).

[16] R.R. Yager, Entropy and specificity in a Mathematical Theory of Evidence, Internat. J. Gen. Systems 9 (1983) 249-260.

[17] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338-353.

Una aplicación del concepto de prototipo deformable a un método que gestiona incertidumbre en un SBC[1]

José Angel Olivas Varela
Área de Lógica y Filosofía de la Ciencia
Universidad de Santiago de Compostela
e-mail: lflgalex@usc.es

Alejandro Sobrino Cerdeiriña
Área de Lógica y Filosofía de la Ciencia
Universidad de Santiago de Compostela
e-mail: lflgalex@usc.es

Resumen

En este trabajo se pretende mostrar la utilidad del concepto de prototipo deformable para métodos de predicción en sucesos que evolucionan en el tiempo o son cíclicos. Este concepto, junto con el de prototipos borrosos, permite averiguar la afinidad entre una situación real y una prototípica y modificando la definición de ésta última tener capacidad para establecer previsiones de lo que puede suceder.

Palabras clave: *Prototipo deformable, prototipo borroso.*

1 Introducción

Tradicionalmente, el concepto de prototipo ha tenido mucha utilidad en el proceso de categorización conceptual del medio, al permitir agrupar a objetos relativamente dispares entres sí en clases comunes. Ello sucede, p. ej., con animales que, aún teniendo propiedades esenciales o accidentales diferentes, como sucede con un murciélago o un gorrión, no obstante son clasificados como pájaros. En este ejemplo, el gorrión es usualmente considerado como el prototipo de la clase 'pájaro', ya que se presupone que reúne todas las características primarias y buena parte de las secundarias que se le atribuye a la clase.

Los psicólogos cognitivos han reparado en los beneficios de la noción de prototipo como mejor ejemplar de la clase y, ya que se habla de clase, han verbalizado frecuentemente el problema en términos de conjuntos. Así, indican que un prototipo es aquel ejemplar que tiene pertenencia absoluta a la categoría o clase de la que es prototipo, clasificándose otros ejemplares por similaridad con él. La similaridad justifica el decir que su pertenencia a la categoría es verdadera con algún tipo de grado, que se cuantifica usualmente en razón de las coincidencias y divergencias entre propiedades esenciales y accidentales del prototipo y el ejemplar comparado.

Esta noción de prototipo, caracterizada por nosotros en un trabajo previo [3] como 'ingenua', ha sido contestada en el ámbito de la lógica fuzzy. Parece discutible que la ubicación conceptual de un objeto se haga siempre respecto a un ente ideal. En el ámbito del sentido común, no parecen existir entes absolutamente ideales, sino objetos que reúnen características de todo tipo, aunque algunas en mayor medida que otras.

Así. p. ej., un automóvil 'Mercedes' coincide con una furgoneta o con un tractor en que posee cuatro ruedas. En esta característica coincide con un ejemplar mediocre (la furgoneta) y malo (tractor), del concepto 'automóvil de turismo'. Un automóvil 'Mercedes' tiene ruedas de un determinado tamaño de radio. En esto coincide con un ejemplar mediocre (la furgoneta), pero no con uno malo (el tractor). Por último, posee una cota superior determinada, mas pequeña que la de los otros ejemplares. No coincide, por tanto, con ninguno de ellos en esta característica. El prototipo de 'automóvil de turismo', -coincidamos en el Mercedes, p. ej.- encierra, por tanto, características que también son comunes a

[1] Financiación a cargo del proyecto PB92-0394, D.G.I.C.Y.T., Ministerio de Educación y Ciencia,.

ejemplares mediocres y malos del mismo. Esta es la idea que esta detrás de la propuesta de Zadeh (1982) [4], y que le valió para formular el concepto de prototipo borroso, una idea quizás menos intuitiva, pero más racional de cómo los humanos manejamos este concepto.

En el ámbito de la lógica fuzzy se han hecho otras aproximaciones al concepto de prototipo. Una de ellas es la de prototipo deformable, originariamente propuesta por Bremermann (1976) [1]. En este contexto, se usa una noción ingenua de prototipo y se considera como deformable cuando se puede hacer un emparejamiento entre él y un objeto, de forma que el prototipo se adapte al objeto dentro de unos límites preestablecidos.

En los métodos que gestionan incertidumbre (algoritmos borrosos), cabe distinguir: a) los que admiten entradas imprecisas, pero su mecanismo de inferencia es preciso y b) los que tratan de simular el razonamiento del experto globalmente, con el manejo de incertidumbre que ello conlleva.

Dentro de estos últimos, el método utilizado [2] para la predicción de incendios forestales en un SBC consiste en emparejar una situación real con un conjunto de prototipos borrosos de un concepto (el de 'siniestralidad progresiva'), ver con que prototipo tiene más afinidad y, según esto, modificar dicho prototipo hasta que describa adecuadamente la situación real.

En este trabajo se pretende hacer este método más fiable y flexible, utilizando conjuntamente características de esas dos concepciones de prototipo.

2 Prototipos deformables

La noción de prototipo deformable surge, fundamentalmente, en el ámbito del reconocimiento de patrones. En ese contexto se considera, p. ej., que los caracteres escritos a mano pueden ser vistos como imitaciones distorsionadas de prototipos ideales, sobre los cuales existe bastante consenso en cuanto a su representación. Así, un carácter cualquiera, puede ser básicamente considerado como compuesto de segmentos cuya posición se controla con una serie de parámetros. Dado un carácter, se puede deformar su prototipo manipulando los parámetros hasta que coincida con el carácter en la mayor medida de lo posible, aunque es previsible la presencia de unos restos que no puedan ser cubiertos por las variaciones del prototipo. Estas discrepancias pueden ser caracterizadas por una función ei de emparejamiento entre el objeto y el prototipo,

$$c_i \text{ (objeto, } p_1 \text{ ,....,} p_n) = | \text{ objeto–prototipo } (p_1 \text{ ....,} p_n) |$$

donde

p_1 ,....p_n	son los parámetros
objeto	es el carácter a ser reconocido
prototipo$_i$	es su prototipo
e_i (objeto)	es la función de emparejamiento
| |	es una medida de la diferencia entre el objeto y el prototipo deformado

y el mínimo de e_i se corresponde con la mejor coincidencia entre el prototipo deformado y el objeto que se puede obtener.

La función e_i no es una buena medida de afinidad entre el objeto y el prototipo porque, p. ej., no tiene en cuenta la relevancia especial de determinados parámetros, no diferencia entre características esenciales y accidentales de objeto a examen o no tiene en cuenta lo que supone que un determinado parámetro sea muy dispar. A fin de salvar estas deficiencias, se introduce una función de distorsión d(p_1 ,....,p_n), que relativiza el peso de los parámetros. Combinando las funciones e_i y d, se obtiene una función, llamada de afinidad que evalúa los efectos combinados. Esta nueva función de afinidad se puede definir como la combinación lineal de e_i y d,

$$\text{afin}_i \text{ (objeto)} = \text{mín } (e_i \text{ (objeto, } p_1,....,p_n) + w\ d(p_1,....,p_n), \quad p_1,....,p_n)$$

donde

w	es una constante de pesos, y
mín	denota el mínimo de todas las combinaciones posibles de parámetros

En este trabajo, el uso de la noción de prototipo deformable no se centra en el ámbito del reconocimiento de patrones, sino en la identificación de situaciones reales con prototipos borrosos que pueden ser parametrizados.

Podemos considerar la función de afinidad de un prototipo consigo mismo, ya que la coincidencia es exacta y no hay ninguna distorsión, como:

$$afin \text{ (prototipo)} = 0$$

Cuando la coincidencia del "objeto" con el prototipo es buena (el nivel de distorsión es bajo) entonces afin ("objeto") tendrá valores bajos. Según esto, como afin por sí misma no es una función de pertenencia a un conjunto borroso, podemos definir la siguiente:

$$\mu(\text{"}objeto\text{"}) = 1 - \frac{afin(\text{"}objeto\text{"})}{\text{má x(afin)}}$$

donde máx(afin) es el mínimo límite superior de afin en el mismo Universo U. Por lo tanto, μ ("objeto") será 1 para el prototipo mismo, y más pequeña cuanto más diferentes sean los objetos del prototipo (podría haber otras funciones que cumplieran este cometido, pero en principio esta parece ser adecuada).

Si consideramos la definición del prototipo en forma de tabla (adquirida del experto) como una parametrización, podremos definir una función de emparejamiento y calculando la distorsión podremos hallar afin("objeto"). (Nótese que si el número de parámetros es grande debemos optimizar la función). A partir de esto, podremos calcular μ ("objeto") al conjunto borroso de cada prototipo.

Con esto obtendremos dos resultados ventajosos:

1. El grado de pertenencia a cada prototipo (derivado del emparejamiento y la distorsión con respecto a él).
2. La variación en los parámetros de la definición inicial.

3 Método que gestiona incertidumbre en un SBC

En [2] se describe un método para emular el razonamiento de un experto en predicción de incendios forestales.

En él se describe que cuando el responsable en la toma de decisiones en incendios forestales (en lo que se refiere a despliegue de recursos o nivel de alerta) tiene que tomar medidas para los siguientes días, lo hace en base a patrones de comportamiento, prototipos de evolución de la siniestralidad: el de periodos de alta siniestralidad, siniestralidad media y siniestralidad baja, ya que ante el inicio de un nuevo periodo de incendios, no tiene otro criterio de ayuda que la experiencia de etapas anteriores.

Además, mediante técnicas de elicitación del conocimiento, se logra que el experto describa cada prototipo en forma de tabla definiendo cada uno de los parámetros de un determinado período, como número de incendios de los días posteriores, intensidad, localización, etc. Pero los factores que se dan en una ciclo real normalmente no coinciden con los de los patrones definidos, por lo que resulta necesario modificar la tabla inicial del prototipo más afín hasta que coincida en el mayor grado posible con la situación real.

Formalmente, el método consiste en:

- representar los prototipos de evolución de la siniestralidad mediante números borrosos triangulares,
- calcular la afinidad de la situación real con cada uno de los prototipos,
- decidir cuál es el prototipo más afín y
- describir el periodo actual modificando la definición (tabla) inicial del prototipo más afín.

Esta modificación se realiza mediante reglas, que normalmente es difícil extraer.

4 Objetivo

El objetivo de este trabajo es proporcionar una mejor caracterización del método descrito en el apartado anterior utilizando conjuntamente la teoría de prototipos borrosos y la noción de prototipo deformable. La mejora pretendida consiste en llevar a cabo la deformación del prototipo más afín usando en vez de reglas una función de afinidad.

5 Nueva caracterización del método

Debido a la naturaleza y complejidad de los factores que configuran el método de predicción de incendios, se hace necesario trabajar con autoajustes y modificaciones puntuales de los ciclos. Pero si reducimos el método a su forma esencial, puede ser utilizado para bastantes problemas en los que resulte adecuado asimilar situaciones reales a prototipos.

Esto resulta válido cuando los prototipos se refieren a modelos de comportamiento temporal. Si se ha definido la evolución de un determinado fenómeno, y se poseen datos hasta un determinado punto, se podrá hacer una predicción de como evolucionará ese fenómeno sabiendo cuál es el prototipo más afín de comportamiento hasta ese momento, y deformándolo hasta que se empareje con los datos reales conocidos. Con ello se podrá describir automáticamente la parte restante del proceso, con mayor fiabilidad cuanto más cercana sea en el tiempo.

La forma concreta de modificar el prototipo más afín es propia de cada caso real, pero en general se realiza en base a funciones que regulen y describan la serie temporal de valores dentro de cada factor. En el caso

que planteamos a continuación esta labor es sencilla y directa, pero en algunos casos reales es necesario utilizar funciones estadísticas para determinar la evolución.

La distorsión se calcula asignando pesos a cada valor dentro de la propia tabla de definición de los prototipos extraída del experto. Estos pesos actúan como factor w en la función de afinidad del objeto con el prototipo.

Aceptemos que los tres prototipos definidos de "Siniestralidad progresiva" de incendios forestales para una zona son los mostrados en las tablas 1, 2 y 3.

Siniestralidad altamente progresiva (P_1)

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Día 1	2	3	2	3	3
Día 2	4	4	3	4	4
Día 3	5	5	5	6	5
Día 4	6	6	7	8	7
Día 5	8	7	9	9	8

Tabla 1

Siniestralidad medianamente progresiva (P2)

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Día 1	1	2	0	1	2
Día 2	3	3	2	3	3
Día 3	4	4	4	4	4
Día 4	6	5	5	5	5
Día 5	7	6	6	7	7

Tabla 2

Siniestralidad escasamente progresiva (P_3)

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Día 1	0	1	1	0	2
Día 2	1	1	2	1	2
Día 3	1	2	2	2	3
Día 4	3	3	3	3	3
Día 5	3	3	4	4	5

Tabla 3

Los factores son "número de incendios", "virulencia", "zona horaria de inicio", etc. Pueden ser borrosos o precisos, lingüísticos o numéricos, pero están normalizados para el ejemplo en valoraciones en el intervalo [0,10].

Supóngase ahora la situación real mostrada en la Tabla 4.

Situación real a XX de XXXXXX de 19XX

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Día 1	1	2	1	2	2
Día 2	2	3	3	3	3
Día 3 (XX)	4	4	5	4	4
Día 4	?	?	?	?	?
Día 5	?	?	?	?	?

Tabla 4

La cuestión que puede plantearse es ¿qué ocurrirá previsiblemente los días 4 y 5?. Si se calcula el valor de e_i para la situación real de cada día con cada prototipo se obtiene lo siguiente:

e_i(XX, día 1, P_1) = (1, 1, 1, 1, 1)

e_i(XX, día 2, P_1) = (2, 1, 0, 1, 1)

e_i(XX, día 3, P_1) = (1, 1, 0, 2, 1)

e_i(XX, día 1, P_2) = (0, 0, 1, 1, 0)

e_i(XX, día 2, P_2) = (1, 0, 1, 0, 0)

e_i(XX, día 3, P_2) = (0, 0, 1, 0, 0)

e_i(XX, día 1, P_3) = (1, 1, 0, 2, 0)

e_i(XX, día 2, P_3) = (1, 2, 1, 2, 1)

e_i(XX, día 3, P_3) = (3, 2, 3, 2, 2)

Por lo que se puede concluir (para simplificar el ejemplo se prescindirá de la distorsión) que el prototipo más afín, el caso en el que se da una mayor pertenencia de la situación al prototipo, es P_2. Si se analizan con detalle los parámetros se puede observar que se podría completar la Tabla 4 de la definición de la forma que representa la Tabla 5, donde lo único que se varía es el Factor 3, aumentándolo en una unidad, como indica la evolución de los parámetros obtenidos.

Situación real a XX de XXXXXX de 19XX

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Día 1	1	2	1	2	2
Día 2	2	3	3	3	3
Día 3 (XX)	4	4	5	4	4
Día 4	6	5	6	5	5
Día 5	7	6	7	7	7

Tabla 5

Referencias

[1] Bremermann, H., Pattern Recognition. En H. Bossel: *Systems Theory in the Social Sciences*. Birkhäuser Verlag. pp. 116 -159, 1976.

[2] Olivas, J. A., Sobrino A., An Application of Zadeh's Prototype Theory to the Prediction of Forest Fire in a Knowledge-based System. *Proceedings of the 5th. International IPMU*. Paris, Francia. VOL. II, pp. 747 - 752, 1994.

[3] Olivas, J. A., Sobrino A., Aproximación Lingüística y teoría de prototipos de Zadeh. *Actas del Primer Congreso Español sobre Tecnologías y Lógica Fuzzy*, Universidad de Granada, pp. 179-183, 1.991.

[4] Zadeh, L. A., A note on prototype set theory and fuzzy sets. *Cognition*, 12, pp. 291-297, 1982.

Medidas Lingüísticas Para El Consenso En Grupo

F. Herrera, E. Herrera-Viedma, J.L. Verdegay
Dpto. de Ciencias de la Computación e Inteligencia Artificial
E.T.S de Ingeniería Informatica
Universidad de Granada, 18071 - Granada
e-mail:herrera,viedma,verdegay@robinson.ugr.es

Resumen

En un contexto lingüístico, se propone un nuevo modelo de consenso para el proceso de la toma de decisión en grupo. El modelo contiene un amplio conjunto de medidas lingüísticas que permiten conocer con gran exactitud el grado de acuerdo existente en el grupo y con ello dirigir el proceso de toma de decisión en grupo. Estas medidas son de dos tipos: grados de consenso, para saber la proximidad existente entre las opiniones de los componentes del grupo y, distancias lingüísticas, para conocer como de lejos está cada individuo de la opinión colectiva existente en el grupo. Este modelo permite desarrollar sistemas soporte de decisión en grupo con mayor consistencia humana.

Palabras Clave: Modelos lingüísticos, consenso, decisión en grupo.

1 Introducción

Un problema de decisión en grupo se establece en entornos donde hay una cuestión común a solucionar, un conjunto de opciones posibles a escoger y, un conjunto de individuos (expertos, jueces, ...) que expresan sus opiniones o preferencias sobre el conjunto de opciones posibles y que tienen la intención de alcanzar una decisión colectiva sobre la solución que resuelva la cuestión.

Una formulación clásica, más rigurosa y práctica de un problema de decisión en grupo es como sigue: hay un conjunto finito de alternativas $X = \{x_1, \ldots, x_n\}$ $(n \geq 2)$ con sus respectivos *grados de relevancia* $\mu_R(i) \in [0,1]$, y un conjunto de individuos $N = \{1, \ldots, m\}$ con sus respectivos *grados de importancia* $\mu_G(k) \in [0,1]$ (si $\mu_G(1) = \mu_G(2) = \ldots = \mu_G(m)$ estamos en el caso de *problemas de decisión en grupo homogéneos*, y si no, en el caso de *problemas de decisión en grupo heterogéneos*). Cada individuo $k \in N$ expresa sus opiniones sobre X con una *relación de preferencia fuzzy* $P^k \subset XxX$, siendo p_{ij}^k el grado de preferencia de la alternativa x_i sobre x_j expresado por el individuo k. Cuando $p_{ij}^k \in [0,1]$, es decir, cuando las relaciones de preferencia fuzzy son valoradas numéricamente, hablamos de *problemas de decisión en grupo en entorno numérico*, y cuando $p_{ij}^k \in S$, siendo S un conjunto de etiquetas lingüísticas, es decir, cuando las relaciones de preferencia fuzzy son valoradas lingüísticamente, hablamos de *problemas de decisión en grupo en entorno lingüístico*. Aquí nosotros trabajaremos en entorno lingüístico.

En los problemas de decisión en grupo son dos los procesos a desarrollar antes de obtener una solución: *proceso de consenso y proceso de selección*. El primero, hace referencia a cómo alcanzar el máximo grado de consenso o acuerdo entre los individuos sobre el conjunto de alternativa(s) solución. El segundo, hace referencia a cómo obtener el conjunto de alternativa(s) solución al problema planteado a partir de los opiniones expresadas por los individuos. Ambos procesos actúan conjuntamente de forma secuencial. Primero, el proceso de consenso actúa para lograr alcanzar el máximo grado de consenso posible en las opiniones de los expertos. En cada momento se calcula el grado de consenso existente. Si el grado es satisfactorio, entonces el proceso de selección se aplica de cara a obtener la solución. Por contra, si el grado de consenso medido no es satisfactorio, entonces los individuos son instados a cambiar sus opiniones de cara a aumentar la proximidad en sus planteamientos. De este modo, un proceso de toma de decisión en grupo se puede definir como un proceso dinámico e iterativo en el que los individuos van cambiando sus opiniones hasta que sus planteamientos sobre la solución son lo suficientemente próximos. Esto, gráficamente, puede ser observado en la Figura 1.

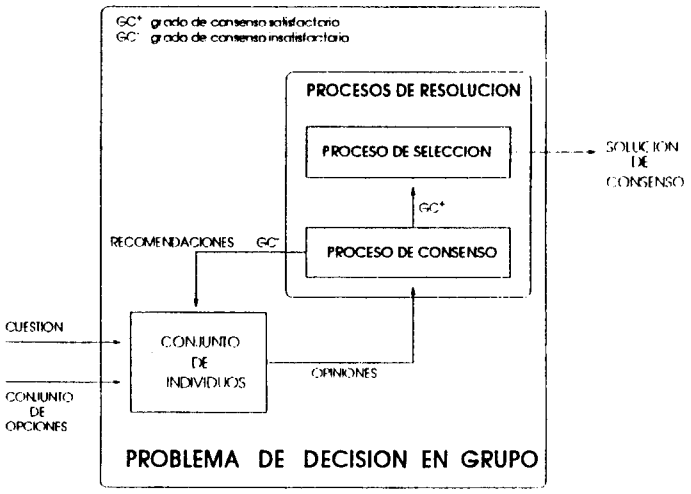


Figura 1: Esquema del proceso de resolución en grupo

Ambos procesos se han estudiado en entorno numérico en [6, 7] y, en entorno lingüístico en [3, 4, 5, 8]. Concretamente, en [8] se presenta un proceso de

consenso en entorno lingüístico guiado por medidas de consenso numéricas. En este trabajo, asumiendo que las opiniones de los individuos se expresan mediante relaciones de preferencia lingüísticas, presentamos un proceso de consenso para grupos heterogéneos, en entorno lingüístico, guiado por varios tipos de ***medidas lingüísticas de consenso*, grados de consenso y distancias lingüísticas**, las cuales describen con gran exactitud la situación de consenso actual y la proximidad de los individuos a la opinión colectiva, respectivamente.

El trabajo está estructurado como sigue: en la Sección 2 se presenta el modelo lingüístico usado; en la Sección 3 se definen las medidas lingüísticas de consenso y se describe su proceso de cálculo; y finalmente, en la Sección 4 se muestra un ejemplo de aplicación.

2 Modelo Lingüístico

En multitud de ocasiones es normal que el individuo tenga un conocimiento vago e impreciso sobre como evaluar el grado de preferencia de una alternativa sobre otra con un valor numérico exacto. Entonces, una forma más normal y práctica de expresar sus opiniones puede ser usar valoraciones lingüísticas en lugar de valores numéricos exactos, es decir, suponer que el dominio de las variables que intervienen en el problema es un conjunto de términos lingüísticos [3, 4, 5, 8]. Para ello, un conjunto apropiado de etiquetas, S, se selecciona de acuerdo al dominio del problema, y con él los individuos expresan sus preferencias.

Siguiendo nuestro modelo presentado en [3, 4, 5], consideremos $S = \{s_i\}, i \in H = \{0, \ldots, T\}$, 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. 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]. El conjunto de etiquetas debe cumplir las siguientes propiedades:

1) $s_i \geq s_j$ si $i \geq j$.

2) $\mathrm{Neg}(s_i) = s_j$ tal que $j = T - i$.

3) $\mathrm{Max}(s_i, s_j) = s_i$ if $s_i \geq s_j$.

4) $\mathrm{Min}(s_i, s_j) = s_i$ if $s_i \leq s_j$.

Un ejemplo de S, puede ser el siguiente conjunto de 9 etiquetas,

s_8	C	*Cierto*
s_7	EP	*Extremadamente_Posible*
s_6	MP	*Muy_Posible*
s_5	BP	*Bastante_Posible*
s_4	P	*Posible*
s_3	PP	*Poco_Posible*
s_2	MPP	*Muy_Poco_Posible*
s_1	EI	*Extremadamente_Imposible*
s_0	I	*Imposible*

cuya semántica asociada es, [1]:

C	$(1, 1, 0, 0)$
EP	$(.98, .99, .05, .01)$
MP	$(.78, .92, .06, .05)$
BP	$(.63, .80, .05, .06)$
P	$(.41, .58, .09, .07)$
PP	$(.22, .36, .05, .06)$
MPP	$(.1, .18, .06, .05)$
EI	$(.01, .02, .01, .05)$
I	$(0, 0, 0, 0)$

donde los primeros dos parámetros indican el intervalo en el que la función de pertenencia vale 1.0; y el tercero y cuarto indican las holguras izquierda y derecha de la función de pertenencia.

Esto, llevado a un problema de decisión en grupo, se traduce en usar relaciones de preferencia lingüísticas en vez de numéricas para expresar las opiniones de los individuos. Por tanto, cada individuo $k \in N$ expresa sus opiniones sobre las alternativas X con relaciones de preferencia lingüística valoradas sobre S,

$$\phi_{Pk} : \mathrm{XxX} \rightarrow S,$$

donde $\phi_{Pk}(x_i, x_j) = p_{ij}^k \in S$ representa el grado de preferencia lingüística de la alternativa x_i sobre la x_j.

3 Medidas Lingüísticas de Consenso

En un problema de toma de decisión en grupo, normalmente, al principio, entre los individuos suelen existir discrepancias en cuanto a qué opción(es) se acomoda(n) mejor al problema. En semejante situación, la solución al problema de decisión pasa por desarrollar un *proceso de consenso* entre los individuos hasta lograr un acuerdo lo más general posible sobre la decisión a adoptar.

El consenso es tradicionalmente entendido como un acuerdo total y unánime de todos los individuos que participan en el proceso, (corresponde a la idea del máximo consenso que se puede alcanzar) [7]. Lógicamente, este tipo de consenso es ideal y muy difícil de alcanzar. Por ello, en los procesos de consenso lo más normal es intentar alcanzar el nivel de consenso más alto o máximo consenso posible sin renunciar al máximo consenso total en los casos que sea alcanzable. Con este fin, un proceso de consenso en grupo se ve como un proceso dinámico donde *un moderador*, mediante el intercambio de información y de argumentos racionales, intenta persuadir a los individuos para que flexibilicen sus posturas y evolucionen a posiciones más cercanas. En cada estado del proceso, el moderador mide el *grado de consenso* existente entre los individuos y la distancia al nivel de *consenso ideal* establecido (puede ser el máximo total u otro valor por debajo que se considere aceptable) y en función de ello dirige el proceso. Esto se repite hasta que el grupo logra acercarse al nivel máximo de consenso establecido, es decir, hasta que la distancia al consenso ideal fijado es lo suficientemente pequeña.

En este trabajo, presentamos dos tipos de medidas lingüísticas de consenso para modelar los procesos de consenso en entorno lingüístico y heterogéneo,

que describen con gran exactitud la situación de consenso actual y ayudan al moderador a conducir el proceso de consenso: **grados de consenso y distancias lingüísticas**. Estas medidas se calculan en base a la idea de contar el número de individuos que están de acuerdo sobre el valor lingüístico asignado a cada preferencia x_i sobre x_j. Como herramientas para calcularlas, usamos el operador LOWA definido en [3, 5] y el concepto de fuzzy mayoría representado por un cuantificador lingüístico [10]. Las medidas actúan en tres niveles de acción, *nivel de preferencia, de alternativa y de relación de preferencia*, y se obtienen en tres pasos, de acuerdo al esquema de consenso presentado en la Figura 2.

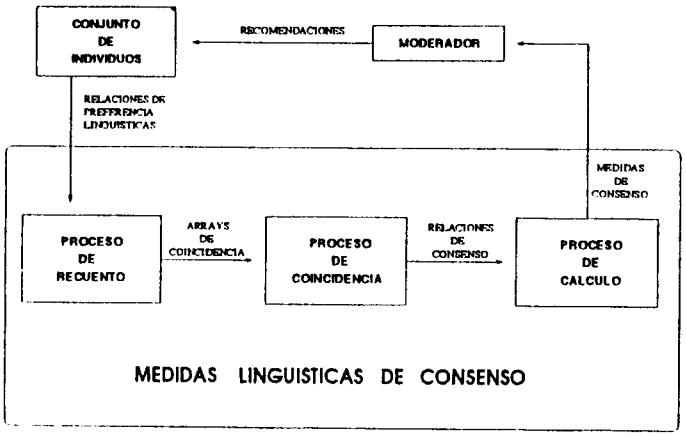


Figura 2: Esquema del Proceso de Consenso.

3.1 Proceso de Recuento

Supongamos que tenemos un conjunto de n alternativas $X = \{x_1, \ldots, x_n\}$ y un conjunto de individuos $N = \{1, \ldots, m\}$. Para cada alternativa, $(x_i$, consideremos definido un grado de relevancia $\mu_R(i) \in [0, 1]$, donde el valor 0 significa "totalmente irrelevante" y el valor 1 "totalmente relevante". De igual modo, para cada individuo, $k \in N$, consideremos conocido su grado de importancia $\mu_G(k) \in [0, 1]$. Cada individuo $k \in N$ expresa su relación de preferencia valorada lingüísticamente en el conjunto de términos S,

$$\phi_{P^k} : \mathrm{XxX} \rightarrow S,$$

donde $\phi_{P^k}(x_i, x_j) = p_{ij}^k \in S$ representa el grado lingüístico de preferencia de la alternativa x_i sobre la x_j. Asumamos, sin pérdida de generalidad, que P^k es recíproca en el sentido de que $p_{ij}^k = \mathrm{Neg}(p_{ji}^k)$, y por definición $p_{ii}^k = \mathit{Indefinido}$ (-).

En base a las relaciones de preferencia lingüística dadas por los individuos,$\{P^1, P^2, \ldots, P^k\}$, se calcula el número de individuos que están de acuerdo en el valor de preferencia de cada par de alternativas (x_i, x_j). Para hacerlo, se definen dos arrays, llamados *arrays de coincidencia* $(V_{ij}^C[s_t], V_{ij}^G[s_t]), \forall s_t \in S$ para las $T + 1$ etiquetas posibles que pueden ser asignadas como valor de preferencia a cada par de alternativas (x_i, x_j).

El primero, simbolizado por $V_{ij}^C[s_t]$ y llamado *array de coincidencia de los individuos*, contiene en cada posición s_t el número de individuos que coinciden en asignar la etiqueta s_t como valor de preferencia al par de alternativas (x_i, x_j). Se obtiene de acuerdo a la siguiente expresión:

$$V_{ij}^C[s_t] = \sharp(V_{ij}[s_t]), \forall s_t \in S.$$

siendo $V_{ij}[s_t] = \{k \mid p_{ij}^k = s_t,\ k = 1..m\}, \forall s_t \in S$, notando $\sharp$ el cardinal de un conjunto.

El segundo, simbolizado por $V_{ij}^G[s_t]$ y llamado *array de coincidencia de grados*, contiene en cada posición s_t la media aritmética de los grados de importancia de los individuos que coinciden en asignar la etiqueta s_t como valor de preferencia al par de alternativas (x_i, x_j). Se obtiene de acuerdo a la siguiente expresión:

$$V_{ij}^G[s_t] = \begin{cases} (\sum_{z \in V_{ij}[s_t]} \mu_G(z))/V_{ij}^C[s_t] & \text{si } V_{ij}^C[s_t] > 1 \\ 0 & \text{en otro caso} \end{cases}$$

3.2 Proceso de Coincidencia

Este proceso está basado en el concepto de *coincidencia* entre individuos. Nosotros consideramos que existe coincidencia sobre la etiqueta asignada a un valor de preferencia cuando hay más de un individuo que la ha escogido como valor de preferencia. En base a este concepto y partiendo de los arrays de coincidencia se calculan dos *relaciones de consenso*. Estas relaciones pueden ser calculadas tomando distintas políticas de consenso en función de lo que se quiera acelerar y de como se entienda el proceso de consenso. Por ejemplo, se puede tomar una *política promedio* en la que se consideren todos los valores en los que existe coincidencia, o una *política estricta* en la que se consideren sólo los valores en los que existe coincidencia con mayor grado, etc,. Aquí calcularemos las relaciones de acuerdo a una política promedio.

Antes de definir las expresiones que nos permiten calcular estas relaciones, introduciremos la definición de cuantificador fuzzy y del operador de agregación de información lingüística LOWA.

1. *Cuantificador Lingüístico.* En nuestro contexto, un cuantificador lingüístico se usa para especificar el concepto de mayoría fuzzy. Los cuantificadores lingüísticos fuzzy fueron presentados por Zadeh en 1983 [10]. Un cuantificador lingüístico es un conjunto fuzzy definido sobre $[0, 1]$ que representa una propiedad linguísticamente cuantificada. Zadeh distinguió dos tipos de cuantificadores, los absolutos como *"alrededor de 7"* y los proporcionales o relativos como *"muchos"*. Utilizamos los cuantificadores proporcionales porque reflejan la esencia de la mayoría, como por ejemplo *muchos, al menos la mitad, todos, tantos como sea posible.*

   Si Q es un cuantificador relativo, entonces Q puede representarse como un subconjunto fuzzy de $[0, 1]$ tal que para cada $r \in [0, 1]$, $Q(r)$ indique el grado en el que la proporción r de objetos satisface el concepto representado por Q.

   La función de pertenencia de un cuantificador relativo , Q, puede valorarse numéricamente,

$$Q : [0, 1] \rightarrow [0, 1],$$

o lingüísticamente,

$$Q : [0,1] \rightarrow L,$$

siendo $L = \{l_i\}$, $i \in J = \{0, \ldots, U\}$, un conjunto de etiquetas. En el primer caso, lo notaremos, Q^1, y su función de pertenencia vendrá dada por la siguiente expresión:

$$Q^1(r) = \begin{cases} 0 & \text{si } r < a \\ \frac{r-a}{b-a} & \text{si } a \leq r \leq b \\ 1 & \text{si } r > b \end{cases}$$

con $a, b, r \in [0,1]$. En el segundo caso, lo notaremos Q^2, y su función de pertenencia vendrá dada por la siguiente expresión:

$$Q^2(r) = \begin{cases} l_0 & \text{si } r < a \\ l_i & \text{si } a \leq r \leq b \\ l_U & \text{si } r > b \end{cases}$$

siendo l_0 y l_U las etiquetas mínima y máxima de L, respectivamentes y, $l_i = Sup_{l_q \in M}\{l_q\}$, con

$$M = \{l_q \in L : \ \mu_{l_q}(r) = Sup_{t \in J}\{\mu_{l_t}(\frac{r-a}{b-a})\}\},$$

con $a, b, r \in [0,1]$. La gráfica 3 muestra el cuantificador *"Al menos la mitad"* con el par (a,b)=(0,0.5).

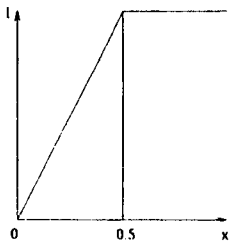


Figura 3: Cuantificador "Al menos la mitad".

2. *Operador LOWA.* El operador LOWA es un operador de agregación de información lingüística por computación directa sobre las etiquetas, basado en el operador de agregación de información numérica OWA de Yager [9] y en la combinación convexa de etiquetas de Delgado et.al. [2]. Se define el operador LOWA en [3, 5] de acuerdo a la siguiente expresión:

$$F(a_1, \ldots, a_m) = W \cdot B^T =$$

$$= C\{w_k, b_k, k = 1, \ldots, m\} =$$

$$= w_1 \odot b_1 \oplus (1 - w_1) \odot C\{\beta_h, b_h, h = 2, \ldots, m\}$$

donde $C\{\ldots\}$ simboliza el operador de combinación convexa de etiquetas lingüísticas; $W = [w_1, \ldots, w_n]$, el vector de pesos, tal que, $a) w_i \in [0,1]$, $b) \Sigma_i w_i = 1$; $\beta_h = w_h / \Sigma_2^m w_k, h = 2, \ldots, m$; y B es el vector de etiquetas ordenado asociado. Cada elemento $b_i \in B$ es la i-ésima mayor etiqueta de la colección de etiquetas a agregar $a_1, \ldots, a_m$. Para el caso de la expresión $w_1 \odot s_j \oplus (1 - w_1) \odot s_i, j \geq i$, resulta la etiqueta s_k tal que $k = \min\{T, i + round(w_1 \cdot (j - i))\}$, donde *round* es la función matemática de redondeo usual; y si $w_j = 1$ y $w_i = 0$ con $i \neq j \ \forall i$, entonces

$$C\{w_i, p_i, i = 1, \ldots, m\} = p_j.$$

Los pesos w_i para la agregación se pueden calcular a partir de la función que describe al cuantificador Q [9]. En el caso de un cuantificador relativo se obtienen mediante la expresión:

$$w_i = Q(i/m) - Q((i-1)/m), i = 1, \ldots, m.$$

En este sentido, cuando usemos un cuantificador Q para hallar los pesos, el operador LOWA será simbolizado por F_Q.

3.2.1 Relación de Consenso de Etiquetas (RCE)

Contiene en cada posición (i, j) la *etiqueta de consenso* (RCE_{ij}) asignada al valor de preferencia del par de alternativas (x_i, x_j). Se obtiene de acuerdo a la siguiente expresión:

$$RCE_{ij} = \begin{cases} F_{Q^1}(l_1, \ldots, l_q) & \text{si } \sharp(M_{ij}) > 1 \\ l_q & \text{si } \sharp(M_{ij}) = 1 \\ Indefinido & \text{en otro caso} \end{cases}$$

$l_k \in M_{ij}$, $k = 1, \ldots, q$, donde,

$$M_{ij} = \{s_y \mid V_{ij}^C[s_y] > 1, s_y \in S\},$$

es el conjunto de etiquetas que han sido escogidas como valor de preferencia por más de un individuo.

3.2.2 La Relación de Consenso de Individuos (RCI)

Contiene en cada posición (i, j) dos componentes:

1. RCI_{ij}^a, *el grado de coincidencia* asignado al valor de preferencia del par de alternativas (x_i, x_j) en función del número de individuos cuyas opiniones se usan para hallar RCE_{ij} y,

2. RCI_{ij}^b, *la media de los grados de importancia* de los individuos cuyas opiniones se usan para hallar RCE_{ij}.

Ambos componentes se obtienen de acuerdo a las siguientes expresiones:

$$RCI_{ij}^a = \begin{cases} \frac{\sum_{s_y \in M_{ij}} (V_{ij}^C[s_y]/m)}{\sharp(M_{ij})} & \text{si } \sharp(M_{ij}) \neq 0 \\ 0 & \text{en otro caso} \end{cases}$$

$$RCI_{ij}^b = \begin{cases} \frac{\sum_{s_y \in M_{ij}} V_{ij}^C[s_y]}{\sharp(M_{ij})} & \text{si } \sharp(M_{ij}) \neq 0 \\ 0 & \text{en otro caso} \end{cases}$$

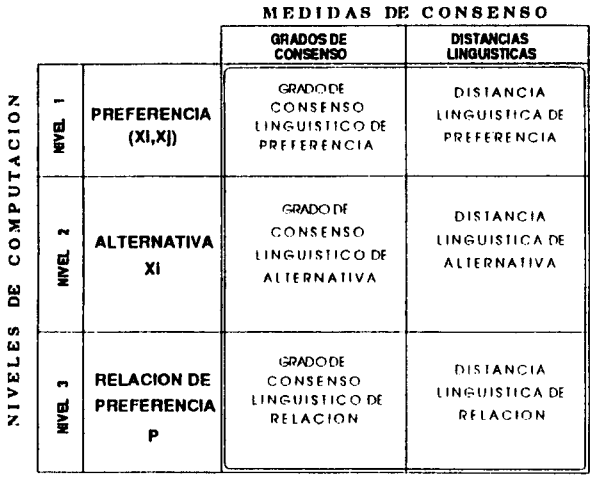


Figura 4: Medidas Lingüísticas de Consenso.

3.3 Proceso de Cálculo

En este paso se calculan las medidas lingüísticas de consenso distinguiendo entre tres niveles de computación como se muestra en la Figura 4.

Como hemos mencionado, trabajamos con dos tipos de medidas de consenso:

1. *Grados de Consenso Lingüísticos.* Sirven para evaluar el grado de consenso existente entre los individuos. Este tipo de medidas se usan por el moderador para decidir sobre la necesidad de continuar el proceso de consenso. Se hallan usando el grado de relevancia de las alternativas, cuantificadores lingüísticos y la relación de consenso RCI. Se pueden distinguir tres grados de consenso:

   (a) Nivel de Preferencia: *Grado de Consenso Lingüístico de Preferencia (GCP_{ij}).* Se define para cada cada par de alternativas (x_i, x_j) e indica el grado de consenso existente entre los individuos en torno al valor de preferencia de ese par. Se obtiene de acuerdo a la siguiente expresión:

   $$GCP_{ij} = Q^2(RCI^a_{ij} \wedge \mu_I(x_{ij}))$$

   con $i, j = 1, \ldots, n$, $i \neq j$, donde $\mu_I(x_{ij})$ simboliza la importancia del grado de consenso alcanzado sobre el valor de preferencia asignado al par de alternativas (x_i, x_j), y se define como:

   $$\mu_I(x_{ij}) = \begin{cases} \frac{\mu_R(i)+\mu_R(j)+RCI^b_{ij}}{3} & \text{si } \mathrm{RCI}^b_{ij} \neq 0 \\ 0 & \text{en otro caso.} \end{cases}$$

   (b) Nivel de Alternativa: *Grado de Consenso Lingüístico de Alternativa (GCA_i).* Se define para cada alternativa x_i e indica el grado de consenso existente entre los individuos en torno a los valores de preferencia de cada par en el que interviene dicha alternativa. Se obtiene de acuerdo a la siguiente expresión:

   $$GCA_i = Q^2[\sum_{j=1,\neq i}^{n} (RCI^a_{ij} \wedge \mu_I(x_{ij}))/(n-1)],$$

   con $i = 1, \ldots, n$.

   (c) Nivel de Relación: *Grado de Consenso Lingüístico de Relación (GC).* Se define para el conjunto completo de alternativas X e indica el grado de consenso existente entre los individuos en torno a los valores de preferencia de todos los pares de alternativas. Se obtiene de acuerdo a la siguiente expresión:

   $$RC = Q^2[(\sum_{i}^{n} \sum_{j=1,\neq j}^{n} (RCI^a_{ij} \wedge \mu_I(x_{ij})))/(n^2-n)].$$

2. *Distancias Lingüísticas.* Sirven para evaluar la proximidad de las opiniones de los individuos a las etiquetas de consenso actuales de los valores de preferencia. Este tipo de medidas se usan por el moderador para identificar qué individuos están más lejos de las posturas colectivas y en qué valores de preferencia existen esas diferencias. Se calculan usando las relaciones de preferencias lingüísticas de los individuos P^k, el operador de agregación lingüístico LOWA, cuantificadores lingüísticos y la relación de consenso de etiquetas RCE. Se pueden distinguir tres distancias lingüísticas:

   (a) Nivel de Preferencia: *Distancia Lingüística de Preferencia (D^k_{ij}).* Se define para cada par de alternativas (x_i, x_j) e indica la proximidad entre la opinión asignada por un individuo k como valor de preferencia del par y la obtenida del grupo de individuos (etiqueta de consenso). Se obtiene de acuerdo a la siguiente expresión:

   $$D^k_{ij} = \begin{cases} p^k_{ij} - RCE_{ij} & \text{si } p^k_{ij} > RCE_{ij} \\ RCE_{ij} - p^k_{ij} & \text{si } RCE_{ij} >= p^k_{ij} \\ s_T & \text{en otro caso} \end{cases}$$

   $i \neq j$, con $i, j = 1, \ldots, n$, y $k = 1, \ldots, m$, donde si $p^k_{ij} = s_t$ y $RCE_{ij} = s_v$ entonces $p^k_{ij} - RCE_{ij}$ se define como la etiqueta s_w, tal que, $w = t - v$.

   (b) Nivel de Alternativa: *Distancia Lingüística de Alternativa (D^k_i).* Se define para cada alternativa x_i e indica la proximidad entre las opiniones asignadas por un individuo k a todos valores de preferencia de los pares de alternativas donde interviene x_i y las obtenidas del grupo de individuos (etiquetas de consenso de cada par). Se obtiene de acuerdo a la siguiente expresión:

   $$D^k_i = F_{Q^1}(D^k_{ij}, j = 1, \ldots, n,\ j \neq i)$$

   con $k = 1..m$, $i = 1..n$.

   (c) Nivel de Relación: *Distancia Lingüística de Relación (D^k_R).* Se define para el conjunto completo de alternativas X, e indica la proximidad entre las opiniones asignadas por un

individuo k a todos los valores de preferencia de todos los pares de alternativas (x_i, x_j) y las obtenidas del grupo de individuos (la relación de consenso RCE). Se obtiene de acuerdo a la siguiente expresión:

$$D_R^k = F_{Q^1}(D_{ij}^k,\ i,j = 1..n,\ j \neq i)$$

con , $k = 1..m$.

4 Ejemplo

Supongamos que tenemos cuatro alternativas y cuatro expertos cuyas preferencias lingüísticas usando el anterior conjunto de 9 etiquetas son:

$$P_1 = \begin{bmatrix} - & PP & BP & MPP \\ BP & - & P & P \\ PP & P & - & MPP \\ MP & P & MP & - \end{bmatrix}$$

$$P_2 = \begin{bmatrix} - & P & P & MPP \\ P & - & BP & P \\ P & PP & - & MPP \\ MP & P & MP & - \end{bmatrix}$$

$$P_3 = \begin{bmatrix} - & P & BP & I \\ P & - & MP & P \\ PP & MPP & - & MPP \\ C & P & MP & - \end{bmatrix}$$

$$P_4 = \begin{bmatrix} - & PP & BP & PP \\ BP & - & MPP & PP \\ PP & MP & - & MPP \\ BP & BP & MP & - \end{bmatrix}$$

y cuyos grados de relevancia y de importancia, respectivamente son:

$$\mu_R(x_1) = 0.4, \mu_R(x_2) = 1, \mu_R(x_3) = 0.3, \mu_R(x_4) = 0.8,$$

y,

$$\mu_G(1) = 0.9, \mu_G(2) = 0.2, \mu_G(3) = 0.5, \mu_G(4) = 0.7.$$

Usando el cuantificador lingüístico *"Al menos la mitad"* con parametros $(0.0, 0.5)$, obtenemos los siguientes grados de consenso:

1. Nivel de Preferencia: $(GCP_{12} = C,\ GCP_{13} = MP,\ GCP_{14} = C,\ GCP_{21} = C,\ GCP_{23} = I,\ GCP_{24} = C,\ GCP_{31} = MP,\ GCP_{32} = I,\ GCP_{34} = C,\ GCP_{41} = C,\ GCP_{42} = C,\ GCP_{43} = C)$.

2. Nivel de Alternativa: $(GCA_1 = EP,\ GCA_2 = MP,\ GCA_3 = BP,\ GCA_4 = C)$.

3. Nivel de Relación: $GC = MP$.

y, por ejemplo, las siguientes distancias lingüísticas correspondientes al individuo 3,

1. Nivel de Preferencia: $(D_{12}^3 = I,\ D_{13}^3 = I,\ D_{14}^3 = MPP,\ D_{21}^3 = EI,\ D_{23}^3 = C,\ D_{24}^3 = I,\ D_{31}^3 = I,\ D_{32}^3 = C,\ D_{34}^3 = I,\ D_{41}^3 = MPP,\ D_{42}^3 = I,\ D_{43}^3 = I)$.

2. Nivel de Alternativa: $(D_1^3 = EI,\ D_2^3 = MP,\ D_3^3 = BP,\ D_4^3 = EI)$.

3. Nivel de Relación: $D_R^3 = PP$.

De estos resultados, podemos concluir que: (i) el grado de consenso existente entre los expertos es muy alto, (ii) el grado de consenso existente sobre la alternativa x_4 es el mayor, aunque sobre las demás existe un alto grado de consenso, (iii) el grado de consenso existente sobre el par (x_2, x_3) es inexistente, (iv) el individuo 3 está muy lejos de las posiciones colectivas sobre las alternativas x_2 y x_3.

Referencias

[1] Bonissone, P.P., Decker, K.S., 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, London, 217-247, 1986.

[2] Delgado, M., Verdegay, J.L., Vila, M.A., On Aggregation Operations of Linguistic Labels, *International Journal of Intelligent Systems* 8 (1993), 351-370.

[3] Herrera, F., Herrera-Viedma, E., Verdegay, J.L., A Linguistic Decision Process in Group Decision Making. Aceptado en *Group Decision and Negotiation* (1994).

[4] Herrera, F., Herrera-Viedma, E., Verdegay, J.L., A Sequential Selection Process in Group Decision Making with Linguistic Assessment. *Int. Journal of Information Sciences* 80 (1995), 1-17.

[5] Herrera, F., Herrera-Viedma, E., Verdegay, J.L., Direct Approach Processes in Group Decision Making Using Linguistic OWA Operators. Aceptado en *Fuzzy Sets and Systems* (1995).

[6] Kacprzyk, J., Group Decision Making with a Fuzzy Linguistic Majority. *Fuzzy Sets and Systems* 18 (1986), 105-118.

[7] Kacprzyk, J., Fedrizzi, M., A 'soft' measure of consensus in the setting of partial (fuzzy) preferences. *European Journal of Operational Research* 34 (1988), 316-323.

[8] Mich, L., Gaio, L., Fedrizzi, M., On Fuzzy Logic-Based Consensus in Group Decision, *Proc. Firth IFSA World Congress*, Seoul, pp. 698-700, 1993.

[9] Yager, R.R., On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol 18 N 1 (1988), 183-190.

[10] Zadeh, L., *A Computational Approach to Fuzzy Quantifiers in Natural Language*, Comps & Maths. with Appls. 9 (1983), 149-184.

Obtención de soluciones en redes de restricciones borrosas

M.A. Cárdenas, R. Marín, M. Balsa, J. L. Sánchez, I. Navarrete
Dept. de Informática y Sistemas
Universidad de Murcia
Campus de Espinardo
30071 Murcia, España
e-mail: roque@dif.um.es

Resumen

En este trabajo, se proponen varios métodos de ordenación de variables que son aplicables a modelos de restricciones métricas borrosas. Se identifican algunos criterios de calidad que establecen propiedades deseables para los métodos de ordenación de números borrosos dependientes, discutiendo las características de los métodos propuestos en relación con estos criterios.

Keywords. Razonamiento temporal, restricciones métricas, ordenación de números borrosos, búsqueda de soluciones en redes de restricciones.

1 Introducción

Las aproximaciones convencionales al problema de la ordenación de números borrosos pueden clasificarse en tres grupos: a) Métodos basados en funciones ordenadoras; b) Métodos basados en funciones de comparación; y c) Aproximaciones lingüísticas. En el primer caso, se establece una función que proyecta cada número borroso en un número de la recta real, de forma que la comparación de dos números borrosos se reduce a identificar el orden natural existente entre los números reales correspondientes. En el segundo caso, se aplica una función de comparación a cada dos números borrosos, obteniendo uno o varios índices reales, que son interpretados mediante un conjunto de reglas de comparación. En el último caso, se establecen relaciones de preferencia lingüística, conservando la subjetividad intrínsecamente asociada a la definición de los conjuntos borrosos [1, 2, 12, 14, 13].

El marco de aplicación usual de los distintos métodos de ordenación corresponde, generalmente, a problemas de decisión. En este contexto, cada número borroso puede interpretarse como una variable real X_i, cuyos posibles valores vienen descritos mediante una distribución de posibilidad π_i, que es independiente de las distribuciones correspondientes a los restantes números borrosos. El problema de la ordenación de variables descritas en forma borrosa puede plantearse en un contexto más general, incluyendo el caso en el que las variables no verifican la condición de independencia. Un modelo particular que permite representar situaciones de dependencia entre variables borrosas es el modelo de redes de restricciones temporales borrosas (FTCN), que hemos introducido en trabajos previos como base para la aplicación de técnicas de razonamiento temporal a dominios médicos [4, 5, 8, 9, 10]. En una FTCN, las variables representan instantes de tiempo imprecisos, cuyos posibles valores se establecen mediante un conjunto de restricciones métricas entre parejas de variables. El formalismo FTCN permite expresar información global sobre las variables mediante la agregación de las piezas de información local proporcionadas por las diferentes restricciones binarias. La información global derivable de la FTCN es justamente el conjunto de posibles soluciones de la red, siendo cada solución una n-tupla de instantes precisos que define una determinada disposición de las variables sobre el eje real del tiempo. El conjunto de soluciones de la red describe los posibles escenarios temporales globales consistentes con las restricciones locales de la red. Una propiedad deseable en los métodos de ordenación de variables, cuando se aplican a las variables de una FTCN, es que conduzcan siempre a la obtención de una solución de la red; en caso contrario, estarán proporcionando escenarios temporales inconsistentes.

En este trabajo, proponemos varios métodos de ordenación de variables que son aplicables a modelos de restricciones métricas borrosas y verifican la propiedad anterior. Presentamos aquí sólo los métodos basados en funciones ordenadoras. De los métodos de este tipo descritos en la bibliografía, la mayoría actúan como métodos de "defuzzificación", de tal forma que la aplicación de una función ordenadora f a una variable X_i independiente proporciona un elemento representativo $f(X_i)$ perteneciente al soporte de la distribución de posibilidad π_i. En otros casos, $f(X_i)$ es sólo un índice real abstracto sin relación con el dominio de X_i, obtenido exclusivamente a efectos de ordenación; a menudo, este índice se obtiene mediante la comparación de π_i con algún patrón relevante al objetivo de la ordenación (por ejemplo, "tan grande como sea posible"). La principal ventaja de los métodos del primer tipo sobre los segundos, en el contexto FTCN, es que, además de resolver el problema de la ordenación, pueden plantearse

de forma que obtengan una solución de la red. La obtención eficiente de soluciones es uno de los problemas centrales del razonamiento temporal, principalmente por sus implicaciones en la aplicación a planificación temporal de tareas [6].

2 Redes de restricciones borrosas

Comenzaremos recordando algunos conceptos básicos que ya han sido introducidos en otros trabajos previos [5, 8]. Formularemos el modelo de redes de restricciones métricas borrosas en forma general, haciendo abstracción del contexto de aplicación del modelo FTCN previo, que es el razonamiento temporal.

Definición 1.- *Una* red de restricciones borrosas *(FCN) $\mathcal{N} = <X, L>$ es una pareja formada por un conjunto finito de n+1 variables $X = \{X_0, X_1 \ldots X_n\}$ y un conjunto finito de restricciones binarias borrosas entre ellas $L = \{L_{ij} / i, j \leq n\}$.*

En ausencia de restricciones, cada variable X_i puede tomar cualquier valor numérico preciso de un dominio real o entero $\mathcal{R}$. Los valores de las variables se establecen mediante asignaciones: $X_i := x_i$, $x_i \in \mathcal{R}$. Las restricciones establecen distancias borrosas entre las variables, limitando los valores que pueden ser asignados a las variables. Cada restricción binaria L_{ij} se define mediante una distribución de posibilidad π_{ij} sobre $\mathcal{R}$, normalizada y convexa, que describe los posibles valores de la diferencia entre las variables X_i y X_j. Para poder realizar las asignaciones $X_i := x_i$ y $X_j := x_j$, establecemos como condición necesaria que se verifique $\pi_{ij}(x_j - x_i) > 0$; sólo será condición suficiente si no hay otras restricciones actuando sobre alguna de las dos variables. La variable X_0 representa un origen preciso, y se le asigna un valor arbitrario x_0, que supondremos igual a cero. De esta forma, se evita la introducción de restricciones unarias, que pueden representarse como restricciones respecto al origen. Cada una de estas restricciones, L_{0i}, limita aisladamente el dominio de los posibles valores de la variable X_i, mientras las restricciones L_{ij} con i, j >0 limitan conjuntamente los dominios de valores asignables a X_i y X_j. Supondremos que las restricciones L_{ij} y L_{ji} se definen en forma simétrica: $\pi_{ij}(m) = \pi_{ji}(-m)$.

Definición 2.- *Una* solución *σ-posible de la FCN $\mathcal{N}$ es una* n-*tupla* $s = (x_1, \ldots x_n) \in \mathcal{R}^n$ *que verifica $\pi_s(s) = \sigma$, donde π_s es:*

$$\pi_s(s) = \min_{i,j \leq n} \pi_{ij}(x_j - x_i)$$

La distribución de posibilidad π_s define el conjunto S de posibles soluciones de la red, que son aquellas que satisfacen todas las restricciones en algún grado no nulo. Restringimos nuestro interés a las redes que admiten soluciones completamente posibles, es decir, con π_s normalizado.

Definición 3.- *Una FCN $\mathcal{N}$ es* consistente *si existe al menos una solución* s *que verifica $\pi_s(s) = 1$.*

Definición 4.- *Dos FCN $\mathcal{H}$ y $\mathcal{N}$ son* equivalentes *si y sólo si toda solución σ-posible de una de ellas es también solución σ-posible de la otra:*

$$\pi_s^H(s) = \pi_s^N(s),\ s \in \mathcal{R}^n.$$

En general, siempre es posible encontrar una red equivalente a otra dada $\mathcal{N}$, pero cuyas restricciones estén incluidas en las correspondientes restricciones de $\mathcal{N}$. La nueva red, sin perder soluciones de $\mathcal{N}$, nos define las distancias entre variables en forma más precisa. La red equivalente a $\mathcal{N}$ cuyas restricciones son mínimas respecto a la inclusión es denominada red *mínima* $\mathcal{M}$ asociada a $\mathcal{N}$. En otros trabajos hemos demostrado que las restricciones M_{ij} de la red mínima pueden obtenerse como:

$$M_{ij} = \bigcap_{k=1}^{n} L_{ij}^k,$$

siendo:

$$L_{ij}^k = \bigcap C_{i_0,i_1,\ldots,i_k}^k,\ i_1 \ldots i_{k-1} \leq n,\ i_0 = i,\ i_k = j;$$

$$C_{i_0,i_1,\ldots,i_k}^k = \sum_{p=1}^{k} L_{i_{p-1},i_p}.$$

Puede demostrarse también que la red $\mathcal{N}$ es consistente si y sólo si las restricciones M_{ij} así obtenidas resultan normalizadas. Además, la red mínima verifica que: $M_{ij} \subseteq M_{ik} \bigoplus M_{kj}$, i, j, k $\leq$ n. La detección de inconsistencias y la obtención de la red mínima se implementan computacionalmente mediante un algoritmo de camino-consistencia que se ejecuta en tiempo polinómico [3, 8, 9].

3 Obtención de soluciones

Consideremos el siguiente problema: Dada una FCN consistente mínima $\mathcal{M}$, obtener una solución s cualquiera de grado de posibilidad $\pi_s(s) \geq \sigma$, $\sigma \in [0, 1]$. Un algoritmo para la obtención de soluciones, que no requiere "backtracking", es el siguiente:

Algoritmo 1:

```
begin
    X0 := 0;
    for i := 1 to n do
                F = ⋂_{p<i} (x_p ⊕ M_pi);
                Seleccionar x_i | π_F(x_i) ≥ σ ;
                X_i := x_i;
    s := (x_1, x_2, ... x_n);
end
```

Teorema 1.- *Dada una FCN consistente mínima $\mathcal{M}$, la* n-*tupla $s = (x_1, x_2, \ldots x_n)$ obtenida por el algoritmo 1 es una solución de la red $\mathcal{M}$ con grado de posibilidad $\pi_s(s) \geq \sigma$.*

Demostración (por inducción):

Se trata de ver que, en cada repetición i del lazo: a) existe un x_i que verifica la condición establecida en la sentencia de selección, y b) las asignaciones realizadas hasta i verifican todas las restricciones que les afectan con grado de posibilidad mayor o igual que σ. Para $i = 1$ es obvio que existe un valor x_i que verifica $\pi_F(x_i) \geq \sigma$, pues, por ser $\mathcal{M}$ consistente, $F = M_{01}$ corresponde a una distribución normalizada, y es la única restricción que afecta a X_0 y X_1. En la i-ésima repetición, por hipótesis, todas las asignaciones previas a i verifican las restricciones que les afectan:

$(x_j - x_k) \in M^{\sigma}_{kj}$, k, j < i.

donde M^{σ}_{kj} es el σ-corte de la restricción M_{kj}. Por ser $\mathcal{M}$ mínima

$M^{\sigma}_{kj} \subseteq [M_{ki} \bigoplus M_{ij}]^{\sigma} = M^{\sigma}_{ki} \bigoplus M^{\sigma}_{ij}$

Consecuentemente, y teniendo en cuenta que los σ-cortes son intervalos cerrados,

$\min M^{\sigma}_{ki} + \min M^{\sigma}_{ij} \leq (x_j - x_k) \leq \max M^{\sigma}_{ki} + \max M^{\sigma}_{ij}$

Teniendo en cuenta que:

$\min M^{\sigma}_{ij} = -\max M^{\sigma}_{ji}$

y:

$x_j + \min M^{\sigma}_{ji} = \min (x_j \bigoplus M_{ji})^{\sigma}$

se obtiene que:

$$\max \{\min [x_k \bigoplus M_{ki}]^{\sigma}, \min [x_j \bigoplus M_{ji}]^{\sigma}\} \leq \min \left\{\max (x_k \bigoplus M_{ki})^{\sigma}, \max (x_j \bigoplus M_{ji})^{\sigma}\right\}$$

lo que significa que:

$[x_k \bigoplus M_{ki}]^{\sigma} \cap [x_j \bigoplus M_{ji}]^{\sigma} \neq \emptyset$. Como esta expresión se verifica para cualesquiera subíndices k, j < i, y los σ-cortes son convexos, se tiene que

$$F^{\sigma} = \bigcap_{p<i} [x_p \bigoplus M_{pi}]^{\sigma} \neq \emptyset.$$

Además, como $\pi_F(x_i) \geq \sigma$, la asignación $X_i := x_i$ satisface todas las restricciones M_{pi}, p < i, con grado de posibilidad mayor o igual que σ.

Por simplicidad, hemos supuesto que las variables se recorren en el orden correspondiente a subíndices crecientes. Es evidente que siempre se obtiene una solución, cualquiera que sea el orden en que se recorran las variables y cualquiera que sea el proceso de selección aplicado en el interior del lazo.

4 Ordenación de variables

Consideremos ahora el problema consistente en la ordenación de las variables de una FCN mínima mediante una función ordenadora f que proyecte cada variable X_i en un número real $f(X_i) \in \mathcal{R}$, de tal forma que $f(X_i)$ sea un valor representativo de la variable y el orden natural de los valores $f(X_i)$ sobre la recta real sea una ordenación representativa de las posibles ordenaciones de las variables. En los métodos convencionales de ordenación de números borrosos, se acepta como hipótesis subyacente que cada número X_i puede ser definido mediante una distribución de posibilidad π_i independiente de las restantes. Esta hipótesis no se verifica en el caso de las FCN.

Podemos interpretar la FCN como una definición *global* de los valores posibles de las n variables $X_1, X_2 \ldots X_n$ a partir de la definición de las restricciones *locales* que deben verificar las variables. En general, no es posible obtener una definición local de las n variables mediante n números borrosos independientes, sin perder información. En particular, las restricciones mínimas relativas al origen, M_{0i}, definen sólo los dominios de valores posibles de las variables, pero no contienen toda la información de la red. Si elegimos un valor al azar de cada dominio mínimo, $x_i \in M_{0i}$, la n-tupla resultante, $(x_1, x_2 \ldots x_n)$, no verificará necesariamente todas las restricciones de la red, y no será, en general, una solución de la FCN. Lo único cierto es que cada valor $x_i \in M_{0i}$ puede formar parte de una solución $s=(x_1, x_2 \ldots x_n)$ cuyo grado de posibilidad es $\pi_s(s) = \pi_{0i}(x_i)$; para obtener esta solución hay que aplicar el algoritmo 1, tomando X_i como variable inicial del lazo. Si X_i no se toma como variable inicial, el valor x_i puede quedar excluido del conjunto F de valores posibles de la variable X_i, que contiene sólo los valores compatibles con las asignaciones previas. Consecuentemente, la aplicación de una función ordenadora convencional conduce, en general, a la obtención de n-tuplas que no son solución de la red.

Existen otros criterios que pueden aplicarse para evaluar la calidad de un método de ordenación de variables de una FCN. Entre ellos, están los criterios exigidos a los métodos convencionales de ordenación de números borrosos: complejidad, facilidad de interpretación, consistencia con la intuición subjetiva, robustez, flexibilidad,... [1, 2]. A estos criterios convencionales añadimos los siguientes cuatro criterios, cuya deseabilidad en relación con el modelo FCN parece obvia:

Criterio C1.-*La n-tupla de valores reales obtenida mediante el método de ordenación* f *debe ser una solución de la FCN:*

$\pi_s(f(X_1), f(X_2) \ldots f(X_n)) > 0.$

Criterio C2.-*Dado un nivel de consistencia* $\sigma \in [0, 1]$ *previamente fijado, el método de ordenación* f *debe ser capaz de obtener una solución cuyo grado de posibilidad sea mayor o igual que* σ :

$\pi_s(f(X_1), f(X_2) \ldots f(X_n)) \geq \sigma, \sigma \in [0, 1]$

Criterio C3.-*La solución s obtenida por el método de ordenación* f *debe ser independiente del orden en que* f *se aplique a las variables. Es decir, para cualquier permutación* $i_1, i_2 \ldots i_n$ *de los subíndices de las n variables, la secuencia de aplicaciones* $f(X_{i_1}), f(X_{i_2}) \ldots f(X_{i_n})$ *debe conducir a la misma solución.*

Criterio C4.- *Si* $f(X_i)$ *se obtiene en forma separable (mediante la aplicación de un operador P a la restricción absoluta* M_{0i}*), debe verificarse que:*

$\forall\ i,j \leq n,\ f(X_i) = P(M_{0i}) \wedge f(X_j) = P(M_{0j}) \Rightarrow f(X_j) - f(X_i) = P(M_{ij})$

Ya hemos discutido la idoneidad del criterio C1. El criterio C2 es una generalización del criterio de flexibilidad establecido por Zhu y Lee (1992) para métodos convencionales, y permite que el usuario fije de antemano el grado de posibilidad σ de la solución buscada. En particular, haciendo $\sigma = 1$ se obtendrá una solución completamente posible; sólo en este caso el grado de certeza podrá ser no nulo. El criterio C3 garantiza que el resultado de aplicar un método algorítmico al conjunto de variables de la red sea independiente del orden en que el algoritmo recorre las variables. Finalmente, el criterio C4 sólo tiene interés para los métodos de tipo separable, que son los que operan únicamente con restricciones absolutas. Un método separa-

ble tiene la ventaja de la eficiencia, pues se reduce a aplicar el operador P a las n restricciones absolutas $M_{01}, M_{02} \ldots M_{0n}$. Sin embargo, y dado que las restricciones absolutas no contienen toda la información de la red, los métodos separables sólo pueden verificar el criterio C4 para ciertas topologías particulares de red. Omitimos aquí un estudio más detallado de estas topologías [3], indicando sólo que corresponden a dos casos de interés en el ámbito del razonamiento temporal (modelo de fechas absolutas y árbol de fechas relativas). Para topologías generales, los métodos separables son válidos sólo como aproximación eficiente. A continuación, presentamos algunos métodos de ordenación de variables en FCN, y analizamos su calidad en relación con estos criterios.

Una primera familia de métodos es la formada por los métodos que consisten en la aplicación del algoritmo 1, usando un operador de selección fijo. El operador de selección puede ser cualquiera que cumpla la condición $\pi_F(x_i) \geq \sigma$ (por ejemplo, selección del punto medio de F^{σ}). Cualquier método basado en el algoritmo 1 verifica los criterios C1 y C2, pero no el criterio C3. Aunque no es un método separable, se trata de un método eficiente. En lo que respecta a la familia de métodos separables, proponemos los siguientes:

Método f_1.- *Dada una FCN consistente mínima $\mathcal{M}$ y dos parámetros σ , $\lambda \in [0,1]$, asignar a cada variable X_i el valor x_i dado por:*

$f_1(X_i) = \lambda \cdot max\{x/\pi_{0i}(x) \geq \sigma\}$

$+(1-\lambda) \cdot min\{x/\pi_{0i}(x) \geq \sigma\}$

Teorema 2.- *El método f_1 obtiene una solución de la FCN consistente mínima $\mathcal{M}$ cuyo grado de posibilidad es mayor o igual que σ.*

Demostración:

Llamemos:

$f_2(X_i) = \max\{x/\pi_{0i}(x) \geq \sigma\} = \max M^{\sigma}_{0i}$

$f_3(X_i) = \min\{x/\pi_{0i}(x) \geq \sigma\} = \min M^{\sigma}_{0i}$

donde M^{σ}_{0i} es el σ-corte de la restricción M_{0i}. Dado que $\mathcal{M}$ es mínima, $M_{0j} \subseteq M_{0i} \bigoplus M_{ij}$,y dado que los σ-cortes son intervalos cerrados:

$M^{\sigma}_{0j} \subseteq [M_{0i} \bigoplus M_{ij}]^{\sigma} = M^{\sigma}_{0i} \bigoplus M^{\sigma}_{ij}$

$f_2(X_j) \leq f_2(X_i) + \max M^{\sigma}_{ij}$

$f_3(X_j) \geq f_3(X_i) + \min M^{\sigma}_{ij}$

De forma similar, a partir de $M_{0i} \subseteq M_{0j} \bigoplus M_{ji}$ se obtiene:

$f_2(X_i) \leq f_2(X_j) + \max M^{\sigma}_{ji}$

$f_3(X_i) \geq f_3(X_j) + \min M^{\sigma}_{ji}$

Como $\max M^{\sigma}_{ji} = -\min M^{\sigma}_{ij}$:

$\min M^{\sigma}_{ij} \leq f_2(X_j) - f_2(X_i) \leq \max M^{\sigma}_{ij}$

$\min M^{\sigma}_{ij} \leq f_3(X_j) - f_3(X_i) \leq \max M^{\sigma}_{ij}$

Sumando las dos expresiones anteriores, multiplicadas por λ y $1-\lambda$, respectivamente, se obtiene:

$\min M^{\sigma}_{ij} \leq f_1(X_j) - f_1(X_i) \leq \max M^{\sigma}_{ij}$;

$\pi_{ij}(f_1(X_j) - f_1(X_i)) \geq \sigma$;

$\pi_s(f_1(X_1) \ldots f_1(X_n)) = \min_{i,j} \pi_{ij}(f_1(X_j) - f_1(X_i))$

$\geq \sigma$

En particular, haciendo $\sigma = 1$, el método f_1 obtiene soluciones completamente posibles, que siempre existirán por ser la red consistente. La solución concreta obtenida para un σ dado, depende del factor de escala λ. Tres casos particulares, correspondientes a distintos valores λ son los métodos siguientes:

Método f_2 ($\lambda = 1$).- *Dada una FCN consistente mínima $\mathcal{M}$, asignar a cada variable X_i el valor:*

$f_2(X_i) = max\{x/\pi_{0i}(x) \geq \sigma\}$

Método f_3 ($\lambda = 0$).- *Dada una FCN consistente mínima $\mathcal{M}$, asignar a cada variable X_i el valor:*

$f_3(X_i) = min\{x/\pi_{0i}(x) \geq \sigma\}$

Método f_4 ($\lambda = 1/2$).- *Dada una FCN consistente mínima $\mathcal{M}$, asignar a cada variable X_i el valor:*

$f_4(X_i) = 1/2\ (f_2(X_i) + f_3(X_i))$

Obsérvese que f_2 corresponde al método propuesto por Adamo (1980). Por tanto, se trata de un método convencional que puede ser aplicado a las restricciones absolutas para obtener una solución. En términos de una aplicación a razonamiento temporal, f_2 proporciona la solución más tardía posible para un nivel de consistencia σ dado. Análogamente, f_3 proporciona la solución más temprana. Tomando $\sigma = 1$, f_2 y f_3 parecen apropiados para resolver consultas del tipo "último evento" o "primer evento", respectivamente. f_4 proporciona una solución representativa obtenida como promedio de los puntos de máxima pertenencia. Si bien los cuatro métodos satisfacen los criterios C1, C2 y C3, ninguno de ellos satisface el criterio C4, como es fácil comprobar.

Finalmente, introducimos un método f_5, que verifica los criterios C1, C2, C3, no es un método separable, y como tal está exento del requerimiento C4, aunque su complejidad es superior a la de los métodos restantes.

Método f_5.- *Dada una FCN consistente mínima $\mathcal{M}$, y un parámetro $\sigma \in [0,1]$, asignar a cada variable X_i el valor $f_5(X_i)$ dado por:*

$$f_5(X_i) = \frac{\int_{I_n}^{E_n} \cdots \int_{I_2}^{E_2} \int_{I_1}^{E_1} X_i \cdot dX_1 dX_2 \ldots dX_n}{\int_{I_n}^{E_n} \cdots \int_{I_2}^{E_2} \int_{I_1}^{E_1} dX_1 dX_2 \ldots dX_n}$$

donde:

$I_j = min\ P^{\sigma}_j$

$E_j = max\ P^{\sigma}_j$

$$P^{\sigma}_j = M^{\sigma}_{0j} \cap \left[\bigcap_{j<p\leq n} (X_p \bigoplus M^{\sigma}_{pj}) \right] =$$

$P^{\sigma}_j(X_{j+1}, X_{j+2} \ldots X_n)$

Teorema 3.- *Dada una FCN consistente mínima $\mathcal{M}$, y un parámetro $\sigma \in [0,1]$,el método f_5 obtiene una solución s cuyo grado de posibilidad es mayor o igual que σ.*

Demostración:

El método f_5 obtiene el centro de gravedad (cdg) de la región S^σ la cual contiene todas las n-tuplas que verifican $\pi_s(s) \geq \sigma$, y $f_5(X_i)$ es la coordenada i-ésima del cdg. Por tanto, basta demostrar que todo σ-corte S^σ es convexo para asegurar que $\hat{s} = (f_5(X_1), f_5(X_2) \ldots f_5(X_n)) \in S^\sigma$. Demostraremos que S es un número borroso n-dimensional convexo. En efecto, para cualquier $\lambda \in [0,1]$ y cualesquiera dos n-tuplas $s_x = (x_1, x_2 \ldots x_n)$ y $s_y = (y_1, y_2 \ldots y_n)$:

$$\pi_s(\lambda \cdot s_x + (1-\lambda) \cdot s_y) =$$
$$\pi_s(\lambda \cdot (x_1, x_2 \ldots x_n) + (1-\lambda) \cdot (y_1, y_2 \ldots y_n)) =$$
$$= \pi_s(\lambda \cdot x_1 + (1-\lambda) \cdot y_1, \lambda \cdot x_2 + (1-\lambda) \cdot y_2,$$
$$\ldots, \lambda \cdot x_n + (1-\lambda) \cdot y_n) =$$
$$= \min_{i,j} \pi_{ij}(\lambda \cdot x_j + (1-\lambda) \cdot y_j - \lambda \cdot x_i - (1-\lambda) \cdot y_i)$$

Por ser M_{ij} una restricción convexa:

$$\pi_{ij}(\lambda \cdot m + (1-\lambda) \cdot m') \geq \min\{\pi_{ij}(m), \pi_{ij}(m')\}, \forall \lambda, m, m', i, j.$$

En particular, para $m = x_j - x_i$ y $m' = y_j - y_i$:

$$\pi_{ij}(\lambda \cdot x_j + (1-\lambda) \cdot y_j - \lambda \cdot x_i - (1-\lambda) \cdot y_i) \geq \min\{\pi_{ij}(x_j - x_i), \pi_{ij}(y_j - y_i)\}$$

Por tanto:

$$\pi_s(\lambda \cdot s_x + (1-\lambda) \cdot s_y) \geq \min\{\pi_s(s_x), \pi_s(s_y)\}.$$

Obsérvese que las componentes de $\hat{s} = (f_5(X_1), f_5(X_2) \ldots f_5(X_n))$ se obtienen mediante integrales de volumen, que son independientes del orden de integración de las variables, verificándose, por tanto, el criterio C3.

5 Conclusiones

En este trabajo hemos presentado varios métodos de ordenación de variables borrosas, aplicables a redes de restricciones borrosas. Hemos identificado algunos criterios de calidad que establecen propiedades deseables para los métodos de ordenación de variables dependientes, discutiendo las características de los métodos propuestos en relación con estos criterios.

El rango principal de aplicaciones de los métodos propuestos es el razonamiento temporal, tanto en problemas de planificación de tareas como en problemas de diagnóstico a partir de información independiente del tiempo. En el primer caso, el problema central es obtener una solución u ordenación temporal de tareas compatibles con las restricciones establecidas. En el segundo caso, el problema central es identificar patrones temporales de síntomas clínicos concordantes con las relaciones causales del proceso fisiopatológico subyacente. Para ello, los agentes de razonamiento sobre el dominio consultan a un especialista temporal sobre la existencia de relaciones temporales concretas entre los síntomas, inferibles a partir de la información temporal imprecisa introducida por el usuario. Las consultas planteadas por los agentes de razonamiento pueden ser complejas, incluyendo preguntas del tipo: "El primer evento del síntoma A ha ocurrido menos de aproximadamente 24 horas antes del último evento del síntoma B?" Este tipo de preguntas, que incluyen selectores temporales, requieren la ordenación previa de las historias de eventos.

En relación con la familia de métodos separables asociados a f_1, es necesario un estudio más exhaustivo sobre la influencia del parámetro λ sobre la calidad subjetiva de la respuesta. El factor de escala λ actúa como un parámetro de posicionamiento sobre los σ-cortes M_{0i}^σ. Es justamente la aplicación de una regla de posicionamiento fija lo que permite respetar la dependencia entre variables, garantizando la obtención de una solución. Existe, obviamente, cierto grado de arbitrariedad en la elección del valor de λ. En general, en aquellos casos en que la respuesta intuitiva es evidente, ésta coincide con f_1 cualquiera que sea el valor de λ. Unicamente en aquellos casos en que la intuición no es evidente, la elección de λ es crítica. A la espera de una evaluación completa, λ=0 y λ=1 parecen apropiados para la resolución de consultas del tipo "primer evento" y "último evento", respectivamente.

El razonamiento temporal no es el único rango de aplicaciones de las redes de restricciones borrosas en el que la ordenación de variables y la obtención de soluciones son necesarias. Nuestro grupo ha propuesto un modelo de adquisición de conocimiento para tareas de decisión multicriterio, en el que los atributos de las alternativas se establecen en forma relativa mediante restricciones métricas borrosas, y en el que se requiere un proceso de "defuzzificación" del conocimiento, como fase previa a la aplicación del proceso de decisión. Este modelo se aplica en un problema de minimización del impacto medioambiental: la selección de los productos fitosanitarios utilizados para el control de plagas de insectos en cultivos bajo invernaderos.

Agradecimientos

Este trabajo ha sido financiado por la Consejería de Educación y Cultura de la Región de Murcia (proyecto PCT94/48), por la Comisión Interministerial de Ciencia y Tecnología (proyecto TIC95-064-C02), y parcialmente por el CDTI (proyecto PACE/PASO PR191/93).

Referencias

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

[2] Q. Zhu, E.S. Lee, Comparison and Ranking of Fuzzy Numbers. Int. J. Kacprzyk and M. Fedrizzi, Eds. Fuzzy Regression Analysis. Onmitech Press Warsan and Physica-Verlag, Heidelberg, 1992, 21-44.

[3] R. Marín, Razonamiento Temporal, XV Escuela de Verano de Informática. Asociación Española de Informática y Automática, 1993.

[4] S. Barro, R. Marín, R.P. Otero, R. Ruiz, J. Mira, On the handling of time in intelligent monitoring of CCU patients. Proc. of the 14th Annual International Conference of the IEEE

Engineering in Medicine and Biology Society ,1992, 871- 873.

[5] S. Barro,R. Marín, J. Mira, A.R. Patón, A model and a language for the fuzzy representation and handling of time, Fuzzy Sets and Systems, 61 ,1994, 153-175.

[6] R. Dechter, I. Meiri, J. Pearl, Temporal constraint networks, Artificial Intelligence 49 ,1991, 61-95.

[7] L. Godo, L. Vila, A temporal reasoning system based on fuzzy temporal constraints. Actas del Cuarto Congreso Español sobre Tecnologías y Lógica Fuzzy ,1994, 43-48.

[8] R. Marín, S. Barro, A. Bosch, J. Mira, Modeling time representation from a fuzzy perspective. Proc. of EUROCAST'93, 1993, 81-84.

[9] R. Marín, S. Barro, A. Bosch, I. Navarrete, M. Cárdenas, A. Bugarín, J. Mira, Un modelo para la representación de información temporal borrosa. Actas del Tercer Congreso Español sobre Tecnologías y Lógica Fuzzy ,1993, 209-216.

[10] R. Marín, S. Barro, A. Bosch, J. Mira, Modeling time representation from a fuzzy perspective. Cybernetics and Systems 25, 2 ,1994, 207-215.

[11] L. Vila, L. Godo, Query answering in fuzzy temporal constraints. Actas del Cuarto Congreso Español sobre Tecnologías y Lógica Fuzzy ,1994, 65-70.

[12] M. Delgado, J.L. Verdegay, M.A. Vila, A procedure for ranking fuzzy relations. Fuzzy Sets and Systems, 26, 1988, 49-62.

[13] M. Jiménez, Ordenación de números borrosos mediante la comparación de sus intervalos esperados. Actas del IV Congreso Español sobre Tecnologías y Lógica Fuzzy, 1994, Blanes, 203-208.

[14] F.Herrera, E. Herrera-Viedma, J.L. Verdegay, Un proceso de decisión en grupo con preferencias lingüísticas y selección secuencial. Actas del IV Congreso Español sobre Tecnologías y Lógica Fuzzy, 1994, Blanes, 215-220.

Patrones Temporales Borrosos en la Supervisión de Pacientes *

P. Félix[1], S. Barro[1], R. Marín [2], M.J. Taboada[1] y A. Engel[3].
[1] Departamento de Electrónica e Computación
Facultade de Física. Universidade de Santiago de Compostela.
e-mail: elsenen@usc.es
[2] Departamento de Informática y Sistemas
Universidad de Murcia
[3] Dept. of Mathematics
Rochester Institute of Technology

Resumen

En este trabajo introducimos el concepto de "Patrón Temporal Borroso" (PTB), que permite representar la imprecisión asociada a la definición de un perfil o patrón de interés en la evolución temporal del valor de una variable. Introducimos mecanismos de razonamiento sobre PTB que permiten su minimización local y global y su equiparación con los datos conocidos para la variable a la que hace referencia. Por último valoramos su utilidad en el contexto de un problema médico: la supervisión de pacientes en la fase aguda del Infarto de Miocardio.

Palabras clave: *Problemas de satisfacción de restricciones, razonamiento temporal, razonamiento cualitativo, aritmética borrosa.*

1 Introducción

En muchos dominios, y el de la medicina es quizás un caso paradigmático, un patrón o perfil específico en la tendencia o evolución de una o más variables (señales) puede tener para un experto del dominio un significado concreto, de modo que su detección y valoración tenga un claro interés en el contexto de un problema dado.

Si se dispone de un modelo computable para la representación del conocimiento del experto, estos patrones podrán ser definidos por él de un modo más o menos preciso, permitiendo su incorporación a un sistema informático. Razonar en base a estos patrones exige, además, disponer de mecanismos adecuados para su comparación o equiparación con las señales o datos introducidos al sistema.

En la medida en que estos patrones estarán a menudo definidos de modo impreciso, creemos que la teoría de conjuntos borrosos es una herramienta formal adecuada para modelar e implementar la adquisición del conocimiento y el razonamiento en base a patrones temporales.

*Este trabajo ha sido financiado por la Xunta de Galicia, por la Consejería de Cultura y Educación de la Región de Murcia y por la CICYT, a través de los proyectos de investigación XUGA-20603B94, PCT94/48 y TIC95-0604, respectivamente.

Con esta idea, en este trabajo introducimos el concepto de "Patrón Temporal Borroso" (PTB), propuesto como una extensión de los modelos de representación precisa y cualitativa de tendencias. Consideramos como punto de partida la descripción de la evolución en el tiempo de una variable en términos de una serie de restricciones borrosas sobre puntos significativos del PTB, definidos éstos a través de un valor borroso y su disposición en el tiempo, también borrosa. Las restricciones borrosas establecidas entre puntos significativos consecutivos restringen de forma directa (influencia local) la información asociada a dichos puntos significativos, y de forma indirecta (influencia global) la asociada al conjunto de puntos significativos del PTB. Considerar la influencia local (minimización local) y global (minimización global) nos lleva a obtener el PTB mínimo, con el que se operará en un dominio problema como con una pieza más de conocimiento, susceptible de formar parte del proceso de razonamiento llevado a cabo en un sistema basado en conocimiento.

2 Representación de patrones temporales borrosos

Un Patrón Temporal Borroso (PTB) será un modelo mediante el que describiremos la evolución de un determinado parámetro físico V en el tiempo. Como es clásico en el razonamiento cualitativo [6], partiremos de una descripción de la evolución problema mediante tramos. Así, definiremos un conjunto de puntos significativos $(X_0, X_1, ..., X_n)$, y aproximaremos la curva evolución mediante tramos lineales entre estos puntos. Cada tramo vendrá definido mediante una restricción imprecisa entre los puntos conectados por el tramo. Esta restricción limitará los posibles valores de distancia temporal entre los puntos (imprecisión en los instantes temporales), de diferencia entre los valores de la variable en los dos puntos (imprecisión en los valores del parámetro físico), y de pendiente del tramo (imprecisión en el ritmo de variación del parámetro).

2.1 Definiciones previas

Introducimos previamente los conceptos de *valor* e *incremento borroso* sobre los que se apoyan los conceptos propios del modelo.

Definición 1 *Dado un universo de discurso U (en lo que nos concierne, $\Re$ ó $[m,n] \subset \Re$) extendemos el concepto de valor a lo que denominaremos* **valor borroso** *c, representado por una distribución de posibilidad π_c, normalizada y unimodal, sobre U. De este modo, dado un valor preciso $v \in U$, $\pi_c(v) \in [0,1]$ representa la posibilidad de que c sea precisamente v. Los valores extremos 1 y 0, representan, respectivamente, la posibilidad absoluta y la posibilidad nula de que c sea v. Por medio de π_c podemos definir un subconjunto borroso C de U, que contiene valores posibles de c, asumiendo que C es un conjunto disjunto, en el sentido de que sus elementos representan alternativas mutuamente excluyentes para c. Considerando μ_C como la función de pertenencia asociada con C, tendremos que*

$$\forall v \in U,\ \pi_c(v) = \mu_c(v).$$

En general, usaremos indistintamente funciones de pertenencia y distribuciones de posibilidad asociadas a los distintos conceptos que definiremos.
Llamaremos **soporte** *de un subconjunto borroso C al conjunto de valores $v \in \Re$ tales que $\mu_C(v) > 0$, y* **núcleo** *de C a cualquier conjunto de valores $v \in \Re$ tales que $\mu_C(v) = 1$. Siempre asumiremos que π_c está* **normalizada**, *esto es, que el núcleo de C es un conjunto no vacío, de modo que existe al menos un v completamente posible para c:*

$$\exists v \in \Re,\ \pi_c(v) = 1.$$

También consideraremos que π_c es **unimodal**. *Esto es:*

$$\forall v, v', v'' \in \Re, v < v' < v'', \pi_c(v') \geq min\{\pi_c(v), \pi_c(v'')\}.$$

Resulta evidente que el concepto de fecha borrosa [1, 2] es un caso particular de valor borroso, cuando U es el dominio del tiempo.

Definición 2 *Introducimos el concepto* **incremento borroso** *con el fin de representar cantidades, como, por ejemplo, la diferencia entre dos valores. Un incremento borroso d se representa por una distribución de posibilidad, π_d, normalizada y unimodal, también definida sobre U. De este modo, dado un $\delta \in U, \pi_d(\delta) \in [0,1]$ representa la posibilidad de que d sea precisamente δ. Como en la definición de valor, por medio de π_d podemos definir un subconjunto borroso disjunto D de U, que contiene todos los posibles valores de d. Dado un par ordenado de valores borrosos (a,e), la distancia entre a y e viene dada por un incremento borroso. Esta distancia se representa por medio de una distribución de posibilidad $\pi_{D(a,e)}$, sobre U, donde dado un $\delta \in U, \pi_{D(a,e)}(\delta)$ representa la posibilidad de que la distancia entre los dos valores sea exactamente d:*

$$\forall \delta \in U,\ \pi_{D(a,e)}(\delta) = \sup_{\delta = t-s} min\ \{\pi_a(s), \pi_e(t)\}$$

Esta definición de incremento no sólo proporciona información sobre la separación, sino sobre la disposición relativa de los valores. Más específicamente, $\pi_{D(a,e)}$ no coincide con $\pi_{D(e,a)}$, excepto, obviamente, cuando la distancia viene dada por la singularidad $\{0\}$. Por otra parte, las distribuciones $\pi_{D(a,e)}$ y $\pi_{D(e,a)}$ son simétricas con respecto al elemento 0, esto es, $\pi_{D(a,e)}(\delta) = \pi_{D(e,a)}(-\delta)$, $\forall \delta \in U$.

2.2 Definición del modelo

Un Patrón Temporal Borroso (PTB) será un modelo mediante el que describiremos la evolución de un determinado parámetro físico V en el tiempo. Como ya es clásico en el razonamiento cualitativo [6], partiremos de una descripción de la evolución problema mediante tramos. Así, definiremos un conjunto de puntos significativos $(X_0, X_1, ..., X_n)$, y aproximaremos la curva evolución mediante tramos entre estos puntos. Cada tramo vendrá definido mediante una restricción imprecisa entre los puntos conectados por el tramo. Esta restricción limitará los posibles valores de distancia temporal entre los puntos (imprecisión en los instantes temporales), de diferencia entre los valores de la variable en los dos puntos (imprecisión en los valores del parámetro físico), y de pendiente del tramo (imprecisión en el ritmo de variación del parámetro).

Definición 3 *Definimos* **punto significativo** *asociado a una variable $v(t)$, y lo designamos como X_i, al par formado por un valor V_i, en general borroso, sobre el dominio de definición de la variable $v(t)$, y un valor temporal T_i.*

$$X_i =< V_i, T_i >$$

En ausencia de cualquier restricción, las variables V_i y T_i pueden tomar cualquier valor v_i y t_i precisos.

Definición 4 *Una restricción unitaria L_i sobre un valor temporal T_i se define mediante una distribución de posibilidad $\pi_i^L(t)$ normalizada y unimodal, cuyo universo de discurso es el eje de tiempo $\Re^+$.*

$$\forall\ t \in\ \Re^+\ :\ \pi_i^L(t)\ \in\ [0,1]$$

Dado un instante de tiempo preciso t_i, $\pi_i^L(t_i)$ representa la posibilidad de que T_i tome precisamente el valor t_i.
La restricción unitaria L_i restringe el dominio de valores que pueden ser asignados a T_i a aquellos instantes de tiempo t_i que satisfacen $\pi_i^L(t_i) > 0$. La distribución de posibilidad $\pi_i^L(t)$ asociada a una restricción unitaria induce un subconjunto borroso en el eje de tiempo, al que denotaremos con el mismo símbolo que la restricción L_i. Formalmente, la distribución $\pi_i^L(t)$ corresponde a la distribución de posibilidad de un valor borroso. Por tanto, podemos interpretar una restricción unitaria L_i, como la asignación de un **instante borroso** a la variable T_i.

Definición 5 *Una restricción unitaria D_i sobre un valor del dominio V_i se define, de modo similar a L_i, mediante una distribución de posibilidad $\pi_i^D(v)$ normalizada y unimodal, cuyo universo es $\Re$.*

$$\forall\ v \in\ \Re\ :\ \pi_i^D(v)\ \in\ [0,1]$$

Formalmente, la distribución $\pi_i^D(v)$ corresponde a la distribución de posibilidad de un valor borroso. Por tanto, podemos interpretar una restricción unitaria D_i como la asignación de un **valor borroso** a la variable V_i.

Definición 6 *Una restricción binaria L_{ij} sobre dos valores temporales se define mediante una distribución de posibilidad π_{ij}^L normalizada y unimodal, cuyo universo de discurso es $\Re$.*

$$\forall\, d \in \Re \;:\; \pi_{ij}^L(d) \;\in\; [0,1]$$

Dado un valor preciso d_{ij}, $\pi_{ij}^L(d_{ij})$ representa la posibilidad de que la distancia temporal del valor T_i al valor T_j tome precisamente el valor d_{ij}. La restricción L_{ij} restringe conjuntamente los dominios de valores posibles de T_i y T_j. En ausencia de otras restricciones, las asignaciones $T_i = t_i$ y $T_j = t_j$ son posibles si se satisface $\pi_{ij}^L(t_j - t_i) > 0$. La distribución de posibilidad π_{ij}^L asociada a una restricción binaria, induce un subconjunto borroso en el dominio de distancias temporales, que representaremos como L_{ij}. Formalmente, la distribución π_{ij}^L corresponde a la distribución de posibilidad de una duración borrosa. Por tanto, podemos interpretar una restricción binaria L_{ij} como la asignación de un incremento borroso, al que denominaremos **duración borrosa**, a la distancia entre T_i y T_j.

Definición 7 *Una restricción binaria D_{ij} sobre dos variables del dominio V_i y V_j se define, de modo similar a D_{ij} mediante una distribución de posibilidad π_{ij}^D normalizada y unimodal, cuyo universo de discurso es $\Re$.*

$$\forall\, \delta \in \Re \;:\; \pi_{ij}^D(\delta) \;\in\; [0,1]$$

Formalmente, la distribución π_{ij}^D corresponde a la distribución de posibilidad de un incremento borroso. Por tanto, podemos interpretar una restricción binaria D_{ij} como la asignación de un **incremento borroso** a la distancia entre V_i y V_j.

Definición 8 *Una restricción cuaternaria M_{ij} sobre dos puntos significativos X_i y X_j, se define mediante una distribución de posibilidad π_{ij}^M normalizada y unimodal, cuyo universo de discurso es $\Re$.*

$$\forall\, p \in \Re \;:\; \pi_{ij}^M(p) \;\in\; [0,1]$$

Dado un valor preciso p_{ij}, $\pi_{ij}^M(p_{ij})$ representa la posibilidad de que la pendiente de la recta que une X_i y X_j sea precisamente p_{ij}. La restricción M_{ij} restringe conjuntamente los dominios de V_i, V_j, T_i y T_j. En ausencia de otras restricciones las asignaciones $V_i = v_i$, $V_j = v_j$, $T_i = t_i$ y $T_j = t_j$ son posibles si se satisface $\pi_{ij}^M((v_j - v_i)/(t_j - t_i)) > 0$. La distribución de posibilidad π_{ij}^M asociada a una restricción cuaternaria induce un subconjunto borroso en el dominio de pendientes, que representaremos como M_{ij}. Formalmente, la distribución π_{ij}^M corresponde a la distribución de posibilidad de un valor borroso. Por tanto, podemos interpretar una restricción cuaternaria M_{ij} como la asignación de un valor borroso al que denominaremos **pendiente borrosa**, a la pendiente de la recta que une X_i y X_j.

Definición 9 *Una restricción borrosa R_{ij} sobre dos puntos significativos X_i y X_j es una tripla formada por una duración borrosa L_{ij}, un incremento borroso D_{ij} y una pendiente borrosa M_{ij}.*

$$R_{ij} =< L_{ij}, D_{ij}, M_{ij} >$$

Definición 10 *Definimos un* **Patrón Temporal Borroso** *(PTB) como un conjunto finito de puntos significativos $\{X_1, X_2, ..., X_n\}$, un conjunto finito de restricciones unitarias $\{L_i, D_i\}_{i=1,...,n}$, binarias $\{L_{ij}, D_{ij}\}_{i,j=1,...,n}$ y cuaternarias $\{M_{ij}\}_{i,j=1,...,n}$ definidas sobre esas variables.*

Un PTB puede representarse mediante un grafo dirigido, en el que los nodos se corresponden con puntos significativos, y los arcos se corresponden con restricciones sobre las variables asociadas a los nodos que éstos unen.
Cualquier restricción unitaria $L_i(D_i)$ sobre $T_i(V_i)$ puede transformarse en una restricción binaria equivalente, que delimite la duración temporal (incremento de valor) del origen $X_0 =< L_0, D_0 >$ a X_i. Por tanto, añadiendo una variable adicional $T_0(V_0)$ que represente un origen preciso del eje de tiempo (eje de valores del dominio del parámetro físico), podemos eliminar todas las restricciones unitarias $L_i(D_i)$ y convertirlas en restricciones binarias equivalentes $L_{0i}(D_{0i})$. A $X_0 =< L_0, D_0 >$ le podemos asignar un valor arbitrario, pero por simplicidad le asignamos el valor $L_0 = 0, D_0 = 0$. De esta forma, $L_{0i} = L_i, D_{0i} = D_i$.

Definición 11 *Decimos que un punto significativo X_i es* **anterior** *a otro X_j si y sólo si se verifica que $L_{ij} > 0$. Sólo consideraremos puntos significativos ordenados mediante la relación $L_{ij} > 0$ o bien $L_{ij} < 0$.*

Supondremos que toda restricción de la forma $L_{ii}(D_{ii})$ es igual al conjunto convencional $\{0\}$. Es decir, se verifica $\pi_{ii}^{L,D}(0) = 1$ y $\pi_{ii}^{L,D}(x) = 0, \forall\, x \neq 0$.

Definición 12 *Llamamos PTB* **secuencial** *(PTBS) a aquél en el que sólo se permite una introducción secuencial de los puntos significativos (en el sentido de que cualquier referencia relativa entre puntos significativos lo será entre aquellos consecutivos y en el orden de ocurrencia, o bien entre el origen y cualquier otro).*

En ocasiones un PTBS puede no resultar suficientemente expresivo, tal es el caso de cuando el experto no dispone de una descripción secuencial de la evolución de la variable de estudio, con referencias del tipo "pocos segundos antes del último máximo". En estos casos, en los que cualquier punto significativo puede ser referencia de otro, configuran un PTB que denominaremos *aleatorio*.

En el problema del dominio médico que luego introduciremos, y en el que estamos interesados, podemos asumir como suficiente la utilización del PTBS, sobre cuyos aspectos concretos nos centraremos a continuación.

3 Minimización de un PTBS

La construcción de un PTBS se llevará a cabo en general a través de un proceso de **elicitación** del conocimiento de un experto, y será normalmente expresado de un modo lingüístico. Este proceso supondrá en general la presencia de información redundante y/o contradictoria. Para detectar la presencia de inconsistencias y eliminar la información superflua se lleva a cabo la minimización del PTBS. Planteamos dicha minimización en dos niveles: minimización local y minimización global.

3.1 Minimización local

La minimización local trata la presencia de redundancia y/o inconsistencia interna en cada restricción. Ésta minimización forma parte fundamental del cálculo de la minimización global, que analizaremos en el siguiente apartado. Mostramos a continuación su solución.

Dada una restricción $R_{ij} =< L_{ij}, D_{ij}, M_{ij} >$ se obtiene la restricción mínima: $W_{ij} =< L'_{ij}, D'_{ij}, M'_{ij} >$ donde[1]:

$$\begin{aligned} M'_{ij} &= M_{ij} \cap (D_{ij}/L_{ij}) \\ D'_{ij} &= D_{ij} \cap (M_{ij} * L_{ij}) \\ L'_{ij} &= L_{ij} \cap \{(-D^N_{ij}/-M^N_{ij}) \cup (D^P_{ij}/M^P_{ij}) \cup \\ &\quad \cup Q(D^Z_{ij}, M^Z_{ij})\} \end{aligned} \tag{1}$$

siendo:

$$\begin{aligned} &D^N_{ij} = N \cap D_{ij} \\ &D^P_{ij} = P \cap D_{ij} \\ &D^Z_{ij} = Z \cap D_{ij} \\ &M^N_{ij} = N \cap M_{ij} \\ &M^P_{ij} = P \cap M_{ij} \\ &M^Z_{ij} = Z \cap M_{ij} \\ &Q(D^Z_{ij}, M^Z_{ij}) = \begin{cases} P & si\ M^Z_{ij} \neq \{\emptyset\}\ y\ D^Z ij \neq \{\emptyset\} \\ \{\emptyset\} & en\ otro\ caso \end{cases} \end{aligned} \tag{2}$$

y correspondiendo P, N y Z con las distribuciones:

$$\mu_P(x) = \begin{cases} 1 & si\ x > 0 \\ 0 & si\ x \leq 0 \end{cases} \quad \mu_N(x) = \begin{cases} 1 & si\ x < 0 \\ 0 & si\ x \geq 0 \end{cases}$$

$$\mu_Z(x) = \begin{cases} 1 & si\ x = 0 \\ 0 & si\ x \neq 0 \end{cases}$$

3.2 Minimización global

Dado un PTBS L, buscamos otro PTBS H que, siendo equivalente a L -conteniendo la misma información-, sea más reducido -esté libre de información superflua-. El PTB H contiene restricciones más reducidas que las de L, acotando más los dominios de posibles valores involucrados en el PTBS. Por tanto, será preferible trabajar con H, dado que nos define relaciones de modo más preciso.

De un modo telegráfico, podríamos decir que la minimización global de un PTB se basa en la combinación, para cada restricción temporal (de valor), de la restricción directa -o explícita- entre cada par de puntos significativos, con la información temporal (de valor) asociada a todas las restricciones inducidas -o implícitas- sobre dichos puntos por las otras restricciones del PTBS. Usaremos la minimización local para combinar información de distinta naturaleza en la minimización de cada restricción R_{ij}.

A continuación presentamos las expresiones finales de minimización de un PTBS, aunque no mostramos el desarrollo matemático que lleva a las mismas.

Usaremos la siguiente notación. Dado un camino de longitud k desde un nodo de subíndice i hasta otro nodo de subíndice j:

$$i_0 = i, i_1, i_2, ...i_k = j,$$

representaremos la correspondiente restricción de valor inducida, por $C^{D,k}_{i_0,i_1,...,i_k}$ y la restricción temporal inducida, por $C^{L,k}_{i_0,i_1,...,i_k}$. Ambas vienen dadas por la composición -traducida en suma borrosa que representaremos por $\oplus$- de las restricciones directas y del mismo tipo entre cada par de nodos consecutivos pertenecientes al camino:

$$C^{D,k}_{i_0,i_1,...,i_k} = D_{i_0 i_1} \oplus D_{i_1 i_2} \oplus ... \oplus D_{i_{k-1} i_k} = \sum_{p=1}^{k} D_{i_{p-1} i_p} \tag{3}$$

$$C^{L,k}_{i_0,i_1,...,i_k} = L_{i_0 i_1} \oplus L_{i_1 i_2} \oplus ... \oplus L_{i_{k-1} i_k} = \sum_{p=1}^{k} L_{i_{p-1} i_p} \tag{4}$$

Representaremos por D^k_{ij} (L^k_{ij}), la intersección de las restricciones inducidas correspondientes a todos los caminos de longitud k desde el nodo de subíndice i al nodo de subíndice j:

$$\begin{aligned} D^k_{ij} &= \bigcap_{\substack{i_0=i \\ i_t\ tq.\ \forall i_t \neq 0, i_{t\pm1} \neq 0 \\ i_k=j}} C^{D,k}_{i_0,i_1,...,i_k} \quad i_t = i_{t\pm1} \pm 1 \\ L^k_{ij} &= \bigcap_{\substack{i_0=i \\ i_t\ tq.\ \forall i_t \neq 0, i_{t\pm1} \neq 0 \\ i_k=j}} C^{L,k}_{i_0,i_1,...,i_k} \quad i_t = i_{t\pm1} \pm 1 \end{aligned} \tag{5}$$

Llamaremos D^n_{ij} (L^n_{ij}) -siendo $n+1$ el número de puntos significativos del PTBS- a la intersección de todas las restricciones inducidas correspondientes a caminos de cualquier longitud entre 1 y n que lleven de X_i a X_j:

$$D^n_{ij} = \bigcap_{k=1}^{n} D^k_{ij} \qquad L^n_{ij} = \bigcap_{k=1}^{n} L^k_{ij} \tag{6}$$

Llegados a este punto minimizamos localmente la restricción:

$$< D^n_{ij}, L^n_{ij}, M_{ij} > \xrightarrow[local]{minim.} \psi_{ij} =< L^{\psi}_{ij}, D^{\psi}_{ij}, M^{\psi}_{ij} > \tag{7}$$

La expresión final se consigue mediante la nueva aplicación de (3-7) a un conjunto de restricciones temporales (de valor) en que aquellas que afecten a puntos

[1] Los operadores aritméticos que aparecen tanto en esta sección como en las siguientes son operadores borrosos [5]

significativos consecutivos (excepto X_0 y X_1) serán las calculadas en (7). Así obtenemos la restricción: $\Omega_{ij} =< L_{ij}^{\Omega}, D_{ij}^{\Omega}, M_{ij}^{\Omega} >$, donde:

$$D_{ij}^{\Omega} = \bigcap_{k=1}^{n} D_{ij}^{\psi,k} \cap M_{ij} * \bigcap_{k=1}^{n} L_{ij}^{\psi,k} \quad (8)$$

$$L_{ij}^{\Omega} = \bigcap_{k=1}^{n} L_{ij}^{\psi,k} \cap \{(\bigcap_{k=1}^{n} D_{ij}^{\psi,k})^{P} / M_{ij}^{P} \cup \cup - (\bigcap_{k=1}^{n} D_{ij}^{\psi,k})^{N} / - M_{ij}^{N} \cup \cup Q((\bigcap_{k=1}^{n} D_{ij}^{\psi,k})^{Z}, M_{ij}^{Z})\} \quad (9)$$

$$M_{ij}^{\Omega} = M_{ij}^{\psi} \cap (\bigcap_{k=1}^{n} D_{ij}^{\psi,k}) / (\bigcap_{k=1}^{n} L_{ij}^{\psi,k}) \quad (10)$$

3.3 Comparación de un PTBS con los datos conocidos

El problema básico aquí es el siguiente: dado un PTBS globalmente mínimo y un conjunto de datos precisos P medidos sobre una variable real en diferentes instantes, encontrar el grado de correspondencia entre el patrón temporal y los datos. El conjunto P será:

$$P = \{(p_1, q_1), ..., (p_w, q_w)\} \quad (11)$$

donde p_r es un valor preciso del parámetro V y q_r es un instante de tiempo preciso, que no coincidirá, en general, con el de ningún punto significativo. Aunque podríamos abordar como caso más general aquel en que tanto p_r como q_r sean valores borrosos, ésto no es preciso en la mayor parte de las aplicaciones y, sin embargo, añade complejidad al tratamiento.

Calcularemos el **grado de concordancia** entre los datos y el PTBS, esto es, la medida en que los datos concuerdan con ("pertenecen a") el conjunto de valores asociados al PTBS. La solución se basa en generar previamente la función de pertenencia asociada a la descripción lingüística del PTBS. Para ello usaremos la información relativa asociada a cada tramo (R_{ij}) y la información absoluta asociada al punto de inicio del tramo (R_{0i}). Dado que el PTBS es mínimo, no hay información redundante, por lo que basta usar la pendiente del tramo (M_{ij}) y el inicio del tramo (D_{0i} y L_{0i}) para generar la función de pertenencia correspondiente a cada tramo. Obtenemos la global como la expresión de un mapa de posiciones y restricciones entre ellas [3]. Dicha descripción se comparará con otra, formada por los datos expresados como un conjunto de estados -puntos precisos- y un conjunto de vínculos entre ellos -pendiente de la recta entre cada dos puntos consecutivos-.

4 Orientación a la planificación de terapia en la fase aguda del infarto de miocardio

Los PTBS formalmente definidos en los apartados previos tienen un interés claro en un dominio médico en el que estamos interesados desde hace algún tiempo: la planificación de terapia y seguimiento de pacientes que se encuentran en la fase aguda del infarto de miocardio. El objetivo de nuestro trabajo es el desarrollo de un sistema que tenga la habilidad de razonar, a partir del estado del paciente, sobre el conjunto de objetivos terapéuticos a alcanzar por el médico, el conjunto de acciones o planes a aplicar y el conjunto de estrategias de planificación que permitan adaptar y refinar estos planes al paciente particular.

Aunque orientado a otro dominio -el de la oncología-, quizás sea conveniente comenzar por reflejar algunos aspectos de interés de un sistema bien conocido: ONCOCIN [8], que persigue algunos objetivos de interés en nuestro dominio.

El conocimiento heurístico utilizado para los procesos de selección y refinamiento de planes se conocen en ONCOCIN como directrices ("guidelines") y se utilizan para contrastar el estado del paciente con las acciones de terapia apropiadas. Suelen ser reglas asociadas a cada tratamiento que causan preguntas a la base de datos del paciente, y que resultan necesarias para constrastar su estado con las acciones terapéuticas que se llevaron a cabo. Nuestro objetivo es desarrollar directrices que se adapten más a la forma de razonamiento del médico, permitiendo representar algún tipo de conocimiento impreciso e incierto, y que presenten algún grado de colaboración entre el sistema y el médico. Las directrices que nosotros vamos a desarrollar serán reglas cuya parte antecedente pueda hacer referencia a cambios temporales en la evolución del estado del paciente. La forma en que nosotros vamos a modelar la evolución temporal del estado de un paciente será similar a cómo los expertos describen la evolución temporal de las manifestaciones clínicas observadas en 'el. Los médicos describen patrones temporales de manifestaciones clínicas como una sucesión de eventos con imprecisión en los valores que pueden tomar dichas manifestaciones, imprecisión en la forma de variación e imprecisión en los instantes temporales en los que éstos se producen. Los patrones temporales borrosos constituyen una aproximación de modelado del tiempo que permiten hacer especial énfasis en la representación de cambio del estado del paciente, capturando los efectos que producirán sobre el paciente las acciones terapéuticas.

Por ejemplo, la administración de Nitroglicerina en perfusión en pacientes con dolor isquémico en la fase aguda del infarto de miocardio, persigue, entre otros objetivos, disminuir el consumo de oxígeno por el miocardio. Esta disminución está asociada con la disminución del llamado Doble Producto (DP) (resultado de multiplicar la Tensión Arterial Sistólica (TAS) por la Frecuencia Cardíaca (FC)). Una respuesta "MUY BUENA" a la administración de la Nitroglicerina está asociada con una "disminución evidente del DP" junto con una "reducción evidente de la Tensión Arterial Media (TAM)" (un 10% en pacientes normotensos y hasta un 30% en pacientes hipertensos) y un "aumento moderado de la FC" (hasta un máximo de 20 latidos por minuto). Si la respuesta es satisfactoria, el tratamiento debe de suprimirse de forma paulatina repitiendo el proceso inverso de su administración. Durante la supresión paulatina se espera que el DP y la TAM se mantengan o aumenten ligeramente y la FC se mantenga o se reduzca ligeramente. Por ejemplo, el patrón temporal borroso asociado al DP, correspondiente a una respuesta buena o muy buena se caracteriza por: "una disminución evidente durante

las primeras 12-24 horas y un mantenimiento o ligera disminución durante las 12-24 horas siguientes aproximadamente".

5 Trabajo futuro

El modelo que se presenta de PTB constituye una primera versión de un objetivo más pretencioso, que incluye la formalización de conceptos referidos a la evolución de una variable y que tienen una expresión lingüística de uso común, como el signo de curvatura, o la presencia de puntos de inflexión [4].

Dedicaremos también esfuerzos a la optimización computacional de los cálculos de minimización.

En su vertiente aplicada extenderemos su proyección en el dominio médico, orientándolo a la detección y valoración de episodios de isquemia.

Referencias

[1] S. Barro, R. Marín, J. Mira y A. R. Patón, *A Model and a Language for the Fuzzy Representation and handling of time*, Fuzzy Sets and Systems, no. 61, pp. 153-175, 1994.

[2] D. Dubois, H. Prade, *Processing Fuzzy Temporal Knowledge*, IEEE Trans. on Systems, Man, and Cybernetics, vol. 19, no. 4, pp. 729-744, Julio/Agosto 1989.

[3] D. Dubois, H. Prade, *Systems of Linear Fuzzy Constraints*, Fuzzy Sets and Systems, no. 3, pp. 37-48, 1980.

[4] P. Félix, S. Barro, R. Marín, *Un modelo borroso de representación de tendencias*, informe interno, Mayo 1995.

[5] A. Kaufmann, M.M. Gupta, *Introduction to Fuzzy Arithmetic. Theory and Applications*, Van Nostrand Reinhold Co., 1985.

[6] B. Kuipers, *Qualitative Simulation*, Artificial Intelligence, vol. 29, pp. 289-338, 1986.

[7] M. Ramoni, M. Stefanelli, L. Magnani y G. Barosi, *An Epistemological Framework for Medical Knowledge-Based Systems*, IEEE Transactions on Systems, Man and Cybernetics, vol. 22, no. 6, pp. 1361-1375, 1992.

[8] S. Tu, M. Kahn, M. Musen, J. Ferguson, E. Shortliffe y L. Fagan, *Episodic Skeletal-Plan Refinement Based on Temporal Data*. Communications of the ACM, vol. 32, no. 12, pp. 1439-1455, 1989.

Las Redes Neuronales Artificiales son Sistemas Difusos Aditivos

J.M. Benítez, J.L. Castro e I. Requena
Dpto. Ciencias de la Computación e Inteligencia Artificial
E.T.S. Ingeniería Informática
Avda. Andalucía, 38
Universidad de Granada
18071 - GRANADA
e-mail: jmbs, castro, requena@robinson.ugr.es

Resumen

Las Redes Neuronales Artificiales son modelos eficaces de cálculo que han demostrado sus capacidades en la resolución de difíciles problemas en el ámbito de la Inteligencia Artificial. También se ha demostrado que estas redes son aproximadores universales. Sin embargo, siempre se ha criticado su propiedad de caja negra, *ya que no se conoce ninguna interpretación satisfactoria de su comportamiento. Múltiples trabajos se han hecho en la búsqueda de esta interpretación y en el establecimiento de una equivalencia entre redes y sistemas basados en reglas difusas. El objetivo de esta nota es simplemente comentar la solución inmediata de tan ampliamente estudiado problema: utilizando un simple cambio de variables se puede construir un sistema basado en reglas difusas que calula lo mismo que una red neuronal. La construcción también nos proporciona una vía para encontrar una interpretación de las redes en base a un tipo particular de reglas difusas. No obstante, habrá que profundizar más en la semántica de las reglas obtenidas, sobre todo en aplicaciones reales.*

Palabras clave: *Redes Neuronales, Sistemas Difusos Aditivos, Adquisición del Conocimiento*

1 Redes Neuronales Artificiales.

Las Redes Neuronales Artificiales (RNA) son modelos masivamente paralelos de cálculo que han demostrado un comportamiento excelente en la resolución de problemas complejos en el ámbito de la Inteligencia Artificial. Destacan entre estos problemas los relativos a Optimización Combinatoria, Clasificación y Aprendizaje. Existen numerosos trabajos describiendo a fondo estos modelos y sus aplicaciones [10].

Las RNA multicapa de propagación hacia adelante son especialmente interesantes, pues son las más frecuentemente utilizadas: se han empleado en cerca del 80% del total de aplicaciones desarrolladas con RNA. En estas redes, las neuronas se organizan en capas y sólo existen conexiones de las neuronas de una capa a las de la siguiente. Consideremos una RNA con una capa de entrada, una oculta y otra de salida. Supongamos que la red tiene n neuronas de entrada $(x_1, \ldots, x_n)$, h neuronas ocultas $(z_1, \ldots, z_n)$ y m neuronas de salida $(y_1, \ldots, y_m)$. Sea w_{ij} el peso de la conexión de la neurona x_i a la z_j y, β_{jk} el peso de la conexión de la neurona z_j a la y_k. Entonces la función que calcula la red es:

$$F : [0,1]^n \to R^m; \quad F(x_1, \ldots, x_n) = (y_1, \ldots, y_m)$$

$$y_k = g_A(\sum_{j=1}^{h} \beta_{jk} z_j)$$

con

$$z_j = f_A(\sum_{i=1}^{n} w_{ij} x_j)$$

g_A y f_A son las funciones de activación. Se emplean funciones continuas monótonas no lineales. Una elección usual es la función logística: $f(x) = \frac{1}{1+e^{-x}}$. Para g_A también se emplean funciones lineales o incluso la identidad.

Una de las propiedades más interesantes de estas redes es que son aproximadores universales, es decir, pueden aproximar con el grado de exactitud que se desee cualquier función continua o con un número numerable de discontinuidades entre compactos. En

[6], se demuestra este resultado para redes de propagación hacia adelante con una única capa oculta y un número suficiente de unidades en ella cuya función de activación sea continua no lineal y considerando como función de activación en la salida, la identidad. Este resultado nos asegura que podemos emplear con éxito las redes neuronales para aprender y reproducir con fidelidad el comportamiento de un sistema a partir de un conjunto suficientemente amplio de ejemplos del mismo. El resultado ha contribuido a aumentar el ya extenso uso de las RNA.

Sin embargo, su uso se ha visto limitado por el que sigue siendo el aspecto más crítico de las RNA: no se dispone de una interpretación de su funcionamiento. Las RNA son dispositivos que operan a modo de cajas negras: Captan perfectamente las relaciones "ocultas" entre datos de entrada y salida, sin embargo, aún no se ha dado una respuesta satisfactoria a la pregunta de cómo funcionan.

2 Sistemas Basados en Reglas Difusas.

Al igual que las RNA, los sistemas basados en reglas difusas (SRD) [8], [12] también son aproximadores universales [5]. Los SRD tienen la ventaja de que el conocimiento que manejan está representado en un modo perfectamente asimilable por el ser humano, las reglas difusas. Las reglas, en general, representan de modo natural las relaciones de causalidad entre entradas y salidas, correspondientes a la estructura lingüística usual "*Si ... entonces ...*". Las reglas difusas tienen la ventaja adicional de permitir un tratamiento adecuado del conocimiento vago o incierto. En [5] se establece que clases amplias de controladores difusos son aproximadores universales. Dentro de estas clases están los controladores de Sugeno [11] que usan reglas del tipo:

$$R_i: \text{ Si } x_1 \text{ es } A_i^1 \text{ y } x_2 \text{ es } A_i^2 \text{ y } \ldots \text{ y } x_t \text{ es } A_i^t \text{ entonces } y \text{ is } p(x_1, \ldots, x_t)$$

donde $p(x_1, \ldots, x_t)$ es una función lineal de las entradas.

Cuando las funciones de activación de una RNA son continuas la función que calcula ésta, F, es continua. Por tanto, una RNA se puede aproximar mediante un SRD. Recíprocamente, una RNA puede aproximar un SRD continuo. De este modo es fácil concluir que las RNA que hemos descrito y los SRD son equivalentes. Una demostración directa de este hecho se ofrece en [4].

Basada en esta equivalencia entre RNA y SRD es inmediata la idea de emplear un sistema híbrido que use los dos paradigmas para realizar la tarea de adquisición del conocimiento. De hecho, en la bibliografía se encuentran múltiples referencias a métodos basados en esta idea que la llevan a la práctica de diversas formas [1], [2], [7]. Una de las más interesantes es la que trata de extraer el conocimiento obtenido por una red tras el aprendizaje almacenado de forma distribuida en los pesos de sus conexiones. Esto no sólo suministraría un método automatizado y eficaz de adquisición de conocimiento, sino que además proporcionaría una interpretación del modo en que operan y aprenden las RNA.

3 Las Redes Neuronales como Sistemas Basados en Reglas Difusas.

La idea expuesta en el último párrafo de la sección anterior ha sido trabajada. El trabajo descrito en [4] tan sólo ofrece un resultado de equivalencia entre RNA y SRD continuos mediante una aproximación recíproca con el nivel que se desee. Además se trata de un resultado puramente teórico, inviable de llevar a la práctica y no ofrece una solución al problema de la interpretación de RNA.

En este trabajo demostramos que es trivial la construcción de un SRD que calcule *exactamente lo mismo*, que una RNA del tipo de las empleadas por Hornik *y otros* [6], no aproximado, sino exactamente lo mismo. No es necesario recurrir a la construcción de ingentes cantidades de reglas difusas. El sistema usa reglas muy parecidas a las empleadas en los controladores de Sugeno.

Para cada peso β_{jk} entre neuronas de la capa oculta y la de salida, se crea una regla R_{jk}:

$$R_{jk}: \text{ Si } \sum_{i=1}^{n} w_{ij}x_i \text{ es } A \text{ entonces } s_{jk} = \beta_{jk} \qquad (1)$$

donde A es un conjunto difuso sobre R cuya función de pertenencia es justamente la función de activación de las neuronas en la capa oculta, f_A.

La salida del SRD es el vector cuyas componentes se obtienen aplicando el método de inferencia que se usa en los sistemas difusos aditivos [9]:

$$y_k = \sum_{j=1}^{h} A\left(\sum_{i=1}^{n} w_{ij}x_i\right) \cdot \beta_{jk}$$

En definitiva, usamos reglas simples del estilo "Si z es B entonces $y = r$", donde $r \in R$ y z son nuevas variables obtenidas a partir de un cambio de variable de las n entradas de la red.

Con el conjunto de reglas y método de inferencia indicados y teniendo en cuenta que g_A es la función identidad, igual que en las redes consideradas en [6], hemos probado la existencia de un sistema basado en

reglas difusas que calcula *exactamente lo mismo* que una red neuronal, con grado de aproximación completo.

4 Conclusión y comentarios finales.

Basándonos en lo expuesto en la sección anterior podemos afirmar que las RNA multicapa de propagación hacia adelante se pueden ver como SRD. Este resultado tiene un interés doble. Por un lado, teórico, pues aunque ya se había demostrado la equivalencia, ahora se establece la igualdad.

Por otro lado, práctico. La demostración consiste en la construcción de un SRD que calcula lo mismo que una RNA y, por tanto, ofrece una interpretación de la RNA basada en reglas difusas. Esto permite desarrollar de forma directa un sistema automático de adquisición del conocimiento. El SRD asociado a cada RNA es manejable pues las reglas no son complejas y su número es justamente $h \cdot m$, pequeño e independiente del conjunto de datos empleados para entrenar la red, con lo que se puede usar un conjunto de datos tan grande como sea necesario sin que ello suponga un incremento en la complejidad del conjunto de reglas resultante.

Podemos añadir, además, que en la reglas usadas, el antecedente se puede descomponer de forma que tengamos expresiones del tipo:

$$R_{jk}: \text{ Si } x_1 \text{ es } A^1_{jk}\ \theta_1\ x_2 \text{ es } A^2_{jk}\ \theta_2\ \ldots\ \theta_{n-1}\ \theta_{n-1}\ x_n \text{ es } A^n_{jk} \text{ entonces } s_{jk} = \beta_{jk} \qquad (2)$$

donde θ_i son conectivos lógicos y A^i_{jk} son conjuntos difusos adecuados obtenidos en función de los pesos w_{ij}.

Si la función de activación es sigmoide, se puede descomponer también de la forma x_i is A_i donde todos los A_i son conjuntos difusos cuya función de pertenencia tiene la forma de una λ-sigmoide, $y = \frac{1}{1+e^{-\lambda x+\sigma}}$, ya que por combinaciones de funciones de este tipo podemos obtener fácilmente la sigmoide normal.

Por otro lado, podemos dar una interpretación sencilla a las reglas del tipo indicado en (1) ya que el difuso A puede claramente ser intepretado como "aproximadamente mayor que r" donde r será un número real positivo, obtenido según un nivel de α-corte prefijado (p.e. para $\alpha = 0'9$, $r = 2'19$). Esto permite interpretar fácilmente el antecedente de (2) como x_i es "aproximadamente mayor que r/w_{ij}".

Referencias

[1] Benítez, J.M., Blanco, A., and Requena, I. *An Empirical Procedure to Obtain Fuzzy Rules using Neural Networks.* 6th IFSA Congress, Brasil, July, 1995.

[2] Benítez, J.M., Blanco, A., Delgado, M. y Requena, I. *Métodos Neuronales para la Extracción de Reglas Difusas.* V Congreso de la Asociación Española de Tecnologías y Lógica Fuzzy, Murcia, Septiembre 1995.

[3] Buckley, J.J. *Sugeno Type Controllers are Universal Controllers.* Fuzzy Sets and Systems, 53, pp. 299–304, 1993.

[4] Buckley, J.J., Hayashi, Y. and Czogala E. *On the equivalence of Neural Network and Fuzzy Expert Systems.* Fuzzy Sets and Systems, 53, pp. 129–134, 1993.

[5] Castro, J.L. *Fuzzy Logic Controllers are Universal Approximators.* IEEE Trans. on Syst., Man and Cybern. 25(4), April, 1995.

[6] Hornik, K., Stinchcombe, M. and White, H. *Multilayer Feedforward Networks are Universal Approximators.* Neural Networks, 2, pp. 359–366, 1989.

[7] Horikawa, S., Furuhashi, T., Okuma, S., and Uchikawa, Y. *A fuzzy controller using a neural network and its capability to learn expert's control rules.* Iizuka, Japan, pp. 103–106, 1990.

[8] Kandel, A. (Ed.)*Fuzzy Expert Systems.* CRC Press, Boca Raton, FL, 1991.

[9] Kosko, B. *Fuzzy Systems as Universal Approximators.* IEEE Trans. on Computers, 43(11), pp. 1324–1333, 1994.

[10] Lippman, R.P. *An introduction to computing with neural nets.* IEEE ASSP magazine, 4, pp. 4–22, 1987.

[11] Sugeno, M. and Nishida, M. *Fuzzy control of model car.* Fuzzy Sets and Systems, 16, pp. 103–113, 1985

[12] Zimmermann, H.J. *Fuzzy Set Theory and Its Applications.* (Second Edition). Kluwer Academic Publishers, Dordrecht-Boston, MA. 1991.

ADQUISICION DE CONOCIMIENTO BASADO EN REDES DE RESTRICCIONES BORROSAS: APLICACION EN AGRICULTURA

S. Túnez[2], R. Marín[1], A. Bosch[2], F. Bienvenido[2], I. del Aguila[2] y M. Taboada[3]

[1] Dept. de Informática y Sistemas. Universidad de Murcia. 30071 Murcia, España. E-mail: roque@dif.um.es

[2] Dept. de Lenguajes y Computación. Universidad de Almería. 04120 Almería, España. E-mail: stunez@ualm.es

[3] Dept. de Electrónica e Computación. Universidade de Santiago. 15706 Santiago, España. E-mail: chus@zfis.usc.es

Abstract

Este trabajo presenta la aplicación de las redes de restricciones vectoriales binarias borrosas para la adquisición de conocimiento e implementación de un modelo de decisión para selección de tratamientos fitosanitarios en un sistema basado en conocimiento para control de plagas.

1.- Introducción

Las técnicas de cultivo intensivo practicadas en invernaderos favorecen la aparición de plagas que se implantan como parásitos de las plantas, y que inciden en la cantidad y calidad de la cosecha. El método de control fitosanitario más utilizado es la lucha química. Sin embargo, la aplicación indiscriminada de plaguicidas produce problemas de aparición de resistencia al producto, deterioro del medio ambiente, riesgo para aplicadores y consumidores, costes elevados, y potenciación de nuevas plagas. Todo ello hace que el control de plagas en cultivos sea un problema complejo.

Nuestro grupo ha desarrollado un sistema basado en conocimiento para asesoramiento en control fitosanitario, que asiste al técnico agrícola en las tres tareas básicas del dominio: establecer la estrategia de control, seleccionar un plaguicida adecuado, y determinar el método de aplicación (Túnez et al., 1990; 1992). En particular, resolvemos el problema de la selección de producto integrando técnicas convencionales de Ingeniería del Conocimiento y de Teoría de Decisión. Esta solución híbrida está basada en reglas de producción y funciones multiatributo. Los principales problemas encontrados, tras la aplicación práctica del modelo, se refieren a: 1) El carácter estático de la base de conocimiento, que colisiona con un dominio en el que las estrategias de selección de productos cambian rápidamente; 2) La necesidad de representar la imprecisión existente en algunos de los criterios de selección empleados por el experto. Ello sugiere la conveniencia de integrar técnicas de decisión borrosas (Bellman y Zadeh, 1970; Bolaños et al., 1988; Delgado et al., 1993; Dubois y Prade, 1988; Yager, 1988; 1992) con utilidades de adquisición del conocimiento que permitan al usuario modificar los elementos del problema de decisión. En este trabajo proponemos una solución que emplea redes de restricciones borrosas como herramienta de preproceso del conocimiento inyectado en un modelo borroso de decisión multicriterio. Está basada en una generalización del modelo FTCN (Fuzzy Temporal Constraint Networks) que hemos propuesto anteriormente para una aplicación distinta, el razonamiento temporal (Marín et al., 1993a; 1993b; 1994a; 1994b; Barro et al., 1994).

2.- Descripción del problema

El problema del dominio que analizaremos aquí es el de la selección de plaguicida. Formalmente, consiste en elegir una acción U de entre un conjunto de posibles alternativas $P = \{P_1, P_2, \ldots P_n\}$ caracterizadas por unos determinados valores de atributos. La acción U debe elegirse de tal forma que sus valores de atributos verifiquen simultáneamente un conjunto de criterios independientes. Supondremos que no todos los criterios son igualmente importantes para la decisión.

En el dominio, el contexto del problema está definido por el tipo de cultivo y el tipo de plaga, así como por variables tales como estado vegetativo, población y estado de desarrollo de los parásitos, climatología, etc ... Las alternativas corresponden a los distintos productos químicos fitosanitarios aplicables en este contexto, y los atributos más significativos son: toxicidad sobre las personas, toxicidad sobre la fauna terrestre, toxicidad sobre la fauna acuícola, toxicidad sobre las abejas, fitotoxicidad, plazo de seguridad antes de la recolección, forma de acción sobre la plaga, forma de acción sobre la planta, polivalencia, coste económico de la aplicación del producto, eficacia, etc ...

Los valores de algunos de estos atributos pueden ser definidos en forma precisa y absoluta. Es el caso de los atributos de toxicidad, que toman valores en categorías discretas bien establecidas en la legislación. Sin embargo, otros atributos deben valorarse en forma imprecisa. Un ejemplo es el coste económico de la aplicación del producto, que depende, no sólo del precio del producto, sino también de factores variables como el precio de la mano de obra de la aplicación, la dosis a emplear, la vía de aplicación, o el número de aplicaciones necesarias. Aún más, el precio del producto puede variar de unos proveedores a otros y puede modificarse dinámicamente por razones comerciales (v.g. ofertas). Sobre estos últimos factores influyen además las rápidas fluctuaciones del mercado hortofrutícola. Dada la dificultad de conocer a priori todos estos factores, resulta aceptable simplificar el problema modelando el coste de aplicación como un

atributo de valoración imprecisa.

Por otra parte, hay atributos que, además de ser imprecisos, son difícilmente valorables en forma absoluta. Un ejemplo claro es el de la eficacia del producto. En la práctica, la información suministrada por los ensayos experimentales de eficacia es de limitada utilidad por diversas razones:

a) El número de combinaciones materia-activa/plaga/planta es demasiado elevado para poder realizar un estudio exhaustivo. No se dispone de datos experimentales para todos los productos.

b) Los resultados de un ensayo de eficacia son dependientes de factores como la climatología, densidad de plantación, aireación del invernadero, técnicas de cultivo, etc ..., factores que no necesariamente coincidirán con los del problema real a resolver.

c) La eficacia está sometida a una enorme variabilidad local: los ensayos realizados no son extrapolables de unos entornos geográficos a otros.

d) Aún más grande es la variabilidad temporal: un producto muy eficaz en una campaña puede ser ineficaz en la campaña siguiente, debido a que los insectos desarrollan resistencia a las materias activas a una gran velocidad.

En la práctica, los ensayos de eficacia son orientativos, y para el técnico agrícola constituyen sólo una fuente adicional de experiencia. Todo ello hace que no sea realista establecer un orden total entre los productos en razón de su eficacia, y menos aún asignar un valor numérico absoluto a la eficacia de cada producto. El experto se siente más cómodo si puede valorar la eficacia de los productos en términos relativos, usando expresiones del tipo: "El producto P_2 es *bastante más* eficaz que el producto P_1, pero *aproximadamente igual* de eficaz que el producto P_3". La posibilidad de dejar indefinida la ordenación relativa entre dos productos (P_2 y P_3 en el ejemplo) es crucial para obtener la colaboración del experto en la elicitación de conocimiento. No son ajenas a ello las connotaciones de tipo comercial que llevaría el decantarse por un producto determinado.

Un aspecto adicional del problema es la necesidad de establecer *perfiles de tratamiento* individualizados. Idealmente, habría que seleccionar el producto que mejor verifique los criterios de "eficacia muy alta", "precio muy bajo", "toxicidad sobre personas en el rango {B, C}", "forma de acción moderadamente selectiva", etc ... Obviamente, se trata de criterios contradictorios que conducen a una sobreespecificación del problema, sin solución. En la práctica, el experto considera prioridades entre los criterios, estableciendo un perfil de tratamiento adaptado a las características individuales del problema concreto. En general, el criterio básico es la alta eficacia, seguido por la baja toxicidad y la alta tolerancia a residuos (plazo de seguridad condicionado por el tiempo disponible hasta la cosecha). Sin embargo, la importancia relativa de los criterios se modifica drásticamente en determinados casos particulares: en planta pequeña lo primordial es la fitotoxicidad; en planta grande de rentabilidad comercial discutible, el criterio más importante es el bajo coste de aplicación del producto, etc ... No se trata sólo de relajar o modificar los criterios, sino también de graduar su importancia en la decisión final, importancia que en muchos casos es expresada en términos relativos. De nuevo, la posibilidad de expresar grados de importancia relativos ("criterio C_1 *mucho más* importante que criterio C_2 pero *aproximadamente igual* de importante que criterio C_3") añade flexibilidad en la especificación del problema y facilita la elicitación de conocimiento.

En definitiva, tanto para representar el conocimiento sobre los atributos como para representar los criterios y su grado de importancia, es conveniente un modelo tal que:

a) Permita representar valores en términos *relativos*, imprecisos y susceptibles de ser expresados lingüísticamente.

b) Su contenido sea fácilmente modificable por el usuario (técnico agrícola) para reflejar las variaciones dinámicas de los atributos y de la importancia de los criterios.

c) Esté dotado de mecanismos automáticos de eliminación de las redundancias y detección de las inconsistencias que el usuario pueda introducir durante los procesos de modificación del conocimiento.

En general, las técnicas borrosas de decisión multicriterio no incluyen mecanismos automáticos de supervisión de las modificaciones y no contemplan la representación de conocimiento relativo. Por otra parte, las técnicas usuales en ingeniería del conocimiento son insuficientes para la resolución de problemas complejos de decisión. Unas y otras persiguen objetivos complementarios. En este trabajo proponemos una solución para su integración.

3.- Modelo integrado

3.1.- Arquitectura general

Resolver el problema de decisión multicriterio previamente planteado supone tres tareas (fig. 1): (*a*) Especificación del problema; (*b*) Representación del problema; y (*c*) Proceso de decisión.

La especificación del problema incluye: (*a*) El establecimiento del contexto del problema, que es generado por un módulo basado en conocimiento (no mostrado en la figura) que se encarga de determinar la estrategia de control fitosanitario adecuada, y es un proceso previo a la selección de producto; (*b*) La definición del espacio de opciones, es decir, el conjunto de alternativas y sus atributos, que modelamos mediante una red de restricciones borrosas y describiremos en la siguiente sección; (*c*) El establecimiento del perfil de tratamiento, centrado en los criterios a aplicar y en su pertinencia respecto a la decisión global.

En particular, la definición del espacio de opciones para el contexto actual se produce a partir de una base de conocimiento de productos fitosanitarios, que contiene un catálogo de productos con atributos independientes del contexto (i.e. válidos para cualquier contexto plaga/planta, como el precio y algunos tipos de toxicidad) y atributos dependientes del contexto (v.g. eficacia, fitotoxicidad, ...) Esta base de conocimiento es dinámica y se actualiza a través de un módulo de adquisición de conocimiento, que analiza la consistencia y redundancia de la información introducida en el espacio de opciones, y actualiza la base de conocimiento. Se sigue un esquema análogo para la definición del perfil de tratamiento.

A partir de la especificación del problema, y mediante un proceso de extracción de información, se obtiene una representación del problema en términos del modelo de decisión usado. Este es un modelo de decisión multicriterio basado en órdenes de prioridad, y proviene del propuesto por Yager en 1992. Finalmente, se aplica el proceso de decisión asociado al modelo usado. El usuario cierra un lazo de realimentación, pudiendo modificar interactivamente el perfil de tratamiento si aparecen problemas de sobreespecificación o subespecificación.

3.2.- Modelo de especificación del problema

El modelo de especificación del problema consta de una *red de atributos*, un conjunto de *criterios*, una *red de grados de importancia* y un *umbral de prioridad*.

La red de atributos es un caso particular del concepto de *red de restricciones vectoriales binarias borrosas* (FVCN). Introduciremos primero el modelo de red de atributos y después haremos algunos comentarios sobre el modelo general. Una *red de atributos* está formada por un conjunto finito de $n+1$ variables $P = \{P_0, P_1, P_2, \ldots P_n\}$ y un conjunto finito de restricciones binarias vectoriales entre ellas $\{R_{ij}\}$. Cada variable P_i está asociada a una de las posibles alternativas, y se representa mediante una m-tupla de atributos $P_i = \langle A^1_i, A^2_i, \ldots A^m_i \rangle$. Cada atributo A^k_i es un parámetro variable que, en ausencia de restricciones, puede tomar cualquier valor preciso en un dominio D^k. Consideraremos que los dominios D^k son subconjuntos del conjunto de los números naturales.

Las restricciones delimitan los valores que pueden tomar los atributos de las distintas variables. Una restricción binaria vectorial R_{ij} entre dos variables P_i y P_j es una m-tupla $R_{ij} = \langle R^1_{ij}, R^2_{ij}, \ldots R^m_{ij} \rangle$ formada por m restricciones elementales. Cada restricción elemental R^k_{ij} se define mediante un número borroso entero, es decir, mediante una distribución de posibilidad π^k_{ij} normalizada y convexa, definida sobre el conjunto de los números enteros. La restricción R^k_{ij} establece los posibles valores para la diferencia numérica entre los valores de los atributos A^k_i y A^k_j: dado un entero preciso l, $\pi^k_{ij}(l)$ representa la posibilidad de que la diferencia entre los valores de A^k_i y A^k_j sea precisamente l. Por tanto, R^k_{ij} permite definir el valor del atributo k-ésimo de la alternativa P_j en forma relativa al valor del mismo atributo de la alternativa P_i, que hace de punto de referencia. Expresado informalmente, la restricción R^k_{ij} representa la "distancia" imprecisa existente entre las alternativas P_i y P_j en la dimensión k. La variable P_0 representa un origen, de forma que cada restricción R_{0i} relativa al origen nos permite representar el valor absoluto impreciso de los atributos de la alternativa P_i.

En el modelo FVCN general, las restricciones elementales R^k_{ij} que componen una restricción R_{ij} pueden ser dependientes entre sí, esto es, pueden aportar información unas sobre otras. Además, el número de restricciones elementales que componen una restricción vectorial R_{ij} puede ser mayor que el número m de atributos. Sin embargo, en la particularización del modelo usada en este trabajo suponemos que las restricciones elementales son independientes, y, por tanto, no pueden ser ni redundantes ni inconsistentes entre sí. Ello nos permite conceptualizar la red de atributos como yuxtaposición de redes de restricciones unidimensionales. Debido a la suposición de independencia interdimensional podemos aplicar separadamente a cada red unidimensional el algoritmo de propagación de restricciones que hemos descrito en otros trabajos (Marín, 1993a; 1994b), y que se ejecuta en tiempo polinómico.

En nuestro dominio de aplicación, cada alternativa P_i corresponde a un producto fitosanitario distinto, con atributos A^k_i tales como toxicidad, eficacia, coste de aplicación, ... El origen P_0 representa a una alternativa ficticia de referencia con atributos nulos. La red de atributos nos permite representar declarativamente conocimiento sobre las características de los productos. El tipo de conocimiento que podemos representar, previa traducción linguística, es el contenido en expresiones como: "P_i es moderadamente eficaz" (valor impreciso expresado en términos absolutos), o "P_i es bastante más eficaz que P_j" (valor impreciso expresado en términos relativos).

Como hemos indicado anteriormente, el proceso de adquisición del conocimiento debe ser dinámico para reflejar las variaciones en las características de los productos. Cada vez que el usuario introduce modificaciones en la red de atributos, aplicamos un algoritmo de propagación de restricciones que detecta las inconsistencias, y si éstas no existen, obtiene una red mínima equivalente en la que las redundancias han sido eliminadas. La red mínima verifica que $R_{ij} \subseteq R_{ih} \oplus R_{hj}$, para cada tres variables P_i, P_j y P_h. Es decir, sus restricciones son mínimamente imprecisas como resultado de combinar globalmente toda la información introducida por el usuario. En particular, las restricciones mínimas respecto al origen representan los valores absolutos de los atributos que se obtienen como consecuencia lógica de la información introducida por el usuario. Estos valores R_{0i} son tomados como punto de partida para el resto del proceso de decisión. Las restantes restricciones R_{ij}, $i, j \neq 0$, ya no aportarán información adicional, dado que, una vez propagadas las restricciones, $R_{0i} \subseteq R_{0j} \oplus R_{ji}$.

Por otra parte, se define un conjunto C de m *criterios*, $C = \{C^1, C^2, \ldots C^m\}$, estando cada C^k representado mediante una distribución de posibilidad π^k_C sobre el dominio D^k correspondiente al atributo k-ésimo. Cada criterio C^k establece un objetivo que la acción U elegida debe satisfacer. Se define también una *red de grados de importancia* como una red de restricciones unidimensional formada por un conjunto finito de $m+1$ variables $I = \{I_0, I_1, \ldots I_m\}$ y un conjunto finito de restricciones binarias borrosas entre ellas $\{\alpha_{ij}\}$. Las variables I_i representan la importancia o pertinencia de cada criterio respecto a la decisión global, y su valor se establece mediante las restricciones α_{ij}. Cada restricción α_{ij} define el grado de importancia relativo del criterio C^i respecto al criterio C^j, y se representa mediante una distribución de posibilidad π^{α}_{ij} normalizada y convexa sobre un intervalo [0, 1] discretizado. La red permite que el perfil de tratamiento pueda ser especificado mediante expresiones del tipo: "El criterio de toxicidad en personas es muy importante" (valor impreciso en términos absolutos) o "el criterio de coste de aplicación es mucho menos importante que el criterio de eficacia" (valor impreciso en términos relativos). Finalmente, se

especifica un *umbral de prioridad* $\beta \in [0, 1]$, que es usado en el modelo de decisión para distinguir entre criterios de primer y segundo orden.

3.3.- Modelo de decisión

El modelo de decisión usado está basado en Yager (1992), pero hemos introducido algunas modificaciones para poder manejar alternativas caracterizadas por atributos imprecisos. Consta de: (a) un conjunto de *alternativas* $P = \{P_1, P_2, \ldots P_n\}$, cada una caracterizada por m atributos borrosos, $P_i = <A^1_i, A^2_i, \ldots A^m_i>$; (*b*) un conjunto de *criterios de primer orden* $F = \{F^1, F^2, \ldots F^r\}$; (*c*) un conjunto de *criterios de segundo orden* $H = \{H^1, H^2, \ldots H^q\}$, con $r + q = m$; y (*d*) una distribución de posibilidad de *grados de importancia* $\pi_\alpha = (\alpha^1, \alpha^2, \ldots \alpha^m)$ normalizada. Para simplificar la notación, supondremos que los atributos se han ordenado de forma que los r primeros se corresponden con los criterios de primer orden y los q últimos con los criterios de segundo orden.

Los r criterios de primer orden establecen objetivos que la acción U elegida debe satisfacer conjuntamente. Llamamos s^k_i al grado de compatibilidad entre el criterio F^k y el atributo A^k_i de la alternativa P_i, que podemos calcular como la posibilidad de que la proposición "U es F^k" sea cierta dada la proposición de referencia "U es A^k_i":

$$s^k_i = \Pi (F^k / A^k_i) = \max_{x \in D^k} \min (\pi^k_F (x), \pi^k_i (x)).$$

La distribución de posibilidad $\pi_\alpha = (\alpha^1, \alpha^2, \ldots \alpha^m)$ establece la pertinencia de cada criterio respecto al objetivo global, y cada α^k suele interpretarse como el grado de importancia del criterio F^k en la decisión. Se han propuesto distintos tipos de operadores de agregación para combinar criterios y grados de importancia (Dubois y Prade, 1988; Yager, 1988; 1992). Hasta ahora, sólo hemos probado en nuestra aplicación el siguiente:

$$D (P_i) = \min_{k=1..r} \max (s^k_i, 1 - \alpha^k)$$

La función de evaluación $D (P_i)$ refleja el grado de satisfacción global de la alternativa P_i respecto al conjunto de criterios de primer orden. En ausencia de otros criterios, seleccionaríamos la alternativa que maximiza a la función D.

Los criterios de segundo orden, o criterios posibilísticos, fueron introducidos por Yager (1992) para modelar patrones de razonamiento por defecto. Representan objetivos que se desea satisfacer, siempre que no interfieran con los criterios de primer orden. Actúan como filtros sobre aquellas alternativas que ya satisfacen los criterios de primer orden. El establecimiento de un criterio de segundo orden H^k corresponde a la proposición: "Si U es H^k es posible, entonces U es H^k". Dado un criterio de segundo orden H^k, y dada la función de evaluación $D (P_i)$ previamente obtenida a partir de los criterios de primer orden, la proposición anterior inducirá una nueva función de evaluación $D'(P_i)$ dada por:

$$D'(P_i) = \min[D (P_i), \text{máx } \{(1 - \Pi (H^k / D)), H^k (P_i)\},$$

siendo:

$$H^k (P_i) = \Pi (H^k / A^k_i) = \max_{x \in D^k} \min (\pi^k_H (x), \pi^k_i (x)),$$

y:

$$\Pi (H^k / D) = \max_{P_i \in P} \min (H^k (P_i), D (P_i))$$

Obsérvese que el grado de satisfacción de la alternativa i-ésima, $D'(P_i)$ depende de la posibilidad $\Pi (H^k / D)$ de satisfacer el criterio H^k una vez satisfechos los criterios de primer orden. En particular, si H^k produce una sobreespecificación que haga imposible satisfacer simultáneamente el criterio de segundo orden y los criterios de primer orden, se tiene:

$$\Pi (H^k / D) = 0;$$
$$D'(P_i) = D (P_i),$$

y el criterio de segundo orden no tendrá efecto. Por el contrario, si es completamente posible satisfacer todos los criterios, incluido el de segundo orden, se tiene:

$$\Pi (H^k / D) = 1;$$
$$D'(P_i) = \min \{D (P_i), H^k (P_i)\}$$

y el criterio H^k es tratado como un criterio de primer orden. En general, H^k actúa como un criterio cuyo grado de importancia depende de su compatibilidad con los criterios de primer orden. La inclusión de criterios de segundo orden circunscribe el riesgo de sobreespecificación a los criterios de primer orden. Por tanto, una estrategia para relajar un problema sobreespecificado consiste en desplazar criterios de primer orden hacia el conjunto de criterios de segundo orden.

La representación del problema en términos de este modelo de decisión se obtiene a partir de los elementos de especificación descritos en la sección anterior. En particular, el valor asignado a cada atributo es igual a cada una de las restricciones absolutas de la red de atributos mínima, $A^k_i = R^k_{0i}$. La distribución de posibilidad de grados de importancia π_α se obtiene como una *solución* máximamente posible de la red de grados de importancia mínima. Una *solución* de una red de restricciones binarias borrosas se define como una m-tupla $(\alpha^1, \alpha^2, \ldots \alpha^m)$ que verifica todas las restricciones de la red:

$$\forall\ i, j: \pi^\alpha_{ij} (\alpha^j - \alpha^i) > 0.$$

La posibilidad de que una m-tupla sea solución de la red viene dada por:

$$\pi_S (\alpha^1, \alpha^2, \ldots \alpha^m) = \min_{i,j} \pi^\alpha_{ij} (\alpha^j - \alpha^i)$$

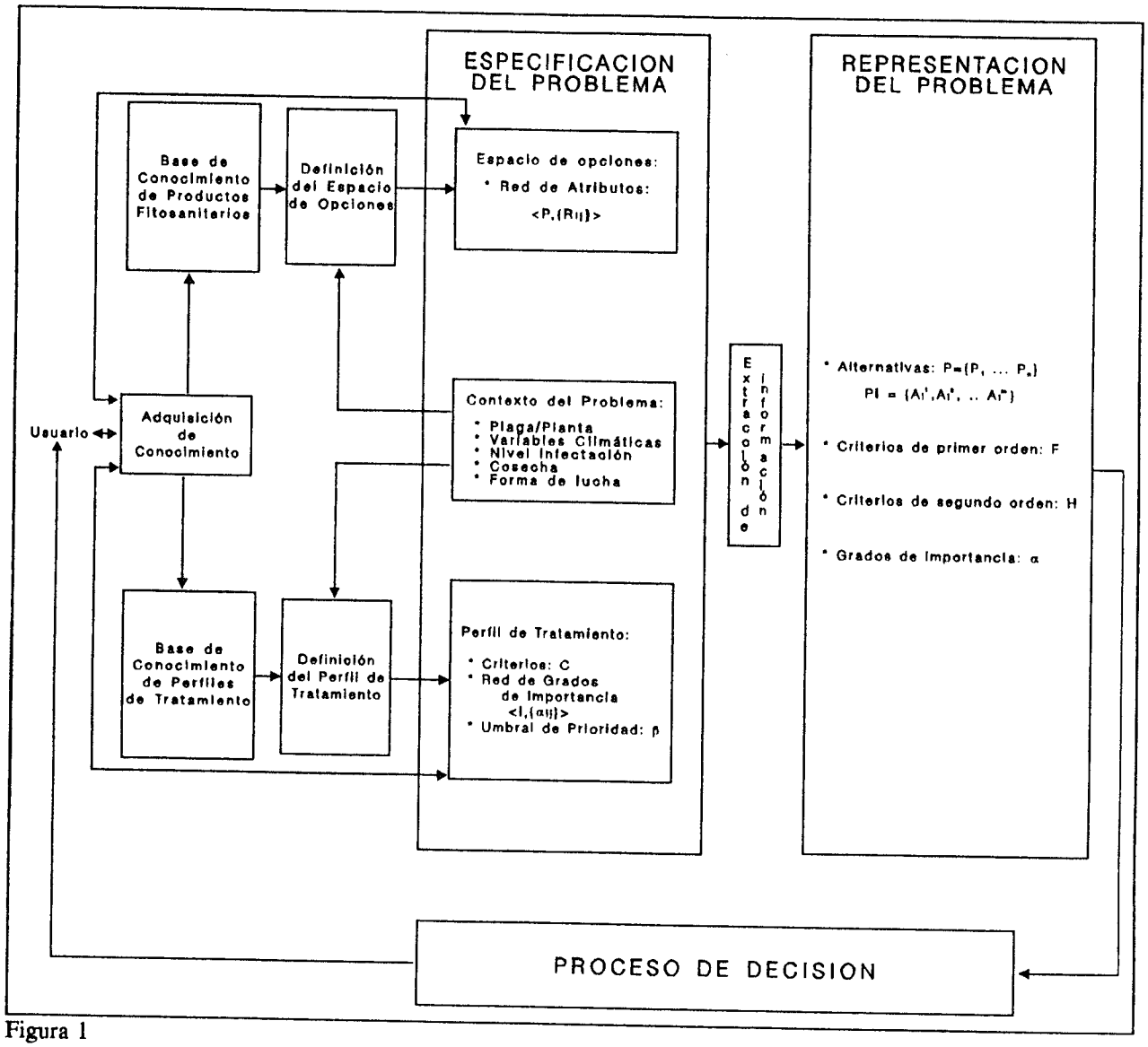


Figura 1

y siempre existen soluciones completamente posibles en una red mínima consistente. Recordemos que el algoritmo de propagación de restricciones aplicado en el módulo de especificación siempre obtenía una red mínima consistente (Marín, 1994b). Finalmente, del conjunto de criterios C se seleccionan como criterios de primer orden aquellos cuyo grado de importancia α^j es mayor o igual que el umbral de prioridad β, y los restantes son introducidos en el conjunto de criterios de segundo orden.

4.- Conclusiones

En este trabajo se propone una solución a problemas de decisión que integra técnicas de la ingeniería del conocimiento y técnicas borrosas de decisión multicriterio. Nuestra aproximación se basa en el uso de redes de restricciones borrosas como modelo representacional para recoger la especificación del problema. Sus principales ventajas se concretan en la posibilidad de introducir conocimiento relativo en las especificaciones y modificarlo dinámicamente. La aplicación de técnicas de propagación de restricciones y detección de inconsistencias, que ya hemos descrito en otros trabajos, proteje contra las posibles redundancias y/o inconsistencias que el usuario pueda introducir durante los procesos de modificación del conocimiento.

Como modelo de decisión utilizamos el propuesto por Yager (1992), basado en estructuras de segundo orden, en el que hemos introducido algunas modificaciones para permitir la representación de alternativas caracterizadas mediante atributos imprecisos. Tomando como punto de partida este modelo básico, caben varias modificaciones: distintos operadores de agregación, obtención de grados de compatibilidad borrosos, otras técnicas de "defuzzificación" de redes de restricciones borrosas, ... Se

requiere un estudio más exhaustivo antes de proceder a una integración definitiva con los módulos preexistentes del sistema experto para asesoramiento en control fitosanitario.

Como línea de trabajo a explorar en el futuro, nos planteamos la introducción de técnicas de análisis de fallos basadas en conocimiento. El objetivo, básicamente, es detectar situaciones de sobreespecificación o subespecificación a partir de la forma de la función de evaluación, y modificar iterativamente la especificación inicial del problema mediante relajación de criterios, modificación de grados de importancia con operadores de contraste, modificación del umbral de prioridad, ... Este módulo cerrará un lazo de control sobre el proceso de decisión, que quedará incluido en un esquema de razonamiento abductivo.

El modelo de especificación presentado aquí constituye una aplicación de un caso particular del concepto de FVCN. Otras aplicaciones del modelo FVCN en las que trabajamos son la representación de curvas cualitativas basadas en conocimiento y el razonamiento temporal.

Agradecimientos

Este trabajo ha sido realizado gracias al soporte del CDTI (proyecto PACE-PASO PR 191/93), la CICYT (proyecto TIC95-0604) y la Consejería de Cultura y Educación de la Comunidad Autónoma de la Región de Murcia (proyecto PCT94/48). Agradecemos también la colaboración del Departamento de Sanidad Vegetal de la Delegación de Agricultura y Pesca de Almería, y en particular, la colaboración de Vicente Aparicio Salmerón.

Referencias

[1] Barro, S., Marín, R., Mira J. and Rodríguez-Patón, A., A model and a language for the fuzzy representation and handling of time, Fuzzy Sets and Systems, 61: 153-175, 1994.

[2] Bellman, R. y Zadeh, L., Decision-making in a fuzzy environment, Manage. Sci., 17, 141-164, 1970.

[3] Bolaños, M. J., Lamata, M.T. y Moral, S., Decision making problems in a general environment, Fuzzy Sets and Systems, 18, 135-144, 1988.

[4] Delgado, M., Verdegay, J., Vila, M. A., On aggregation operations of linguistic labels, Int. J. of Intelligent Systems, 1993, 8, 351-370.

[5] Dubois, D. y Prade, H., Possibility Theory: an approach to computerized processing of Uncertainty, Plenum Press, New York, 1988.

[6] Marín, R., Barro, S., Bosch, A., and Mira, J., Modeling the representation of time from a fuzzy perspective. Cybernetics and Systems: an International Journal, 25: 217-231, 1994.

[7] Marín, R., Barro, S., Mira J., y Delgado, A.E., Razonamiento temporal borroso. En "Estudios de lógica borrosa y sus aplicaciones" (cap. 7), eds. A. Sobrino y S. Barro. Servicio de Publicaciones de la Universidad de Santiago, Santiago de Compostela, 1993. ISBN: 84-8121-018-8.

[8] Marín, R., Barro, S., Bosch, A., Navarrete, I., Cárdenas, M., Bugarín A., y Mira, J., Un modelo para la representacion de informacion temporal borrosa. Actas del Tercer Congreso Español sobre Tecnologías y Lógica Fuzzy, 209-216, Santiago de Compostela, 1993.

[9] Marín, R., Ruiz, R., Martín, F., Barro, S., Palacios, F., Una aproximacion al razonamiento temporal borroso en medicina, Actas del IV Congreso Español sobre Tecnologías y Lógica Fuzzy, 53-58, Blanes, 1994.

[10] Túnez, S., Bosch, A., Bienvenido, F., García, J., Marín, R., Mira, J., and Soria, F., Temporal knowledge modelling in plagues control in greenhouses. Cybernetics and Systems: an International Journal, 21, 367-374, 1990.

[11] Túnez, S., Bosch, A., Bienvenido, F., Marín, R., Cañabate, F., Mira J., and Soria, F., Strategic knowledge in a expert system for agriculture. Cybernetic and Systems: an International Journal, 23, 401-415, 1992.

[12] Yager, R., On ordered weighted averaging aggregation operators in multicriteria decision making, IEEE Trans. on Systems, Man and Cybern., 18, 183-190, 1988.

[13] Yager, R. Second order structures in multi-criteria decision making, Int. J. Man-Machine Studies, 36, 553-570, 1992.

Indice de autores

Abril, J., 273
Arenas, M., 139

Balsa, M., 93
Bandera, G., 295
Barriga, A., 331
Barros, S., 65, 99
Benítez, J.M., 105, 123
Bienvenido, F., 43
Bilbao, A., 139
Blázquez, J.M., 289
Blanco, A., 123
Bolaños, M.J., 157
Bosch, A., 43
Braunstingl, R., 279
Brdys, M.A., 253
Bugarín, A., 65
Burillo, P., 215
Burusco, A., 179
Bustince, H., 215

Cárdenas, Mª A., 93
Cabrera, J., 3
Cadenas, J.M., 29, 133
Calvo, O., 273
Camacho, A., 29
Castillo, F., 303
Castro, J.L., 59, 105
Clares, B., 203, 209
Conejo, R., 203, 209
Corral, N., 147
Cubero, J.C., 23
Cubillo, S., 171

De Miguel, L.J., 15
De Salvador, L., 325
De Soto, A.R., 171
Del águila, L., 43
Del Campo, C., 227
Del Val, D., 285
Delgado, M., 111, 123
Deschamps, J.P., 319

Engel, A., 99
Eraso, M.L., 245
Escriche, M., 267
Esteva, F., 221

Félix, P., 99
Falcón, A., 3
Fariña, F., 9
Fernández, C., 289
Fernández, J., 15
Fernández, L., 29
Fernández, S., 227
Ferreiro, R., 129, 285
Fuentes, R., 179

Gómez, A., 29, 111
Galán D., 331
García, P, 55
García, P., 221
Garmendia, A., 233
Garmendia, L., 239
Gil, Mª A., 147
Gil, P., 163
Godo, L., 55, 221
González de Mendivil, J.R., 9
González, L., 185, 191
Gutiérrez, J., 325

Herencia, J.A., 77
Hernández, F., 3
Hernández, L.D., 157
Herrera Viedma, E., 87
Herrera, F., 87

Jacas, J., 73
Jiménez, C.J., 331
Jiménez, F., 133
Jiménez, M., 139
Juanes, R., 285

López, H., 147
López, J., 285
López, Mª J., 163
Lamata, M^a. T., 77

Méndez, J., 3
Magdalena, L., 261
Marín, A., 185, 191
Marín, R., 43, 93, 99
Martín, F., 29, 111
Martínez, J.I., 319
Matía ,F., 267
Matellán,V., 289
Mayor, G., 175
Medina, J.M., 23
Mohedano, V., 215
Molero, M^a., 233, 239
Molina, J.M., 289
Monasterio, F., 261
Monreal, A., 73
Montes, S., 163
Montseny, E., 35
Moraleda, E., 267
Morales, R., 203, 209
Mujika, J., 279

Navarrete, I., 93

Olivas, J.A., 81, 197
Otero, A., 117

Pérez, J.L., 203, 209
Perán, J.R., 15
Pons, O., 23
Portilla, M.I., 245
Puente, E.A., 267
Puig, D., 35
Puyo, I., 303

Quevedo, J., 253

Recasens, J., 73
Requena, I., 105, 123
Reyes, M., 313
Rodrígez, M., 139
Rodríguez, A, 307
Rodríguez, R., 55

Sánchez, J.C., 285
Sánchez, J.L., 93
Sánchez, L., 117
Sánchez, S., 331
Salmerón , A., 157
Salom, J., 273
Salvador, A., 233, 239
Sanz, A., 9
Sim, W.L., 253
Sobrevilla, P., 35
Sobrino, A., 81, 197, 227
Suarez, E., 151

Túnez, S., 43
Taboada, M., 43
Taboada, M.J., 99
Torra, V., 49
Trillas, E., 171
Tuya, J., 117

Ullastre, S., 313
Uribe, J.P., 279

Valderrama, E., 313
Valverde, L., 167
Verdegay, J.L., 87
Vicente, J.M., 295
Vidal, F., 307
Vila, M.A., 23
Villadangos, J., 9

Yañez, V., 129

Zurita, J.M., 59