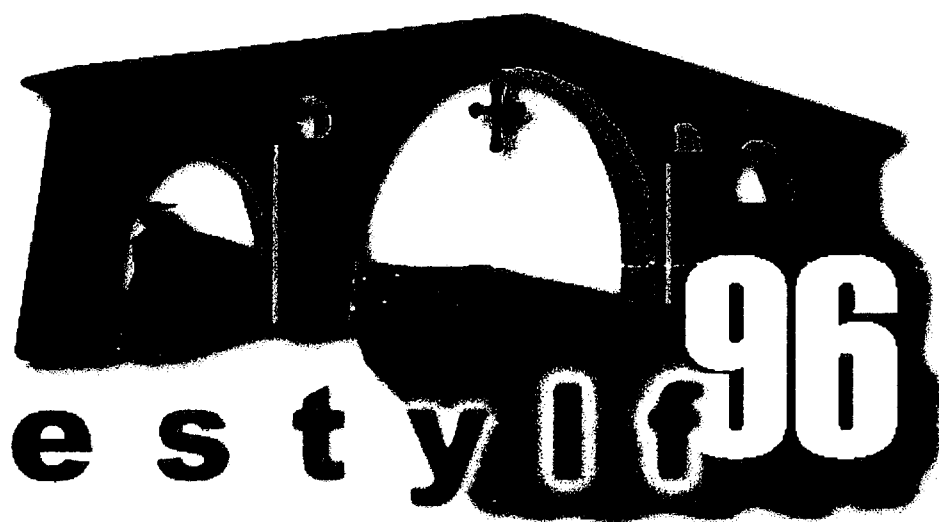




VI Congreso Español Sobre Tecnologías y Lógica Fuzzy

ACTAS

Asturias, 10-13 Septiembre

Servicio de Publicaciones. Universidad de Oviedo.

D.L.: AS/2124-96.

I.S.B.N.: 84-7468-946-5.

Indice

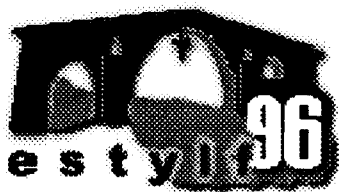
Prefacio	v
Comité de Programa y Organizador	vii
COMUNICACIONES	
Un modelo de clasificación con estructuras binarias de preferencia difusa	1
<i>Amo, A. del; Montero, J.</i>	
Programación lineal posibilística con restricciones flexibles	7
<i>Arenas, M.M.; Bilbao, A.; Jiménez, M.; Rodríguez, M.V.</i>	
Reglas difusas para la interpretación de redes neuronales artificiales entrenadas	13
<i>Benítez, J.M.; Castro, J.L.; Requena, I.</i>	
Measure of a fuzzy set based on an arbitrary continuous t-conorm: the finite case	19
<i>Bertoluzza, C.; Bodoni, A.</i>	
Modelo para codificación de reglas IF-THEN expresadas en términos lingüísticos.....	23
<i>Blanco, A.; Delgado, M.; Fajardo, W.</i>	
Inferencia gramatical difusa mediante redes neuronales	29
<i>Blanco, A.; Delgado, M.; Pegalajar, M.C.</i>	
Obtención de sistemas basados en reglas con valores de verdad mediante redes neuronales.....	35
<i>Blanco, A.; Fajardo, W.; Pegalajar, M.C.</i>	
Hacia una semántica de la defuzzyficación	41
<i>Boixader, D.; Jacas, J.; Recasens, J.</i>	
Definición de parámetros estadísticos utilizando la integral de Choquet	47
<i>Bolaños, M.J.; Reche, F.</i>	
Sobre extensiones de los conceptos mayorante, minorante, máximo, mínimo, supremo e ínfimo a subconjuntos difusos de un referencial con un preorden difuso	53
<i>Burusco, A.; Fuentes-González, R.</i>	
Método de los mínimos cuadrados aplicado a la inferencia fuzzy	59
<i>Bustince, H.; Calderón, M.; Mohedano, V.</i>	
Aprendizaje de hipergrafos posibilísticos	65
<i>Campos, L.M. de; Huete, J.F.</i>	
Sobre algunas propiedades en ternas de De Morgan	71
<i>Carbonell, M.; Mas, M.; Mayor, G.; Suñer, J.; Torrens, J.</i>	
Ejecución eficiente de reglas borrosas: inferencia parametrizada	77
<i>Cariñena, P.; Bugarín, A.; Barro, S.</i>	
Aprendizaje inductivo utilizando algoritmos genéticos	83
<i>Castro, J.L.; Castro-Sánchez, J.J.; Zurita, J.M.</i>	

Un proceso evolutivo para el aprendizaje de bases de conocimientos de controladores difusos..	89
<i>Cordón, O.; Herrera, F.</i>	
Sistema de clasificación con reglas difusas utilizando algoritmos genéticos	95
<i>Cordón, O.; del Jesús, M.J.; Herrera, F.</i>	
Arboles difusos de clasificación y regresión	101
<i>Delgado, M.; Gómez-Skarmeta, A.F.; Linares, L.J.</i>	
Optimización de operadores reticulares en un entorno de síntesis automática de controladores borrosos	107
<i>Deschamps, J.P.; Martínez Torre, J.I.; Moreno, R.</i>	
Approximate-based multi-modal systems and possibilistic logic: a comparative study	113
<i>Esteva, F.; Godo, L.; García, P.; Rodríguez, R.</i>	
Algunos tipos particulares de ecuaciones en relaciones difusas	119
<i>Fernández, M.J.; Suárez, F.</i>	
Diagnosis de equipos mecánicos basada en lógica difusa	125
<i>Ferreiro, R.; Pardo, X.C.</i>	
Un algoritmo para el refinamiento de reglas difusas de control	131
<i>González, A.; Pérez, R.</i>	
Conjuntos, puntos y números graduales	137
<i>Herencia, J.A.</i>	
Una generalización de la entropía a través de la teoría de Dempster-Shafer	143
<i>Herencia, J.A.; Lamata, M.T.</i>	
Distancias e índices parciales de medidas difusas	149
<i>Hernández Molinero, L.D.; Salmerón, A.</i>	
Medidas lingüísticas de racionalidad para alcanzar el consenso en grupo	155
<i>Herrera, F.; Herrera-Viedma, E.; Verdegay, J.L.</i>	
Subconjuntos difusos de un anillo. Ideales difusos	161
<i>Jiménez, J.; Montes, S.; Suárez, F.</i>	
XFL: un lenguaje para la definición de sistemas difusos	167
<i>López, D.R.; Sánchez-Solano, S.; Barriga, A.</i>	
Red neuronal para fuzzyficación y reconocimiento de patrones	173
<i>López Aligüé, F.J.; Acevedo, M.I.</i>	
Aplicación de técnicas temporales difusas a la discriminación de vocablos simples	179
<i>Martínez, D.; Mas, O.; Rolando, M.</i>	
Razonamiento aproximado en presencia de conceptos temporales difusos	185
<i>Mas, O.; Palà, P.; Miró, J.M.</i>	
Evolución genética de controladores borrosos	191
<i>Matellán, V.; Molina, J.M.; Fernández, C.</i>	
Juegos simples difusos	197
<i>Molina, E.; Tejada, J.</i>	

ε -particiones y α -equivalencias	201
<i>Montes, S.; Jiménez, J.; Gil, P.</i>	
Aplicación de técnicas difusas al diseño de algoritmos para visión por computador	207
<i>Montseny, E.; Sobrevilla, P.</i>	
FUZZY-LISP: un modelo de cálculo funcional borroso	213
<i>Pérez de la Cruz, J.L.; Morales, R.; Conejo, R.</i>	
Una arquitectura para bases de datos relacionales difusas deductivas	219
<i>Pons, O.; Medina, J.M.; Vila, M.A.; Cubero, J.C.</i>	
ID3f+A. Algoritmo de aprendizaje inductivo borroso con división intervalar automática de los atributos	225
<i>Ramos, G.; López Muñoz, J.</i>	
Segmentación difusa de clientes bancarios usando redes neuronales artificiales	231
<i>Requena, I.; López, E.</i>	
Gramáticas formales para el procesamiento de lenguaje natural vago	235
<i>Sobrino, A.</i>	
Integración difusa para medidas del tipo s y ns . Teorema de Radon-Nikodym	241
<i>Suárez Díaz, E.; Suárez, F.</i>	
Discretización de atributos cuantitativos en sistemas basados en el conocimiento	247
<i>Valls, A.; Torra, V.</i>	

II Jornada sobre Transferencia de Tecnología Fuzzy

Controlador borroso secuencial segmentado para lógica borrosa implementado en FPGA	255
<i>Aranguren, G.; Rodríguez Gómez, M.; Barrón, M.</i>	
RICE: un sistema fuzzy en la conservación y estudio del comportamiento del arroz ensilado ...	261
<i>Buldain, G.; Buldain, P.; Lafarga, A.; Pina, A.; Serón, F.J.</i>	
Localización y clasificación borrosa de defectos de fabricación en planchas de vidrio	267
<i>Junco, L.A.; Sánchez, L.</i>	
INFEDEC 2.1: sistema de ayuda en la decisión de diagnóstico y tratamiento de enfermedades infecciosas	271
<i>Lago, R.; Olivas, J.A.; Pérez, J.; Grávalos, C.; Megido, J.A.</i>	
Un prototipo de sistema experto difuso para la valoración visual de paisajes	277
<i>Martínez-Béjar, R.; Cadenas, J.M.; Martín, F.</i>	
FREE: un entorno visual para edición de consultas difusas sobre servidores de fuzzy SQL.....	281
<i>Medina, J.M.; Rashid, D.; Torralbo, J.L.; Vela, R.; Vila, M.A.</i>	
Optimizador fuzzy para paletizado en la industria conservera	287
<i>Puyosa, H.D.; Soto Alarcón, P.; Guardiola, F.; Tomás, L.M.</i>	
Índice de autores	291



Asturias 10 - 13 Septiembre

Ya llegamos a la sexta edición del Congreso Español sobre Lógica y Tecnologías Fuzzy y a la II Jornada sobre Transferencia de Tecnología Fuzzy. Parece que ambas celebraciones se consolidan y ello se debe al esfuerzo de todos los que colaboran: desde aquí quiero agradecer a los miembros de los comités de programa y de organización, así como a los autores de trabajos y a los participantes en general, su labor muchas veces callada y sin lucimiento personal.

Espero que en estas páginas se reflejen sólo una parte de las vivencias de estos días asturianos, días que deseo que queden, en la mente de todos, en el archivo de los más gratos recuerdos.

Pedro Gil

Comité de Programa

Dr. Pedro Albertos
Dr. Claudi Alsina
Dr. Senén Barro
Dr. José M. Cadenas
Dr. Luis M. Campos
Dr. Juan Luis Castro
Dr. Norberto Corral
Dr. Miguel Delgado
Dr. Francesc Esteva
Dr. Ricardo García Rosa
Dra. M. Carmen García Alegre
Dra. M. Angeles Gil
Dr. Antonio González
Dr. Julio Gutiérrez
Dr. Joan Jacas
Dr. Ramón L. de Mántaras
Dr. Roque Marín
Dr. Serafín Moral
Dr. Javier Montero
Dr. Aníbal Ollero
Dra. M. Angeles de Prada
Dr. Alejandro Sobrino
Dr. Enric Trillas
Dr. Llorenç Valverde
Dr. José Luis Verdegay
Dra. Amparo Vila

Comité Organizador

D. Pedro Gil
D^a. M. Angeles Gil
D. José Labra
D^a. M. Teresa López
D^a. Susana Montes
D^a. Antonia Salas
D. Luciano Sánchez

UN MODELO DE CLASIFICACION CON ESTRUCTURAS BINARIAS DE PREFERENCIA DIFUSA

Ana del Amo

Dpto. Estadística e I. Operativa
Facultad de CC. Matemáticas
Universidad Complutense
Madrid 28040
aamo@sunes1.mat.ucm.es

Javier Montero

Dpto. Estadística e I. Operativa
Facultad de CC. Matemáticas
Universidad Complutense
Madrid 28040
monty@sunes1.mat.ucm.es

Resumen

La mayoría de los modelos al uso para clasificación asumen, de manera directa o indirecta, la posibilidad de una representación de los objetos a clasificar dentro de un espacio con buenas propiedades, sobre el que es factible definir algún tipo de distancia. Esa representación subyacente, que con frecuencia es la base de los argumentos de carácter intuitivo que hace el decisor, es la mayor parte de las veces un producto elaborado a partir de la comparación sistemática entre las diferentes opciones que se le han presentado al decisor. En este trabajo se persigue modelizar un problema de clasificación cuando este análisis comparativo forma parte esencial de la información disponible, junto con una somera descripción de la *calidad* del conjunto de opciones presentado.

Palabras Clave: agregación, relaciones de preferencia, clasificación.

1 Introducción

El proceso de un problema estandar de clasificación supone unas *clases* previamente definidas, siendo el objetivo la asignación de una de estas clases a cada objeto que se nos pueda presentar,

dentro de una determinada familia de objetos. En general, estas *clases* no están perfectamente definidas. Quizá simplemente se dispone de una vaga descripción de unos *prototipos* básicos, de tal modo que cada objeto puede no estar al cien por cien en ninguna de esas clases, o estar en varias en intensidades variables. Incluso puede ocurrir que a lo largo del proceso se tenga evidencia de que la familia de clases inicialmente propuesta es insuficiente (algún aspecto de la realidad puede no estar siendo capturado).

La asignación de una clase a cada objeto pasa de hecho por un análisis global de todas las clases disponibles. Este análisis global se puede modelizar en un primer estadio considerando que el individuo lleva a cabo la comparación sistemática de todas las parejas de clases, definiendo por tanto una relación binaria de preferencia difusa, como la arriba descrita. Si esta relación determinase un orden completo nítido, la asignación de una clase podría resultar inmediata (asignar a cada objeto la clase que ocupa el máximo de dicha relación). Aún así, este método de asignación -análogo al de máxima verosimilitud en estadística- puede inducir a error, pues como ya se ha comentado, los objetos no necesariamente han de estar al cien por cien en una y sólo una de las clases consideradas.

La hipótesis de que el análisis global se lleva a cabo comparando sólo parejas es sin duda restrictiva, y puede ser causa de una complejidad de cálculo excesiva (de hecho, lo usual es restringir el análisis de las comparaciones sólo a una subfamilia relevante que contiene las clases más plausibles para cada objeto). Se trata sin embargo de una hipótesis estandar desde el punto de vista matemático. En la realidad, el método utilizado por el decisor para llevar a cabo un análisis

global del conjunto de alternativas o clases depende de su capacidad para comprender una mayor o menor complejidad. La información excesivamente compleja se descompone en unidades asequibles, para luego agregar de alguna manera a todos esos análisis parciales. En este proceso de agregación la condición de asociatividad suele jugar un papel clave, pero sin duda existen condiciones más débiles que garantizan también la operatividad del proceso de cálculo (cfr., por ejemplo [4,19] y también [5,17]). Es precisamente la visión global del problema la causa frecuente de la violación del principio de alternativas irrelevantes en la toma de decisiones. La introducción de una nueva clase puede provocar profundos cambios en la clasificación hasta ese momento llevada a cabo.

En una primera fase, basándonos en una distancia D definida en el espacio real de representación, calcularemos bien la distancia a unos prototipos definidos a partir de experimentaciones previas, bien los centroides para una clasificación inicial. A partir de los prototipos o de la clasificación inicial, se podrá proponer algún algoritmo -preferiblemente interactivo- para obtener una clasificación difusa del conjunto de objetos.

Por ejemplo, todos los colores que vemos en la vida real son combinaciones de tres colores básicos (rojo, azul y amarillo), que en la naturaleza aparecen muy rara vez en su versión pura. Todo color que se nos presente será una combinación de esos tres colores básicos, pero probablemente no sea ninguno de ellos. La asignación de la combinación más adecuada de colores pasa por un análisis comparativo entre los colores puros, para determinar en qué medida el objeto dado es más de un color que de otro. La situación se complicaría desde luego si nos hubiésemos olvidado de incluir un determinado color básico en el listado de colores disponibles para la clasificación. La adición posterior de ese nuevo color básico provocaría la revisión automática de todas las asignaciones anteriores.

Además, no es tan frecuente que *a priori* se disponga de una buena familia de *clases*, sino que esta familia de clases se encuentra normalmente a partir de una a veces larga experiencia. Con el tiempo esperamos, precisamente a partir de la experiencia, encontrar una descomposición *canónica* en términos de unas características *elementales*.

Al principio, casi cada objeto es una clase, y luego vamos descubriendo que necesitamos más clases, o que las ya definidas son de alguna manera *redundantes* -o presentan otras propiedades poco convenientes o simplemente no deseables- y deben ser por tanto revisadas. Considerar los colores *azul*, *amarillo* y *verde* como sistema de clasificación será simultáneamente insuficiente y redundante. Además, la eliminación eventual de la clase *verde* presentaría previsiblemente más ventajas que la eliminación de la clase *azul* (todo objeto *verde* es simultáneamente *algo amarillo* y *algo azul*).

La observación secuencial de los diferentes objetos obliga en general a modificar no sólo el número de clases disponibles -creando clases nuevas- sino que también las clases que fueron previamente introducidas son susceptibles de ser modificadas. Se trata de un proceso de mejora sucesiva en el que cada paso puede afectar al modelo particular de clasificación utilizado en el estadio anterior.

En este trabajo vamos a considerar uno de los eslabones de ese proceso de aprendizaje. La decisión sobre la clase se elaborará agregando la información contenida en las relaciones binarias asociadas a cada objeto (en qué medida está cada objeto más en una clase que en otra), pero no supondremos que la familia de clases está ya lograda sea cual sea el nuevo objeto que se presente en el futuro (hipótesis por otra parte normalmente inverificable). A estas relaciones binarias se les dotará de una estructura de preferencia [6,7]. No se pretende desarrollar un modelo de clasificación en base a otras relaciones binarias como las de similaridad (ver, por ejemplo, [18]).

Junto a la clasificación propuesta, debemos simultáneamente conocer de alguna manera hasta qué punto se ha logrado un buen sistema de clasificación. No basta con tener la mejor de las clasificaciones posibles para la familia de clases que hemos considerado, porque aún siendo la mejor de entre las posibles, dicha clasificación puede estar muy lejos de ser considerada *buen*a.

Por tanto, junto a la información de carácter binario (probablemente según más de un criterio), debe existir una información no binaria que ha de valorar el modelo concreto que hemos usado. La clasificación que se lleve a cabo intentará ser en algún sentido *la mejor* de entre las posi-

bles. No se debe de obviar el hecho de que por unos motivos u otros se haya introducido *a priori* alguna importante restricción en la familia de clasificaciones que serán consideradas.

Se plantea de este modo un problema de clasificación que tiene en cuenta dos tipos de información. De una parte, las relaciones binarias que establecen la comparación entre clase y clase, para cada objeto. Y de otra parte, una información no binaria que valora la calidad del modelo en cuanto tal. La clasificación se llevará a cabo teniendo en cuenta estas dos informaciones complementarias (muchos métodos de decisión, como los basados en la idea de *outranking* (ver, por ejemplo, la introducción a los métodos ELECTRE en Roy [14,15]) elaboran a partir de una única valoración numérica unidimensional sendas relaciones binarias con significados diferentes, información multidimensional que luego ha de ser coherentemente agregada). Esta información de carácter global, como ya se ha comentado, nos permitirá evitar versiones excesivamente fuertes de los axiomas de independencia de alternativas (clases) irrelevantes.

2 Estructuras de Preferencias

Sea $C_1, C_2, \dots, C_p$ la familia C de clases inicialmente dada. Sea $X_1, X_2, \dots, X_q$ la familia X de objetos bajo consideración. Cada objeto X_i lo supondremos representado mediante un vector $(x_{i1}, x_{i2}, \dots, x_{in}) \in \mathbb{R}^n$, siendo X_{ic} el valor de la característica f_c para el objeto X_i . Estos valores podrán ser subproducto elaborado de una serie de valores directamente observados.

Nuestro objetivo es, dados el conjunto de objetos y la familia de *clases*, poder asignar a cada objeto una clasificación difusa en el conjunto de *clases*.

Para ello, a partir de la información agregada mediante un operador basado en los modelos OWA de Yager [20], se define para cada objeto X_i una relación de preferencia difusa [21]

$$\mu_{X_i} : C \times C \rightarrow [0, 1]$$

que asocie a cada par de clases (C_j, C_k) la intensidad de preferencia

$$\mu_{X_i}(C_j, C_k)$$

en la que el objeto X_i está mejor clasificado dentro de la clase C_j que en la clase C_k .

Estas preferencias entre clases puede ser entendida en sentido estricto (ver, por ejemplo, [12], aunque como bien se puntualiza en [7,11], debieran considerarse todos los posibles tipos de preferencia, no sólo la preferencia estricta, sino también la indiferencia, la preferencia débil y por supuesto la incomparabilidad. En este sentido, la aproximación propuesta en [6] nos da unas condiciones en las que todas estas relaciones de preferencia quedan determinadas -salvo isomorfismo en el intervalo unidad- por una cualquiera de ellas.

Nuestro modelo de clasificación difusa debe ser capaz de procesar la información contenida en todas esas relaciones binarias de preferencia difusa que conforman nuestra estructura de preferencias.

Nuestra clasificación difusa es entendida como un conjunto difuso

$$\mu : X \times C \rightarrow [0, 1]$$

que nos indique el grado $\mu(X_i, C_j)$ en el que cada objeto X_i está en una clase C_j .

A veces la familia de clases define una partición difusa (cfr. [1]) en el sentido de [16], de tal modo que sea cual sea el objeto dado se tiene

$$\sum_{C_j \in C} \mu(X_i, C_j) = 1$$

Pero es evidente que ésta no debe ser una hipótesis impuesta *a priori* sin más, sino más bien el resultado de un elaborado análisis de resultados y consideraciones previas.

Como se ha puesto ya de manifiesto en el ejemplo de los colores, ni en general suman la unidad, ni tienen que sumar al menos la unidad, ni han de sumar como mucho la unidad. Una clasificación en términos de una partición difusa es el producto de una experiencia previa que ha permitido, tras aproximaciones sucesivas, la definición de una familia de clases con buenas propiedades. Si las funciones de pertenencia no llegan a sumar la unidad, algún aspecto de nuestro objeto no queda suficientemente bien explicado, y quizá se requiera una clase adicional. Si las funciones de pertenencia suman más de la unidad, convendría en principio revisar la definición de cada una de las

clases, buscando posibles redundancias. La aparición de incomparabilidades a la hora de comparar dos clases podría deberse a un problema de heterogeneidad, lo cual nos podría sugerir la revisión de la familia de clases con la que estamos trabajando.

A partir de una estructura de preferencia, que ha sido definida para cada objeto, se ha de obtener el grado en que ese objeto está en cada clase. Este conjunto difuso se puede obtener mediante un proceso de agregación, justificable axiomáticamente. Sin embargo, ésta no debe ser la única información a tener en cuenta. La relación binaria de preferencia nos habla de la posición relativa entre las clases. Existe en la realidad una información adicional de carácter global, que tenida en cuenta, puede modificar sustancialmente las preferencias que de hecho serán utilizadas para la clasificación. Si la familia de clases es *mala*, esto crea tensiones en la clasificación final que, a modo de *desconfianza* en la misma, puede y debe quedar reflejada de alguna manera.

3 Relación de Outranking Difuso

La relación *estar más en una clase que en otra* es en muchos contextos claramente de naturaleza difusa. No sólo porque las preferencias suelen presentar grados de intensidad, sino porque a veces el concepto de clase no está bien definido. Las fronteras entre una clase y otra no son estrictas, como se puede ver por ejemplo, en la clasificación de cubiertas vegetales en Teledetección (cfr. [8], o el texto en castellano del Prof. Chuvieco [3]). Los problemas de clasificación son sin duda uno de los campos en los que la Teoría de Conjuntos Difusos ha encontrado una más rápida aceptación.

Sin embargo, no es menos cierto que una información parcial puede tener consecuencias directas en el proceso final de evaluación. Basta citar la relación de *outranking* [14,15] (ver también, por ejemplo, Perny-Roy [13] y Martel *et al.* [10]).

Se dice que C_j *outranks* (desplaza a) C_k si la preferencia de C_j sobre C_k ha sido establecida de una manera tan fuerte bajo una de las características f_c , que nunca se aceptará que la alternativa C_k sea desplazada por C_j en el proceso final de selección.

Se trata éste de un enfoque muy adecuado ante un problema de decisión entre varias alter-

nativas concretas, cuando estamos forzados a escoger necesariamente alguna de entre estas alternativas listadas.

Sin embargo, en los problemas de clasificación sobre los que estamos trabajando, éstas clases no están totalmente definidas en muchos casos, sino que se admite la posibilidad de definir nuevas clases. El hecho de tener información discordante es esencial para este objetivo. Parte de esta discordancia debe ser capturada en forma de incomparabilidad, pero también se ha de guardar en mente el hecho de que una clase se adapte mucho mejor bajo un criterio que otra clase, y que esta última clase en ningún caso deba convertirse en la clase finalmente escogida. Se trata pues de un enfoque dual al de *outranking* que tiende a eliminar clases. Al contrario de los modelos de *outranking*, este proceso puede perfectamente conducir a descubrir que ninguna de las clases tenidas en cuenta se adapta bien al objeto considerado. Pero consideramos este enfoque no sólo más acorde con la realidad, sino que además nos aporta información muy útil a la hora de definir las requeridas nuevas clases.

La crítica anterior es también aplicable a los modelos de Toma de Decisiones que hacen uso del concepto de *outranking* (de hecho, parte de los análisis gráficos desarrollados para los modelos PROMETHEE-GAIA [2,9] persiguen de algún modo una mejor visualización de las alternativas para mejor negociar y encontrar nuevas alternativas o soluciones de compromiso).

4 Objetivo de este Trabajo

En este trabajo se propone un algoritmo adaptado a problemas de clasificación en Teledetección, a partir de procedimientos ya existentes basados en el concepto de *outranking*, y que tiene en cuenta las consideraciones anteriores.

En particular, se presenta un primer algoritmo sobre el que se pueda construir un proceso más elaborado de análisis para clasificación que, de una parte, aproveche la potencialidad inherente de las estructuras de preferencia (incomparabilidad incluida), y de otra, sea suficientemente versátil ante las propias sugerencias del análisis realizado.

El procedimiento presentado es sólo una pieza dentro de un complejo proceso de clasificación, intuido éste no como una *caja negra*, sino como

un proceso sobre el que el propio decisor debe ir actuando, redefiniendo el problema a la luz de los resultados que se van obteniendo.

References

- [1] J.C. Bezdek and J.D. Harris, Fuzzy partitions and Relations: an axiomatic basis for clustering, *Fuzzy Sets and Systems* 1:111-127 (1978)
- [2] J.P. Brans, B. Mareschal and Ph. Vincke, How to select and how to rank projects: the PROMETHEE method for MCDM, *European Journal of Operational Research* 24:228-238 (1986)
- [3] E. Chuvieco, *Fundamentos de Teledetección Espacial* (Rialp, Madrid, 1990)
- [4] V. Cutello and J. Montero, Recursive families of OWA operators, *FUZZ-IEEE'94* (IEEE Press, Piscataway, NJ, 1994)
- [5] V. Cutello and J. Montero, Hierarchical aggregation of OWA operators, basic measures and related computational problems, *Uncertainty, Fuzziness and Knowledge-Based Systems* 3:17-26 (1995)
- [6] V. Cutello and J. Montero, A general model for deriving preference structures from data, *European Journal of Operational Research* (to appear)
- [7] J.C. Fodor and M. Roubens, Valued preference structures, *European Journal of Operational Research*, 79:277-286 (1994)
- [8] J.R. Jensen, *Introductory Digital Image Processing: a Remote Sensing Perspective*, (Prentice Hall, Upper Saddle River, NJ, 1995)
- [9] B. Mareschal, Geometrical representations for MCDA: the GAIA procedure, *European Journal of Operational Research* 34:69-77 (1988)
- [10] J.M. Martel, G.R.D. D'Avignon and J. Couillard, Fuzzy outranking relation in multicriteria decision making, *European Journal of Operational Research*, 25:258-271 (1986)
- [11] S. Ovchinnikov, Structure of fuzzy binary relations, *Fuzzy Sets and Systems* 6:169-195 (1981)
- [12] S. Ovchinnikov and M. Roubens, On strict preference relations, *Fuzzy Sets and Systems* 43:319-326 (1991)
- [13] P.P. Perny and B. Roy, The use of fuzzy outranking relations in preference modelling, *Fuzzy Sets and Systems* 49:33-53 (1992)
- [14] B. Roy, Partial preference analysis and decision-aid: the fuzzy outranking relation concept, in: D. Bell, R. Keeney and H. Raiffa (eds.) *Conflicting Objectives in Decisions*, Wiley, New York (1977)
- [15] B. Roy, The outranking approach and the foundations of ELECTRE methods, *Theory and Decisions* 31:49-73 (1991)
- [16] E.H. Ruspini, A new approach to clustering, *Information and Control* 15:22-32 (1969)
- [17] T.L. Saaty, Exploring the interface between hierarchies, multiple objectives and fuzzy sets, *Fuzzy Sets and Systems* 1:57-68 (1978)
- [18] S. Tamura, S. Higuchi and K. Tanaka, , *IEEE Trans. Syst., Man, Cybern.* 1:61-66 (1971)
- [19] R.R. Yager, Quasi-associative operations in the combination of evidence, *Tech. Rep. MII-505* (Iona College, New Rochelle, NY)
- [20] R.R. Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making, *IEEE Transactions on Systems, Man and Cybernetics* 18:183-190 (1988)
- [21] L.A. Zadeh, Similarity relations and fuzzy orderings, *Information Science* 3:177-200 (1971)

PROGRAMACIÓN LINEAL POSIBILISTA CON RESTRICCIONES FLEXIBLES

M.Mar Arenas*, Amelia Bilbao*, Mariano Jiménez**, M. Victoria Rodríguez*

*Dpto. Matemáticas
Universidad de Oviedo
e-mail: VRODRI@hp845.econo.univi.es

**Dpto. Economía Aplicada I
Universidad del País Vasco
e-mail: eupjilom@se.ehu.es

Resumen

En este trabajo, partiendo de las ideas de Buckley, estudiamos un programa general de programación lineal posibilista en el que todos los parámetros pueden ser fuzzy, así como las restricciones. Obtenemos la distribución de posibilidad de la solución en función del grado de cumplimiento de las restricciones. Con esta información se le plantea al decisor que especifique una meta para su objetivo, y estudiamos como alcanzar dicha meta con un nivel satisfactorio sopesándolo con el grado de cumplimiento de las restricciones.

Palabras Clave: Distribución de posibilidad, Principio de extensión, Programación lineal fuzzy, Toma de decisiones.

1 INTRODUCCIÓN

Presentamos en este trabajo una extensión de la programación lineal posibilista. Desde que Zadeh [17] introdujo la teoría de posibilidad se han desarrollado muchos trabajos en este campo. Los modelos de decisión posibilista han jugado un importante papel en el tratamiento de problemas prácticos de decisión. Nuestro trabajo se enmarca dentro de las aplicaciones de la teoría de la posibilidad a problemas de programación lineal con coeficientes imprecisos o inciertos, pero cuya distribución de posibilidad puede ser estimada, de manera que sus valores vienen delimitados por la función de pertenencia de un número fuzzy. Estos problemas han sido objeto de varios trabajos con enfoques alternativos [13]. Por ello parece necesario precisar cual va a ser nuestro entorno de trabajo:

Nosotros partimos fundamentalmente de los artículos de Buckley [3, 4, 5, 7] y Julien [11]. Estos autores plantean un programa lineal posibilista (PLP) utilizando la siguiente notación:

$$\begin{aligned} \max Z &= \bar{c}x \\ \text{s.a. } \bar{A}x &\leq \bar{b} \\ x &\geq 0 \end{aligned} \quad (1)$$

donde $\bar{c} = (c_1, \dots, c_n)$ es un vector $1 \times n$ de números fuzzy, $\bar{b} = (b_1, \dots, b_m)$ es un vector $m \times 1$ de números fuzzy y $\bar{A} = [\bar{a}_{ij}]$ es una matriz $m \times n$ de números fuzzy.

Para determinar la distribución de posibilidad de la solución del problema (1) Buckley y otros autores [Julien 1994, Arenas, Bilbao, Jiménez y Rodríguez 1995] consideran el siguiente programa lineal crisp (PL):

$$\begin{aligned} \max Z &= cx \\ \text{s.a. } Ax &\leq b \\ x &\geq 0 \end{aligned} \quad (2)$$

de manera que si Z^* es el valor máximo de Z y $\Pi(c, A, b)$ es la distribución de posibilidad conjunta de todos los parámetros. Entonces la posibilidad de que un determinado valor z sea solución es la siguiente

$$\Pi(Z^* = z) = \sup_{c, A, b} \{ \Pi(c, A, b) \mid \text{la solución de (2) es } z \}$$

una observación respecto al enfoque utilizado por Buckley, es que no es necesario definir una relación de orden entre números fuzzy, ya que la desigualdad ($\leq$), que aparece en el problema (1), se ha de interpretar como un simbolismo que hace referencia a la desigualdad ordinaria del problema (2).

Puede demostrarse que bajo condiciones poco restrictivas la distribución de posibilidad de Z^* , se corresponde con un número fuzzy [2].

2 UN MODELO GENERAL DE PROGRAMACIÓN LINEAL POSIBILISTA

Trataremos en este trabajo de abordar un problema lineal posibilista más general que el estudiado por Buckley, en el cual el decisor permita una cierta violación de las restricciones. Este programa lo denotaremos así:

$$\begin{aligned} \max Z &= \bar{c}x \\ \text{s.a. } \bar{A}x &< \bar{b} \\ x &\geq 0 \end{aligned} \quad (3)$$

La desigualdad $<$ la interpretamos a partir de un vector difuso de tolerancia máxima $\bar{t}_0$, dado por el decisor.

De forma que el siguiente problema lineal posibilista con coeficientes imprecisos: $\bar{c}, \bar{A}, \bar{b} + \bar{t}$,

$$\begin{aligned} \max Z &= \bar{c}x \\ \text{s.a. } \bar{A}x &\leq \bar{b} + \bar{t} \\ x &\geq 0 \end{aligned} \quad (4)$$

con $\bar{t} = (1 - \beta)\bar{t}_0$, se entenderá que es una concreción de (3) cuando se cumplen las restricciones con un grado igual a β .

Al igual que hicimos entre (1) y (2), para determinar la distribución de posibilidad de la solución del problema (4), consideramos el siguiente programa crisp

$$\begin{aligned} \max Z &= cx \\ \text{s.a. } Ax &\leq b + t \\ x &\geq 0 \end{aligned} \quad (5)$$

Con el fin de darle al decisor la capacidad de admitir un mayor o menor grado de cumplimiento de las restricciones (véase apartado 3), daremos a β un tratamiento distinto al de otros grados de pertenencia y no lo integraremos en la función de pertenencia global. Aplicando el principio de extensión determinamos la distribución de posibilidad de la solución Z_β^* para cada valor del parámetro β :

$$\Pi(Z_\beta^* = z_\beta) = \sup_{c, A, b+t} \{ \Pi(c, A, b+t) \mid \text{la solución de (5) es } z_\beta \}$$

y según lo comentado anteriormente la distribución de posibilidad de cada Z_β^* se corresponde con un número fuzzy.

Entonces si consideramos algunos valores para el grado de cumplimiento de las restricciones: $\beta=0, 0.25, 0.50, 0.75, 1$, obtendríamos un resultado similar a lo representado en la siguiente figura (obsérvese que cuanto mayor sea el grado de cumplimiento de las restricciones menor será el valor óptimo):

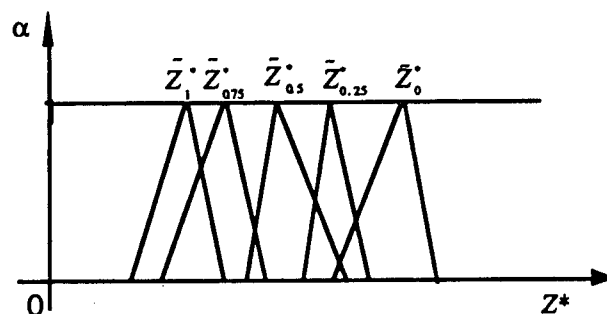


figura 1

Este gráfico aporta una información muy valiosa para decisor.

Ejemplo numérico

Con el fin de comparar nuestra aproximación al problema con la propuesta por Delgado, Verdegay y Vila [9, 10], vamos a estudiar el ejemplo numérico propuesto por los mencionados autores:

$$\begin{aligned} \max Z &= 5x_1 + 6x_2 \\ \text{s.a. } \bar{3}x_1 + \bar{4}x_2 &\leq \bar{18} \\ \bar{2}x_1 + \bar{1}x_2 &\leq \bar{7} \\ x_1, x_2 &\geq 0 \end{aligned}$$

donde:

$$\begin{aligned} \bar{3} &= (3, 2, 4), \quad \bar{4} = (4, 2.5, 5.5), \quad \bar{18} = (18, 16, 19) \\ \bar{2} &= (2, 1, 3), \quad \bar{1} = (1, 0.5, 2), \quad \bar{7} = (7, 6, 9) \end{aligned}$$

el decisor permite un límite máximo para la violación de cada restricción de:

$$\bar{t}_1 = (3, 2.5, 3.5), \quad \bar{t}_2 = (1, 0.5, 1.5)$$

Obsérvese que en este ejemplo los coeficientes que intervienen en la función objetivo son crisp. En el método de resolución de Delgado, Verdegay y Vila esto debe ser así, sin embargo con el método propuesto en el presente trabajo no es necesario.

α	$\tilde{z}_0^*$		$\tilde{z}_{0.25}^*$		$\tilde{z}_{0.5}^*$		$\tilde{z}_{0.75}^*$		$\tilde{z}_1^*$	
0	19.5	56	19.13	53.83	18.75	51.66	18.17	49.5	17.53	47.33
0.25	22.53	48.2	21.79	46.37	21.05	44.55	20.31	42.73	19.57	40.9
0.50	25.37	41.92	24.51	40.36	23.66	38.81	22.8	37.26	21.94	35.71
0.75	28.68	36.81	27.69	35.48	26.7	34.15	25.71	32.82	24.72	31.48
1	32.6	32.6	31.45	31.45	30.3	30.3	29.15	29.15	28	28

Tabla 1

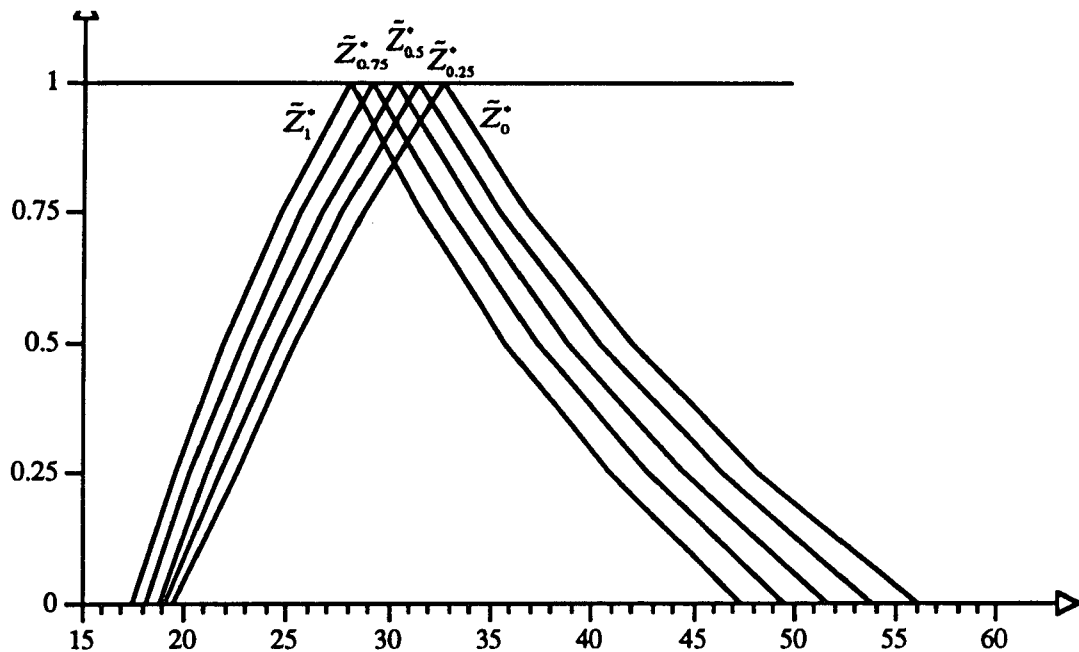


figura 2

Basándonos en el principio de extensión utilizamos las operaciones con α -cortes [Buckley 1992], [Arenas, Bilbao, Jiménez, Rodríguez 1995] para determinar las distribuciones de posibilidad $\tilde{z}_\beta^*$ de la solución de (3), para cada valor β de cumplimiento de las restricciones, (nosotros consideraremos $\beta=0, 0.25, 0.5, 0.75, 1$).

La tabla 1 corresponde a los resultados obtenidos para algunos α -cortes. La figura 2 representa los resultados de dicha tabla.

Comparemos estos resultados con los obtenidos por Delgado, Verdegay y Vila: dependiendo del criterio de clasificación de números fuzzy empleado, estos autores obtienen los siguientes resultados:

$$z^{(18)} = 28 + 4.6(1 - \beta)$$

ó

$$z^{(19)} = 17.53 + 2.56(1 - \beta)$$

Puede observarse que estos resultados son un caso particular de los obtenidos por nosotros:

$z^{(18)}$ se corresponde con los núcleos de las diferentes distribuciones de posibilidad obtenidas por nosotros. Es decir en este caso el decisor estaría contemplando las situaciones de mayor posibilidad de ocurrencia dentro de cada grado de cumplimiento de las restricciones.

$z^{(19)}$ se corresponde con la situación más pesimista es decir con los extremos inferiores de los 0-cortes de cada $\tilde{z}_\beta^*$.

Como vemos nuestro enfoque permite al decisor obtener mayor información, puesto que existen otras muchas posibilidades, que el decisor también debe sopesar.

3 CASO EN EL QUE EL DECISOR ESPECIFICA UNA META COMO OBJETIVO

Cuando se presentan restricciones flexibles, es frecuente en la literatura admitir que el decisor se plantea la resolución del problema como la obtención de un determinado nivel de aspiración [18, 19]. Sin embargo como dicen Cadenas y Verdegay [8], "no es realista empezar por pedirle al decisor que proporcione la meta y su máxima tolerancia, sin proporcionarle ninguna información acerca de la misma".

A la vista de la información contenida en la figura 1, se le solicita al decisor que especifique una meta $\bar{M}$ y su límite de tolerancia $\underline{M}$. De forma que si $Z^* \geq \bar{M}$, le resultará completamente satisfactorio, pero si $Z^* \leq \underline{M}$, su grado de satisfacción será nulo. es decir la meta viene expresada mediante un conjunto fuzzy $\bar{M}$, cuya función de pertenencia es la siguiente:

$$\mu_{\bar{M}}(Z^*) = \begin{cases} 0 & \text{si } Z^* \leq \underline{M} \\ \lambda \in [0, 1] & \text{si } \underline{M} \leq Z^* \leq \bar{M} \\ 1 & \text{si } Z^* \geq \bar{M} \end{cases}$$

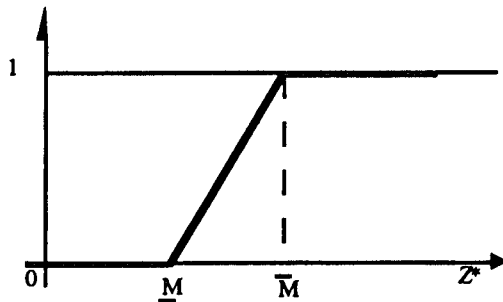


figura 3

El decisor, lógicamente, pedirá obtener un grado de satisfacción máximo. Ahora bien, la consecución de un valor mayor para la función objetivo será a costa de un menor nivel de cumplimiento de las restricciones. En estas condiciones puede que el decisor se conforme con un menor grado de satisfacción en sus aspiraciones a cambio de que las restricciones se verifiquen en mayor grado.

Vamos pues a establecer un criterio que permita al decisor sopesar el nivel β de cumplimiento de las restricciones, con la consecución de la meta en un grado satisfactorio. Para ello nos planteamos comparar el subconjunto fuzzy $\bar{M}$, que representa la aspiración del decisor con cada $\bar{Z}_\beta^*$, que representan las distribuciones de posibilidad del

óptimo obtenido para los diferentes grados de cumplimiento de las restricciones.

Si superponemos la figura 3 sobre la figura 1, obtenemos lo siguiente:

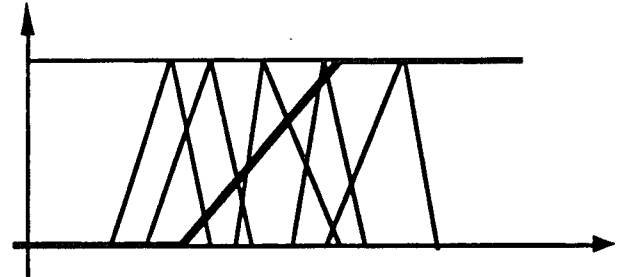


figura 4

Como hemos dicho debemos establecer un índice que nos permita comparar cada $\bar{Z}_\beta^*$ con $\bar{M}$, es decir que nos proporcione el grado en el que $\bar{Z}_\beta^*$ cumple las expectativas del decisor.

Proponemos el siguiente índice [12]. (Véase figura 5)

$$k(\bar{Z}_\beta^*, \bar{M}) = \frac{\text{área } \zeta_1}{\text{área } \zeta}$$

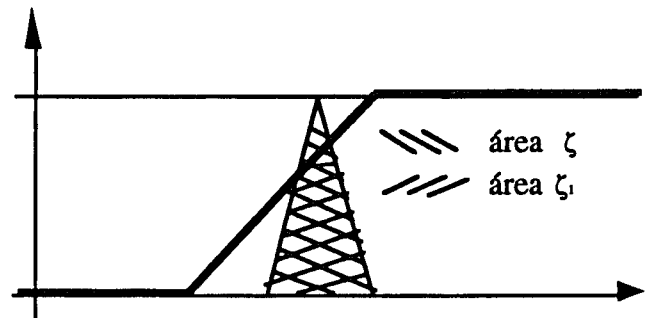


figura 5

Entonces si Z_α^L y Z_α^R son las funciones inversas de la parte izquierda y derecha de la función de pertenencia de $\bar{Z}_\beta^*$, y M_α es la inversa de la función de pertenencia de $\bar{M}$, escribiremos que

$$k(\bar{Z}_\beta^*, \bar{M}) = \frac{S_1 - S_2}{S_3 - S_4}$$

donde

$$s_1 = \int_{\{\alpha | z_{\alpha}^R \geq M_{\alpha}\}} (z_{\alpha}^R - M_{\alpha}) d\alpha$$

$$s_2 = \int_{\{\alpha | z_{\alpha}^L \geq M_{\alpha}\}} (z_{\alpha}^L - M_{\alpha}) d\alpha$$

$$s_3 = \int_0^1 z_{\alpha}^R d\alpha$$

$$s_4 = \int_0^1 z_{\alpha}^L d\alpha$$

Este índice de comparación es una generalización del índice de Yuan [15] para la comparación de números fuzzy. Lo hemos elegido porque verifica las propiedades de: *robustez, racionalidad, poder de discriminación y expresión fuzzy o lingüística de la relación de preferencia.*

Llegados a este punto el decisor deberá sopesar el grado de satisfacción de sus expectativas con el grado en que permita se violen las restricciones.

A la vista de la información contenida en la figura 4, podemos solicitar al decisor que establezca un umbral de aceptación de sus aspiraciones, por ejemplo $k(\tilde{Z}_{\beta}^*, \tilde{M}) = 0.7$, que en términos lingüísticos podría expresarse como "bastante satisfactorio".

Entonces se calcularía:

$$\beta^* = \max_{\beta} \{k(\tilde{Z}_{\beta}^*, \tilde{M}) \geq 0.7\}$$

Así β^* , sería el grado de cumplimiento de las restricciones que el decisor está dispuesto a admitir, y por tanto $\tilde{Z}_{\beta^*}$ representaría la distribución de posibilidad de la solución del problema (3) inicialmente planteado.

4 CONCLUSIONES

Hemos abordado en este trabajo un problema lineal posibilista más general que los que suelen estudiarse en la literatura: aquel en el que todos los coeficientes pueden ser fuzzy y además las restricciones son flexibles. Utilizando la teoría de posibilidades hemos calculado la distribución de posibilidad de la solución de este problema para diferentes grados de cumplimiento de las restricciones, y comprobado que algunos de los resultados obtenidos por otros autores pueden considerarse como un caso particular del obtenido por nosotros. Cuando el decisor se plantea la solución del problema como la obtención de un determinado nivel de aspiración, hemos estudiado como alcanzar la meta propuesta en un grado satisfactorio sopesándolo con el grado de cumplimiento de las restricciones.

Referencias

- [1] M.M. Arenas, A. Bilbao, M.Jiménez y M.V. Rodríguez (1995). Resolución de un programa lineal difuso mediante la teoría de posibilidad. *Actas ESTILF'95*. Murcia.
- [2] M.M. Arenas, A. Bilbao, M.Jiménez y M.V. Rodríguez. A theory of possibility approach to the solution of a fuzzy linear programming. (Pendiente de publicación)
- [3] J.J. Buckley (1988). Possibilistic linear programming with triangular fuzzy numbers. *Fuzzy Sets and Systems* 26 135-138.
- [4] J.J. Buckley (1989). Solving possibilistic linear programming problems. *Fuzzy Sets and Systems* 31 329-341.
- [5] J.J. Buckley (1990). Multiobjective possibilistic linear programming. *Fuzzy Sets and Systems* 26 135-138.
- [6] J.J. Buckley (1992). Solving fuzzy equations. *Fuzzy Sets and Systems* 50 1-14.
- [7] J.J. Buckley (1995). Joint solution to fuzzy programming problems. *Fuzzy Sets and Systems* 72 215-220.
- [8] J.M. Cadenas and J.L. Verdegay (1995). PROBO: an interactive system in fuzzy linear programming. *Fuzzy Sets and Systems* 76 319-322.
- [9] M. Delgado, J. Kacprzyk, J.L. Verdegay and M.A. Vila (1994). *Fuzzy Optimization. Recent Advances*. Springer-Verlag.
- [10] M. Delgado, J.L. Verdegay and M.A. Vila (1989). A general model for fuzzy linear programming. *Fuzzy Sets and Systems* 29 21-29
- [11] B. Julien (1994). An extension to possibilistic linear programming. *Fuzzy Sets and Systems* 64 195-206.
- [12] A. Kaufmann, y J. Gil Aluja (1987). *Técnicas operativas de gestión para el tratamiento de la incertidumbre*. Hispano Europea. Barcelona.
- [13] Y. Lay and C. Hwang (1992). *Fuzzy Mathematical programming. Methods and applications*. Springer-Verlag.

- [14] B. Werners (1987). An interactive fuzzy programming system. *Fuzzy Sets and Systems* 23 131-147.
- [15] Y. Yuan (1991). Criteria for evaluating fuzzy ranking methods. *Fuzzy Sets and Systems* 44 139-157.
- [16] L. Zadeh (1975). The concept of a linguistic variable and its appication to approximate reasoning-I. *inform. Sci..* 8 199-249.
- [17] L. Zadeh (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems* 1 3-28.
- [18] H.J. Zimmermann (1976). Description and optimization of fuzzy systems. *Internat. J. general Systems* 2 209-15
- [19] H.J. Zimmermann (1993). *Fuzzy Sets Theory and its Applications* . Kluber Academic Publishers.

Reglas Difusas para la interpretación de Redes Neuronales Artificiales entrenadas*

José M. Benítez, Juan L. Castro, Ignacio Requena

Dpto. Ciencias de la Computación e I. A. ETS Ingeniería Informática

Universidad de Granada. 18071 - GRANADA

e-mail: {J.M.Benitez, castro, requena}@decsai.ugr.es

Resumen

En trabajos anteriores hemos presentado un sistema difuso basado en reglas, obtenido a partir de una red neuronal artificial (RNA) entrenada, que calcula exactamente lo mismo que la RNA. Teníamos así un sistema difuso, no equivalente sino igual a la RNA. Las reglas obtenidas tienen una estructura compleja, y planteábamos la posibilidad de descomponerlas para hacerlas más sencillas y entendibles fácilmente. En este trabajo presentamos esta descomposición, a través de un operador lógico, que llamamos o-interactivo, y que tiene sorprendentes propiedades.

Palabras Clave: Sistemas difusos aditivos, reglas difusas, redes neuronales, operador o-interactivo, interpretación de redes neuronales.

1 INTRODUCCIÓN

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

Una metodología muy usada en la identificación de sistemas, a partir de un conjunto de ejemplos de su com-

portamiento son las técnicas con Redes Neuronales Artificiales (RNAs).

A pesar de la gran variedad de métodos de representación del conocimiento existentes 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 muchos casos las reglas que describen el sistema pueden ser proporcionadas por un experto sobre el sistema. Pero en otras muchas ocasiones, en realidad la mayoría de las veces, o no hay experto o no nos puede aportar un conjunto de reglas adecuado. De lo que sí solemos disponer es de un conjunto de ejemplos del funcionamiento del sistema.

Se plantea entonces la posibilidad de usar algún método de aprendizaje que "aprehenda" la esencia del funcionamiento del sistema y sea capaz de reproducir su comportamiento. Y a partir de este método obtener un conjunto de reglas difusas que nos permitan desempeñar la misma tarea con un Sistema Basado en Reglas Difusas (SBRD). Esta idea ha sido desarrollado por múltiples investigadores utilizando algoritmos de lo más variado.

Las características de las RNAs, 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 han

*Este trabajo ha sido financiado en parte por el Proyecto PB 92-0945 de la DGICYT. Madrid

utilizado mucho para la obtención de un conjunto de reglas difusas que describan el sistema en forma "entendible" por los posibles usuarios.

Sin embargo, a las RNAs se les achaca su característica de caja negra, lo que impide la "explicación" de su funcionamiento interno. Ha habido muchos intentos de explicar este funcionamiento interno, básicamente mediante la obtención de sistemas (difusos), generalmente basados en reglas, equivalentes o idénticos a la RNA.

En [2] presentamos un sistema difuso aditivo que calcula exactamente lo mismo que una RNA entrenada, y dimos una idea de la forma que tenían las reglas del sistema, y la posibilidad de descomponerlas en reglas más sencillas y "entendibles".

En este trabajo hacemos un paseo rápido por algunos de los enfoques que usan RNAs, para obtener reglas difusas de un sistema, y profundizamos en la descomposición de las reglas obtenidas en [2], con la introducción de un operador, o-interactivo, con unas propiedades muy interesantes.

2 RNA PARA OBTENER REGLAS DIFUSAS DE SISTEMAS

En [6], los autores proponen tres modelos de redes neuronales para los que los pesos de las conexiones tienen un significado particular en la adquisición de reglas de pertenencia y para el ajuste de funciones de pertenencia, lo que desarrollan simultáneamente. Las redes se entrenan usando el algoritmo de retropropagación de errores. Entre sus ventajas cuenta con que las reglas producidas son sencillas y fáciles de interpretar por un ser humano.

En [7] se presenta una arquitectura de sistema de inferencia difusa basada en una red adaptativa de 5 capas denominada ANFIS. El sistema se ofrece como base para construir un conjunto de reglas difusas capaces de representar el conocimiento contenido en un conjunto de pares (entrada, salida).

Tan sólo los nodos de las capas 1 y 4, que contienen los antecedentes y los consecuentes, son adaptativos. Los parámetros de los nodos adaptativos se ajustan me-

diante gradiente descendente. Está considerado como uno de los más eficientes de funcionamiento, una vez entrenado.

En [1], los autores presentan una metodología para obtener las reglas difusas que describen un sistema, a partir de un conjunto de ejemplos de su comportamiento. Partiendo del conjunto de todas las reglas posibles formadas con las variables relevantes al sistema y sus valores (etiquetas lingüísticas) se eliminan las que no se adaptan a la RNA entrenada con los ejemplos del sistema y después, con un algoritmo tipo greedy se obtiene la mejor combinación de reglas.

El procedimiento indicado en [3] es constructivo y se puede aplicar en dos ámbitos: a un conjunto de datos reales o a un conjunto de datos difusos del comportamiento del sistema. Obtiene primero las variables relevantes del sistema y forma con ellas las estructuras de las reglas, que llama pre-reglas. Posteriormente se obtienen las etiquetas (números difusos) de cada variable en cada pre-regla, para obtener las reglas definitivas.

Destacan como principales aportaciones la translación al ambiente difuso del modo de establecer las dependencias o relaciones causales entre entradas y salidas y definir la metodología para procesar información expresada ya en términos difusos.

3 EQUIVALENCIA ENTRE RNAs Y SISTEMAS DIFUSOS

El estudio de la relación entre las redes neuronales artificiales y los sistemas basados en reglas difusas ha dado diversos frutos. Primero, Buckley [4] estableció la equivalencia de ambos modelos en los siguientes términos: Dada una red neuronal (Perceptron multicapa) para cualquier $\epsilon > 0$ es posible encontrar un sistema basado en reglas difusas que aproxime la red con un error menor que ϵ . Y, recíprocamente, dado un sistema basado en reglas difusas es posible encontrar una red neuronal artificial que los aproxime con un error menor que un cierto $\epsilon > 0$. Kosko [8] demostró la existencia de la misma relación entre sistemas basados en reglas del tipo Takagi-Sugeno-Kang (TSK) con infe-

rencia suma ponderada, los llamados Sistemas Difusos Aditivos. Castro [5] extendió los resultados previos a un conjunto más amplio de sistemas basados en reglas difusas (controladores).

Sin embargo, en todos estos casos los resultados quedan a un nivel puramente teórico. La aproximación arbitraria de la red requiere un número de reglas que tiende a infinito, y la aproximación del sistema requiere un número de neuronas ocultas que también tiende a infinito. Esto no es aplicable en problemas reales.

Profundizando en esta relación que vincula redes neuronales y sistemas basados en reglas difusas, Benítez, Castro y Requena [2] han establecido la igualdad entre ambos, no tan sólo una aproximación más o menos cercana, sino la igualdad exacta. Además, la demostración que proveen es constructiva. Dan un método para obtener un conjunto de reglas tal que un sistema difuso aditivo basado en ellas calcula exactamente lo mismo que la red. Por último, la ventaja está en que el número de reglas es finito y proporcional al tamaño de la red. El resultado mencionado es el siguiente:

Teorema Sea una red neuronal de propagación hacia adelante con tres capas (entrada, oculta y salida), y con función de activación sigmoide en la capa oculta y función de activación lineal en la salida. Entonces existe un sistema difuso aditivo que calcula exactamente la misma función que la red.

Demostración: Para describir el sistema nos basta con dar una base de reglas. Se utilizan reglas del tipo TSK. Por cada par de unidades (oculta, salida) se añade una regla difusa R_{jk} :

$$R_{jk} : \text{Si } \sum_{i=1}^n x_i w_{ij} \text{ es } A \text{ entonces } y_k = \beta_{jk} \quad (1)$$

donde A es un subconjunto difuso de $\mathbb{R}$ cuya función de pertenencia es justamente la función de activación de las neuronas ocultas.

Dado que se trata de un sistema difuso aditivo, el grado de activación de la regla R_{jk} , v_{jk} , es $A(\sum_{i=1}^n w_{ij} x_i)$, y la salida del sistema es el vector cuyas componentes están dadas por:

$$y_k = \sum_{j=1}^h A(\sum_{i=1}^n x_i w_{ij}) \cdot \beta_{jk} \quad (2)$$

De (1) y (2), fácilmente se puede comprobar que cada salida y_k del sistema difuso es exactamente la misma que la salida correspondiente de la red. c.q.d.

Se pueden hacer un par de observaciones. Primero, se utilizan reglas simples de la forma "Si z es A , entonces $y = v$ " donde $v \in \mathbb{R}$ y z es una nueva variable obtenida por un simple cambio de variable de las n entradas. Segundo, el subconjunto difuso A se puede interpretar como "mayor que aproximadamente r ", donde r es un número real positivo obtenido a partir de un α -corte preestablecido. Dado que la función logística sólo puede alcanzar los valores 0 y 1 asintóticamente, la convención usual en los trabajos sobre redes neuronales es considerar los niveles de activación de 0'1 y 0'9, respectivamente. Por tanto, podemos establecer un α -corte para $\alpha = 0'9$ e interpretar el subconjunto difuso con función de pertenencia A como "mayor que aproximadamente 2'19", dado que $A(2'19) = 0'9$.

De la relación establecida entre RNAs y SBRDs se derivan dos conclusiones importantes: Por un lado, cualquier resultado descubierto para uno de los modelos se puede aplicar al otro. Por otro lado, el conocimiento que una RNA codifica en sus pesos sinápticos se puede expresar en una forma más inteligible para los seres humanos. Lo que a su vez, nos lleva a dos ideas interesantes: es posible establecer una interpretación de las RNAs y también obtener un método para la adquisición automática del conocimiento.

Las reglas usadas en el sistema son difusas. Tienen una significación clara para un matemático o un experto en lógica difusa. Sin embargo, son deseables reglas que se puedan interpretar más fácilmente. Por ejemplo, encontrando una descomposición de las premisas de las reglas de modo que se puedan reescribir como:

$$R_{jk} : \text{Si } x_1 \text{ es } A_{jk}^1 \theta x_2 \text{ es } A_{jk}^2 \theta \dots \theta x_n \text{ es } A_{jk}^n \text{ entonces } y_k = \beta_{jk} \quad (3)$$

donde θ es un conectivo lógico y A_{jk}^i son subconjuntos difusos obtenidos de A y los pesos w_{ij} .

El primero paso iría en la dirección de intentar una descomposición de las reglas obtenidas al tipo más simple y ampliamente utilizado, o sea, reglas cuyos antecedentes están conectados con el operador AND. Claramente, esta sería la situación ideal, pero esta descomposición no es posible. Es fácil comprobar

la respuesta negativa. Las reglas difusas se pueden ver como relaciones difusas. La descomposición que se intenta significa que la relación correspondiente a cada regla (1) debería ser expresada como una composición de relaciones. Sin embargo, Yager demostró que esto no es posible en todos los casos. Con respecto a la conectiva OR aparece un problema similar. Consiguientemente, se buscan conectivos alternativos.

La solución que incorporamos aquí, viene dada por un operador difuso nuevo que permite reescribir las reglas de forma más comprensible. Este operador se nota por $*$, recibe el nombre de *o-interactivo* y viene dado por:

$$a * b = \frac{a \cdot b}{(1-a) \cdot (1-b) + a \cdot b} \quad (4)$$

El operador *o-interactivo* cumple algunas propiedades interesantes como:

1. $((0, 1), *)$ es un grupo abeliano.
2. $*$ es fácilmente extensible a n argumentos:

$$a_1 * a_2 * \dots * a_n = \frac{a_1 \cdot a_2 \cdot \dots \cdot a_n}{(1-a_1) \cdot (1-a_2) \cdot \dots \cdot (1-a_n) + a_1 \cdot a_2 \cdot \dots \cdot a_n}$$

3. $\lim_{a_i \rightarrow 0} a_1 * a_2 * \dots * a_n = 0 \quad \forall a_1, \dots, a_n \in (0, 1) \quad \forall i \in \{1, \dots, n\}$
4. $\lim_{a_i \rightarrow 1} a_1 * a_2 * \dots * a_n = 1 \quad \forall a_1, \dots, a_n \in (0, 1) \quad \forall i \in \{1, \dots, n\}$

5. $*$ es estrictamente creciente en cada argumento.

La definición de este operador es una construcción con indudable elegancia desde el punto de vista puramente matemático. Pero también tiene una interpretación clara desde el enfoque de agregación de enunciados modelados mediante lógica difusa.

Lo más interesante en este punto es que permite descomponer las reglas difusas propuestas inicialmente como:

$$R_{jk} : \text{Si } x_1 w_{1j} \text{ es } A * x_2 w_{2j} \text{ es } A * \dots * x_n w_{nj} \text{ es } A \text{ entonces } y_k = \beta_{jk} \quad (5)$$

Y, aceptando que " $x_i w_{ij}$ es A " se puede interpretar como " x_i es A/w_{ij} ", entonces se pueden reescribir

como se indica en la ecuación (3):

$$R_{jk} : \text{Si } x_1 \text{ es } A_{jk}^1 * x_2 \text{ es } A_{jk}^2 * \dots * x_n \text{ es } A_{jk}^n \text{ entonces } y_k = \beta_{jk} \quad (6)$$

Los subconjuntos difusos A_{jk}^i se obtienen de A y w_{ij} . Su función de pertenencia es:

$$\mu_{A_{jk}^i}(x) = \mu_A(x w_{ij})$$

Entonces es fácil comprobar que:

$$x_1 \text{ es } A_{jk}^1 * x_2 \text{ es } A_{jk}^2 * \dots * x_n \text{ es } A_{jk}^n = \sum_{i=1}^n x_i w_{ij} \text{ es } A \quad (7)$$

Obviamente, su interpretación se construye a partir de la de A y de los pesos aprendidos, w_{ij} . Si w_{ij} es positivo, entonces " x_i es A_{jk}^i " se puede leer como " x_i es mayor que aproximadamente r/w_{ij} ," donde r es un número real positivo obtenido de un α -corte previamente establecido. Si w_{ij} es negativo, entonces " x_i es A_{jk}^i " se puede entender como " x_i es menor que aproximadamente r/w_{ij} ." En lo relativo al valor absoluto, cuando éste aumenta, la proposición se hace más precisa y el punto de referencia, r/w_{ij} , se hace más pequeño. El efecto es el inverso cuando el valor absoluto decrece.

Además, cuando el peso es negativo, la proposición también se puede expresar en términos de una relación "mayor que". Esto se obtiene escribiendo la proposición negada "no x_i es $\neg A_{jk}^i$ ", donde $\neg A_{jk}^i$ es el complemento de A_{jk}^i , cuya función de pertenencia es

$$\mu_{\neg A_{jk}^i}(x) = 1 - \mu_{A_{jk}^i}(x) \quad (8)$$

Es fácil comprobar que $\mu_{\neg A_{jk}^i}$ es una función logística. De este modo, " x_i es menor que aproximadamente r " es lo mismo que " x_i no es mayor que aproximadamente $-r$."

Estas reglas permiten expresar el conocimiento extraído de la red de una forma significativa. Por tanto, ofrecen una interpretación de las RNAs.

Una de las propiedades más interesantes de esta interpretación es su independencia del algoritmo de aprendizaje. El método de entrenamiento utilizado no es

relevante, tan sólo lo es el conocimiento que la red ha aprendido.

Por último, exponemos un ejemplo de la vida real que proporciona una aplicación interesante del operador o-interactivo.

Consideremos un editor de una revista, que manda un trabajo para revisión a dos referees. Estos puntúan el trabajo para su aceptación de 0 a 1. El editor, normalmente, si los dos referees dan puntuaciones cercanas a 1 (cercanas a 0) acepta (rechaza) el trabajo. Si uno de ellos indica aproximadamente 0.5, el editor acepta lo que indica el otro referee. Si hay opiniones contradictorias (o los dos indican sobre 0.5), el editor queda indeciso.

Si tenemos en cuenta que el elemento neutro para el o-interactivo es precisamente 0.5, y recordamos su comportamiento, se refleja claramente lo que el editor hace normalmente.

4 CONCLUSIONES

En trabajos anteriores habíamos presentado un Sistema Aditivo Difuso, que calculaba exactamente lo mismo que una red neuronal entrenada, descrito con reglas TSK, y se indicaron posibilidades de descomposición de estas reglas para hacerlas más comprensibles.

Ahora hemos presentado un nuevo operador de la lógica difusa, el operador o-interactivo, que permite realizar esta descomposición de una forma sencilla. El operador o-interactivo tiene propiedades interesantes, que se han puesto de manifiesto, y una interpretación bastante natural.

Entonces, a través del Sistema Aditivo Difuso construido a partir de la RNA de propagación hacia adelante entrenada, y con la ayuda del o-interactivo, conseguimos una interpretación interesante de la RNA.

Referencias

- [1] Benítez, J.M., Blanco, A., y Requena, I. (1995). *An Empirical Procedure to Obtain Fuzzy Rules*

using Neural Networks. Proceedings of IFSA'95. Brasil, Julio.

- [2] Benítez, J.M., Castro, J.L, y Requena, I. (1995). *Las Redes Neuronales Artificiales son Sistemas Difusos Aditivos* Proceedings de ESTYLF'95. Murcia. Septiembre.
- [3] Benítez, J.M., Blanco, A., Delgado, M., y Requena, I. (1996). *Neural Methods for Obtaining Fuzzy Rules*. Aceptado para su publicación en Mathware.
- [4] Buckley, J.J., Hayashi, Y., and Czogala E. (1993). *On the equivalence of Neural Nets and Fuzzy Expert Systems*. Fuzzy Sets and Systems, **53**, pp. 129-134.
- [5] Castro, J.L. (1995). *Fuzzy Logic Controllers are Universal Approximators*. IEEE Trans. on Systems, Man, and Cybernetics. **25**(4), April.
- [6] Horikawa, S., Furuhashi, T., y Uchikawa, Y. (1992). *On Fuzzy Modeling Using Fuzzy Neural Networks with the Back-Propagation Algorithm*. IEEE Trans. on Neural Networks **3**(5), pp. 801-806, Septiembre.
- [7] Jang, J.S.R. (1993). *ANFIS: Adaptive-Network-Based Fuzzy Inference System*. IEEE Trans. on Systems, Man, and Cybernetics, **23**(3), pp. 665-685.
- [8] Kosko, B. (1994). *Fuzzy Systems as Universal Approximators*. IEEE Trans. on Computers, **43**(11), pp. 1324-1333.

Measure of a Fuzzy Set based on an arbitrary continuous t-Conorms: the Finite Case

Carlo Bertoluzza and Antonella Bodini
Dipartimento di Informatica e Sistemistica, Università
via Ferrata 1, Pavia, Italy

Abstract

The aim of this research is the construction of a compositive measure of information (see e.g.[2]) on a class (an algebra) of fuzzy subsets of an universe Ω . In a previous work (see [3]) we analyzed and solved the problem in two particular cases, corresponding to the two best known forms of the composition law. The present paper deals with the general case and uses the results obtained in [3]. In particular we try to construct a compositive information measure over the fuzzy space $(\Omega, \tilde{\mathcal{A}})$, starting from a compositive information measure defined on the crisp space $(\Omega, \mathcal{A})$, where $\mathcal{A}$ and $\tilde{\mathcal{A}}$ are suitable algebras of, respectively, crisp and fuzzy subsets of Ω .

As in [3] the problem has been analyzed in the case of membership functions with finitely many values. We think that the continuous case may be solved with a procedure analogous to the one developed in [7].

Key Words Decomposable Measure, t-Conorms, Sugeno's Integral

1. BASIC DEFINITIONS

Let $L = \{\alpha_0 \dots \alpha_n\}$ a finite subset of the interval $[0, 1]$ such that $0 \in L$ and $\alpha \in L \implies 1 - \alpha \in L$. A *L-fuzzy subset* of a given universe Ω is a mapping $\tilde{A}$ from Ω to L . So a fuzzy set $\tilde{A}$ is represented by a map

$$\tilde{A} : \Omega \longrightarrow L. \quad (1)$$

For each $\alpha \in [0, 1]$, the α_i -level of a fuzzy set $\tilde{A}$ is the crisp subset defined by

$$A_i = \{\omega \in \Omega \mid \tilde{A}(\omega) = \alpha_i\} \quad (2)$$

Union and intersection of two fuzzy sets are defined by means of the classical *max* and *min* operations, and it is evident that

$$\begin{aligned} (\tilde{A} \cup \tilde{B})_i &= A_i \cup B_i \quad [a] \\ (\tilde{A} \cap \tilde{B})_i &= A_i \cap B_i \quad [b] \end{aligned} \quad (3)$$

A *compositive information space* is a structure $(\Omega, \mathcal{S}, J)$ where

- $(\Omega, \mathcal{S})$ is a measurable space
- $J : \mathcal{S} \rightarrow \mathbb{R}^+$ is a map (*information measure*) with the following properties

$$J(\Omega) = 0, \quad J(\emptyset) = +\infty \quad [a]$$

$$A, B \in \mathcal{S}, \quad A \subset B \implies J(A) \geq J(B) \quad [b] \quad (4)$$

$$\begin{aligned} A \cap B = \emptyset &\implies \\ J(A \cup B) &= J(A) \top J(B) \end{aligned} \quad [c]$$

Composition law $\top$ has all the properties which follow from the compatibility with the union operation. It is easy to recognize that it is a triangular norm (*t-norm*) and its continuous general form (see e.g.[7]) is given by the following result:

Let Δ be a closed subset of $\mathbb{R}^+$. Its complementary subset is the union of a finite or countable open intervals $\Delta^c = \cup_{i \in I}]a_i, b_i[$, $I \subset \mathbb{N}$. For each $i \in I$ let $f_i : [a_i, b_i] \rightarrow \mathbb{R}$ be a continuous and strictly increasing function with $f_i(a_i) = 0$. Let us consider the following composition law:

$$x \top y = \begin{cases} f_i^{(-1)}[f_i(x) + f_i(y)] & \text{if } (x, y) \in [a_i, b_i] \times [a_i, b_i] \\ \max(x, y) & \text{otherwise} \end{cases} \quad (5)$$

where $f_i^{(-1)}(\xi) = f_i^{-1}[\min\{\xi, f_i(b_i)\}]$. Every function defined by (5) is a continuous t-conorm, and vice-versa every continuous t-conorm has the form (5) with a

suitable choice of the subset Δ and of the functions f_i . The proof is based on a result due to C.H.Ling ([4], see also [5]).

Now let $\tilde{\mathcal{S}}$ be an algebra of fuzzy subset of Ω . An information measure over $(\Omega, \tilde{\mathcal{S}})$ is a map $\tilde{J} : \tilde{\mathcal{S}} \rightarrow \mathbb{R}^+$ with the same properties of the crisp case, that is

$$\begin{aligned} \tilde{J}(\Omega) &= 0, \quad \tilde{J}(\emptyset) = +\infty & [a] \\ \tilde{A}, \tilde{B} \in \tilde{\mathcal{S}}, \quad \tilde{A} \subset \tilde{B} &\implies \tilde{J}(\tilde{A}) \geq \tilde{J}(\tilde{B}) & [b] \\ \tilde{A} \cap \tilde{B} = \emptyset &\implies & [c] \\ \tilde{J}(\tilde{A} \cup \tilde{B}) &= \tilde{J}(\tilde{A}) \tilde{J}(\tilde{B}) \end{aligned} \quad (6)$$

It is easy to recognize that $\tilde{T}$ is also a t-conorm and therefore it has the same representation of the law $\tilde{T}$, possibly with different structure of the closed subset $\tilde{\Delta}$ and of the functions $\tilde{f}_j$. More precisely, if we indicate by $\cup_{j \in J} [\alpha_j, \beta_j]$ the complement of $\tilde{\Delta}$, then we have

$$x \tilde{T} y = \begin{cases} \tilde{f}_j^{(-1)}[\tilde{f}_j(x) + \tilde{f}_j(y)] & \text{if } (x, y) \in [\alpha_j, \beta_j] \times [\alpha_j, \beta_j] \\ \max(x, y) & \text{otherwise} \end{cases} \quad (7)$$

Now we formulate our problem as follow. Suppose that a compositive information measure is defined on the crisp space $(\Omega, \mathcal{S})$, where $\mathcal{S}$ is the family of the α -levels of the elements of $\tilde{\mathcal{S}}$. We will construct the information measure of a fuzzy subset $\tilde{A}$ by means of the crisp information of its α -levels, that is

$$\tilde{J}(\tilde{A}) = V_n(J_1, \dots, J_n), \quad \text{where } J_i = J(A_i) \quad (8)$$

2. ONE-LEVEL SUBSET'S MEASURE

In order to simplify the exposition, let us introduce the following notations:

$$\alpha_k A(\omega) = \begin{cases} \alpha_k & \text{if } \omega \in A \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

$$\begin{aligned} x_i &= J(A_i), \quad y_i = J(B_i) & [a] \\ h_k(\xi) &= V_n(\underbrace{0, \dots, 0}_{k-1}, \xi, \underbrace{0, \dots, 0}_{n-k}) & [b] \end{aligned} \quad (10)$$

The α -levels of $\tilde{A} = \alpha_k A$ are

$$A_i = \begin{cases} A & \text{if } i = k \\ \emptyset & \text{otherwise} \end{cases} \quad (11)$$

It is easy to recognize that $\tilde{J}(\alpha_k A) = h_k[J(A)]$ and that function h_k has the following properties

$$\begin{aligned} h_k(+\infty) &= +\infty & [a] \\ x < y &\implies h_k(x) \leq h_k(y) & [b] \\ k < m &\implies h_k(x) \leq h_m(x) & [c] \end{aligned} \quad (12)$$

As we will see later the knowledge of the functions h_k will allow us to construct the measure of any fuzzy subset. Therefore we dedicate this paragraph to the study of the form of the function h_k . We omit, for sake of simplicity, subindex k from now till to the end of the paragraph.

Let us note, first of all, that we have

$$\alpha(A \cup B) = (\alpha A) \cup (\alpha B) \quad (13)$$

From the above identity we obtain immediatly the following compatibility condition (which we will denote as **fundamental equation**):

$$h[J(A)] \tilde{T} h[J(B)] = h[J(A) \tilde{T} J(B)] \quad (14)$$

This means that the subsets Δ , Δ^c and the families $\{f_i\}$, $\{\tilde{f}_j\}$ have to satisfy some suitable conditions in order that (14) would be always satisfied. In particular the following result hold (the proofs may be found in [8]).

Lemma 1 If $x \in \Delta$, then $h(x) \in \tilde{\Delta}$, that is $h(\Delta) \subset \tilde{\Delta}$, and consequently $\tilde{\Delta}^c \subset h(\Delta^c)$.

Lemma 2 For each $[a, b] \subset \Delta^c$, function $f([a, b])$ is either a point $\alpha \in \tilde{\Delta}$, or an interval $[\alpha, \beta]$ with $[\alpha, \beta] \subset \tilde{\Delta}^c$.

Lemma 3 In the same conditions of lemma 2, if $\alpha < \beta$, let $c = \inf\{x | h(x) = \beta\}$. Then we have

$$\begin{aligned} h(x) &= \beta \quad \forall x \in [c, b] & [a] \\ h([a, c]) &= [\alpha, \beta] & [b] \end{aligned} \quad (15)$$

Lemma 4 If $h([a, b]) = [\alpha, \beta]$, $\alpha < \beta$ and $f, \tilde{f}$ are the functions which determine the restriction of $\tilde{T}, \tilde{T}$ respectively to $[a, b]$, $[\alpha, \beta]$, then

$$\tilde{f} \circ h \circ f^{-1}(u) = \sigma \cdot u \quad (16)$$

Remark 1 Since the functions $\varphi, f^{-1}, \tilde{f}$ are strictly monotonic and continuous in the intervals where are defined, we can conclude that function h is also strictly monotonic in $[a, c]$ and continuous.

Remark 2 Let us remember that h is strictly monotonic in $[a, c]$ and $h([a, b]) = [\alpha, \beta]$. This fact imposes some constraint to the constant σ ; more precisely it is easy to recognize that $0 < \sigma \leq \frac{\tilde{f}(\beta)}{\tilde{f}(c)}$.

Now let us remember that: (i) function h depend on the index level k , (ii) function f refers to a particular subset $]a_i, b_i[$ of Δ^c , function $\tilde{f}$ is associated to the image $]\alpha_{j(i)}, \beta_{j(i)}[$ of the subset $]a_i, b_i[$: consequently σ depends on the indice i, k . So we can state in a more precise notation the following:

Corollary (of lemma 4). The functions $h_k, f_i, \tilde{f}_{j(i)}$ are related together by the following relations

$$\begin{aligned} \tilde{f}_{j(i)}(t) &= \sigma_{i,k} \cdot f_i \circ h_k^{-1}(t) & [a] \\ f_i(t) &= \frac{1}{\sigma_{i,k}} \tilde{f}_{j(i)} \circ h_k(t) & [b] \\ h_k(u) &= \tilde{f}_{j(i)}^{-1}[\sigma_{i,k} \cdot f_i(u)] & [c] \end{aligned} \quad (17)$$

3. INFORMATION OF THE FUZZY SET

Let us observe that a fuzzy subset $\tilde{A}$ can be represented by means of its α -levels as follows

$$\tilde{A} = \bigcup_{i=1}^n \alpha_i A_i \quad (18)$$

Since the α -levels are pairwise disjoint, we can obtain the fuzzy information of $\tilde{A}$ by means of

$$\begin{aligned} \tilde{J}(\tilde{A}) &= \tilde{J}(\bigcup_{i=1}^n \alpha_i A_i) = \\ \tilde{T}^n \tilde{J}(\alpha_i A_i) &= \tilde{T}^n h_i[J(A_i)] \end{aligned} \quad (19)$$

Definition We will call **discriminant element** of the subset $\tilde{A}$ the value

$$\gamma(\tilde{A}) = \gamma = \min\{\tilde{J}(\alpha_i A_i)\} \quad (20)$$

Theorem 3.1 If $\gamma \in \tilde{\Delta}$ then

$$\tilde{J}(\tilde{A}) = \gamma \quad (21)$$

The proof is an immediate consequence of expression (7) of the t-norm $\tilde{T}$.

If condition of theorem 3.1 is not satisfied, then γ is an interior point of a suitable interval $]\alpha, \beta[\subset \Delta^c$. Let us indicate by $S = S(\tilde{A})$ the subset of $\{1 \dots n\}$ composed by the indices s for which $h_s[m(A_s)] \in]\alpha, \beta[$. We have

Theorem 3.2 If $\gamma \notin \tilde{\Delta}$, then the information of the fuzzy set $\tilde{A}$ is given by

$$\tilde{J}(\tilde{A}) = \tilde{f}^{-1} \left[\sum_{s \in S} \sigma_{i,s} f_i \circ J(A_s) \right] \quad (22)$$

where $\tilde{f}$ is the function which determines the t-norm $\tilde{T}$ in the interval $]\alpha, \beta[$.

The proof is immediate: it is easy to recognize, from (7), that only the terms $h_s[J(A_s)] < \beta$ contributes to

the last term of equation (19); the others are "killed" by the fact that β is an $\tilde{T}$ -idempotent. Moreover if $y_s = h_s[J(A_s)] \in]\alpha, \beta[$, then we can use lemma 1 for deducing that $J(A_s)$ belong to a suitable interval $]a_i, b_i[\subset \Delta^c$ whose image, via h_s , is y_s . Then by using formula (17.c) of corollary of lemma 4 we obtain directly (22).

References

- [1] L.A.Zadeh, "Fuzzy sets", *Inform. Control* **8**(1965), 338-353.
- [2] B.Forte, "Measures of information; the General Axiomatic Theory" *RIRO, R2*(1969) 63-90
- [3] C.Bertoluzza, T.Brezmes, M.L.Capodieci, G.Naval, "Compositive information Measure of a Fuzzy Set" *manuscript*
- [4] C.-H.Ling "Representation of associative functions", *Publ. Mathematicae* **12** (1965), 189-212.
- [5] P.S.Mostert and A.L.Shields, "On the structure of semigroups on a compact manifold with boundary", *Ann. of Math.* **65**(1957), 117-143.
- [6] J.Aczel, "Lectures on functionalequations and their applications", *Academic Press* (1966).
- [7] C.Bertoluzza, D.Cariolaro, "Measure of s Fuzzy Set based on Continuous t-Conorms" to appear in *Fuzzy Set and Systems*
- [8] A.Bodini, "Sugli spazi sfumati di informazione e di incertezza", *Tesi di Laurea Università di Pavia* (1995).

Modelo para Codificación de Reglas IF - THEN Expresadas en Términos Lingüísticos.

A. Blanco
Dpto. Ciencias de la
Computación e I. A.
E.T.S.I. Informática
Universidad de Granada.
18071 Granada.
aragom@robinson.ugr.es

M. Delgado
Dpto. Ciencias de la
Computación e I. A.
E.T.S.I. Informática
Universidad de Granada.
18071 Granada.
aragom@robinson.ugr.es

W. Fajardo
Dpto. Ciencias de la
Computación e I. A.
E.T.S.I. Informática
Universidad de Granada.
18071 Granada.
aragom@robinson.ugr.es

Resumen

Presentamos un modelo de codificación para implementar un sistema de relaciones difusas sobre cualquier sistema de decisión discreto. La presentación comienza con la exposición de un modelo de codificación discreta de información expresada en términos lingüística, para continuar con su extensión a la codificación de reglas lingüísticas.

Palabras Clave: Codificación, Conjuntos Difusos, Reglas, Redes Neuronales.

1 INTRODUCCIÓN.

Los sistemas automáticos de tratamiento de datos han sido generalmente diseñados de forma que necesitan de un protocolo más o menos especializado de captación y devolución de información.

Parece pues más razonable diseñar sistemas de forma que admitan la información tal y como el usuario es capaz de proporcionarla, mediante conceptos vagos e imprecisos. Para esto, tenemos dos opciones:

- Diseñar nuevos sistemas capaces de trabajar directamente con información vaga o imprecisa, abandonando todos los obtenidos hasta el momento (con todas sus ventajas e inconvenientes).

- Diseñar algún método que a modo de *filtro* admita la información tal y como la proporciona el usuario y la convierta en datos conforme a las exigencias del protocolo específico del sistema a utilizar, permitiendo la reutilización de los sistemas ya existentes.

Desde un punto de vista formal el manejo de información y conceptos vagos es el objetivo de la parcela de la inteligencia artificial denominada Razonamiento Aproximado. Este pretende, partiendo de una colección de hechos (quizás vagos) y empleando premisas constituidas por afirmaciones o reglas imprecisas, obtener consecuencias (posiblemente también vagas o imprecisas). Las bases conceptuales para el desarrollo del razonamiento aproximado se encuentran en la teoría de subconjuntos difusos, desarrollada por Zadeh y, a partir de sus trabajos, por otros muchos investigadores contemporáneos [5], [9], [10], [11], [12].

Nuestro objetivo es investigar formas inmediatas de extender modelos *clásicos* (no difusos) de Inteligencia Artificial al campo difuso. Debido a que los modelos son básicamente discretos en origen (el tratamiento de información continua requiere realmente su discretización) parece razonable pensar en que lo óptimo es emplear una representación discreta de la información lingüística sobre modelos pensados en su origen como discretos. A partir de esta idea, es obvio que es más deseable manejar la información difusa con una representación discreta en lugar de la continua que se emplea habitualmente.

Trabajo financiado por el proyecto
PB 920945 de la DGICYT

Por todo lo expuesto hasta el momento pasamos a desarrollar un **método de codificación de datos imprecisos representados por medio de etiquetas lingüísticas**, de forma que podamos reutilizar modelos crisp ya existentes y de demostrada eficacia en modelos capaces de trabajar directamente con información lingüística.

2 CODIFICACION DE ETIQUETAS LINGÜISTICAS.

Un patrón puede representarse en forma discreta por múltiples y diferentes protocolos, en función del objetivo concreto que se pretenda cubrir con la correspondiente codificación. Las posibles codificaciones discretas son múltiples (Blanco [2], Ishibuchi [4], ...), en particular sus características dependen del problema a solucionar y del modelo específico para el que han sido diseñados. Por ejemplo Blanco propone una codificación para ser utilizada sobre sistemas de *Backpropagation*. Codifica la información difusa mediante una discretización de forma que si A representa a un conjunto difuso (y su correspondiente función de pertenencia), cuyo soporte es el intervalo $[u_0, u_1]$, se elige un k natural y se hace una partición del dominio, obteniendo los elementos:

$$s_i = u_0 + (i-1) \frac{u_1 - u_0}{k-1}, \quad i = \{1, 2, \dots, K\}$$

a partir de los cuales se asocia el vector:

$$\begin{aligned} \vec{a} &= (a_1, a_2, \dots, a_k) \\ &= (A(s_1), A(s_2), \dots, A(s_k)), \quad a_i \\ &= A(s_i) \in [0, 1] \end{aligned}$$

al vector

$$\vec{s} = (s_1, s_2, \dots, s_k)$$

Entonces cuando un sistema difuso se describe mediante reglas con antecedente y consecuente difuso, se discretiza de forma que si las reglas se componen de r y t variables en el antecedente

y el consecuente respectivamente, cada regla puede ser vista como una función de $[0, 1]^{n_1 + \dots + n_r}$ en $[0, 1]^{m_1 + \dots + m_t}$, donde n_i y m_j son los correspondientes tamaños de la discretizaciones referenciadas por $i = 1, \dots, r$ y $j = 1, \dots, t$. De esta forma, y para cualquier sistema difuso, la entrada puede ser modelada por un vector n -dimensional $(x_1, x_2, \dots, x_n)$ y la salida por un vector m -dimensional $(y_1, y_2, \dots, y_m)$ con $x_i, y_j \in [0, 1]$, $i = 1, \dots, n$; $j = 1, \dots, m$.

Observamos pues que la codificación propuesta por Blanco solventa el problema de la representación de conjuntos difusos. No utiliza la representación en forma de etiqueta, de ahí que cuando se restringe al enfoque lingüístico de los problemas solo puede representar información que venga expresada en términos lingüísticos. Por ello vamos a proponer otra codificación que permite no solo representar las etiquetas, sino cualquier valor de la variable lingüística, venga o no expresado en términos puramente lingüísticos.

3 CODIFICACION DE VARIABLES LINGÜISTICAS MEDIANTE DISCRETIZACION INCREMENTAL.

Consideremos una variable lingüística descrita por una quintupla $(H, T(H), V, G, M)$, donde:

- H es el nombre de la variable.
- $T(H)$ es el conjunto de términos.
- V es el universo de discurso de H .
- G es una regla sintáctica para generar cualquier elemento t de $T(H)$.
- M es una regla semántica para asociar a cada $t \in T(H)$ su significado $M(t)$ que es un conjunto difuso de V .

Basándonos en una forma de representación de información difusa altamente aceptada y muy extendida en el razonamiento en presencia de datos difusos (ver Delgado [3], Pedrycz [6], [7], [8]) en la que cualquier elemento u de V se representa en términos de $T(H)$ como:

$$u \equiv \alpha_1 t_1, \dots, \alpha_n t_n \quad (1)$$

$$\alpha_i \in [0, 1], t_i \in T(H)$$

siendo α_i la compatibilidad de u con t_i (medida empleando una función ad-hoc, en particular posibilidad, necesidad, o cualquier otro tipo de medida de incertidumbre).

A partir de esta representación proponemos codificar los valores de la variable tal y como se muestra a continuación.

Supongamos que $T(H)$ se compone de n elementos. Dotamos a $T(H)$ de un orden arbitrario y a cada término de $T(H)$ se asocia un vector de dimensión m (supuesto que se desee una precisión global de orden $\frac{1}{m}$). A la totalidad de $T(H)$ se asocia un vector de dimensión $m \times n$.

Entonces u se codifica en un vector binario (o bipolar) de dimensión $m \times n$:

$$C(u) = (C_{11}, \dots, C_{1m}, \dots, C_{n1}, \dots, C_{nm}) \quad (2)$$

donde:

$$\left\{ \begin{array}{l} \alpha_i = 0 \rightarrow C_{ij} = 0; \quad i = 1, \dots, n; \quad j = 1, \dots, m \\ \alpha_i \neq 0 \rightarrow \left\{ \begin{array}{l} \exists j \text{ t.q. } \frac{j}{m} \leq \alpha_i < \frac{j+1}{m} \\ C_{il} = \begin{cases} 1 & \text{si } l \leq j \\ 0 & \text{si } l > j \end{cases} \end{array} \right. \end{array} \right. \quad (3)$$

o lo que es lo mismo $C(u)$ es un vector cuyos componentes C_{ij} se calculan siguiendo el algoritmo expresado en el esquema 1.

Hemos obtenido un sistema de codificación capaz de expresar cualquier elemento del universo de discurso V en términos del conjunto $T(H)$.

Con esta idea como base podemos extender nuestra codificación al caso en que el valor de la variable venga dado por un subconjunto difuso A

sobre V . Basta seguir considerando α_i como el grado de compatibilidad de A con t_i medido adecuadamente.

En particular si $A = t_k$ entonces se codifica con:

$$C_{ij} = \begin{cases} 1 & \text{si } i = k, \forall j \\ 0 & \text{en otro caso} \end{cases} \quad (4)$$

En este caso es redundante utilizar un número elevado de bits por etiqueta, por lo que podemos ganar en economía utilizando un único bit ($m=1$) por término $t_i \in T(H)$.

```

1.- Desde i = 1 hasta n
  1.1.- Desde j = 1 hasta m
    1.1.1.- Si j ≤ Parte entera de (αi * m)
      1.1.1.1.- Entonces Cij = 1
    1.1.2.- Si j > Parte entera de (αi * m)
      1.1.2.1.- Entonces Cij = 0
  1.2.- Fin
2.- Fin

```

Esquema 1.

Al fijar el número de bits utilizados en la codificación estamos determinando la precisión máxima que vamos a permitir desde el principio al final del problema, por lo que tanto si utilizamos 1 b 10 bits por etiqueta estamos determinando que las apreciaciones más finas con las que podemos trabajar a lo largo del desarrollo son unarias o decimales respectivamente. Por lo cual hemos de prever desde el principio la precisión máxima con la que pretendemos trabajar.

4- CODIFICACION DE REGLAS LINGÜISTICAS MEDIANTE DISCRETIZACION INCREMENTAL.

Un sistema difuso se puede identificar mediante un sistema basado en reglas (Blanco [1]) y una regla es una asociación causa-efecto susceptible

de ser codificada de forma discreta. Además, una regla puede ser vista como una asociación de patrones, cuya complejidad es directamente proporcional a la de la regla en sí.

Tras ver en el apartado anterior el método de codificación de los valores que puede tomar una variable lingüística y las distintas casuísticas que pueden presentarse; en función de que la información venga expresada en términos de compatibilidad, en términos *crisp* o en términos puramente lingüísticos no debemos de encontrar problema alguna para codificar reglas expresadas en términos lingüísticos. Estas al igual que en el caso de las variables podrán presentar tres casuísticas distintas con respecto a los valores de su antecedente y consecuente. De esta forma una regla con antecedente y consecuente difuso, de la forma (A, B) donde A y B se describen mediante las n y p etiquetas sobre las que puede tomar valor las respectivas variables lingüísticas, se expresarán mediante vectores de características que tienen por componentes las n y p etiquetas lingüísticas sobre las que se evalúan A y B respectivamente. Así mismo, cada componente del vector (etiqueta lingüística) quedará descrita como una cadena de bits, más o menos larga en función de la sutileza de las posibles apreciaciones de las descripciones. Si la precisión del corte en todas las etiquetas lingüísticas del antecedente A es j y en las del consecuente B es k , la regla difusa (A, B) puede ser expresada como un par de vectores compuestos de $n \times j$ elementos y $p \times k$ respectivamente.

De esta forma si pretendemos modelar un sistema para memorizar pares de patrones de la forma (A, B) , en el cual A se describe mediante n etiquetas y B mediante p y si la precisión del corte en las etiquetas de A es j y k para las de B , entonces, (A, B) se expresa como un par de vectores de dimensión $n \times j$ y $p \times k$.

De forma más esquemática, (A, B) , descritos inicialmente como:

$$A_i = (a_{i1}, a_{i2}, \dots, a_{in})$$

$$B_i = (b_{i1}, b_{i2}, \dots, b_{ip})$$

Pasan a codificarse tal y como se muestra a continuación.

$$A = a_{11}, \dots, a_{1j}, \dots, a_{n1}, \dots, a_{nj}$$

$$B = b_{11}, \dots, b_{1j}, \dots, b_{p1}, \dots, b_{pj}$$

Puesto que ha quedado expuesto con la suficiente claridad en el apartado 3 como codificar una variable lingüística, pasamos ahora a mostrar con un ejemplo como obtener los vectores binarios (o bipolares) correspondientes a la representación de reglas como asociaciones causa efecto expresadas en términos lingüísticos.

EJEMPLO.

Supongamos que pretendemos modelizar un sistema de frenado. Dicho sistema pretende que un móvil se detenga en un determinado punto de una línea. Se encuentra caracterizado por el conjunto de reglas que se muestran en la Tabla I. Pretendemos además integrar dicho modelo de decisión (expresado en términos lingüísticos) sobre una máquina que únicamente trabaja con información *crisp*.

Podemos codificar la posición, la velocidad y la fuerza conforme a las etiquetas presentadas en Figura 1, Figura 2 y Figura 3.

Puesto que estamos interesados en una codificación con precisión decimal discretizaremos las etiquetas con 10 bits en función de sus α -cortes (fruto de las funciones de pertenencia a los correspondientes conjuntos difusos) para las diferentes casuísticas que se puedan presentar.

Trataremos únicamente el caso en el que la información venga expresada en términos de un grado de compatibilidad sobre cada etiqueta que puede tomar un valor perteneciente al intervalo

$[0, 1]$, la codificación, por ejemplo, resultante para la componente Y_1 del conjunto de patrones de entrenamiento, que es "una fuerza negativa media con grado de pertenencia 1" será:

1111111111 0000000000 0000000000 0000000000 0000000000

De esta forma codificamos los m pares de patrones de entrenamiento correspondientes a las reglas como pares de asociaciones (X, Y) donde X corresponde a Posición - Velocidad, e Y a Fuerza.

Tabla I

- 1- Si posición positiva media y velocidad cero, entonces fuerza negativa media.
- 2- Si posición positiva baja y velocidad positiva baja, entonces fuerza negativa baja.
- 3- Si posición positiva baja y velocidad negativa baja, entonces fuerza cero.
- 4- Si posición negativa media y velocidad cero, entonces fuerza positiva media.
- 5- Si posición negativa baja y velocidad negativa baja, entonces fuerza positiva baja.
- 6- Si posición negativa baja y velocidad positiva baja, entonces fuerza cero.
- 7- Si posición cero y velocidad cero entonces fuerza cero.

Con esta representación, parece lógico el hecho de que ateniéndonos a las limitaciones del correspondiente sistema anfitrión en cuestión de codificación y capacidad, podemos implementar sistemas difusos mediante sistema discretos, utilizando las técnicas suministradas por los diferentes autores para procesar los datos suministrados al sistema garantizando la recuperación de los patrones suministrados en el entrenamiento. Así pues tendremos la información difusa representada de forma discreta y tratarla conforme al protocolo propio del modelo elegido como anfitrión para el sistema difuso.

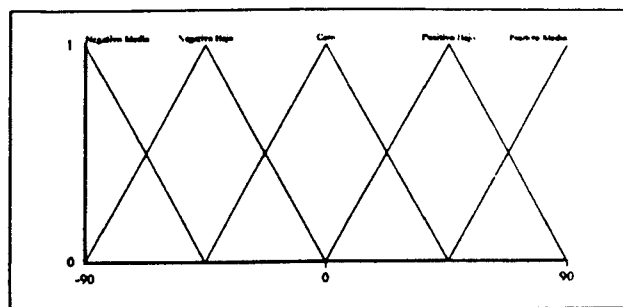


Figura 1: Representación Semántica de la Variable Lingüística Posición.

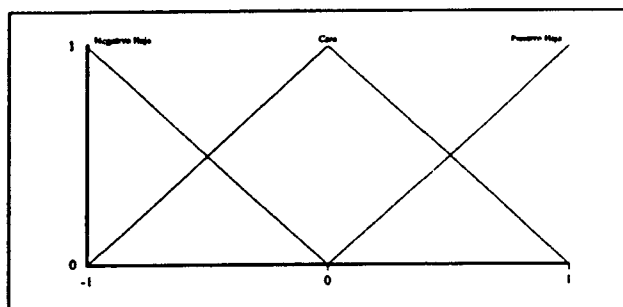


Figura 2: Representación Semántica de la Variable Lingüística Velocidad.

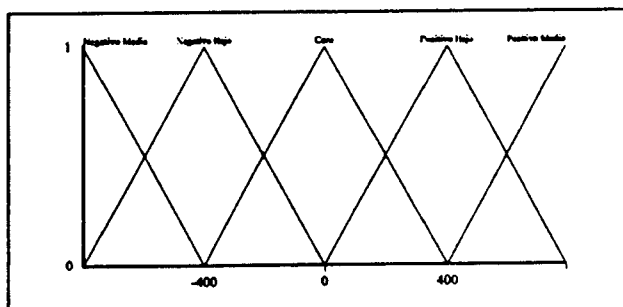


Figura 3: Representación Semántica de la Variable Lingüística Fuerza.

Una vez codificada la información presentaremos la información en forma de grados de pertenencia a las etiquetas lingüísticas del antecedente, de forma que si el grado de pertenencia de un determinado elemento a una determinada etiqueta es 0.3, el correspondiente sub-vector que representa este hecho tendría sus tres primeros bits a 1 y el resto (hasta el décimo, pues suponemos que la precisión es decimal) a 0. En particular y según el ejemplo si quisiéramos saber que fuerza hay que aplicar en el caso de una velocidad 0.2 negativa baja, 0.8 cero y una

posición 0.7 negativa media y 0.3 negativa baja, presentaríamos al sistema el siguiente vector:

1100000000 1111111100 0000000000
1111111000 1110000000 0000000000 0000000000 0000000000

al cual el sistema nos devolvería la correspondiente solución (en función de sus características propias). Solución que interpretaríamos tal y como hemos comentado en el apartado con anterioridad.

5 CONCLUSIONES.

Una vez descrito nuestro procedimiento de codificación, pasamos a destacar las ventajas del mismo, las cuales se han ido haciendo patentes a lo largo de la exposición.

- Hemos desarrollado un método para representar en binario o bipolar información expresada en términos de un sistema lingüístico (en términos de etiquetas previas).
- Con esta codificación podemos trabajar con problemas lingüísticos utilizando técnicas desarrolladas para problemas discretos.
- Una ventaja destacable de nuestro método de codificación, su simplicidad. Los modelos son fáciles de implementar, ya que se fundamentan en un mecanismo fácilmente inteligible.
- Análogamente hemos de destacar la facilidad de operación. El método no solo es fácil de manipular, sino que no supone un gran coste computacional a la hora de incrementar el número de patrones a codificar.

Referencias.

- [1] Blanco, A.; (1993). *Identificación de Sistemas Mediante Redes Neuronales*. Tesis Doctoral. Universidad de Granada.
- [2] Blanco, A.; (1994). *A Learning Procedure to Identify Weighted Rules by Neural Networks*. Fuzzy Sets and Systems, por publicar.
- [3] Delgado, M.; Verdegay, J. L.; Vila, M. A.; (1992). *A Linguistic Version of the Compositional Rule of Inference*. Preprints Sicica'92, Symposium of Intelligent Components and Instruments for Control Applications. 141 - 148.
- [4] Ishibuchi, H.; Tanaka, H.; Okada, H.; (1994). *Interpolation of Fuzzy If-Then rules by Neural Networks*. International Journal of Approximate Reasoning 1994; 10: 3-27.
- [5] Mizumoto, M.; Zimmermann, H. J.; (1982). *Comparison of Fuzzy Reasoning Methods*. Fuzzy Sets and Systems, 8, 253 - 283.
- [6] Pedrycz, W.; (1983). *Numerical and Application Aspects of Fuzzy Relational Equations*. Fuzzy Sets and Systems, 11, 1-18.
- [7] Pedrycz, W.; (1985). *Applications of Fuzzy Relational Equations for Method of Reasoning in Presence of Fuzzy Data*. Fuzzy Sets and Systems, 16, 163-175.
- [8] Pedrycz, W.; (1995). *Fuzzy Sets Engineering*. CRC Press, Inc.
- [9] Yanakawa, T.; (1987). *A Simple Fuzzy Controller Hardware System Employing MIN & MAX Operations, a Challenge to 6th Generation Computer*. Preprints of Second IFSA Congress, Tokyo, July 20 - 25, 827 - 830.
- [10] Zadeh, L. A.; (1973). *Outline of a New Approach to the Analysis of Complex Systems and Decision Processes*. IEEE Transactions on Systems Man and Cybernetics, SSMC-3 82 - 44.
- [11] Zadeh, L. A.; (1975). *The Concept of a Linguistic Variable and its Applications to Approximate Reasoning, Part I*, Information Sciences, vol. 8, 199 - 249, *Part II*. Information Sciences, vol. 8, 301 - 357, *Part III*. Information Sciences, vol. 9, 43 - 80.
- [12] Zadeh, L. A.; (1979). *A Theory of Approximate Reasoning*. Machine Intelligence 9 (J. E. Hayes, D. Michie, L. I. Mikulich, eds.) Elsevier, 149 -194.

Inferencia Gramatical Difusa mediante Redes Neuronales

Blanco A.

Dpto. Ciencias de la Computación
e Inteligencia Artificial
E.T.S.I Informática
Universidad de Granada
armando@decsai.ugr.es

Delgado M.

Dpto. Ciencias de la Computación
e Inteligencia Artificial
E.T.S.I Informática
Universidad de Granada
delgado@decsai.ugr.es

Pegalajar M.C.

Dpto. Ciencias de la Computación
e Inteligencia Artificial
E.T.S.I Informática
Universidad de Granada
mcarmen@decsai.ugr.es

Resumen

Mostramos un modelo de red neuronal difusa capaz de inferir gramáticas regulares difusas a partir de los ejemplos de entrenamiento. La red es entrenada mediante una retro-propagación del error basado en la derivada suavizada [1]. Una vez entrenada la red las relaciones difusas asociadas a las transiciones del autómata se encuentran reflejadas en los pesos.

Palabras Clave: Automatas Difusos, Redes Neuronales Recurrentes Difusas, Inferencia Gramatical Difusa.

1 Introducción.

1.1 Gramáticas, Inferencia Gramatical y Redes Neuronales Recurrentes.

En los últimos años la inferencia gramatical mediante redes neuronales recurrentes ha sido un problema extensamente tratado. En particular se ha demostrado que las redes neuronales recurrentes (RNN) tienen la capacidad de inferir gramáticas regulares [2, 4, 5, 10]. Durante el entrenamiento la red recurrente particiona su espacio de estados en regiones bien definidas que representan los estados correspondientes del AFD (autómata finito determinístico) que se intenta aprender. Sin embargo la inferencia gramatical difusamediante redes neuronales es un problema que aún sigue abierto. En este trabajo presentamos un modelo de red neuronal recurrente difusa mediante la cual se puede inferir, a partir de secuencias con grados de pertenencia a los estados del autómata la gramática difusa asociada. El esquema del trabajo es el siguiente : en primer lugar se describirá la arquitectura de la red

neuronal difusa que se propone, a continuación veremos el procedimiento de entrenamiento y cómo extraer el autómata. Finalmente se presentan los resultados obtenidos de la simulación para un ejemplo concreto.

1.2 Gramáticas Difusas e Inferencia Gramatical Difusa.

Una gramática regular difusa $\mathcal{G}$ es una cuádrupla $\mathcal{G} = \langle N, T, P, S \rangle$ donde N y T son símbolos no terminales y terminales, respectivamente. P es un conjunto de producciones de la forma $A \rightarrow^\theta a$ o $A \rightarrow^\theta aB$ donde $A, B \in N$, $a \in T$ y $\theta \in [0, 1]$.

En una gramática difusa $\mathcal{G}$ las secuencias pertenecen al lenguaje generado por la gramática con un grado de pertenencia. El grado de pertenencia $\mu_{\mathcal{G}}(x)$ de una secuencia $x \in T^*$ es el máximo valor de cualquier derivación de x donde el valor de una derivación específica de x es igual al mínimo grado de pertenencia de la producción usada:

$$\mu_{\mathcal{G}}(x) = \mu_{\mathcal{G}} \Rightarrow x = \max_{S \Rightarrow^* x} [\min \{\mu_{\mathcal{G}}(S \rightarrow \alpha_1), \mu_{\mathcal{G}}(\alpha_1 \rightarrow \alpha_2), \dots, \mu_{\mathcal{G}}(\alpha_m \rightarrow x)\}]$$

Un autómata se denomina difuso cuando sus estados están caracterizados por conjuntos difusos y la producción de respuestas y estados siguientes son facilitados por las apropiadas relaciones difusas. Un automata difuso, $\mathcal{A}$, es un sistema relacional difuso definido por la quintupla $\mathcal{A} = \langle \mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{R}, S \rangle$ donde $\mathcal{X}$ es un conjunto finito no vacío de estados de entrada (alfabeto de entrada), $\mathcal{Y}$ es un conjunto finito no vacío de estados de salida (alfabeto de salida), $\mathcal{Z}$ es un conjunto finito de estados internos, $\mathcal{R}$ es la relación difusa en $\mathcal{Z} \times \mathcal{Y}$ y S es una relación difusa en $\mathcal{X} \times \mathcal{Z} \times \mathcal{Z}$.

El autómata difuso opera de la siguiente manera:

Para cualquier estado de entrada difuso, A^t , la relación de transición de estados ternaria S es convertida en una relación binaria S_{A^t} , sobre $\mathcal{Z} \times \mathcal{Z}$ por la fórmula:

$$S_{A^t}(z_i, z_j) = \max_{k \in N_n} (\min(A^t(x_k), S(x_k, z_i, z_j)))$$

para todos los pares $(z_i, z_j) \in Z \times Z$

Suponiendo que el estado difuso actual es C^t , el siguiente estado E^t y el estado de salida difuso B^t es determinado por las composiciones max-min:

$$E^t = C^t \circ S_{A^t}; B^t = C^t \circ R$$

Consideremos, por ejemplo, una secuencia $A^1, A^2, \dots, A^S$ de r estados de entrada difusos aplicados a un estado difuso inicial dado C^1 . Entonces el automata difuso produce la secuencia de estados internos difusos:

$$E^1 = C^1 \circ S_{A^1}; E^2 = E^1 \circ S_{A^2}; \dots; E^r = E^{r-1} \circ S_{A^r}$$

y la correspondiente secuencia de estados de salida:

$$B^1 = C^1 \circ R; B^2 = E^2 \circ R; \dots; B^r = E^{r-1} \circ R$$

Si solamente estamos interesados en el estado final y el estado final de salida, podemos usar las fórmulas:

$$E^r = C^1 \circ S_{A^1} \circ S_{A^2} \circ \dots \circ S_{A^r}; B^r = C^1 \circ S_{A^1} \circ S_{A^2} \circ \dots \circ R$$

Dada una gramática difusa $\mathcal{G}$ existe un autómata difuso $\mathcal{A}$ tal que $L(\mathcal{G}) = L(\mathcal{A})$.

Por tanto para nosotros la inferencia gramatical será obtener el autómata difuso a partir de pares de ejemplos (α, β) donde α es una secuencia de caracteres a procesar por el autómata y β es un conjunto difuso que representa los grados de pertenencia a cada uno de los estados del autómata una vez leída la cadena de entrada.

2 La Red Neuronal Recurrente Difusa.

La red que proponemos consta de N neuronas recurrentes difusas etiquetadas S_j ; L neuronas de entrada no recurrentes I_k y $N^2 \times L$ pesos etiquetados W_{ijk} . Esta red acepta una secuencia de entrada ordenada en el tiempo y su dinámica se desenvuelve del siguiente modo:

$$S_i^{t+1} = \max_{j,k} \min(W_{ijk}, S_j^t, I_k^t) \quad (1)$$

Cada símbolo de la secuencia de entrada es codificado en las neuronas de entrada en cada paso de tiempo t de la siguiente forma:

a) Si el caracter de entrada es un 0, entonces $I_0 = 1$ e $I_1 = 0$.

b) Si el caracter de entrada es un 1, entonces $I_0 = 0$ e $I_1 = 1$.

La ecuación 1 es evaluada en cada neurona difusa S_i para calcular el valor del estado siguiente en el tiempo $t + 1$.

El valor inicial, antes de que cualquier entrada sea vista es $S_0^0 = 1$ y $S_j^0 = 0$ para todo $j \neq 0$.

2.1 Procedimiento de entrenamiento.

El objetivo de entrenar la red es ajustar los pesos para que el conjunto de ejemplos de entrada produzcan el conjunto de salidas deseadas. El error cometido por cada neurona difusa es calculado minimizando el cuadrado de la diferencia entre la salida deseada T_i , y la salida obtenida por la red S_i^f para cada uno de los patrones utilizados en el aprendizaje. S_i^f será, por tanto, el valor de cada una de las neuronas una vez finalizada la presentación de una secuencia de ejemplo. El error lo calcularemos atendiendo a la siguiente expresión:

$$E_i = \frac{1}{2} (T_i - S_i^f)^2 \quad (2)$$

donde

$$S_i^f = \max_{j,k} \min(W_{ijk}, S_j^{f-1}, I_k^{f-1})$$

siendo T_i el grado de pertenencia de la secuencia al estado i del autómata $\mathcal{A}$ y S_i^f el grado de pertenencia al estado i que la red da como resultado.

Para ajustar los pesos utilizamos un algoritmo de entrenamiento que actualiza los pesos al final de la presentación de cada secuencia de ejemplo. Actualizaremos los pesos si el error es mayor que ϵ , que será elegido dependiendo de la precisión deseada en los resultados. As, la regla de actualización de los pesos será:

$$\nabla W_{ilm} = -\alpha \frac{\partial E_i}{\partial W_{ilm}} = \alpha (T_i - S_i^f) \cdot \frac{\partial S_i^f}{\partial W_{ilm}} \quad (3)$$

$$\frac{\partial S_i^f}{\partial W_{ilm}} = \frac{\partial \max_{j,k} \min(W_{ijk}, S_j^{f-1}, I_k^{f-1})}{\partial \min(W_{ilm}, S_m^{f-1}, I_n^{f-1})} \quad (4)$$

$$\frac{\partial \min(W_{ilm}, S_m^{f-1}, I_n^{f-1})}{\partial W_{ilm}} \quad (5)$$

Llamaremos P1 a:

$$\frac{\partial \max_{j,k} \min(W_{ijk}, S_j^{f-1}, I_k^{f-1})}{\partial \min(W_{imn}, S_m^{f-1}, I_n^{f-1})} = \quad (6)$$

$$\frac{\partial \max_{\substack{j \neq m \\ k \neq n}} [\min(W_{imn}, S_m^{f-1}, I_n^{f-1}), \min(W_{ijk}, S_j^{f-1}, I_k^{f-1})]}{\min(W_{imn}, S_m^{f-1}, I_n^{f-1})} \quad (7)$$

y P2:

$$P2 = \frac{\partial \min(W_{imn}, S_m^{f-1}, I_n^{f-1})}{\partial W_{imn}} \quad (8)$$

Las funciones $\min(y, p)$ y $\max(y, p)$ son derivables en los intervalos abiertos $y < p$ y $y > p$, sin embargo, sus derivadas no están definidas en $y = p$:

$$\frac{\partial \min(y, p)}{\partial y} = \begin{cases} 1 & \text{si } y < p \\ 0 & \text{si } y > p \end{cases}; \quad (9)$$

$$\frac{\partial \max(y, p)}{\partial y} = \begin{cases} 1 & \text{si } y > p \\ 0 & \text{si } y < p \end{cases} \quad (10)$$

Como es usual[7], asignaremos en cualquier caso en $y = p$ el valor a la derecha o a la izquierda de la derivada respectivamente:

$$\frac{\partial \min(y, p)}{\partial y} = \begin{cases} 1 & \text{si } y \leq p \\ 0 & \text{si } y > p \end{cases}; \quad (11)$$

$$\frac{\partial \max(y, p)}{\partial y} = \begin{cases} 1 & \text{si } y \geq p \\ 0 & \text{si } y < p \end{cases} \quad (12)$$

Sin embargo hemos cambiado este comportamiento "crisp" por uno "difuso" capaz de capturar el significado real de $(y \leq p)$ ó $(y \geq p)$ en un contexto vago. Utilizaremos una medida del grado de inclusión de p en y , dada por la implicación de Gödel. De esta manera nos queda:

$$\frac{\partial \min(y, p)}{\partial y} = \begin{cases} 1 & \text{si } y \leq p \\ y & \text{si } y > p \end{cases}; \quad (13)$$

$$\frac{\partial \max(y, p)}{\partial y} = \begin{cases} 1 & \text{si } y \geq p \\ p & \text{si } y < p \end{cases} \quad (14)$$

Teniendo en cuenta las ecuaciones anteriores obtenemos los siguientes resultados:

$$P1 = \begin{cases} 1 & \text{si } \min(W_{imn}, S_m^{f-1}, I_n^{f-1}) \geq \max_{\substack{j \neq m \\ k \neq n}} [\min(W_{ijk}, S_j^{f-1}, I_k^{f-1})] \\ \min(W_{imn}, S_m^{f-1}, I_n^{f-1}) & \text{si } \min(W_{imn}, S_m^{f-1}, I_n^{f-1}) < \max_{\substack{j \neq m \\ k \neq n}} [\min(W_{ijk}, S_j^{f-1}, I_k^{f-1})] \end{cases} \quad (15)$$

$$P2 = \begin{cases} 1 & \text{si } W_{imn} \leq \{S_m^{f-1}, I_n^{f-1}\} \\ S_m^{f-1} & \text{si } S_m^{f-1} \leq \{W_{imn}, I_n^{f-1}\} \\ I_n^{f-1} & \text{si } I_n^{f-1} \leq \{W_{imn}, S_m^{f-1}\} \end{cases} \quad (16)$$

Combinando los valores de P1 y P2:

$$\frac{\partial S_i^f}{\partial W_{imn}} = \begin{cases} \text{si } MIN = W_{imn} \\ \left\{ \begin{array}{l} W_{imn} \geq MAX \rightarrow C1 \\ W_{imn} < MAX \rightarrow C2 \end{array} \right\} \\ \text{si } MIN = S_m^{f-1} \\ \left\{ \begin{array}{l} S_m^{f-1} \geq MAX \rightarrow C3 \\ S_m^{f-1} < MAX \rightarrow C4 \end{array} \right\} \\ \text{si } MIN = I_n^{f-1} \\ \left\{ \begin{array}{l} I_n^{f-1} \geq MAX \rightarrow C5 \\ I_n^{f-1} < MAX \rightarrow C6 \end{array} \right\} \end{cases} \quad (17)$$

$$MAX = \max_{\substack{j \neq m \\ k \neq n}} [\min(W_{ijk}, S_j^{f-1}, I_k^{f-1})] \quad (18)$$

$$MIN = \min(W_{imn}, S_m^{f-1}, I_n^{f-1}) \quad (19)$$

As, los resultados obtenidos son:

$$C1 = 1; C2 = W_{imn}; C3 = S_m^{f-1}; C4 = S_m^{f-1} * S_m^{f-1}; C5 = I_n^{f-1}; \quad (20)$$

$$C6 = I_n^{f-1} * I_n^{f-1} \quad (21)$$

Una vez calculado este incremento utilizamos un término momentum, teniendo en cuenta en el cálculo del incremento actual el incremento inmediatamente anterior:

$$\nabla W_{ijk} = \mu \cdot \nabla W_{ijk}^{old} + (1 - \mu) \cdot \nabla W_{ijk} \quad (22)$$

Generalmente hemos tomado $\mu = 0.5$.

2.2 Presentación de los ejemplos de entrenamiento.

Inicialmente los pesos son elegidos aleatoriamente en el intervalo $[0, 1]$. Los datos de entrenamiento, como

hicimos referencia anteriormente, consisten en pares (α, β) , donde α es una secuencia de entrada $x_1 x_2 \dots x_l$, $x_i \in \{0, 1\}$ y $\beta = (\mu_1(x), \mu_2(x), \dots, \mu_N(x))$, siendo N el número de estados del autómata y el número de neuronas difusas de la red neuronal. El valor inicial para las neuronas difusas S_i^0 es cero excepto para la neurona S_0^0 que será igual a 1. La red comienza el entrenamiento con sólo 20 ejemplos. La porción restante de ejemplos de entrenamiento es separada en otro conjunto que pasará a estudiar la red una vez que haya aprendido los 20 primeros ejemplos o realice un número determinado de épocas, entendiendo por épocas el período que consume la red en procesar sólo una vez todas las secuencias de entrenamiento. Cuando la red ha aprendido su conjunto de patrones de entrenamiento o ha superado el número máximo de épocas, pasa a estudiar el conjunto restante de ejemplos que todavía no ha visto, actuando de la siguiente manera: si una secuencia de este conjunto no es reconocida ($E > \epsilon$) es incluida en el conjunto de entrenamiento y, en caso contrario, pasa a procesar otra. El número de secuencias a introducir es limitado a 10 con el objeto de que la red no entre en sobreespecialización. Al conjunto de acciones de procesar las secuencias del conjunto de entrenamiento y posteriormente estudiar el conjunto de secuencias que no ha visto e incluirlas si no las reconoce se le denomina ciclo. Usualmente utilizamos como máximo 20 ciclos para entrenar la red. Siempre que una secuencia del conjunto de entrenamiento sea procesada y su error sea mayor que epasaremos a ajustar los pesos mediante las ecuaciones dadas anteriormente.

2.3 Extracción del Autómata Difuso.

Una vez que la red ha aprendido los ejemplos de entrenamiento, el autómata difuso aprendido puede ser extraído estudiando los pesos de conexión de las neuronas recurrentes difusas, es decir, el peso W_{ijk} representa el grado de pertenencia de la transición del estado j al estado i del autómata, habiendo leído el símbolo de entrada k . Por tanto, podemos construir las relaciones difusas $S[0]$ y $S[1]$ en función de la entrada k , $k \in \{0, 1\}$:

$$S[0] = \begin{pmatrix} W_{000} & W_{100} & \dots & W_{N00} \\ W_{010} & W_{110} & \dots & W_{N10} \\ \vdots & \vdots & \ddots & \vdots \\ W_{0N0} & W_{1N0} & \dots & W_{NN0} \end{pmatrix} \quad (23)$$

$$S[1] = \begin{pmatrix} W_{001} & W_{101} & \dots & W_{N01} \\ W_{011} & W_{111} & \dots & W_{N11} \\ \vdots & \vdots & \ddots & \vdots \\ W_{0N1} & W_{1N1} & \dots & W_{NN1} \end{pmatrix} \quad (24)$$

3 Simulación.

Hemos escogido como ejemplo el autómata cuyo funcionamiento queda expresado por las siguientes relaciones difusas:

$$S[0] = \begin{pmatrix} 0.8 & 0.2 & 0.1 & 0.2 & 0.1 \\ 0.3 & 0.1 & 0.9 & 0.2 & 0.1 \\ 0 & 0.1 & 0.2 & 0.85 & 0.1 \\ 0.1 & 0 & 0.9 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0.2 & 0.1 & 0.7 \end{pmatrix}$$

$$S[1] = \begin{pmatrix} 0.1 & 0.9 & 0.2 & 0.1 & 0.2 \\ 0.8 & 0 & 0.1 & 0.2 & 0 \\ 0.2 & 0.1 & 0 & 0.1 & 0.9 \\ 0.1 & 0.8 & 0.1 & 0 & 0.1 \\ 0.1 & 0.1 & 0 & 0.2 & 0.8 \end{pmatrix}$$

Según queda reflejado en las matrices, el autómata tiene 5 estados, de los cuales los estados 0, 1 y 3 son finales. El estado inicial será el estado 0, y lo representaremos mediante $I_0 = (1 \ 0 \ 0 \ 0 \ 0)$. La red utilizada para aprender este autómata ha sido una red neuronal con 5 neuronas difusas recurrentes y dos neuronas de entrada. Los parámetros utilizados para el aprendizaje han sido una razón de aprendizaje, $\alpha = 0.2$, una tolerancia de respuesta, $\epsilon = 0.0005$ y un término momentum, $\mu = 0.5$. En esta simulación fueron necesarios 3 ciclos de 500 épocas para obtener unos resultados de aprendizaje del 100% de los ejemplos. Los pesos obtenidos una vez entrada fueron:

$$\begin{aligned} W_{000} &= 0.8000 & W_{001} &= 0.1000 & W_{010} &= 0.3000 \\ W_{011} &= 0.7999 & W_{020} &= 0.0019 & W_{021} &= 0.2000 \\ W_{030} &= 0.2002 & W_{031} &= 0.0126 & W_{040} &= 0.0007 \\ W_{041} &= 0.0323 & W_{100} &= 0.2000 & W_{101} &= 0.9000 \\ W_{110} &= 0.0887 & W_{111} &= 0.0198 & W_{120} &= 0.0609 \\ W_{121} &= 0.0999 & W_{130} &= 0.1080 & W_{131} &= 0.8695 \\ W_{140} &= 0.0886 & W_{141} &= 0.0999 & W_{200} &= 0.1000 \\ W_{201} &= 0.2000 & W_{210} &= 0.9014 & W_{211} &= 0.0000 \\ W_{220} &= 0.2000 & W_{221} &= 0.0000 & W_{230} &= 0.2000 \\ W_{231} &= 0.0001 & W_{240} &= 0.1464 & W_{241} &= 0.0000 \\ W_{300} &= 0.1210 & W_{301} &= 0.1260 & W_{310} &= 0.1197 \\ W_{311} &= 0.1696 & W_{320} &= 0.1214 & W_{321} &= 0.2026 \\ W_{330} &= 0.2516 & W_{331} &= 0.2374 & W_{340} &= 0.1259 \\ W_{341} &= 0.1998 & W_{400} &= 0.0000 & W_{401} &= 0.1999 \\ W_{410} &= 0.0005 & W_{411} &= 0.0000 & W_{420} &= 0.0000 \\ W_{421} &= 0.8744 & W_{430} &= 0.2512 & W_{431} &= 0.0000 \\ W_{440} &= 0.6782 & W_{441} &= 0.8281 \end{aligned}$$

Así por ejemplo, el significado del peso W_{231} es que el autómata tiene una transición con grado de pertenencia 0.0001 del estado 3 al estado 2 leyendo el símbolo de entrada 1. Con lo cual podemos extraer el autómata difuso asociado a los ejemplos de entrenamiento construyendo las matrices difusas con la semántica del

ejemplo anterior. De esta forma se obtienen las siguientes relaciones difusas:

$$S[0] = \begin{pmatrix} 0.8000 & 0.2000 & 0.1000 & 0.1210 & 0.0000 \\ 0.3000 & 0.0887 & 0.9014 & 0.1197 & 0.0005 \\ 0.0019 & 0.0609 & 0.2000 & 0.1214 & 0.0000 \\ 0.2002 & 0.1080 & 0.2000 & 0.2516 & 0.2512 \\ 0.0007 & 0.0886 & 0.1464 & 0.1259 & 0.6782 \end{pmatrix}$$

$$S[1] = \begin{pmatrix} 0.1000 & 0.9000 & 0.2000 & 0.1260 & 0.1999 \\ 0.7999 & 0.0198 & 0.0000 & 0.1696 & 0.0000 \\ 0.2000 & 0.0999 & 0.0000 & 0.2026 & 0.8744 \\ 0.0126 & 0.8695 & 0.0001 & 0.2374 & 0.0000 \\ 0.0323 & 0.0999 & 0.0000 & 0.1998 & 0.8281 \end{pmatrix}$$

4 Conclusiones.

Hemos propuesto una red neuronal recurrente difusa capaz de inferir el autómata asociado a partir de secuencias de ejemplo de dicho lenguaje difuso. Simulamos el comportamiento de la red en la fase de aprendizaje para un ejemplo concreto, mostrando que la red aprende a inferir el autómata asociado a los ejemplos de entrenamiento, calculando en sus pesos las relaciones difusas que caracterizan el comportamiento del autómata a aprender.

Referencias

- [1] Blanco A., Delgado M., Requena I., Identification of fuzzy relational equations by fuzzy neural networks, *Fuzzy Sets And Systems*, Vol.71, p.215-226, (1995).
- [2] Cleeremans, A., Servan-Schreiber, D., and McClelland, J. 1989. Finite state automata and simple recurrent networks. *Neural Computation*. 1(3), 372-381.
- [3] Fu, K. S., *Syntactic Pattern Recognition and Applications*, Prentice Hall, Englewood Cliffs, N.J. (1982).
- [4] Giles, C.L., and Omlin, C.W. 1992. Inserting rules into recurrent neural network Proc 1992 IEEE Workshop Neural Networks Signal Process, pp. 13-22. Copenhagen, Denmark.
- [5] Giles, C.L., Miller, C.B., Chen, D., Chen, H.H., Sun, G.Z., and Lee, Y.C. 1992. Learning and extracting finite state automata with second-order recurrent neural networks. *Neural Computation* 4, 393-405.
- [6] Harrison, M.H., *Introduction to Formal Language Theory*, Addison-Wesley, Reading, Mass., (1978).
- [7] Ikoma N., Pedrycz W. and Hirota K. 1993. Estimation of fuzzy relational matrix by using probabilistic descent method. *Fuzzy Set Systems* 57, 335-349.
- [8] Watrous, R.L., and Kuhn, G.M. 1992. Induction of finite-state languages using second-order recurrent networks. *Neural Computation* 4, 406-414.
- [9] Zeng, Z., Goodman, R., and Smyth, P. 1993. Learning finite state machines with self-clustering recurrent networks. *Neural Computation* 5(6), 976-990.
- [10] Zeng, Z., Goodman, R., and Smyth, P., 1994. Discrete recurrent neural networks for grammatical inference. *IEEE Transact. Neural Networks* 5(2), 320-330.

OBTENCIÓN DE SISTEMAS BASADOS EN REGLAS CON VALORES DE VERDAD MEDIANTE REDES NEURONALES*

A. Blanco

Dpto. Ciencias de la Computación
e Inteligencia Artificial
E.T.S de Ingeniería Informática
18071 Granada España
armando@decsai.ugr.es

W. Fajardo

Dpto. Ciencias de la Computación
e Inteligencia Artificial
E.T.S de Ingeniería Informática
18071 Granada España
aragorn@decsai.ugr.es

M. C. Pegalajar

Dpto. Ciencias de la Computación
e Inteligencia Artificial
E.T.S de Ingeniería Informática
18071 Granada España
mcarmen@decsai.ugr.es

Resumen

En el presente trabajo, presentamos la posibilidad de identificar un sistema real, mediante un sistema basado en reglas a las que se les asocian valores de verdad, con la posibilidad de realizar posteriormente inferencia difusa utilizando dicho sistema; hemos utilizado una red neuronal con una capa oculta, que puede ser entrenada con valores crisp obtenidos por cualquier sistema.

Palabras Clave: Sistemas basados en reglas, Valores de verdad, Redes Neuronales.

primitivos, construyendo a continuación el conjunto de valores de verdad formado por los valores: $\{true, fairly\ true, neutral, fairly\ false, false\}$, en el que "neutral" puede ser considerado como "neither true nor false", pudiendo incrementar el número de valores aplicando las conectivas definidas en QNL, con la condición de que los valores de verdad sean parcialmente ordenados. El significado cuantitativo de los valores de verdad se asume que son números difusos que normalizados se representan por los números difusos de la figura 1

1 INTRODUCCIÓN.

Es conocido que la aproximación lingüística permite más versatilidad y aplicabilidad a los sistemas reales cuando se plantea la identificación mediante sistemas basados en reglas (S.B.R.), como puede verse en ([5], [6], [8] y [9]) sin embargo desde el punto de vista semántico aparecen varios problemas, como pueden ser: la negación y la antonimia de un término que a veces pueden confundirse; expresiones demasiado complejas como *very very very old* pueden llevar a confusión; la interpretación de cabeceras lingüísticas que pueden ser inapropiadas en determinados casos como *very true*. Para resolver estos problemas fue formulado un lenguaje Cuasi Natural (QNL) por [3]

La verdad de una expresión vaga ha sido un concepto problemático en la lógica convencional, ya que la lógica bivalente sólo es apropiada en sistemas precisos.

Niskanen en [2] justifica una aproximación a la métrica de verdad, que llama Lógica de la métrica difusa (FML), y la inclusión en el conjunto de valores de verdad "true" y "false" como términos

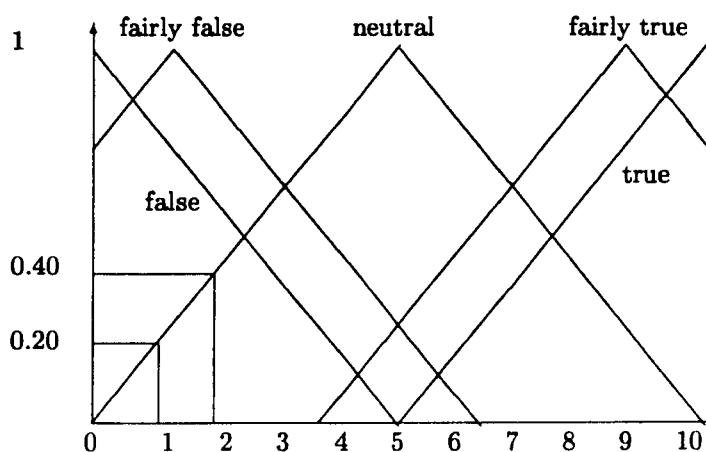


Fig. 1

Normalmente la adquisición del conocimiento se ha realizado a través de expertos, pero actualmente se está utilizando ampliamente la adquisición automática del conocimiento aplicando técnicas para identificar sistemas difusos. Entre otras técnicas, se encuentra la identificación mediante S.B.R.. En el mundo real, los S.B.R. no tienen que ser necesariamente crisp, sino que por el contrario algunas reglas que componen el sistema, pueden contener intrínsecamente una cierta imprecisión, que normalmente no es considerada computacionalmente. En definitiva, en un S.B.R. no todas

las reglas tienen que ser absolutamente verdaderas para que identifiquen el sistema original; por consiguiente se plantea el problema de asignar valores de verdad en la identificación de un sistema original mediante un sistema basado en reglas.

Para conseguir este objetivo, utilizamos como herramienta una red neuronal con una capa oculta, donde las entradas se obtienen a partir de los datos crisp proporcionados por cualquier sistema, y las salidas son los valores de verdad considerados.

2 FORMULACIÓN DEL PROBLEMA.

Consideremos el sistema original que queremos identificar, caracterizado por un conjunto de entradas-salidas relacionadas mediante la función continua

$v = f(u)$, $u \in U \subseteq R^n$, $v \in V \subseteq R^m$, la función f es desconocida "a priori", sin embargo disponemos de un conjunto de ejemplos crisp (u_i, v_i) , $i = 1, 2, \dots, N$. En lo que sigue, nos limitaremos al caso más simple $n = m = 1$ para que se entienda mejor la resolución del problema, pero puede ser fácilmente ampliable.

Consideremos H_U y H_V las variables lingüísticas definidas sobre los referenciales U y V que identifican a los valores de entrada y salida del sistema; sean $\{h_{u1}, h_{u2}, \dots, h_{us}\}$ and $\{h_{v1}, h_{v2}, \dots, h_{vt}\}$ el conjunto de etiquetas lingüísticas de dichas variables que vandrán dadas cada una por un subconjunto difuso definido sobre los referenciales correspondientes U y V

Las posibles reglas que pueden identificar el sistema original serán de la forma:

R_{ij} : IF H_U is h_{ui} THEN H_V is h_{vj} $i = 1, 2, \dots, s$ $j = 1, 2, \dots, t$

todas las reglas posibles se obtienen a partir del producto cartesiano $\mathcal{R} = \{\{h_{u1}, h_{u2}, \dots, h_{us}\} \times \{h_{v1}, h_{v2}, \dots, h_{vt}\}\}$

Los conjuntos de etiquetas h_{ui} (valores de H_U en las reglas) y h_{vj} (valores de H_V en las reglas) son obtenidos de los conjuntos $\{h_{u1}, h_{u2}, \dots, h_{us}\}$, $\{h_{v1}, h_{v2}, \dots, h_{vt}\}$ donde $s * t$ es el número de todas las posibles reglas obtenidas mediante el producto Cartesiano $L = \{\{h_{u1}, h_{u2}, \dots, h_{us}\} \times \{h_{v1}, h_{v2}, \dots, h_{vt}\}\}$. La semántica de h_{ui} y h_{vj} será dada por conjuntos difusos de R^n y R^m respectivamente.

Consideramos

T , una variable lingüística definida sobre un referencial $E \subseteq R^p$ y que dicha variable lingüística puede tomar valores en el conjunto de etiquetas $\{t_1, t_2, \dots, t_K\}$, en particular el conjunto que consideraremos será $\{true, fairly\ true, neutral, fairly\ false, false\}$

Los t_{ij} representan posibles valores que toma la variable lingüística T definida sobre el referencial E , es decir, elementos del conjunto de valores de verdad, representados por conjuntos difusos de R^p .

En [1] presentamos un procedimiento para asociar a cada posible regla, un grado de incertidumbre, sin embargo parece más razonable asociar un grado de verdad a cada regla.

Nuestro objetivo es obtener el valor de verdad de cada una de las posibles reglas R_{ij} que pueden identificar el sistema continuo representado por los ejemplos proporcionados. A partir de la obtención de los valores de verdad de cada regla, podemos elegir un umbral a partir del cual se seleccionaran las reglas con sus respectivos valores de verdad, que serán de la forma:

R_{11} : If H_U is h_{u1} then H_V is h_{v1} con valor de verdad t_{11}

R_{12} : If H_U is h_{u1} then H_V is h_{v2} con valor de verdad t_{12}

:

R_{ij} : If H_U is h_{ui} then H_V is h_{vj} con valor de verdad t_{ij}

2.1 IDENTIFICACIÓN DEL SISTEMA.

Para identificar el sistema se seguirán los siguientes pasos:

- Discretizamos los conjuntos referenciales U , V y E , obteniendo $U = \{u_1, u_2, \dots, u_n\}$, $V = \{v_1, v_2, \dots, v_m\}$ y $E = \{l_1, l_2, \dots, l_p\}$
- Suponemos que las variables H_U y H_V pueden tomar valores en los conjuntos $\{h_{u1}, h_{u2}, \dots, h_{us}\}$, $\{h_{v1}, h_{v2}, \dots, h_{vt}\}$ de etiquetas lingüísticas.
- Introducimos la variable T (valores de verdad de una regla) que toma valores en el conjunto $\{true, fairly\ true, neutral, fairly\ false, false\}$, conjuntos difusos que discretizados toman valores en el intervalo $[0, 1]^p$.
- A cada regla crisp, le asociamos el valor de la variable T *true*
- Definimos el referencial $W = U \times V$ mediante los pares ordenados proporcionados (u_i, v_r) para la representación del sistema. $l = \{1, 2, \dots, n\}$, $r = \{1, 2, \dots, m\}$
- Sobre W establecemos el conjunto $\mathcal{R} = \{h_{u1} \times h_{v1}, h_{u1} \times h_{v2}, \dots, h_{us} \times h_{vt}\}$ producto Cartesiano a partir de las etiquetas lingüísticas.

- Sobre el nuevo referencial W se define la variable $H_{U \times V}$ que toma valores en el conjunto $\mathcal{R}$.

A continuación definimos un nuevo sistema, que llamaremos Intermedio, que estará definido en

$W \rightarrow [0, 1]^p$ mediante el conjunto de reglas

R_1 : If $H_{U \times V}$ is (u_1, v_1) then T is true

R_2 : If $H_{U \times V}$ is (u_2, v_2) then T is true

⋮

R_N : If $H_{U \times V}$ is (u_N, v_N) then T is true

Nuestro objetivo es identificar el sistema Intermedio mediante un sistema basado en reglas de la forma:

R_{ij} : If $H_{U \times V}$ is (h_{ui}, h_{vj}) then T is t_{ij} donde

$t_{ij} \in \{true, fairly true, neutral, fairly false, false\}$,

Una vez que este sistema es obtenido, tendremos todas las posibles reglas que pueden identificar el sistema con el valor de verdad correspondiente a cada regla.

Mediante el proceso de discretización de los conjuntos difusos asociados a los valores de las etiquetas lingüísticas y a los valores de verdad, se traslada el problema, a un recubrimiento de $[0, 1]^{n+m}$ en $[0, 1]^p$ por lo que podemos utilizar una red neuronal feedforward con una capa oculta y con función de activación $f(x) = \frac{1}{1+e^{-x}}$ en la capa oculta y en la capa de salida, ya que el sistema se considera continuo. La red neuronal constará de $n+m$ neuronas en la capa de entrada y p neuronas en la capa de salida

Para el entrenamiento de la red escogemos un método apropiado, como el algoritmo de backpropation en [4]

3 RED NEURONAL

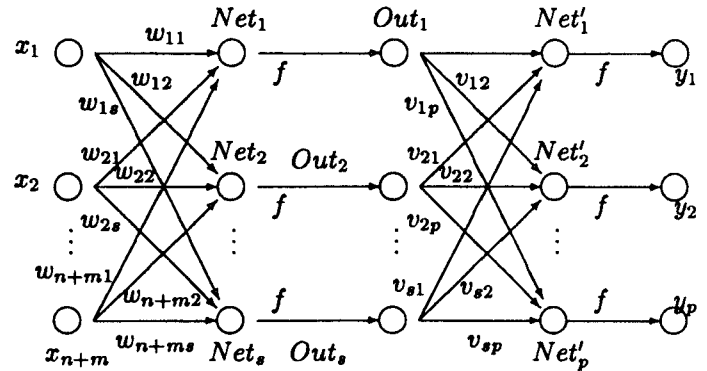
3.1 TOPOLOGÍA DE LA RED.

El modelo de red elegido para el proceso de aprendizaje debe tener en cuenta la discretización de los subconjuntos correspondientes a los ejemplos que se proporcionen, de forma que si cada elemento de entrada es representado por un elemento del conjunto $[0, 1]^n$ y cada elemento de salida es representado por un elemento de $[0, 1]^m$, la red neuronal tendrá $n+m$ neuronas en la capa de entrada.

La capa de salida estará formada por un conjunto de p neuronas correspondientes a la discretización de los posibles valores de verdad, es decir de sus respectivos conjuntos difusos, en el caso de que cada conjunto difuso se represente por un elemento del conjunto $[0, 1]^p$

De esta forma, a cada vector de $n+m$ componentes, es decir a cada regla se le asociará un vector de p componentes, un valor de verdad.

La red neuronal, es una red feedforward con una capa oculta y para su entrenamiento utilizamos la función sigmoide $f(x) = \frac{1}{1+e^{-x}}$



3.2 MODELOS DE ENTRENAMIENTO.

Los modelos de entrenamiento se obtienen a partir del siguiente conjunto de pares ordenados:

I- Un número finito de pares ordenados $((u_i, v_i), true)$, que son las reglas crisp que representan al sistema.

II- Debido a la continuidad del sistema Intermedio, ya que el sistema Inicial lo es, disponemos de pares ordenados de la forma $((u_i, v_i \pm \epsilon), true)$, donde ϵ es un número pequeño.

III- También por la continuidad, escogemos pares ordenados de la forma $((u_i, v_i \pm \rho), fairly true)$ donde ρ es un número real no muy pequeño.

IV- Ya que queremos construir un sistema difuso, también podemos proporcionar modelos en los que se asocie a una observación de U , un intervalo de V , obteniendo pares mediante la fusificación $((u_i, [v_i - \epsilon, v_i + \epsilon]), true)$

V- A partir de los pares del tipo I, obtenemos los pares de entrenamiento negativos $((u_i, v_i \pm \beta), false)$, donde β es escogido de forma que $v_i \pm \beta$ esté alejado de v_i .

VI- A partir de la fusificación de los pares negativos, obtenemos pares de la forma $(u_i, [v_i - \beta - \epsilon, v_i - \beta + \epsilon], false)$ y $(u_i, [v_i + \beta - \epsilon, v_i + \beta + \epsilon], false)$.

En los modelos de entrenamiento, las entradas son vectores $n+m$ dimensionales escogidos de entre los pares ordenados señalados anteriormente, en los que sus componentes son todas nulas, salvo las situadas en la posición dada por los pares ordenados, a las que se les asigna 1. Las salidas son los conjuntos difusos discretizados asociados a las etiquetas lingüísticas, como se muestra en la tabla 1.

De esta forma construimos un sistema donde las entradas son reglas y las salidas son valores de verdad.

Podemos generalizar el proceso anterior para el caso en que los universos del discurso U y V sean producto de un número finito de universos, donde exista más de una variable.

4 EJEMPLO.

Consideremos un sistema que esté definido por la relación $X = Y$ previamente desconocida, definida sobre los universos del discurso $U = \{1, 2, \dots, 11\}$ $V = \{1, 2, \dots, 11\}$, queremos identificar el sistema por un S.B.R. con valores de verdad asociados a cada una de las reglas, las variables lingüísticas del sistema son $H_U = H_V$, y el conjunto de etiquetas que pueden tomar, se encuentra restringido al conjunto Large, Medium and Small

Del sistema que queremos identificar sólo conocemos las entradas y salidas crisp (1,1), (2,2)... (11,11).

4.1 ENTRENAMIENTO DE LA RED.

Cada etiqueta tiene asociada una función de pertenencia y mediante el proceso de discretización se obtiene el vector correspondiente

Small $\rightarrow$ (1.00, 0.81, 0.60, 0.41, 0.19, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00)

Medium $\rightarrow$ (0.01, 0.20, 0.39, 0.60, 0.80, 1.00, 0.80, 0.60, 0.41, 0.20, 0.00)

Large $\rightarrow$ (0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.21, 0.40, 0.61, 0.80, 1.00)

Podemos obtener los siguientes pares ordenados, a partir de los tipos de pares expresados anteriormente

I.- (1, 1), (2, 2) ... (11, 11).

II.- (1, 2), (2, 3) ... (10, 11), (2, 1), (3, 2) ... (11, 10).

III.- (2, (1, 2, 3)), (3, (2, 3, 4)) ... (10, (9, 10, 11)).

IV.- (1, 5), (2, 6) ... (11, 1), (1, 9), (2, 10) ... (11, 3).

V.- (1, (4, 5, 6)), (2, (5, 6, 7)) ... (1, (9, 10, 11))...

VI.- (1, 3), (2, 4) ... (8, 10), ... (3, 1), ... (10, 8).

El número de modelos a entrenar, depende de la precisión que queramos obtener.

A continuación expresamos un ejemplo de la codificación de algunos modelos.

(3,4) $\rightarrow$ (0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0)

(3, (2, 3, 4)) $\rightarrow$ (0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

0, 0, 0, 0, 0, 0, 0)

(3, (6, 7, 8)) $\rightarrow$ (0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)

(2, 4) $\rightarrow$ (0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)

Las salidas de la última capa, para los modelos anteriores son los vectores obtenidos al discretizar las etiquetas *true*, *true*, *false* y *fairly false* respectivamente.

Después del entrenamiento, la obtención de los valores de verdad asociados a las posibles reglas se obtienen de la siguiente forma

Small $\rightarrow$ Small

Input: (1.00, 0.81, 0.60, 0.41, 0.19, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 0.81, 0.60, 0.41, 0.19, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00)

Output: (0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.03, 0.19, 0.38, 0.65, 0.82, 0.96) *true*

Small $\rightarrow$ Medium

Input: (1.00, 0.81, 0.60, 0.41, 0.19, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00, 0.01, 0.20, 0.39, 0.60, 0.80, 1.00, 0.80, 0.60, 0.41, 0.20, 0.00)

Output: (0.02, 0.19, 0.39, 0.60, 0.82, 0.97, 0.81, 0.59, 0.40, 0.22, 0.03) *neutral*

Small $\rightarrow$ Large

Input: (1.00, 0.81, 0.60, 0.41, 0.19, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.21, 0.40, 0.61, 0.80, 1.00)

Output: (0.97, 0.81, 0.60, 0.39, 0.18, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00) *false*

5 CONCLUSIONES.

Hemos desarrollado una metodología para identificar sistemas con entradas-salidas crisp, mediante sistemas basados en reglas, en los que cada regla tiene asociado un valor de verdad. La extensión a problemas más complejos, se puede realizar directamente aplicando el mismo método

Tabla 1: Conjuntos difusos asociados a las etiquetas lingüísticas.

Universo	true	fairly true	neutral	fairly false	false
0	0.00	0.00	0.00	0.78	1.00
1	0.00	0.00	0.20	0.88	0.80
2	0.00	0.00	0.40	0.68	0.60
3	0.00	0.00	0.60	0.48	0.40
4	0.00	0.01	0.80	0.28	0.20
5	0.00	1.08	1.00	0.08	0.00
6	0.20	0.28	0.80	0.01	0.00
7	0.40	0.48	0.60	0.00	0.00
8	0.60	0.68	0.40	0.00	0.00
9	0.80	0.88	0.20	0.00	0.00
10	1.00	0.78	0.00	0.00	0.00

Referencias

- [1] Blanco A., Delgado M., Requena I. (1995). A learning procedure to identify weighted rules by networks. *Fuzzy Sets and Systems*. 69, 29-36
- [2] Niskanen V. A. (1993) Metric truth as a basis for fuzzy linguistic reasoning. *Fuzzy Sets and Systems*. 57, 1-25
- [3] Niskanen V. A. (1986) A fuzzy system with linguistic values: Some logical and methodological considerations and applications. *Research Reports 16. University of Helsinki, Computing Centre*
- [4] Rumelhart, D., Hinton, G. and Williams, R. (1986). Learning internal representations by error propagation *Parallel distributed processing* Vol. 1, pp. 318-362. Cambridge, MA: MIT Press.
- [5] R. Bellman y L.A. Zadeh, Decision Making in a Fuzzy Environment. *Management Science*, 17, (1970), 141-164
- [6] M. Tong y P. Bonissone, Linguistic Solutions to Fuzzy Decision Problems. *TIMS/Studies in the Management Sciences*, 20, 323-334, (1984)984
- [7] Zadeh L. (1975) *Fuzzy logic and approximate reasoning*. *Synthese* 30, 407-428
- [8] L.A. Zadeh, The Concept of a Linguistic Variable and its Applications to Approximate Reasoning, Part I, *Information Sciences*, vol. 8, 199-249, Part II, *Information Sciences*, vol. 8, 301-357, Part III, *Information Sciences*, vol.9, 43-80, (1975). 5).
- [9] L.A. Zadeh, Fuzzy Sets and Information Granularity. En (M.M. Gupta et al. eds.), *Advances in Fuzzy Sets Theory and Applications*, North Holland Publishing Company, Nueva York, (1979), 3-18

Hacia una semántica de la defuzzyficación*

D. Boixader
Sec. Matemàtiques i Informàtica.
E.T.S. Arquitectura del Vallés
Univ. Politècnica de Catalunya
e-mail: boixader@ea.upc.es

J. Jacas
Sec. Matemàtiques i Informàtica.
E.T.S. Arquitectura de Barcelona
Univ. Politècnica de Catalunya
e-mail: jacas@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

La defuzzyficación es un problema esencial en los sistemas borrosos que se resuelve siempre de forma puramente heurística. Este trabajo pretende dar interpretación semántica a este proceso mediante el uso de relaciones de indistinguibilidad.

Palabras clave: Defuzzyficación, control borroso, razonamiento aproximado, relación de indistinguibilidad, indistinguibilidad natural.

1 INTRODUCCIÓN

La defuzzyficación es un problema esencial en los sistemas borrosos que se resuelve siempre de forma puramente heurística.

Los sistemas borrosos contienen una primera fase que consiste en la fuzzyficación de los datos. Esta fase es necesaria por diversos motivos como, por ejemplo, vaguedad lingüística en los *inputs* o la presencia de una indistinguibilidad en el conjunto de los mismos. Partiendo de un conjunto clásico, se genera un conjunto borroso que lo describa teniendo en cuenta la borrosidad del sistema. El proceso de defuzzyficación puede pensarse como el problema inverso: Dado un *output* en forma de subconjunto borroso, buscar el conjunto clásico que mejor lo describa al eliminar la borrosidad del sistema. El resultado del proceso de defuzzyficación es, en este contexto, un conjunto. Si, por necesidades del sistema, se necesita un único valor como *output*, se deberá usar otro algoritmo, fuera ya del sistema borroso, para elegirlo del conjunto obtenido en la defuzzyficación.

En este trabajo se proponen dos interpretaciones semánticas de la defuzzyficación (dependiendo de la naturaleza del sistema) que dan lugar a sendas técnicas para ella. La primera (sección 4) da por supuesta

una indistinguibilidad inherente en el conjunto de los *outputs* del sistema. La segunda (sección 5) utiliza el concepto de indistinguibilidad natural entre subconjuntos borrosos [2,3].

2 RELACIONES DE INDISTINGUIBILIDAD Y CLASES DE EQUIVALENCIA BORROSAS

En esta sección se interpretará el conjunto H_E de subconjuntos borrosos extensionales de una relación de indistinguibilidad E sobre un conjunto X como el conjunto de las partes de las clases de equivalencia borrosas de X respecto de E . En este contexto, la aplicación ϕ_E que asigna a cada subconjunto borroso el menor extensional que lo contiene puede considerarse la proyección canónica. Si restringimos ϕ_E al conjunto $P(X)$ de las partes clásicas no vacías de X , $\phi_E(P(X))$ es el subsemirretículo H_E^C de H_E generado por subfamilias de $\{E(x, \cdot)\}_{x \in X}$ vía supremos [10,12,14,15,16,17].

Definición 2.1. Dada una t -norma T , una relación de indistinguibilidad en un conjunto X es una relación borrosa de X que para todo $x, y, z \in X$ satisface

$$2.1.1 \quad E(x, x) = 1 \text{ (Reflexividad)}$$

$$2.1.2 \quad E(x, y) = E(y, x) \text{ (Simetría)}$$

$$2.1.3 \quad T(E(x, y), E(y, z)) \leq E(x, z) \text{ (T-transitividad)}$$

Las relaciones de indistinguibilidad extienden el concepto de relación de equivalencia a contextos borrosos [16].

Lemma 2.2. Dado un subconjunto borroso h de X y una t -norma T continua, la relación borrosa E_h en X definida para todo $x, y \in X$ por

$$E_h(x, y) = \hat{T}(\text{Max}(h(x), h(y)) | \text{Min}(h(x), h(y))),$$

donde $\hat{T}$ es la quasi-inversa de T , es una T -indistinguibilidad.

*Financiado parcialmente por el proyecto número PB94-1208 de la DGICYT

Demostración. [10]

Definición 2.3. Sea E una T-indistinguibilidad en el conjunto X . Se define la aplicación $\phi_E : [0, 1]^X \rightarrow [0, 1]^X$ por

$$\phi_E(h)(x) = \sup_{y \in X} T(E(x, y), h(y)).$$

Proposición 2.4. ϕ_E es un operador de clausura borroso, i.e.: ϕ_E satisface las siguientes propiedades:

$$2.4.1. h \leq \phi_E(h)$$

$$2.4.2. \phi_E(h_1 \vee h_2) = \phi_E(h_1) \vee \phi_E(h_2)$$

$$2.4.3. \phi_E^2(h) = \phi_E(h)$$

$$2.4.4. \phi_E(k) = k \text{ para todo borroso constante } k(x) = k.$$

Demostración. [10,11]

Definición 2.5. $h \in [0, 1]^X$ es extensional respecto E si, y sólo si, $\phi_E(h) = h$.

El conjunto H_E ha sido ampliamente estudiado [10,12,14,15] y sus elementos han sido caracterizados como los vectores propios [15] y los generadores de E [10], los puntos fijos [14] de ϕ_E , los estados lógicos asociados a la relación E [4] y sus elementos extensionales [17,19].

Con esta última interpretación, los elementos de H_E son los únicos subconjuntos borrosos observables en X si se tiene en cuenta la indistinguibilidad E y ϕ_E es la forma en que E distorsiona los subconjuntos tanto borrosos como clásicos de X .

Así, por ejemplo, se tienen

Proposición 2.6. $\forall x \in X, \phi_E(\{x\}) = E(x, \cdot)$.

Demostración. [10]

Corolario 2.7. $\forall A \subset X, \phi_E(A) = \bigvee_{x \in A} E(x, \cdot)$.

Demostración. [10]

Corolario 2.8. Sea $P(X)$ el conjunto de las partes no vacías de X . $\phi_E(P(X))$ es el subsemirretículo H_E^C de H_E generado por subfamilias de $\{E(x, \cdot)\}_{x \in X}$ vía supremos.

Así H_E puede interpretarse como el conjunto de los subconjuntos borrosos del conjunto cociente X/E ($H_E = [0, 1]^{X/E}$) y $\phi_E : [0, 1]^X \rightarrow H_E = [0, 1]^{X/E}$ es

la proyección canónica. Nótese que si la indistinguibilidad E es clásica (i.e. es una relación de equivalencia clásica), entonces $\phi_E|_X$ es la proyección canónica clásica $\pi : X \rightarrow X/E$.

Dados dos subconjuntos disjuntos de X , borrosos o clásicos es posible que sus extensiones vía ϕ_E no lo sean debido a la imprecisión que se genera al tener en cuenta E . Conviene tener alguna medida al solapamiento o similitud entre subconjuntos borrosos. En [2,3] se introduce la indistinguibilidad natural E_T entre borrosos y se aplica al razonamiento aproximado.

Definición 2.9. Sea X un conjunto y T una t-norma continua. La indistinguibilidad natural E_T entre subconjuntos borrosos de X se define por

$$E_T(h_1, h_2) = \inf_{x \in X} \hat{T}(\text{Max}(h_1(x), h_2(x)) | \text{Min}(h_1(x), h_2(x))).$$

Una propiedad interesante de E_T es su carácter isométrico universal para las T-indistinguibilidades E en X vía la isometría ϕ_E .

Proposición 2.10. Sea E una T-indistinguibilidad en X y E_T la indistinguibilidad natural en $[0, 1]^X$. Para todo $x, y \in X$ se tiene

$$E(x, y) = E_T(\phi_E(\{x\}), \phi_E(\{y\})).$$

Demostración. [2]

3 RELACIONES DE INDISTINGUIBILIDAD Y NEGACIONES FUERTES

En esta sección demostraremos algunos resultados que relacionan las negaciones fuertes con las relaciones de indistinguibilidad.

Empezamos por recordar el teorema de representación de t-normas arquimedianas de C.H. Ling y el teorema de representación de negaciones fuertes de E. Trillas.

Teorema 3.1. (Ling). [20]. Una t-norma T es arquimediana si, y sólo si, existe una función continua y estrictamente decreciente t de $[0, 1]$ en $[0, +\infty]$ con $t(1) = 0$ tal que

$$T(x, y) = t^{[-1]}(t(x) + t(y))$$

donde $t^{[-1]}$, la pseudoinversa de t , se define como

$$t^{[-1]}(x) = \begin{cases} t^{-1}(x), & \text{si } x \in [0, t(0)] \\ 0, & \text{en caso contrario.} \end{cases}$$

Definición 3.2. Una *negación fuerte* es una biyección φ decreciente de $[0, 1]$ en $[0, 1]$ tal que $\varphi(0) = 1$, $\varphi(1) = 0$ y es involutiva ($\varphi^2 = \text{id}$).

Teorema 3.2. (Trillas). [23]. Una función φ de $[0, 1]$ en sí mismo es una negación fuerte si, y sólo si, existe una función continua y estrictamente decreciente t de $[0, 1]$ en $\mathbb{R}^+$ con $t(1) = 0$ tal que

$$\varphi(x) = t^{-1}(t(0) - t(x)) \quad \forall x \in [0, 1].$$

Las funciones t de los teoremas anteriores se llaman generadores de la t -norma y la negación fuerte respectivamente.

Si bien el generador de una t -norma es único, salvo una constante multiplicativa, no ocurre lo mismo con las negaciones. Sería interesante estudiar la relación existente entre t -normas cuyos generadores dieran lugar a la misma negación.

El Teorema 3.2 tiene la siguiente interpretación dentro de las MV -álgebras:

Teorema 3.3. φ es una negación fuerte si, y sólo si, existe una t -norma arquimediana no estricta T con generador t tal que $\varphi(x) = x \rightarrow 0$.

Demostración. $x \rightarrow 0 = \hat{T}(x|0) = t^{-1}(t0 - tx)$.

Un estudio de la relación entre las t -normas y negaciones fuertes con igual generador nos lleva a los teoremas 3.4 y 3.5.

Teorema 3.4. Sea E_T la indistinguibilidad natural en $[0, 1]^X$. $E_T(h_1, h_2) = E_T(\varphi h_1, \varphi h_2) \quad \forall h_1, h_2 \in [0, 1]^X$ con φ una negación fuerte si, y sólo si, T es arquimediana no estricta y φ tiene el mismo generador que T .

Demostración. Sean h_1, h_2 dos subconjuntos borrosos de X tales que $a = h_1(x) \geq h_2(x) = b$. Para un cierto $x \in X$ y $h_1(y) = h_2(y) \quad \forall y \in X, y \neq x$

$$E_T(h_1, h_2) = E_T(\varphi h_1, \varphi h_2) \text{ sii}$$

$$t^{[-1]}(tb - ta) = t^{[-1]}(t\varphi a - t\varphi b) \text{ sii}$$

$$tb - ta = t\varphi a - t\varphi b \text{ sii}$$

$$t\varphi a + ta = t\varphi b + tb = \text{cte.}^*$$

Tomando $a = 1$, $t0 = \text{cte} < \infty$ y volviendo a *

$$\varphi a = t^{-1}(t0 - ta).$$

Fijándose de nuevo en la aplicación ϕ_E , observamos que $\phi_E h$ es el menor de los subconjuntos borrosos extensionales respecto E que contiene h . Así que dicho conjunto es la mejor aproximación superior de h si se tiene en cuenta la indistinguibilidad E . Este es el concepto de aproximación superior de un rough set [21].

De forma análoga se puede definir la mejor aproximación inferior $\underline{h}$ de h respecto E como el supremo de las extensionales contenidas en h . Este conjunto existe porque H_E es cerrado por supremos (e ínfimos) arbitrarios.

Se tiene definida de esta forma el *rough set* borroso $(\phi_E h, \underline{h})$ asociado a h respecto E .

No se conoce una fórmula general para hallar $\underline{h}$. Usualmente aplicando ideas de lógica modal [6], se define la aproximación inferior como $\varphi\phi_E(\varphi h)$. Sin embargo, en general este conjunto no es extensional respecto a E . (De hecho, $\varphi\phi_E(\varphi h)$ es un abierto de la topología borrosa asociada a E de la cual H_E es el conjunto de cerrados [11,14]).

El siguiente teorema determina bajo qué condiciones $\varphi\phi_E(\varphi h)$ es extensional respecto E .

Teorema 3.5. $\varphi\phi_E(\varphi h)$ es extensional respecto a una T -indistinguibilidad E definida en un conjunto X para todo subconjunto borroso h de X con φ una negación fuerte si y sólo si T es no estricta con mismo generador que φ .

4 DEFUZZIFICACIÓN EN PRESENCIA DE INDISTINGUIBILIDAD

El proceso de *defuzzificación* lo consideraremos como un proceso inverso al de *fuzzificación*.

La necesidad de *fuzzificación* en un sistema borroso se puede deber a la existencia de una relación de indistinguibilidad E en el conjunto X de los *inputs* (por ejemplo, en *fuzzy control*). En este caso se substituyen los *inputs* por subconjuntos borrosos de X . Por lo visto en la sección 2, una manera razonable de *fuzzificar* el *input* clásico $A \subset X$ consiste en asignarle $\phi_E(A)$. Esta *fuzzificación* es prudente, porque $\phi_E(A)$ es menos específico que A y por tanto no se aumenta la información en el proceso. De hecho, $\phi_E(A)$ es el menor de los extensionales menos específicos que A . En este sentido, es la elección prudente óptima.

Cuando en el conjunto Y de los *outputs* también hay una indistinguibilidad inherente F , el proceso de *defuzzificación* debe ser el inverso. Idealmente, *defuzzificar* el subconjunto borroso μ de Y es hallar el conjunto clásico $B \subset Y$ tal que $\phi_F(B) = \mu$.

Nótese que al *defuzzificar* pretendemos obtener un subconjunto y no un elemento de Y . *Defuzzificar* significa eliminar la vaguedad, pero no la imprecisión.

Sin embargo, en general tal B no existe. Actuando de forma prudente otra vez (es decir sin incrementar información), el subconjunto B idóneo para la *defuzzificación* de μ es

$$\overline{\text{defuzz}}(\mu) = B = \inf\{C \subset Y \mid \phi_F(C) \supset \mu\}$$

B es el conjunto de los posibles valores clásicos del *output* correspondiente a μ . En el contexto de la lógica modal, se puede definir el conjunto $B' \subset Y$ de los valores que de forma *necesaria* deben ser *outputs* clásicos de μ :

$$\text{defuzz}(\mu) = B = \sup\{C \subset Y \mid \phi_F(C) \subset \mu\}$$

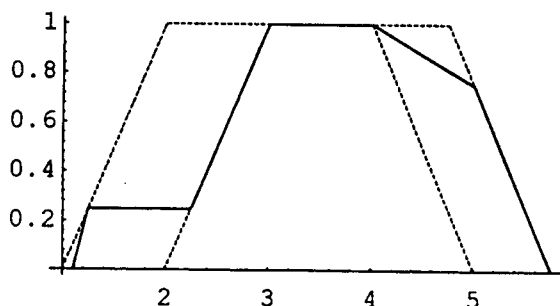
Proposición 4.1. Sean $A \subset Y$ y $\mu \in [0, 1]^Y$

$$\begin{aligned}\overline{\text{defuzz}}(\phi_E(A)) &= A, \\ \text{defuzz}(\phi_E(A)) &= A, \\ \phi_E(\text{defuzz}(\mu)) &= \phi_E(\mu), \\ \phi_E(\overline{\text{defuzz}}(\mu)) &= \mu,\end{aligned}$$

Demostración. Trivial.

Nota: Si *fuzzyficamos* y después *defuzzyficaremos* un conjunto clásico, obtenemos el propio conjunto. Y si *defuzzyficamos* y después *fuzzyficamos* un conjunto borroso, obtenemos su extensión o su aproximación inferior.

Ejemplo 4.2. $Y = \mathbb{R}$ y $E(x, y) = \text{Max}(1 - |x - y|, 0)$. En la figura se muestran un conjunto borroso μ de $\mathbb{R}$ y sus correspondientes $B = [1, 5.75]$ y $B' = [2, 5]$.



5 DEFUZZYFICACIÓN EN PRESENCIA DE AMBIGÜEDAD LINGÜÍSTICA

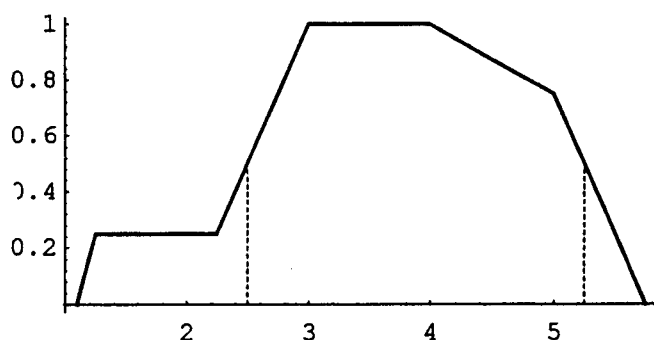
La borrosidad, en un sistema borroso puede deberse a la ambigüedad inherente a *inputs* lingüísticos. En este caso, el grado de semejanza entre dos *inputs* se puede medir mediante la indistinguibilidad natural E_T definida en el conjunto X de los *inputs* borrosos del sistema. Análogamente, los *outputs* también podrán relacionarse mediante la indistinguibilidad natural F_T en el conjunto Y de los *outputs*.

En este contexto, *defuzzyficar* un *output* μ de Y es buscar el conjunto clásico $\text{defuzz}(\mu) \subset Y$ más similar a μ por F_T . Si hay varios conjuntos con esta característica, el *defuzzyficado* se toma como la reunión de todos ellos para obtener la menor especificidad.

Proposición 5.1. Si μ es un clásico, entonces $\text{defuzz}(\mu) = \mu$.

Demostración. Trivial.

Ejemplo 5.2. $Y = \mathbb{R}$. En la figura se muestran el conjunto borroso μ del ejemplo 4.2 y su *defuzzyficado* $\text{defuzz}(\mu) = [2.5, 5.25]$.



Una propiedad que parece interesante en este contexto es que dada una negación fuerte φ , entonces $\text{defuzz}(\mu) \cap \text{defuzz}(\varphi\mu)$ sea lo menor posible. Nótese que esa intersección como mínimo contendrá los $y \in Y$ tales que $\mu(y) = \varphi\mu(y)$.

Proposición 5.2. Sea φ una negación fuerte y F_T la indistinguibilidad natural en $[0, 1]^Y$. $\text{defuzz}(\mu) \cap \text{defuzz}(\varphi\mu) = \{y \in Y \mid \mu(y) = \varphi\mu(y)\}$ si, y sólo si, T es una t-norma arquimediana no estricta y el generador de T también es generador de φ .

Demostración. Basta ver que es equivalente a $F_T(h_1, h_2) = F_T(\varphi h_1, \varphi h_2)$ y aplicar el teorema 3.4.

6 CONCLUSIONES

Este trabajo es un primer paso en la búsqueda de una semántica de la *defuzzyficación*. Se han dado dos modelos o alternativas dependiendo de la naturaleza del sistema borroso y en especial de su conjunto de *outputs*.

El primer método está relacionado con la resolución de ecuaciones de relaciones borrosas ($F \cdot B = \mu$) [5] y con rough sets y lógica modal.

El segundo se basa en la existencia de la indistinguibilidad natural F_T en el conjunto de los *outputs* del sistema.

Referencias

- [1] Alsina, C., Castro, J.L., Trillas, E. (1995), On the characterization of S and R Implications. *Proc. IFSA '95*. Sao Paulo, 317-319.
- [2] Boixader, D., Jacas, J. (1995), Generators and Dual T-indistinguishabilities, in *Fuzzy Logic and Soft Computing*, B. Bouchon et altri eds. World Scientific. Singapore, 283-291.
- [3] Boixader, D., Jacas, J. (1996), T-indistinguishability Operators and Approximate Reasoning via CRI, *Linz'96*, 14-18.
- [4] Castro, J.L., Trillas, E., Cubillo, S. (1994), On Consequence in Approximate Reasoning. *Journal of Applied Non-Classical Logics* 4, 91-103.
- [5] Di Nola, A., Sessa, S., Pedrycz, W., Sanchez, E. (1989), *Fuzzy Relation Equations and their Applications to Knowledge Engineering*. Kluwer Academic Publishers, Dordrecht.
- [6] Dubois, D., Prade, H. (1994), Similarity-Based Approximate Reasoning, en: *Computational Intelligence Imitating Life*. Zurada, J.M., Marles, R.J., Robinson, C.J. eds. IEEE Press, New York, 69-80.
- [7] Esteva, F. (1981), Negaciones en la teoría de conjuntos difusos. *Stochastica* 5, 33-44.
- [8] Esteva, F., Domingo, X. (1980), Sobre funciones de negación en $[0,1]$. *Stochastica* 4, 141-165.
- [9] Esteva, F., Trillas, E., Domingo, X. (1981), Weak and strong negations for Fuzzy Sets Theory, *Proc. XI ISMVL*, Oklahoma. 23-26.
- [10] Jacas, J. (1989), On the Generators of a T-indistinguishability Operator. *Stochastica* 12, 49-63.
- [11] Jacas, J. (1993), Fuzzy topologies induced by S-metrics. *The Journal of Fuzzy Mathematics* 1, 173-191.
- [12] Jacas, J., Recasens, J. (1992) Eigenvectors and generators of fuzzy relations. *Proc. IEEE International Conference on Fuzzy Systems*. San Diego, 687-694.
- [13] Jacas, J., Recasens, J. (1993) Fuzzy numbers and equality relations. *Proc. IEEE International Conference on Fuzzy Systems*. San Francisco.
- [14] Jacas, J., Recasens, J. (1994) Fixed points and generators of fuzzy relations. *J. Math. Anal. Appl.* 186 21-29.
- [15] Jacas, J., Recasens, J. (1995) Fuzzy T-transitive relations: eigenvectors and generators. *Fuzzy Sets & Systems* 72 147-154.
- [16] Jacas, J., Valverde, L. (1990) On Fuzzy Relations, Metrics and Cluster Analysis. J.L. Verdegay, M. Delgado, Eds. *Approximate Reasoning Tools for Artificial Intelligence* (Verlag TUV, Rheinland) 15R, 96.
- [17] Klawoon, F., Castro, J.L. (1995) Similarity in Fuzzy Reasoning, *Mathware & Soft Computing* 2, 197-228.
- [18] Klawoon, F., Kruse, R. (1993) Equality Relations as a Basis for Fuzzy Control. *Fuzzy Sets and Systems* 54, 147-156.
- [19] Kruse, R., Gebhard, J., Klawoon, F. (1994) *Foundations of Fuzzy Systems*. Wiley, Chichester.
- [20] Ling, C.H. (1965) Representation of Associative Functions. *Publ. Math. Debrecen*, 12, 182-212.
- [21] Lowen, R. (1976), Fuzzy topological spaces and fuzzy compactness. *J. Math. Anal. Appl.* 56, 621-633.
- [22] Pawlak, Z. (1991) *Rough Sets*. Kluwer. Dordrecht.
- [23] Trillas, E. (1979). Sobre funciones de negación en la teoría de conjuntos difusos. *Stochastica* 3, 47-60.
- [24] Valverde, L. (1985) On the structure of F-indistinguishability operators. *Fuzzy Sets & Systems* 17, 313-328.
- [25] Zadeh, L.A. (1971) Similarity Relations and Fuzzy Orderings. *Information Sciences* 3, 177-200.

DEFINICIÓN DE PARÁMETROS ESTADÍSTICOS UTILIZANDO LA INTEGRAL DE CHOQUET

Manuel Jorge Bolaños Carmona
Dpto. Estadística e I.O.
Universidad de Granada
e-mail: mjbca@robinson.ugr.es

Fernando Reche Lorite
Dpto. Estadística y Matemática Aplicada
Universidad de Almería
e-mail: freche@stat.ualm.es

Resumen

Es este trabajo se presentan algunas definiciones de parámetros estadísticos en un contexto no-aditivo utilizando medidas difusas. Así pues, se define el concepto de Varianza Monótona y de Momentos Monótonos de funciones no negativas. Se sugiere una transformación de funciones utilizando la Integral de Choquet para facilitar y, en algún sentido, explicar la agregación de información.

Palabras Clave: Medida difusa, Integral de Choquet, Esperanza Monótona, Varianza Monótona.

1 INTRODUCCIÓN

La integral difusa de Sugeno ha sido ampliamente estudiada como instrumento para la evaluación de funciones acotadas sobre medidas no aditivas (medidas difusas) [9, 10, 13]. Más recientemente, la integral de Choquet o Esperanza Monótona se ha utilizado como alternativa a aquélla con características más cercanas a los operadores integrales habitualmente usados en la Matemática y la Estadística clásicas [1, 3, 6]. Precisamente por ser una generalización de la Esperanza Matemática en el contexto no-aditivo, la integral de Choquet parece un prometedor instrumento para establecer las bases de una teoría estadística a partir de las medidas difusas, lo que fué sólo parcialmente desarrollado con la integral de Sugeno [10], mucho más rígida a estos efectos.

El presente trabajo presenta las definiciones básicas, las propiedades más relevantes y las dificultades que pueden encontrarse para generalizar algunos de los parámetros estadísticos más importantes al contexto de la medida e integración difusa. En concreto, se definen los momentos centrados y no centrados y la varianza monótona, para abordar a continuación el

problema clave de la agregación o acumulación de información, a través de la modificación de las funciones originales. Finalmente, se indican algunos de los caminos futuros de estudio.

2 DEFINICIONES Y RESULTADOS CONOCIDOS

Sea $X = \{x_1, x_2, \dots, x_n\}$, diremos que $g : {}^1\mathcal{P}(X) \rightarrow [0, 1]$ es una *medida difusa* si cumple:

1. $g(\emptyset) = 0$
2. $g(X) = 1$
3. $\forall A, B \in \mathcal{P}(X)$ tales que $A \subset B$ se cumple que $g(A) \leq g(B)$

En lo sucesivo, consideraremos siempre un referencial X finito.

Sea $(X, \mathcal{P}(X), g)$ un espacio de medida y h una función $h : X \rightarrow \mathbb{R}^+$, se define la *integral de Choquet* de h con respecto a g como

$$C(h) = \int g(H_\alpha) d\alpha = \sum_{i=1}^n (\alpha_i - \alpha_{i-1}) g(H_{\alpha_i})$$

donde H_α es el α -corte de h ($H_\alpha = \{x \in X / h(x) \geq \alpha\}$), $\alpha_i = h(x_i)$, $\alpha_0 = 0$ y suponemos que $0 = \alpha_0 \leq \alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_n$, lo cual nos simplifica la notación y no conlleva ninguna pérdida de generalidad.

Este funcional también se conoce con el nombre de *Esperanza Monótona* [5] de la función h con respecto a la medida difusa g que notaremos por $E_g(h)$ y cuyas propiedades son una extensión de la Esperanza Matemática [2, 6].

¹ $\mathcal{P}(X)$ denota el conjunto de las partes de X

3 VARIANZA MONÓTONA

De Campos y Bolaños [5] caracterizaron las medidas difusas mediante probabilidades de tal forma que en un referencial finito de cardinal n , una medida difusa es equivalente a un conjunto formado como máximo por $n!$ distribuciones de probabilidad diferentes, que vienen determinadas por los distintos órdenes que se pueden establecer en el referencial.

Si observamos la definición de la Esperanza Monótona, reordenando los términos de la suma obtenemos:

$$E_g(h) = \sum_{i=1}^n \alpha_i [g(H_{\alpha_i}) - g(H_{\alpha_{i+1}})]$$

donde $H_{\alpha_{n+1}} = \emptyset$. Ello nos permite singularizar la función siguiente:

Definición Sea $(X, \mathcal{P}(X), g)$ un espacio de medida y $h : X \rightarrow \mathbb{R}^+$. Denominaremos *Función de Probabilidad asociada a la medida g y a la función h* al conjunto $P_{g,h} = \{p_1, p_2, \dots, p_n\}$ donde

$$p_i = g(H_{\alpha_i}) - g(H_{\alpha_{i+1}}) \quad \text{si } i < n \\ p_n = g(H_{\alpha_n})$$

$\alpha_i = h(x_i)$, $p_i = p(x_i)$ y $H_{\alpha_i} = \{x \in X / h(x) \geq \alpha_i\}$

Si analizamos esta definición podremos observar que $P_{g,h}$ podría generar una distribución de probabilidad, ya que todos sus elementos están entre cero y uno y la suma total vale uno. Además $P_{g,h}$ es una de las $n!$ distribuciones de probabilidad que caracterizan a la medida difusa y viene determinada por el orden que induce h sobre el referencial.

Utilizando $P_{g,h}$ obtenemos una nueva expresión para el Esperanza Monótona: $E_g(h) = \sum_{i=1}^n h(x_i) p_i$, es decir, la Esperanza Monótona de h con respecto a la medida g no es más que la Esperanza Matemática de la función h con respecto a la Función de Probabilidad asociada a la medida y a dicha función ($E_g(h) = E_P(h)$).

Este hecho nos permite definir la varianza de una forma coherente con el concepto de Esperanza Monótona. Nótese que no es posible definir la varianza de la forma:

$$\text{Var}_g(h) = E_g[(h - E_g(h))^2]$$

porque no está garantizado que los valores de la función permitan usar la misma medida P de probabilidad que se utiliza en el cálculo de $E_g(h)$, lo que puede dar lugar a enormes problemas e incoherencias en el planteamiento. Parece conveniente, pues, cuidar que las definiciones de los parámetros utilicen la misma función de probabilidad cuando se trabaja con una medida g y sobre una función h determinadas. Con este criterio, se tiene lo siguiente:

Definición Sea $(X, \mathcal{P}(X), g)$ un espacio de medida y $h : X \rightarrow \mathbb{R}^+$; se define la *Varianza Monótona* de h con respecto a la medida difusa g como

$$\text{Var}_g(h) = \sum_{i=1}^n [h(x_i) - E_g(h)]^2 p_i$$

con $p_i \in P_{g,h}$.

Esta varianza es no negativa por definición. Veamos algunas propiedades.

Teorema 1 Sea $(X, \mathcal{P}(X), g)$ un espacio de medida, $h : X \rightarrow \mathbb{R}^+$ y $p_i \in P_{g,h}$, entonces las siguientes condiciones son equivalentes:

1. $\text{Var}_g(h) = 0$
2. Existe un único i ($1 \leq i \leq n$) de tal forma que $p_i = 1$ y $p_j = 0 \forall j \neq i$
3. Existe i ($1 \leq i \leq n$) de forma que g es una medida tal que

$$g(H_{\alpha_j}) = 1 \quad \forall j \leq i \quad \text{y} \quad g(H_{\alpha_j}) = 0 \quad \forall j > i$$

Demostración.

(1) $\Rightarrow$ (2)

Como $\sum_{i=1}^n p_i = 1$ y $p_i \geq 0$ tiene que haber al menos un $p_i \neq 0$.

Supongamos que $\text{Var}_g(h) = 0$ y que hay dos p_i no nulos.

Entonces $\exists j, k$ tales que $p_j \neq 0$ y $p_k \neq 0$.

Sea $h(x_j) \leq h(x_{j+1}) \leq \dots \leq h(x_{k-1}) \leq h(x_k)$

Si $p_j \neq 0 \Rightarrow h(x_j) = E_g(h)$

Si $p_k \neq 0 \Rightarrow h(x_k) = E_g(h)$

De esto se deduce que $E_g(h) = h(x_j) \leq h(x_{j+1}) \leq \dots \leq h(x_{k-1}) \leq h(x_k) = E_g(h)$ por lo tanto $h(x_j) = h(x_{j+1}) = \dots = h(x_k) = E_g(h) \Rightarrow H_{\alpha_j} = H_{\alpha_{j+1}} = \dots = H_{\alpha_k} \Rightarrow p_j = p_{j+1} = \dots = p_{k-1} = 0$, en contra de lo supuesto, por lo que sólo hay un único $p_i \neq 0$ y además $p_i = 1$.

(2) $\Rightarrow$ (3)

Supongamos que existe un único i tal que $p_i \neq 0$, entonces

$$p_1 = p_2 = \dots = p_{i-1} = p_{i+1} = \dots = p_n = 0$$

lo que implica que $g(H_{\alpha_1}) = g(H_{\alpha_2}) = \dots = g(H_{\alpha_i})$, por un lado y $g(H_{\alpha_{i+1}}) = \dots = g(H_{\alpha_n})$, además $g(H_{\alpha_i}) - g(H_{\alpha_{i+1}}) = 1$, por lo tanto $g(H_{\alpha_j}) = 1$ si $j \leq i$ y $g(H_{\alpha_j}) = 0$ si $j > i$.

(3) $\Rightarrow$ (1)

Evidente por la definición de Varianza Monótona.

Corolario Si h es constante entonces $\text{Var}_g(h) = 0$, siendo g cualquier medida difusa.

Los dos resultados siguientes nos proporcionan propiedades lógicas que cabrían esperarse de un operador que funcionase como una varianza.

Teorema 2 Para cualquier medida difusa g y h una función no negativa de X se verifica que

$$\text{Var}_g(h) = E_g(h^2) - E_g^2(h)$$

Demostración.

$$\begin{aligned} \text{Var}_g(h) &= \sum_{i=1}^n (h(x_i) - E_g(h))^2 p_i \\ &= \sum_{i=1}^n h^2(x_i) p_i + E_g^2(h) - 2E_g(h) \sum_{i=1}^n h(x_i) p_i \\ &= \sum_{i=1}^n h^2(x_i) p_i - E_g^2(h) \end{aligned}$$

Este es un resultado interesante, ya que nos establece una relación directa entre la Varianza Monótona y la Esperanza Monótona.

Teorema 3 Si $T = ah + b$ con $a \in \mathbb{R}_0^+$ y $b \in \mathbb{R}^+$, siendo h una función no negativa de X y g una medida difusa arbitraria se cumple que

$$\text{Var}_g(T) = a^2 \text{Var}_g(h)$$

Demostración.

$$\begin{aligned} \text{Var}_g(T) &= \sum_{i=1}^n [(ah(x_i) + b) - E_g(ah + b)]^2 p_i \\ &= \sum_{i=1}^n [ah(x_i) + b - aE_g(h) - b]^2 p_i \\ &= a^2 \sum_{i=1}^n (h(x_i) - E_g(h))^2 p_i \\ &= a^2 \text{Var}_g(h) \end{aligned}$$

En resumen, hemos podido observar que nuestro operador tiene un comportamiento similar al que puede tener una varianza en el sentido clásico, por lo que podremos utilizar la Varianza Monótona como un indicador de "dispersión" respecto de la "media" en el ámbito de las medidas difusas.

Es importante hacer notar que, aunque la definición "natural" de varianza como integral de la función cuadrado de las desviaciones es completamente inoperativa en este contexto, la que hemos elegido forzando

la utilización de la Función de Probabilidad asociada tiene las características teóricas básicas de la varianza, lo que reafirma como, al menos teóricamente, la definición más adecuada.

4 MOMENTOS

En este punto parece natural hacer una extensión de los conceptos introducidos anteriormente. Si utilizamos la función de probabilidad asociada a una cierta función y a una medida difusa, podemos construir definiciones de conceptos tan fundamentales en la Estadística como son los momentos.

Definición Sea $(X, \mathcal{P}(X), g)$ un espacio de medida $h : X \rightarrow \mathbb{R}^+$, se define el *Momento monótono respecto al origen* de orden k de la función h considerando la medida difusa g de la forma

$$m_g^k(h) = E_g(h^k) = \sum_{i=1}^n h^k(x_i) p_i$$

donde $p_i \in P_{g,h}$

Esta definición no conlleva ninguna contradicción ya que la función $f(x) = x^k$ es monótona para cualquier k y, por lo tanto la función de probabilidad asociada es la misma sea cual sea el orden del momento.

Como se ve, esta definición generaliza de forma directa el habitual concepto de momento respecto del origen dada una distribución de probabilidad; lo que clásicamente se hace con la Esperanza Matemática, se realiza aquí con la Esperanza Monótona.

Sin embargo, la definición de momentos centrales es más problemática, como se ha visto en el caso particular de la varianza. Para estos no tiene mucho sentido definir directamente $E_g(h - E_g(h))^k$ como momento central de orden k , ya que no se utiliza la misma función de probabilidad, o los valores a promediar pueden ser negativos cuando se trate de un orden impar.

Parece más razonable generalizar las definiciones que se han dado anteriormente para la Varianza Monótona, asumiendo la utilización de la Función de Probabilidad asociada a la medida difusa y a la función que se trate.

Definición Sea $(X, \mathcal{P}(X), g)$ un espacio de medida y $h : X \rightarrow \mathbb{R}^+$; se define el *Momento Monótono central* de orden k de la función h con respecto a la medida difusa g de la forma

$$\mu_g^k(h) = \sum_{i=1}^n (h(x_i) - E_g(h))^k p_i$$

donde $p_i \in P_{g,h}$

Teorema 4 Sea g una medida difusa arbitraria y h una función no negativa de X , entonces se verifica

$$\mu_g^k(h) = \sum_{j=0}^k (-1)^j \binom{k}{j} [m_g(h)]^j m_g^{k-j}(h)$$

Demostración.

$$\begin{aligned} \mu_g^k(h) &= \sum_{i=1}^n (h(x_i) - E_g(h))^k p_i \\ &= \sum_{i=1}^n \sum_{j=0}^k (-1)^j \binom{k}{j} E_g^j(h) h^{k-j}(x_i) p_i \\ &= \sum_{j=0}^k (-1)^j \binom{k}{j} E_g^j(h) \sum_{i=1}^n h^{k-j}(x_i) p_i \\ &= \sum_{j=0}^k (-1)^j \binom{k}{j} [m_g(h)]^j m_g^{k-j}(h) \end{aligned}$$

A pesar de que, aparentemente, las definiciones anteriores parecen basarse en principios diferentes, se mantiene la relación clásica entre momentos centrales y no centrales.

De esta forma obtenemos una relación funcional entre el Momento Monótono central y los Momentos Monótonos respecto al origen de orden inferior, lo que será de gran utilidad en el cálculo de los parámetros y nos refuerza en la idea de que las definiciones dadas son adecuadas en este contexto.

En esta situación resulta inmediato definir parámetros similares a los existentes en la Estadística Clásica, como pueden ser coeficientes de asimetría, kurtosis, etc. . .

5 FUNCIÓN TRANSFORMADA POR UNA MEDIDA DIFUSA

En la sección 3 hemos visto como la Esperanza Monótona de una función h es equivalente a la Esperanza Matemática de dicha función con respecto a su función de probabilidad asociada. Ahora vamos a buscar una transformación de la función de tal forma que la Esperanza Monótona sea la Esperanza Matemática de la función transformada con respecto a una medida de probabilidad conocida.

Definición Sea $(X, \mathcal{P}(X), g)$ un espacio de medida, $h: X \rightarrow \mathbb{R}^+$ y $P_{g,h}$ la Función de Probabilidad asociada a la medida g y a la función. Denominaremos *Función Transformada de h por la medida g* a la transformación punto a punto de h construida de la siguiente forma:

$$h^*(x_i) = h(x_i) \frac{p_i}{p_i^U}$$

donde n es el cardinal del referencial, $p_i \in P_{g,h}$ y p_i^U es la distribución de probabilidad uniforme.

Con esta definición lo que hacemos es trasladar la estructura de no aditividad de la medida a la función.

El resultado más inmediato es que la Esperanza Monótona de una función con respecto a una medida difusa se convierte en la media aritmética de la función transformada por la medida difusa.

Por lo tanto, la medida subyacente de la función transformada es la medida de probabilidad uniforme.

El uso más inmediato que le podemos dar a esta función deformada es el de la agregación de información en el siguiente sentido:

Supongamos que tenemos $\{h_1, h_2, \dots, h_k\}$ funciones tales que $h_i: X \rightarrow \mathbb{R}^+$ para $1 \leq i \leq k$, entonces cada una de esas funciones inducirá una función de probabilidad P_{g,h_i} , con la que podremos construir su correspondiente función transformada $\{h_1^*, h_2^*, \dots, h_k^*\}$. Si calculamos la media de cada h_i^* , tendremos la Esperanza Monótona de cada h_i con respecto a la medida difusa, y si hacemos el promedio en todas las h_i^* para cada x_j obtendremos una valoración promedio que nos dan todas las funciones transformadas sobre el elemento x_j del referencial.

6 CUESTIONES DE INTERÉS Y LÍNEAS ABIERTAS

Es fácil deducir que la Esperanza Monótona de la función promedio de varias funciones agregadas coincide con la media aritmética de las esperanzas de cada una de ellas. Así, la transformación que acabamos de definir es coherente con el concepto de integral de Choquet.

Como era de esperar, no ocurre algo similar con la Varianza Monótona, ya que la varianza (en el sentido clásico) de la transformada de una función no puede expresarse en términos de la Varianza Monótona de la misma. Sin embargo, la definición elegida, tanto para la Varianza Monótona como para los Momentos Monótonos centrales, permite que la transformación de una función sea de idéntica estructura que la transformación de la función de los cuadrados de las "desviaciones" que se usa para definir la varianza, lo que permitirá un análisis más fructífero de las relaciones entre todos estos parámetros. Tal estudio de las varianzas monótonas y clásicas implicadas en la agregación será, junto con la extensión al caso bivalente, objeto de futuros desarrollos.

Referencias

- [1] Bolaños, M.J, Hernández, L.D., Salmerón, A. (1995) Experimentación numérica y comparación de integrales difusas. *Actas del Congreso ESTYLF95. Anales de Informática* Vol I, 89-94.
- [2] Bolaños, M.J, de Campos, L.M., González, A. (1989) Convergence properties of the monotone expectation and its application to the extension of fuzzy measures. *Fuzzy Sets and Systems* **33**, 201-213.
- [3] Bolaños, M.J, Lamata, M.T., Moral, S. (1987) Extensión de medidas difusas usando la esperanza monótona. *Stochastica* **XI-2,3**, 75-92.
- [4] Choquet, G. (1953/54) Theory of capacities. *Ann. Ins. Fourier* (Grenoble) **5**, 131-293.
- [5] de Campos, L.M, Bolaños, M.J. (1989) Representation of fuzzy measures through probabilities. *Fuzzy Sets and Systems* **31**, 23-36.
- [6] de Campos, L.M, Bolaños, M.J. (1992) Characterization and comparison of Sugeno and Choquet integrals. *Fuzzy Sets and Systems* **52**, 61-67.
- [7] de Campos, L.M., Lamata, M.T., Moral, S. (1990) The concept of conditional fuzzy measure *International Journal of Intelligent Systems* **5**, 237-246.
- [8] Dubois, D., Prade, H. (1980) Fuzzy Sets and Systems: Theory and Application *Academic Press, Inc.*
- [9] Friedman, M., Schneider, M., Kandel, A. (1989) The use of weighted fuzzy expected value in fuzzy expert systems. *Fuzzy Sets and Systems* **31**, 37-45.
- [10] Kandel, A. (1988) 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.
- [11] Murofushi, T., Sugeno, M. (1989) An interpretation of fuzzy measures and the Choquet integral a fuzzy measure. *Fuzzy Sets and Systems* **29**, 201-227.
- [12] Murofushi, T., Sugeno, M., Machida, M. (1994) Non-monotonic fuzzy measures and the Choquet integral. *Fuzzy Sets and Systems* **64**, 73-86.
- [13] Suarez, F., Gil, P. (1986) Two families of fuzzy integrals. *Fuzzy Sets and Systems* **18**, 67-81.

SOBRE EXTENSIONES DE LOS CONCEPTOS MAYORANTE, MINORANTE, MAXIMO, MINIMO, SUPREMO E INFIMO A SUBCONJUNTOS DIFUSOS DE UN REFERENCIAL CON UN PREORDEN DIFUSO

A. Burusco y R. Fuentes-González
Departamento de Automática y Computación
Universidad Pública de Navarra
Campus de Arrosadía s/n. 31006. Pamplona. Navarra
E-mail: {burusco, rfuentes}@si.upna.es

Resumen

Considerando una negación $'$, una t -norma T y el sistema relacional $(X, \tilde{R})$ determinado por un referencial ordinario X con un preorden difuso $\tilde{R}$, se construye un par de operadores $"$ y $'$ en la clase $[0,1]^X$ de los subconjuntos difusos de X , tales que asocian a cada subconjunto difuso $\tilde{A}$ del referencial X dos nuevos subconjuntos difusos, llamados *mayorante* $\tilde{A}''$ y *minorante* $\tilde{A}'$ de $\tilde{A}$.

Se demuestra que, en relación al subconjunto difuso $\tilde{A}$ en $(X, \tilde{R})$, $\tilde{A}''$ y $\tilde{A}'$ juegan un papel análogo al que desempeñan los subconjuntos ordinarios de mayorantes M'' y de minorantes M' de un subconjunto M de un referencial parcialmente ordenado $(X, \leq_X)$.

Después, y utilizando una nueva t -norma T_1 , (que puede coincidir con la primera T), se emplean los operadores $"$ y $'$ para dar las definiciones de los nuevos subconjuntos difusos asociados a $\tilde{A}$: $MAX\tilde{A}$, $MIN\tilde{A}$, $SUP\tilde{A}$ e $INF\tilde{A}$. Finalmente se dan algunos ejemplos.

Palabras clave: Cotas difusas, Extremos difusos, Preórdenes difusos.

1 INTRODUCCIÓN

En la teoría de conjuntos parcialmente ordenados, dado el sistema relacional $(X, \leq_X)$ formado por un referencial X y una relación de orden parcial $\leq_X$ nítida, se asocia a cada subconjunto ordinario M de X , el par de subconjuntos ordinarios M'' de sus mayorantes y M' de sus minorantes definidos respectivamente por:

$$M'' = \{u \in X / (\forall m \in M) (m \leq_X u)\} \text{ y}$$

$$M' = \{l \in X / (\forall m \in M) (l \leq_X m)\},$$

definiciones que determinan dos operadores $"$ y $'$ en el conjunto de las partes $\wp(X)$ de X y que son operadores antítonos para el orden $\subseteq$ en $\wp(X)$.

El subconjunto $M \subseteq X$ tiene elemento máximo $\max M$ si y solo si $M \cap M'' \neq \emptyset$, y en este caso esta intersección es el subconjunto: $M \cap M'' = \{\max M\}$. Un resultado

análogo es válido para el elemento mínimo $\min M$ de M , en este caso en función de la intersección $M \cap M'$.

Tanto la existencia del elemento mínimo del subconjunto M'' de mayorantes de M como del elemento máximo del subconjunto M' de sus minorantes, asegura a su vez la existencia de los elementos supremo $\sup M \in X$ e infimo $\inf M \in X$ ([4]) del subconjunto M :

Si existe el elemento mínimo $\min M''$ de M'' , se dice que es el supremo $\sup M$ de M en X . Por lo tanto, si M'' representa el subconjunto ordinario $(M'')'$, la existencia del elemento $\sup M$ implica la igualdad entre subconjuntos $M'' \cap M'' = \{\sup M\}$ donde $M'' = (M'')'$.

Análogamente $\inf M$ es el elemento máximo $\max M'$ de M' si existe. De ser así, $M' \cap M' = \{\inf M\}$.

Estos conceptos *sup* e *inf* pueden extenderse a sistemas relacionales (X, R) en los que R es una relación binaria sobre X no necesariamente de orden. Concretamente, en ([5]) se definen utilizando una relación $\tilde{R}$ a la que solo se le exige ser transitiva.

En ([1]), considerando una relación de orden $\leq_X$ nítida en el referencial X y la negación de Zadeh, se extienden los conceptos de mayorante y de minorante a la clase $[0,1]^X$ de los subconjuntos difusos $\tilde{A}$ del referencial X , extensiones que aparecen en forma de operadores sobre $[0,1]^X$. Después, utilizando estos conceptos y con la ayuda del mínimo $\wedge$ en $[0,1]$, se extienden también los conceptos de supremo SUP y de infimo INF a la clase de subconjuntos difusos.

En esta comunicación se presentan definiciones más generales de operadores MAX , MIN , SUP e INF en $[0,1]^X$. Estos dos últimos generalizan los mencionados en el párrafo anterior ya que, por una parte, están asociados a la consideración de una relación binaria difusa $\tilde{R}$ en el referencial de partida X (en lugar de un orden nítido $\leq_X$) y por otra, a la utilización de t -normas en sustitución de la operación mínimo $\wedge$ usada en las definiciones que aparecen en ([1]).

Estos nuevos operadores están basados en la introducción previa de los operadores *mayorante* $"$ y *minorante* $'$ en $[0,1]^X$, a su vez relacionados con los

operadores de la *teoría de conceptos difusos* ([2],[3]) en el caso particular de contextos difusos en los que coinciden los conjuntos de objetos y atributos con el referencial X de partida.

En las definiciones básicas que aparecen aquí, además de la t -norma τ y de la relación binaria difusa $\tilde{R}$ en X , juega también un papel significativo tanto la consideración de una *negación* ' como la $(\vee, \tau)$ -composición usual de subconjuntos difusos $\tilde{A} \cdot \tilde{R}$ definida a partir de la t -norma τ y del operador supremo $\vee$.

En lo que sigue, y para cualquier referencial ordinario, el símbolo $\tilde{\subset}$ representará la *inclusión difusa* usual definida puntualmente en la clase $[0,1]^X$ a partir del orden $\leq$ de la cadena $[0,1]$. Los símbolos $\tilde{0}$ y $\tilde{1}$ representarán respectivamente los elementos mínimo y máximo en el retículo completo $([0,1]^X, \tilde{\subset})$.

Por otra parte, los símbolos $\vee, \vee, \wedge$ y $\wedge$ representarán respectivamente los operadores máximo y supremo, mínimo e ínfimo en $[0,1]$ y los símbolos $\tilde{\cup}, \tilde{\cup}, \tilde{\cap}$ y $\tilde{\cap}$ sus extensiones puntuales a clases de subconjuntos difusos. Reservamos así la notación *MAX*, *SUP*, etc., para expresar los conceptos que se definen aquí.

Consideraremos también una *negación* en $[0,1]$, representada por '. Si la t -norma en el mismo intervalo viene representada por el operador infijo τ , en lo que sigue s_τ representará la t -conorma asociada al par $(\tau, ')$ definida por

$$(\forall (\alpha, \beta) \in [0,1]^2) (\alpha s_\tau \beta = (\alpha \tau \beta)')$$

Escribimos $\tilde{A} \tau \tilde{B}$ para representar el conjunto difuso:

$$(\forall x \in X) ((\tilde{A} \tau \tilde{B})(x) = \tilde{A}(x) \tau \tilde{B}(x)).$$

Dados $\tilde{A} \in [0,1]^X$ y $\tilde{R} \in [0,1]^{X \times X}$, $\tilde{A}'$ y $\tilde{R}'$ serán sus *complementarios* y $\text{sop } \tilde{A} = \{x \in X / \tilde{A}(x) \neq 0\}$, $\text{sop } \tilde{R} = \{(x,y) \in X \times X / \tilde{R}(x,y) \neq 0\}$ sus *soportes*. Un *punto difuso* $\tilde{A} = \tilde{x}_\alpha$ tiene como soporte $\{x\}$ y es tal que $\tilde{A}(x) = \alpha$. Si $\tilde{A}, \tilde{R}$ son *nítidos*, (es decir si $\tilde{A}(X), \tilde{R}(X \times X)$ están incluidos en $\{0,1\}$), los identificamos con las funciones características de sus respectivos soportes.

Consideremos finalmente el *sistema relacional difuso* $(X, \tilde{R})$ determinado por un referencial $X \neq \emptyset$ y una *relación difusa* $\tilde{R}: X \times X \longrightarrow [0,1]$.

Aunque gran parte de los resultados que aparecen aquí son válidos para relaciones difusas en general, estamos interesados en *preórdenes* y *órdenes parciales difusos* ([6]) en el referencial X . En ésta línea, en lo que sigue suponemos que $\tilde{R}$ es al menos *reflexiva* y τ -*transitiva*.

2 OPERADORES MAYORANTE Y MINORANTE EN SISTEMAS DIFUSOS

Sean τ y ' una t -norma y una negación y sea $(X, \tilde{R})$ un sistema relacional difuso de los citados en el apartado anterior. Proponemos las siguientes:

2.1 DEFINICIONES

Sea $\tilde{A} \in [0,1]^X$. Llamaremos *mayorante de $\tilde{A}$ asociado a $(\tau, ', \tilde{R})$* , (o simplemente *mayorante de $\tilde{A}$*), al subconjunto difuso $\tilde{A}''$ definido por:

$$\tilde{A}'' = (\tilde{A} \cdot \tilde{R}')'$$

es decir, tal que

$$(\forall y \in X) (\tilde{A}''(y) = (\vee \{ \tilde{A}(x) \tau \tilde{R}'(x,y) / x \in X \})'),$$

Al operador $''$ que hace corresponder a cada subconjunto difuso $\tilde{A}$ su mayorante $\tilde{A}''$ lo llamaremos *operador mayorante asociado a $(\tau, ', \tilde{R})$* .

Si $\tilde{B} \in [0,1]^X$ y si $\tilde{R}^{op}$ representa la relación difusa opuesta de $\tilde{R}$, llamaremos *minorante de $\tilde{B}$ asociado a $(\tau, ', \tilde{R})$* , (o simplemente *minorante de $\tilde{B}$*), al subconjunto difuso $\tilde{B}'$ tal que:

$$\tilde{B}' = (\tilde{B} \cdot (\tilde{R}^{op})')'.$$

El operador $'$ que hace corresponder a cada subconjunto difuso $\tilde{A}$ su minorante $\tilde{A}'$ es el *operador minorante asociado a $(\tau, ', \tilde{R})$* .

De la igualdad $(\tilde{R}^{op})^{op} = \tilde{R}$ se deduce que $\tilde{B}$ es el mayorante de $\tilde{A}$ asociado a $(\tau, ', \tilde{R})$ si y solo si es el minorante de $\tilde{A}$ asociado a $(\tau, ', \tilde{R}^{op})$, por lo que basta analizar propiedades de uno cualquiera de los dos conceptos introducidos. En esta línea, en lo sucesivo nos limitamos al estudio del *operador mayorante* $''$.

La expresión dada aquí para el operador mayorante coincide con una caracterización del llamado en ([2],[3]) *operador derivación ponderado por una negación*. Por tanto son válidos los siguientes resultados que aparecen en la teoría de conceptos difusos ([2],[3]) y que proporcionan, por una parte, otras expresiones del operador mayorante y, por otra, su carácter antitono en el conjunto ordenado $([0,1]^X, \tilde{\subset})$:

2.2 PROPOSICIÓN

1.- Los enunciados siguientes son equivalentes:

(i) $\tilde{A}''$ es el mayorante de $\tilde{A}$.

(ii) Para todo $y \in X$ se verifica:

$$\tilde{A}''(y) = \bigwedge \{ \tilde{A}'(x) \text{ s, } \tilde{R}(x,y) / x \in X \},$$

siendo s_t la t -conorma asociada a $(\tau, ')$.

2.- Si $\tilde{A} \tilde{\subset} \tilde{B}$ entonces $\tilde{B}'' \tilde{\subset} \tilde{A}''$, es decir el operador mayorante $''$ es una aplicación antitona en $([0,1]^X, \tilde{\subset})$.

Además se prueba que tanto en el apartado (ii) de 2.2 como en la expresión puntual de la definición 2.1 de mayorante, la condición de pertenencia $x \in X$ en el segundo miembro de la igualdad puede sustituirse por $(x \in X) \& (x \in \text{sop } \tilde{A})$ o por $(x \in X) \& (\tilde{R}(x,y) \neq 1)$.

De la proposición 2.2 se deduce que, para toda familia $(\tilde{A}_i)_{i \in J}$ de subconjuntos difusos, se verifican la inclusiones difusas:

$$\tilde{U}\{(\tilde{A}_i)'' / i \in J\} \tilde{\subset} (\tilde{\cap}\{\tilde{A}_i / i \in J\})'',$$

$$(\tilde{U}\{\tilde{A}_i / i \in J\})'' \tilde{\subset} \tilde{\cap}\{(\tilde{A}_i)'' / i \in J\}.$$

Para la segunda de éstas, la igualdad se verifica si se consideran ciertas t -normas entre las que se encuentra el operador $\wedge$ en $[0,1]$:

2.3 PROPOSICIÓN

Si la t -norma τ es tal que $(\forall M) \tau \alpha = \vee (M \tau \alpha)$ para todo $M \subseteq [0,1]$ y para todo $\alpha \in [0,1]$, entonces se verifica la igualdad entre los subconjuntos difusos:

$$(\tilde{U}\{\tilde{A}_i / i \in J\})'' = \tilde{\cap}\{(\tilde{A}_i)'' / i \in J\}.$$

Demostración.

Para cada $(x,y) \in [0,1]^2$ escribimos

$M_x = \{\tilde{A}_i(x) / i \in J\}$ y $\alpha_{xy} = \tilde{R}'(x,y)$. Obtenemos:

$$\begin{aligned} (\tilde{U}\{\tilde{A}_i / i \in J\})''(y) &= \\ &= (\vee \{ ((\tilde{U}\{\tilde{A}_i / i \in J\})(x) \tau \tilde{R}'(x,y)) / x \in X \})' = \\ &= (\vee \{ ((\vee \{\tilde{A}_i(x) / i \in J\}) \tau \tilde{R}'(x,y)) / x \in X \})' = \\ &= (\vee \{ ((\vee M_x) \tau \alpha_{xy}) / x \in X \})' = \\ &= (\vee \{ ((\vee M_x \tau \alpha_{xy})) / x \in X \})' = \\ &= (\vee \{ ((\vee \{\tilde{A}_i(x) \tau \tilde{R}'(x,y)) / i \in J\}) / x \in X \})' = \\ &= (\vee \{ ((\vee \{\tilde{A}_i(x) \tau \tilde{R}'(x,y)) / x \in X\}) / i \in J \})' = \\ &= \bigwedge \{ (\vee \{\tilde{A}_i(x) \tau \tilde{R}'(x,y)) / x \in X \})' / i \in J \} = \\ &= \bigwedge \{ (\tilde{A}_i)''(y) / i \in J \} = (\tilde{\cap}\{(\tilde{A}_i)'' / i \in J\})(y), \end{aligned}$$

que demuestra el enunciado. $\square$

Nótese que para cada punto difuso $\tilde{x}_\alpha$:

$$(\forall y \in X) ((\tilde{x}_\alpha)''(y) = \alpha' \text{ s, } \tilde{R}(x,y)).$$

Demostraremos que, con respecto al subconjunto vacío $\emptyset$, los operadores $''$ y $'$ aquí definidos se comportan

como los operadores mayorante $''$ y minorante $'$ usuales asociados al orden de un conjunto ordenado $(X, \leq_X)$:

2.4 PROPOSICIÓN

Para toda negación, toda t -norma y toda relación difusa $\tilde{R}$ se verifican las igualdades

$$\tilde{0}'' = \tilde{0}' = \tilde{1},$$

Demostración.

Basta demostrarlo para el operador $''$. Sea y un elemento genérico de $[0,1]$. De la igualdad

$$\tilde{0}''(y) = (\vee \{0 \tau \tilde{R}'(x,y) / x \in X\})' = (\vee \{0\})' = 0' = 1,$$

se deduce $\tilde{0}'' = \tilde{1}$. $\square$

En la proposición siguiente demostramos en primer lugar que si $\tilde{A}$ es nítido, $\tilde{A}''$ no depende de la t -norma τ ni de la negación $'$; y en segundo lugar que, en el caso particular en que la relación binaria considerada sea una relación nítida $\tilde{R}$, las definiciones incluidas en 2.1 tampoco dependen de la t -norma τ , (solo de $\tilde{R}$ y de la negación $'$).

2.5 PROPOSICIÓN

1.- Si $\tilde{A}$ es nítido, entonces:

$$(\forall y \in X) (\tilde{A}''(y) = \bigwedge \{ \tilde{R}(x,y) / x \in \text{sop } \tilde{A} \}).$$

2.- Si $\tilde{R}$ es una relación nítida y si $R = \text{sop } \tilde{R}$ representa su soporte, entonces para todo subconjunto difuso $\tilde{A}$ su mayorante $\tilde{A}''$ no depende de la t -norma τ considerada:

$$(\forall y \in X) (\tilde{A}''(y) = (\vee \{ \tilde{A}(x) / x R y \})'),$$

o la expresión equivalente

$$(\forall y \in X) (\tilde{A}''(y) = \bigwedge \{ \tilde{A}'(x) / x R y \}),$$

donde $R \subseteq X \times X$ es la relación binaria ordinaria determinada por el complementario R^c del soporte R .

Demostración.

1. Teniendo en cuenta los resultados del apartado 1.(ii) de la proposición 2.2, y que $\tilde{A}(x) \in \{0,1\}$ para todo x y finalmente que $0 \text{ s, } \alpha = \alpha$ para todo α ,

$$\begin{aligned} \tilde{A}''(y) &= \bigwedge \{ \tilde{A}'(x) \text{ s, } \tilde{R}(x,y) / x \in \text{sop } \tilde{A} \} = \\ &= \bigwedge \{ 0 \text{ s, } \tilde{R}(x,y) / x \in \text{sop } \tilde{A} \} = \bigwedge \{ \tilde{R}(x,y) / x \in \text{sop } \tilde{A} \}. \end{aligned}$$

2. Teniendo en cuenta que $\tilde{R}(X \times X) \subseteq \{0,1\}$, que para toda t -norma τ y para todo elemento α se verifica $\alpha \tau 1 = \alpha$ y finalmente el apartado 1 de la proposición 2.2, se obtiene:

$$\begin{aligned} \tilde{A}''(y) &= (\vee \{ \tilde{A}(x) \tau \tilde{R}'(x,y) / (x \in X) \& (\tilde{R}(x,y) \neq 1) \})' = \\ &= (\vee \{ \tilde{A}(x) / (x \in X) \& (\tilde{R}(x,y) = 0) \})' = \\ &= (\vee \{ \tilde{A}(x) / x R y \})'. \quad \square \end{aligned}$$

La demostración de la segunda igualdad es análoga a la anterior. $\square$

3.4 DEFINICIONES

Llamaremos τ_1 -supremo de $\tilde{A}$ y lo representaremos por $(\tau_1 - SUP)\tilde{A}$, al subconjunto difuso $(\tau_1 - MIN)\tilde{A}^u$ es decir, si $\tilde{A}^u$ representa el subconjunto difuso $(\tilde{A}^u)^l$:

$$(\tau_1 - SUP)\tilde{A} = (\tau_1 - MIN)\tilde{A}^u = \tilde{A}^u \tau_1 \tilde{A}^u.$$

Y si $\tilde{A}^u$ representa el subconjunto difuso $(\tilde{A}^l)^u$ llamamos τ_1 -ínfimo del subconjunto difuso $\tilde{A}$ al subconjunto difuso $(\tau_1 - MAX)\tilde{A}^l$:

$$(\tau_1 - INF)\tilde{A} = (\tau_1 - MAX)\tilde{A}^l = \tilde{A}^l \tau_1 \tilde{A}^l.$$

Como antes, si la t -norma τ_1 coincide con el operador mínimo $\wedge$, hablaremos solo de *supremo* e *ínfimo*:

$$SUP\tilde{A} = \tilde{A}^u \tilde{\sim} \tilde{A}^u, \quad INF\tilde{A} = \tilde{A}^l \tilde{\sim} \tilde{A}^l.$$

Nótese que, para toda t -norma τ_1 y para todo $\tilde{A}$, se verifica

$$(\tau_1 - SUP)\tilde{A} \tilde{\sim} SUP\tilde{A} \text{ y } (\tau_1 - INF)\tilde{A} \tilde{\sim} INF\tilde{A},$$

y que, de forma análoga a lo que ocurre con $sup\emptyset$ y con $inf\emptyset$ en conjuntos ordinarios parcialmente ordenados y acotados, para toda t -norma τ_1 :

$$(\tau_1 - SUP)\tilde{0} = (\tau_1 - MIN)\tilde{1}, \quad (\tau_1 - INF)\tilde{0} = (\tau_1 - MAX)\tilde{1}.$$

Obtenemos así operadores *supremo* SUP e *ínfimo* INF en $[0,1]^X$.

Teniendo en cuenta éstas definiciones, la proposición 2.5 y el corolario 2.6, se concluye que las definiciones de los mismos conceptos dadas en ([1]) para el caso en el que $\tilde{R}$ sea la función característica de un orden $\leq_X$, constituyen un caso particular de las dadas aquí. Por tanto es válida la proposición demostrada en ([1]):

3.5 PROPOSICIÓN

Sea $\tilde{R}$ la función característica del orden $\leq_X$ de un retículo completo $(X, \leq_X)$. Si $\tilde{A}$ es un nítido entonces $SUP\tilde{A}$ también lo es, y su soporte $sop(SUP\tilde{A})$ es un subconjunto unitario de X tal que

$$sop(SUP\tilde{A}) = \{ \sup_X(sop\tilde{A}) \},$$

donde $\sup_X$ representa el supremo asociado al orden $\leq_X$ del retículo completo.

Por tanto, los operadores SUP e INF definidos aquí pueden considerarse extensiones a la clase $[0,1]^X$ de los análogos en conjuntos ordenados completos.

Es evidente que, al igual que ocurría con los operadores ^{u} y ^{l} , existe una dualidad entre las definiciones de los operadores MAX , MIN y entre las de los operadores

SUP , INF . Basta pues estudiar en cada caso las propiedades de uno solo de ellos.

4 EJEMPLOS

Para ilustrar estos conceptos, consideremos en $[0,1]$ las t -normas τ y τ_1 iguales al mínimo ($\tau = \tau_1 = \wedge$), la negación de Zadeh y el orden difuso $\tilde{R}$ en el referencial $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ dado por Zadeh ([6]):

Tabla 1: Ejemplo de orden difuso.

$\tilde{R}$	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1	0.8	0.2	0.6	0.6	0.4
x_2	0	1	0	0	0.6	0
x_3	0	0	1	0	0.5	0
x_4	0	0	0	1	0.6	0.4
x_5	0	0	0	0	1	0
x_6	0	0	0	0	0	1

y que representa por medio del grafo con vértices x_i y aristas etiquetadas por valores de la relación $\tilde{R}$:

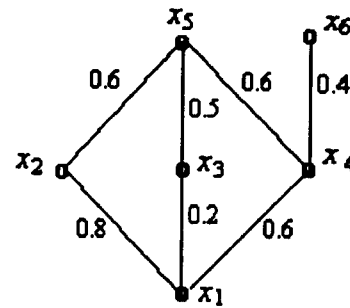


Figura 1: Representación del orden difuso $\tilde{R}$.

Teniendo en cuenta que valores que no aparecen en el grafo de la figura 1, como por ejemplo $\tilde{R}(x_1, x_5) = 0.6$, se pueden calcular a partir de los que sí lo hacen y utilizando la igualdad $\tilde{R} = \tilde{R} \circ \tilde{R}$.

Dado un subconjunto difuso genérico:

$$\tilde{A} = \alpha_1/x_1 + \alpha_2/x_2 + \alpha_3/x_3 + \alpha_4/x_4 + \alpha_5/x_5 + \alpha_6/x_6$$

donde $\alpha_i = \tilde{A}(x_i)$, podemos calcular los subconjuntos difusos asociados $\tilde{A}^u$, $\tilde{A}^l$ y posteriormente $MAX\tilde{A}$, $MIN\tilde{A}$, $SUP\tilde{A}$ e $INF\tilde{A}$ utilizando las definiciones dadas en los apartados 2.1, 3.1 y 3.4. Por ejemplo, si $\tilde{A} = 1/x_2 + 0.8/x_4$, (se omiten las expresiones $0/x_i$ para todo $x_i \notin sop\tilde{A}$), se obtiene:

$$\tilde{A}^u = 0.2/x_2 + 0.6/x_5, \quad \tilde{A}^l = 0.6/x_1 + 0.2/x_2,$$

$$\tilde{A}^{ul} = 0.6/x_1 + 0.6/x_2 + 0.5/x_3 + 0.6/x_4 + 0.8/x_5 + 0.4/x_6,$$

$$\tilde{A}^{lu} = 0.8/x_1 + 0.8/x_2 + 0.4/x_3 + 0.6/x_4 + 0.6/x_5 + 0.4/x_6,$$

$$MAX\tilde{A} = MIN\tilde{A} = 0.2/x_2,$$

$$SUP\tilde{A} = 0.2/x_2 + 0.6/x_5 \text{ e } INF\tilde{A} = 0.6/x_1 + 0.2/x_2.$$

Si $\tilde{B}$ es el nftido $\tilde{B}=1/x_2+1/x_3$, obtenemos:

$$\begin{aligned}\tilde{B}'' &= 0.5/x_5, \quad \tilde{B}' = 0.2/x_1, \\ \tilde{B}''' &= 0.6/x_1+0.6/x_2+0.5/x_3+0.6/x_4+1/x_5+0.5/x_6, \\ \tilde{B}^{iv} &= 1/x_1+0.8/x_2+0.8/x_3+0.8/x_4+0.8/x_5+0.8/x_6, \\ \text{MAX } \tilde{B} &= \text{MIN } \tilde{B} = \tilde{0}, \\ \text{SUP } \tilde{B} &= 0.5/x_5 \text{ e } \text{INF } \tilde{B} = 0.2/x_1.\end{aligned}$$

Para los nftidos $\tilde{1}$ y $\tilde{0}$ se obtiene:

$$\begin{aligned}\tilde{1}'' &= \tilde{0}, \quad \tilde{1}' = 0.2/x_1, \quad \tilde{1}''' = \tilde{1}, \\ \tilde{1}^{iv} &= 1/x_1+0.8/x_2+0.8/x_3+0.8/x_4+0.8/x_5+0.8/x_6, \\ \text{MAX } \tilde{1} &= \text{SUP } \tilde{1} = \tilde{0}, \quad \text{MIN } \tilde{1} = \text{INF } \tilde{1} = 0.2/x_1, \\ \tilde{0}'' &= \tilde{0}' = \tilde{1}, \quad \tilde{0}''' = 0.2/x_1, \quad \tilde{0}^{iv} = \tilde{0}, \\ \text{MAX } \tilde{0} &= \text{MIN } \tilde{0} = \text{INF } \tilde{0} = \tilde{0} \text{ y } \text{SUP } \tilde{0} = 0.2/x_1.\end{aligned}$$

5 CONCLUSIONES

Se han dado aquí extensiones de ciertos conceptos clásicos ligados al orden usual en conjuntos ordinarios a la clase de subconjuntos difusos, utilizando tanto subconjuntos difusos como preórdenes difusos. Estas definiciones están inspiradas en propiedades que cumplen los conceptos análogos usuales en el caso ordinario y que, en función de los resultados que se obtienen y de los ejemplos, parecen definiciones aceptables. Podemos así caracterizar estructuras $(X, \tilde{R})$ de forma análoga a como se hace en el caso ordinario hablando de acotación, completitud, etc., ahora desde una perspectiva difusa.

Estos resultados, que como se ha señalado en el apartado 1 son válidos en su mayoría para relaciones difusas no necesariamente preórdenes, pueden extenderse a otros contextos L -difusos: por ejemplo sustituyendo el conjunto $[0,1]$ de llegada de las aplicaciones $\tilde{A}$ por retículos completos L con una negación y con t -normas apropiadas. Podría ser L tanto finito del tipo $\{0, 1/n, \dots, (n-1)/n, 1\}$, como un retículo producto $[0,1]^n$ o un retículo de subintervalos $[\alpha, \beta]$ del intervalo $[0,1]$. En todos estos casos con t -normas, negación y relaciones de orden inducidas a partir de las consideradas en $([0,1], \leq)$.

En otra línea, estos operadores $''$, MAX , SUP , etc. también se pueden extender fácilmente a sistemas $(W \times X, \tilde{R})$ donde la relación $\tilde{R}$ sigue perteneciendo a $[0,1]^{X \times X}$ pero considerando ahora relaciones difusas $\tilde{Q} \in [0,1]^{W \times X}$ en lugar de subconjuntos difusos $\tilde{A}$ de X .

Como continuación de este trabajo, estamos realizando el estudio de propiedades de los operadores aquí definidos, así como el análisis de ciertos subconjuntos difusos

$\tilde{A}$ y $\tilde{B}$ distinguidos con respecto a estos operadores; concretamente los que verifican $\text{MAX } \tilde{A} = \text{SUP } \tilde{A} \neq \tilde{0}$ o $\text{MIN } \tilde{A} = \text{INF } \tilde{A} \neq \tilde{0}$ así como los que son puntos fijos del tipo: $\tilde{A} = \tilde{A}'''$ y $\tilde{B} = \tilde{B}^{iv}$. Basamos este análisis en la teoría de conceptos difusos desarrollada en ([2], [3]).

Por otra parte estamos estudiando las posibles aplicaciones de las ideas desarrolladas aquí a problemas de cálculo de extremos condicionados de funciones del tipo $y = f(x)$ con valores y recorrido en un referencial X , (funciones en principio ordinarias), considerándolas en un contexto difuso; es decir, expresando la relación entre elementos de X por preórdenes difusos y expresando las condiciones a satisfacer por los extremos de la función f también por medio de subconjuntos difusos.

Agradecimientos

Los autores agradecen las observaciones e indicaciones hechas por el evaluador de este trabajo.

Referencias

- [1] A. Burusco y R. Fuentes-González (1995) Sur une extension des concepts sup et inf associés à l'ordre d'un treillis complet aux sous-ensembles flous. En *Rencontres Francophones sur la Logique Floue et ses Applications. LFA95. Paris. Cépaduès Éditions*. 207-311.
- [2] A. Burusco y R. Fuentes-González (1994) The Study of the L-Fuzzy Concept Lattice. *Mathware* vol. 1, n.3 : 209-218.
- [3] A. Burusco y R. Fuentes-González (1995) Retículos de conceptos asociados a W -contextos L -Fuzzy. En *ESTYLF'95. Murcia* : 111-115.
- [4] B.A.Davey y H.A. Priestley (1990). *Introduction to lattices and order*. Cambridge University Press.
- [5] G. Gierz, K.H.Hofmann, K.Keimel, J.D.Lawson, M. Mislove y D. Scott (1980). *A Compendium of Continuous Lattices*. Springer-Verlag.
- [6] L.A. Zadeh (1971) Similarity Relations and Fuzzy Orderings. *Inform. Sci.* vol. 3 : 177-200.

MÉTODO DE LOS MÍNIMOS CUADRADOS APLICADO A LA INFERENCIA FUZZY

H. Bustince

Dep. Automática y Computación
Universidad Pública de Navarra
Campus de Arrosadía s/n
31006 Pamplona, España
e-mail: bustince@si.upna.es

M. Calderón

Dep. Automática y Computación
Universidad Pública de Navarra
Campus de Arrosadía s/n
31006 Pamplona, España
e-mail: fcalderon@interplanet.es

V. Mohedano

Dep. Automática y Computación
Universidad Pública de Navarra
Campus de Arrosadía s/n
31006 Pamplona, España
e-mail: vmohedano@si.upna.es

Resumen

En este trabajo, el método de los mínimos cuadrados es aplicado a las reglas de inferencia fuzzy. Empezamos estudiando las condiciones bajo las cuales un conjunto difuso puede ser construido a partir de otro, también difuso, por medio del método de los mínimos cuadrados. Posteriormente, aplicamos esta técnica para evaluar las conclusiones del modus ponens generalizado y del modus tollens generalizado. Presentamos diferentes teoremas que demuestran las principales ventajas del método estudiado.

Palabras Clave: Razonamiento aproximado, reglas de inferencia difusas, modus ponens generalizado, modus tollens generalizado, modelo de los mínimos cuadrados, lógica difusa, función de potencia.

1 INTRODUCCIÓN

El modus ponens clásico viene expresado por

$$\begin{array}{c} A \rightarrow B \\ A \\ \hline B \end{array}$$

esto significa que si:

*A implica B y A es verdad, entonces
B también es verdad.*

Este esquema de razonamiento fue extendido al razonamiento borroso por Zadeh [11-14] como sigue:

A) La implicación $A \rightarrow B$ es reemplazada por la regla de inferencia borrosa

si x es A entonces y es B

donde A es un conjunto fuzzy sobre un universo de discurso X y B también borroso en un universo de discurso Y ; x es una variable que toma valores en X , e y una variable que toma valores en Y . La regla borrosa representa la relación entre las variables x e y .

B) De igual forma, la premisa A es reemplazada por la premisa fuzzy:

x es A'

expresando el conocimiento que tenemos sobre el valor de x . A' es un conjunto fuzzy en un universo de discurso X .

Combinando la regla y la premisa, es posible deducir nueva información, escribiendo

y es B'

donde B' es un conjunto borroso sobre un universo de discurso Y .

Las funciones de pertenencia de A y A' las escribiremos como $\mu_A(x)$ y $\mu_{A'}(x)$ respectivamente; las funciones de pertenencia de B y B' las denotaremos con $\mu_B(y)$ y $\mu_{B'}(y)$ respectivamente.

Otro método de inferencia clásico es el modus tollens que viene expresado como sigue:

$$\begin{array}{r} A \rightarrow B \\ \text{no } B \\ \hline \text{no } A \end{array}$$

A partir del modus ponens y del modus tollens clásico y utilizando la teoría de conjuntos difusos podemos realizar representaciones de las imprecisiones inherentes al lenguaje humano. De este modo, el modus ponens generalizado (MPG) y el modus tollens generalizado (MTG) fueron introducidos para obtener una conclusión cuando partimos de premisas fuzzy. Estas reglas pueden ser expresadas como sigue

$$\begin{array}{r} \text{si } x \text{ es } A \text{ entonces } y \text{ es } B \\ x \text{ es } A' \\ \hline y \text{ es } B' \end{array} \quad (\text{MPG})$$

y

$$\begin{array}{r} \text{si } x \text{ es } A \text{ entonces } y \text{ es } B \\ y \text{ es } B' \\ \hline x \text{ es } A' \end{array} \quad (\text{MTG})$$

La principal ventaja de esta extensión al razonamiento borroso es que podemos deducir nueva información, aún cuando la pertenencia no es exactamente idéntica a la condición de la regla o cuando la información que consideramos no es exacta. Es conocido que en (MPG) cuando $A = A'$, y en (MTG) cuando $B' = \text{no } B$, entonces el modus ponens generalizado y el modus tollens generalizado se reducen a los casos del modus ponens y el modus tollens clásicos respectivamente.

Diferentes métodos han sido sugeridos por autores como Zadeh [11,14], Fukami [4], Mizumoto y Zimmermann [7], Ezawa y Kandel [3], para el estudio de las reglas de inferencia fuzzy.

En este trabajo vamos a estudiar el modus ponens generalizado y el modus tollens generalizado siguiendo un razonamiento diferente a cada uno de los empleados por los autores anteriormente mencionados. Las bases de nuestro trabajo serán la ideas dadas por J.F. Baldwin [1],

A. Nafarieh [8], M.B. Gorzalczyk [5], etc., las cuales pueden ser resumidas en los dos pasos siguientes:

- 1) Relacionar primero A y A'
- 2) Construir la consecuencia B' usando el resultado de la comparación anterior y B .

Llevaremos a cabo el punto 1) siguiendo el método presentado por T.D. Pham y S. Valliappan [9], analizando y posteriormente formalizando sus principales consecuencias.

Empezamos el trabajo estudiando la forma de obtener a partir de un conjunto borroso otro borroso, por medio del método de los mínimos cuadrados. Nos centramos sobre todo en el estudio de las condiciones bajo las cuales dicha construcción puede ser llevada a cabo. Concluimos esta sección analizando las principales consecuencias del método de construcción presentado. Debemos dejar claro que en este trabajo no consideramos si la aproximación que hacemos es apropiada o no, esto es, no estudiamos el coeficiente de Pearson, etc., como se verá en la siguiente sección. Estamos interesados única y exclusivamente en los coeficientes calculados por la aproximación llevada a cabo.

En la sección siguiente presentamos un método para la obtención de la consecuencia del MPG y otro para la del MTG. En ambos casos estudiamos las principales características de dichos métodos, resaltando, su gran aplicabilidad cuando las funciones de pertenencia en las premisas son dadas por los operadores lingüísticos borrosos definidos por Zadeh [11-14].

Como T. C. Chang, K. Hasegawa y C. W. Ibbes [2], por conveniencia, tomamos los universos de discurso como sigue:

$$1) X = \{x_1, \dots, x_n\}, Y = \{y_1, \dots, y_m\} \text{ y } Z = \{z_1, \dots, z_p\} \\ x_i, y_j, z_k \in (0,1] \text{ para todo } i=1, \dots, n, j=1, \dots, m, \\ k=1, \dots, p$$

$$2) x_i < x_{i+1}, y_j < y_{j+1}, \text{ y } z_k < z_{k+1} \text{ para todo } i, j, k.$$

Designaremos con $FS(X)$ a todos los conjuntos fuzzy sobre X , denotaremos con $FS^*(X)$ a todos los conjuntos borrosos sobre X tales que para todo $x_i \in X$ se verifica que $\mu(x_i) \neq 0$. Designaremos con A_c al conjunto complementario del conjunto borroso A y para $p \in \mathbb{R}^+ \cup \{0\}$, escribiremos $A^p = \{ \langle x_i, \mu_A^p(x_i) \rangle \mid x_i \in X \}$, evidentemente, p tiene que ser mayor o igual que cero.

2 CONSTRUCCIÓN DE UN FS(X) A PARTIR DE UN FS*(X)

Nuestro objetivo es aproximar las funciones de pertenencia de un conjunto borroso $A \in FS^*(X)$ a funciones del tipo ax_i^b , donde para cualquier conjunto $A \in FS^*(X)$ las constantes "a" y "b" se obtienen por medio de las ecuaciones lineales:

$$(1) \begin{cases} n \ln a + \left(\sum_{i=1}^n \ln x_i \right) b = \sum_{i=1}^n \ln \mu_A(x_i) \\ \left(\sum_{i=1}^n \ln x_i \right) \ln a + \left(\sum_{i=1}^n \ln^2 x_i \right) b = \sum_{i=1}^n \ln x_i \ln \mu_A(x_i) \end{cases}$$

obtenidas al tomar logaritmos en la expresión $\mu_A(x_i) = ax_i^b$ y aplicar el método de los mínimos cuadrados.

Dada la naturaleza del problema con el que estamos tratando, es evidente que la constante "a" tiene que ser mayor o igual que cero y del mismo modo debe cumplirse siempre que, $ax_i^b \leq 1$, para todo $i=1, \dots, n$.

El siguiente teorema establece las condiciones bajo las cuales la condición anteriormente mencionada se cumple.

Teorema 1. Sea $A \in FS^*(X)$ y dados "a" y "b" soluciones de (1).

$ax_i^b \leq 1$ para todo $i=1, \dots, n$, si y sólo si

$$\begin{cases} 1) \text{ si } b \geq 0, \text{ entonces } \left| \sum_{i=1}^n \ln \mu_A(x_i) \right| \geq (n \ln x_{\max} - \sum_{i=1}^n \ln x_i) b \\ 2) \text{ si } b < 0, \text{ entonces } \left| \sum_{i=1}^n \ln \mu_A(x_i) \right| \geq (n \ln x_{\min} - \sum_{i=1}^n \ln x_i) b. \end{cases}$$

Corolario 1. En las mismas condiciones que en el teorema 1, el conjunto

$$\tilde{A} = \{ \langle x_i, \mu_{\tilde{A}}(x_i) \rangle \mid x_i \in X \} \text{ con } \mu_{\tilde{A}}(x_i) = ax_i^b$$

es un conjunto fuzzy sobre X.

Observación

Hasta ahora hemos considerado conjuntos borrosos tales que para todo $x_i \in X$ se verifica que $\mu(x_i) \neq 0$. Sin embargo, la construcción analizada anteriormente puede ser

aplicada a cualquier conjunto fuzzy como se indica a continuación.

Si en el conjunto $A \in FS(X)$ considerado existe un x_i cuya función de pertenencia es nula, $\tilde{A}$ lo construimos siguiendo los dos pasos siguientes:

1) Con el método anterior construimos las nuevas pertenencias de los elementos cuya pertenencia sea distinta de cero.

2) Para los elementos con pertenencia nula, esta es mantenida.

Por lo tanto, de ahora en adelante y hasta que no se diga lo contrario, tomaremos $A, A' \in FS(X)$ y $B, B' \in FS(Y)$ sin considerar la existencia de elementos con pertenencia nula, ya que los ignoraremos cuando construyamos $\tilde{A}$, $\tilde{B}$, $\tilde{A}'$ y $\tilde{B}'$.

Teorema 2. Sea $\tilde{A}$ el conjunto borroso construido a partir de $A \in FS(X)$ en las mismas condiciones que las indicadas en el corolario 1,

si $A = \text{constante}$, entonces $A = \tilde{A}$.

Teorema 3. Sea $A \in FS(X)$.

$b=0$ si y sólo si

$$\left(\sum_{i=1}^n \ln x_i \right) \cdot \left(\sum_{i=1}^n \ln \mu_A(x_i) \right) = n \cdot \left(\sum_{i=1}^n \ln x_i \ln \mu_A(x_i) \right)$$

El resultado de este teorema será muy importante en la siguiente sección, ya que, siempre que b sea igual a 0, tendremos que tener en cuenta, en los métodos que presentamos más adelante que, evidentemente algunos coeficientes no será posible calcularlos.

Teorema 4. Sean $A, A' \in FS(X)$ y sean $\tilde{A}$ y $\tilde{A}' \in FS(X)$ los conjuntos construidos a partir de A y A' respectivamente de acuerdo con el corolario 1. Dado $p \in \mathbb{R}^+ \cup \{0\}$,

si $A' = A^p$, entonces $\tilde{A}' = (\tilde{A})^p$

Una consecuencia de este teorema es que al tomar p los valores utilizados por Zadeh en [12,13] se tiene lo siguiente:

a) Si $A' = A$, entonces $\tilde{A}' = \tilde{A}$

b) Si $A' = A^2$, entonces $\tilde{A}' = \tilde{A}^2$

c) Si $A' = A^{\frac{1}{2}}$, entonces $\tilde{A}' = \tilde{A}^{\frac{1}{2}}$

d) Si $A' = A^q$, entonces $\tilde{A}' = \tilde{A}^q$

3 INFERENCIA BORROSA UTILIZANDO EL MÉTODO DE LOS MÍNIMOS CUADRADOS

Como resultado del teorema 3, el conjunto A de la condicional si x es A entonces y es B , será tal que

$$\left(\sum_{i=1}^n \ln x_i \right) \cdot \left(\sum_{i=1}^n \ln \mu_A(x_i) \right) \neq n \cdot \left(\sum_{i=1}^n \ln x_i \ln \mu_A(x_i) \right)$$

de esta forma estamos seguros de que el coeficiente b calculado aplicando (1) es distinto de cero.

Antes hemos dicho que en la lógica borrosa, el razonamiento exacto puede ser extendido al razonamiento aproximado, el cual contiene imprecisiones inherentes al lenguaje humano, de forma que, el modus ponens generalizado (MPG) y el modus tollens generalizado (MTG) son introducidos para buscar una conclusión a partir de premisas borrosas.

Zadeh [12,13] presentó los operadores *muy, mayor, menor, más, menos*, etc. expresados en términos de las funciones de pertenencia, y de esta forma le asoció a ellos valores numéricos, lo cual le permitió trabajar de forma más sencilla.

A partir de esto, T. D. Pham y S. Valliappan, presentaron en [9] un método para obtener la conclusión del MPG.

Este método consiste en los tres pasos siguientes:

MÉTODO PARA EL MPG

1. Introducir una función $\mu_{A'}(x_i)$ en términos de $\mu_A(x_i)$ y obtener los coeficientes de la relación entre las dos funciones de pertenencia. Llevaremos a cabo este paso aproximando las funciones de pertenencia del conjunto A , a funciones del tipo ax_i^b y las del conjunto A' , a funciones del tipo $a'x_i^{b'}$; esto lo realizaremos resolviendo (1) para A y A' .

Es importante resaltar que en esta sección no nos hemos parado a mirar si $\tilde{A}$ y $\tilde{A}'$ pueden ser construidos, porque no los utilizamos, simplemente hemos calculado a, a', b y b' (independientemente del teorema 1). No obstante, debemos tener en cuenta que

$$\mu_{\tilde{A}}(x_i) = ax_i^b \text{ y } \mu_{\tilde{A}'}(x_i) = a'x_i^{b'}$$

de lo cual deducimos que $\frac{1}{a^b} \mu_{\tilde{A}}^b(x_i) = x_i$ para todo $i=1, \dots, n$, y por tanto

$$\mu_{\tilde{A}'}(x_i) = \frac{a'}{b'} \mu_{\tilde{A}}^{\frac{b'}{b}}(x_i).$$

2.- Calculamos $c_1 = \frac{a'}{b'}$ (obviamente $0 < c_1$) y $c_2 = \frac{b'}{b}$,

así que $\mu_{\tilde{A}'} = c_1 \mu_{\tilde{A}}^{c_2}(x_i)$.

3.- Estos coeficientes de relación son utilizados en $\mu_B(y_j)$ para la construcción de $\mu_{B'}(y_j)$, de la siguiente forma $\mu_{B'}(y_j) = c_1 \mu_B^{c_2}(y_j)$ para todo $j=1, \dots, m$

T. D. Pham y S. Valliappan en [9] no consideraron la posibilidad de que para un y_j con $j=1, \dots, m$, $c_1 \mu_B^{c_2}(y_j)$ pudiera ser mayor que 1. Por este motivo enunciamos el siguiente teorema.

Teorema 5. En las condiciones de los tres puntos anteriores tenemos:

$0 \leq c_1 \mu_B^{c_2}(y_j) \leq 1$ para todo $j=1, \dots, m$ si y solo si

$$\begin{cases} 1) c_2 \geq 0 \text{ y } c_1 \mu_{B_{\max}}^{c_2}(y_j) \leq 1 \\ 2) c_2 < 0 \text{ y } c_1 \mu_{B_{\min}}^{c_2}(y_j) \leq 1. \end{cases}$$

nota : en todo lo que queda de trabajo vamos a utilizar la siguiente notación.

$$\mu_{B_{\max}}(y_j) = \bigvee_j (\mu_B(y_j)) \text{ y } \mu_{B_{\min}}(y_j) = \bigwedge_j (\mu_B(y_j)).$$

Teorema 6. En las condiciones del teorema anterior, se cumple

si $A' = \text{constante}$, entonces $B' = \{ \langle y, a' \rangle \mid y_j \in Y \}$

Teorema 7. Sea $p \in \mathbb{R}^+ \cup \{0\}$. En las condiciones del teorema 5, se cumple que

si $A' = A^p$, entonces $B' = B^p$

En el caso de que este método puede ser aplicado, es decir, si se cumplen las condiciones del teorema 5, nos

encontramos ante un mecanismo de inferencia que tiene las dos ventajas siguientes:

- 1) Puede ser aplicado cuando las funciones de pertenencia en las premisas están caracterizadas por los operadores lingüísticos borrosos como fueron definidos por Zadeh [11,14].
- 2) Las operaciones matemáticas para el modus ponens generalizado son simples y apropiadas para la implementación.

MÉTODO PARA EL MTG

1.- De (1), (aplicado al conjunto B) calculamos g y h .

2.- Por (1), (aplicado a B') calculamos g' y h' .

3.- Aplicamos el método de los mínimos cuadrados al conjunto

$$\{ \langle y_j, 1 - g h_j^{h'} \rangle \mid y_j \in Y \} \quad j = 1, \dots, m$$

así resolviendo el correspondiente sistema (1) calculamos g'' y h''

4.- Calculamos $c_3 = \frac{g''}{g}$ y $c_4 = \frac{h''}{h}$.

5.- Construimos A' tomando

$$\mu_{A'}(x_i) = 1 - c_3 \mu_A^{c_4}(x_i), \text{ para todo } i = 1, \dots, n.$$

Teorema 8. Sea $A \in FS(X)$ y $B, B' \in FS^*(Y)$. El conjunto A' obtenido siguiendo los pasos anteriores satisface:

- i) Si $B' = Bc$, entonces $A' = Ac$.
- ii) Si $B' = B^2c$, entonces $A' = A^2c$.
- iii) Si $B' = B_c^{\frac{1}{2}}$, entonces $A' = A_c^{\frac{1}{2}}$.
- iv) Si $B' = B^4c$, entonces $A' = B^4c$.

4 CONCLUSIONES

En este trabajo queda claro que el método presentado por T. D. Pham y S. Valliappan da buenos resultados cuando se aplica a las variables lingüísticas tal y como las introdujo Zadeh en [11,14]. Sin embargo, hay que tener en cuenta que dicho método no siempre es posible aplicarlo, ya que puede ocurrir que el conjunto fuzzy obtenido tome valores para la función de pertenencia no válidos. Probamos que cuando se cumplen las condiciones del teorema 5 siempre es posible aplicar este método.

Presentamos también el algoritmo para la obtención del modus tollens generalizado cuando aplicamos el método de los mínimos cuadrados, viendo que también en este caso se conservan los valores esperados de las variables lingüísticas.

En los dos métodos que presentamos para la obtención de la conclusión, es destacable, que la implementación computacional es muy simple en ambos casos.

Referencias

- [1] J. F. Baldwin (1980). Feasible algorithms for approximate reasoning using fuzzy logic. *Fuzzy Sets and Systems* 3: 225-251.
- [2] T. C. Chang, K. Hasegawa y C. W. Ibbs (1991). The effects of memberships function on fuzzy reasoning. *Fuzzy sets and systems* 44: 169-186.
- [3] Y. Ezawa y A. Kandel (1991). Robust fuzzy inference. *Int. J. Intelligent Systems* 6: 185-197.
- [4] S. Fukami, M. Mizumoto y K. Tanaka (1980). Some considerations on fuzzy conditional inference. *Fuzzy Sets and Systems* 4: 243-273.
- [5] M. B. Gorzalczy (1987). A method of inference in approximate reasoning based on interval-valued fuzzy sets. *Fuzzy Sets and Systems* 21: 1-17
- [6] M. Mizumoto (1985). Extended fuzzy reasoning in: Gupta, Kandel, Blander and Kiszka. *Reasoning in Expert Systems* : 71-85.
- [7] M. Mizumoto y H.-J. Zimmermann (1982). Comparison of fuzzy reasoning methods. *Fuzzy Sets and Systems* 8: 253-283.
- [8] A. Nafarieh (1988). A new approach to inference in approximate reasoning and its application to computer vision. *Ph D. Dissertation, University of Columbia*.
- [9] T. D. Pham y S. Valliappan (1994). A least squares model for fuzzy rules of inference. *Fuzzy Sets and Systems* 64 : 207-212.
- [10] K. S. Ray, J. Ghoshal (1996). Approximate reasoning to pattern recognition. *Fuzzy Sets and Systems* 77 : 125-150.
- [11] L. A. Zadeh (1988). Fuzzy logic. *IEEE Computer*: 83-93.
- [12] L. A. Zadeh (1975). The concept of a linguistic variable, and its application to the approximate reasoning. Part III, *Information Sciences* 8 : 43-80
- [13] L. A. Zadeh (1975). The concept of a linguistic variable, and its application to the approximate reasoning. Part II, *Information Sciences* 8 : 301-357.
- [14] L.A. Zadeh (1972). A fuzzy-sets theoretic interpretation of linguistic hedges. *Cybernetics* 2 : 4-34.

APRENDIZAJE DE HIPERGRAFOS POSIBILISTICOS.*

Luis M. de Campos

Dpto. Ciencias de la Computación e I.A.
Universidad de Granada
18071-Granada, Spain
E-mail: lci@robinson.ugr.es

Juan F. Huete

Dpto. Ciencias de la Computación e I.A.
Universidad de Granada
18071-Granada, Spain
jhg@robinson.ugr.es

Resumen

Un Hipergrafo es una estructura gráfica que nos permite representar distintas componentes de información sobre un determinado problema. En un Hipergrafo posibilístico, a cada componente de información le asociamos una distribución de posibilidad conjunta. En particular, el trabajo se centra en el estudio de algoritmos que nos permitan recuperar un Hipergrafo utilizando las relaciones de independencia existentes entre las variables.

Palabras Clave: Hipergrafos Posibilísticos, Independencia Posibilística, Aprendizaje.

1 Introducción.

El uso de modelos gráficos es de gran utilidad a la hora de tener una representación eficiente del conocimiento incierto. En general, mediante la estructura del modelo se representan relaciones de dependencia / independencia entre variables, junto con una representación cuantitativa de la información. Un aspecto importante, cuando trabajamos con incertidumbre, es la selección de un formalismo adecuado para representar cuantitativamente la información. En este sentido, podemos encontrar distintos formalismos como la Teoría de la Probabilidad, Teoría de la Evidencia [16] o la Teoría de la Posibilidad [7, 18]. En este trabajo nos centramos en el estudio de sistemas posibilísticos. Estos sistemas permiten diseñar técnicas de razonamiento aproximado, esto es, con un comportamiento poco sensible a pequeños cambios de la información. Basándonos en este hecho, se puede pensar que la teoría de la posibilidad puede representar el mismo papel dentro del

Este trabajo ha sido financiado por la DGICYT bajo el proyecto n. PB92-0939

razonamiento con incertidumbre que el control difuso en el marco de la ingeniería de control [14].

Cuando trabajamos con incertidumbre, podemos encontrar distintos modelos gráficos. En especial, podemos destacar (1) los modelos causales [3, 14, 15] donde cada variable es un nodo del modelo y donde la ausencia de una conexión directa representa una relación de independencia entre las variables y (2) los modelos basados en Hipergrafos [4, 8, 17] donde las variables se agrupan en nodos que representan relaciones de dependencia. Nuestro interés se centra en el estudio de algoritmos que nos permitan realizar el aprendizaje de estructuras gráficas en entornos posibilísticos. Por ejemplo, Kruse et al. [9] proporciona un algoritmo de aprendizaje para modelos causales posibilísticos. En este trabajo, presentamos un algoritmo de aprendizaje para hipergrafos posibilísticos basado en el estudio de relaciones de independencia entre variables.

En la siguiente sección introducimos brevemente la teoría de la posibilidad. La sección 3 presenta el Hipergrafo posibilístico como una herramienta que permite construir un sistema basado en el conocimiento. En la sección 4 vemos como un Hipergrafo puede verse como la descomposición de una distribución de posibilidad conjunta en términos de conjuntos de variables condicionalmente independientes y, utilizando esta propiedad, presentamos un algoritmo que permite recuperar un Hipergrafo a partir de un conjunto de datos. Para finalizar, se presentan las conclusiones y líneas de trabajo futuras.

2 Teoría de la Posibilidad.

Cuando partimos de un conocimiento impreciso, una medida de posibilidad es una herramienta natural para formalizar la incertidumbre. En este trabajo consideraremos que una distribución de posibilidad establece, esencialmente, un orden entre los valores que una determinada variable puede tomar. Formalmente,

si D es el conjunto de referencia, entonces una medida de posibilidad puede definirse como una función $\Pi : \mathcal{P}(D) \rightarrow [0, 1]$ verificando

$$\forall A, B \subseteq D \quad \Pi(A \cup B) = \max\{\Pi(A), \Pi(B)\} \quad (1)$$

donde $\Pi(\emptyset) = 0$ y $\Pi(D) = 1$. Cuando el conjunto D es finito, una medida de posibilidad Π puede obtenerse a través de los valores en los elementos atómicos de D

$$\forall A, \Pi(A) = \max\{\pi(w) \mid w \in A\}$$

donde $\pi(w) = \Pi(\{w\})$ y π es una función de D en $[0, 1]$ llamada *distribución de posibilidad*. Como condición de normalización, se exige que exista un elemento w tal que $\pi(w) = 1$ indicando que existe al menos un valor en D que es completamente posible. Cuando tenemos una variable que toma valores en D , $\pi(w)$ se interpreta como el grado en que $w \in D$ puede ser un valor posible de la variable y por tanto $\Pi(A)$ puede considerarse como la posibilidad de que la variable tome como valor cualquier elemento de A .

Sean X e Y dos variables y sea π una distribución de posibilidad definida sobre el producto cartesiano de sus conjuntos de referencia, $D_X \times D_Y$. Considerando la semántica asociada a la distribución de posibilidad, utilizaremos como operador de condicionamiento el dado por Hisdal [6, 11]. Esto es, la distribución de posibilidad de X , condicionada al suceso $[Y = y]$, y lo notamos por $\pi_h(\cdot \mid y)$, viene dada por

$$\pi_h(x \mid y) = \begin{cases} \pi(x, y) & \text{si } \pi(x, y) < \pi_Y(y). \\ 1 & \text{si } \pi(x, y) = \pi_Y(y). \end{cases}$$

donde π_Y representa la distribución de posibilidad marginal para la variable Y , es decir $\pi_Y(y) = \max_{x \in D_X} \pi(x, y)$.

Otro concepto importante a considerar es el de independencia posibilitista. En este caso, utilizamos un criterio de independencia basado en la imprecisión. Es decir, diremos que la variable X es condicionalmente independiente de la variable Y dado que conocemos el valor de Z , y lo notamos por $I(X \mid Z \mid Y)$, cuando al condicionar obtenemos una información coherente con la original, pero menos precisa. Formalmente podemos decir que

$$I(X \mid Z \mid Y) \Leftrightarrow \pi_h(x \mid yz) \geq \pi_h(x \mid z), \quad \forall x, y, z.$$

Notemos que esta definición de independencia es equivalente [1, 12] a la no interactividad posibilitista [18], es decir, equivale a $\pi(xyz) = \pi(xz) \wedge \pi(yz)$, $\forall x, y, z$.

3 Hipergrafos Posibilitistas.

En teoría de grafos [10], el concepto de Hipergrafo (ver Figura 1) ha sido ampliamente estudiado. Formalmente, dado un conjunto de variables de referencia

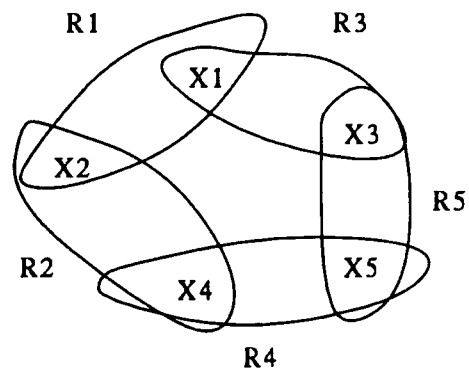


Figura 1: Hipergrafo

$S = \{X_1, \dots, X_n\}$, un Hipergrafo $\mathcal{H}$ sobre S se puede definir como un conjunto no vacío de subconjuntos de S . A los elementos de $\mathcal{H}$ se les llama hiperlados, y a los elementos de S se les denomina vértices. Un Hipergrafo se puede considerar como una estructura gráfica que permite representar distintas componentes de información sobre un determinado problema. En este caso, cada componente (hiperlado) agrupará a un conjunto de variables.

Cuando el conocimiento de partida es impreciso o vago, la teoría de la posibilidad se presenta como un formalismo razonable para representar cuantitativamente la información. En particular, para cada componente de información, R_j , tendremos una distribución de posibilidad (relación difusa) sobre el conjunto de variables que la componen, π_{R_j} . Por tanto, un Hipergrafo Posibilitista [4, 8] se puede considerar como un hipergrafo, donde asociado a cada hiperlado tenemos una distribución de posibilidad marginal.

Este tipo de estructuras permiten realizar razonamiento posibilitista. Esto es, a la luz de nueva información, obtener un conjunto de conclusiones lo más precisas posibles. A la hora de realizar inferencias, son dos las operaciones básicas [8]: la primera es la combinación de información (mediante el operador mínimo), que permite obtener la distribución de posibilidad menos restrictiva compatible con las restricciones y la segunda es una proyección (mediante el operador máximo) de los resultados sobre las variables de interés.

Hemos de notar que en un entorno posibilitista, se satisfacen los axiomas de Shafer y Shenoy [17] y por tanto es posible utilizar técnicas locales para realizar una propagación de la información en árboles de Markov: Un árbol de variables donde dos variables están separadas si existe una relación de independencia condicional, dado el conjunto de variables que las separan. Dubois y Prade [8] proporcionan un conjunto de técnicas que permiten transformar un Hipergrafo posibilitista en

un Hiperárbol, cuyos hiperlados se pueden organizar como un árbol de Markov.

4 Aprendizaje de Hipergrafos Posibilísticos a partir de Datos.

A la hora de realizar tareas de razonamiento en un hipergrafo, es necesario tener como punto de partida la estructura gráfica ya construida. Sin embargo, para muchos problemas, el proceso de elicitación de la estructura a partir de un experto puede ser una tarea bastante costosa. En esta sección nos centraremos en el estudio de algoritmos que permitan, a partir de una base de casos, obtener este tipo de estructuras, junto con los valores cuantitativos asociados a las mismas. Este tipo de algoritmos puede ser de gran ayuda en el proceso de diseño de un hipergrafo.

En concreto, partimos de una base de casos sobre un conjunto de variables $S = \{X_1, X_2, \dots, X_n\}$, de la que obtenemos la distribución de posibilidad que siguen los datos, π . Nuestro objetivo será construir el Hipergrafo que mejor represente esta distribución. Para ello consideramos que en un hipergrafo, además de representarse relaciones de dependencia entre variables, también se representan relaciones de independencia condicionales.

En este proceso, tratamos de factorizar la distribución de posibilidad original en componentes condicionalmente independientes. Así, supongamos que π es una distribución de posibilidad sobre S y que existe un conjunto de variables $C = \{X_i, \dots, X_j\}$ verificando la relación $I(X_1, \dots, X_{i-1} | C | X_{j+1}, \dots, X_n)$. Entonces, podemos factorizar la distribución original en dos distribuciones, π_1 sobre $S_1 = \{X_1, \dots, X_{i-1}, X_i, \dots, X_j\}$ y π_2 sobre $S_2 = \{X_i, \dots, X_j, X_{j+1}, \dots, X_n\}$. Basándonos en el criterio de independencia utilizado, tenemos que la distribución original se puede recuperar mediante $\pi = \pi_1 \wedge \pi_2$. Aplicando este esquema de forma recursiva para los conjuntos S_1 y S_2 , obtenemos una estructura arbórea, que denominaremos árbol posibilístico.

La Figura 2 muestra un árbol posibilístico, donde en cada nivel se presentan junto con el conjunto de nodos, las relaciones de independencia que permiten construirlo. En estas estructuras, la información se localiza únicamente en las hojas del árbol, $L_j, j = 1 \dots m$, en forma de distribuciones de posibilidad marginales. La distribución de posibilidad original se puede recuperar [2] mediante

$$\pi(x_1, x_2, \dots, x_n) = \pi_{L_1} \wedge \pi_{L_2} \wedge \dots \wedge \pi_{L_m} = \bigwedge_{j=1}^m \pi_{L_j},$$

con π_{L_j} siendo la distribución de posibilidad marginal almacenada en la hoja L_j

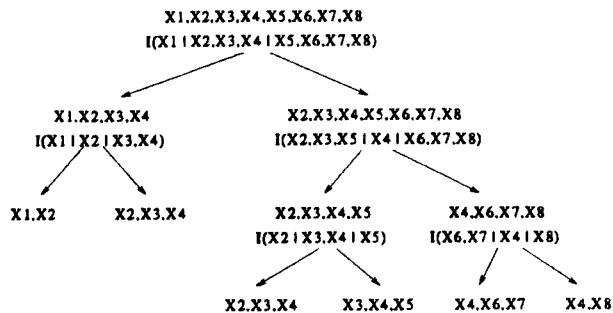


Figura 2: Árbol Posibilístico

Asociado a un árbol posibilístico, es posible construir un Hipergrafo posibilístico (Figura 3). Para ello, basta con considerar como hiperlados las hojas del árbol posibilístico. En este caso, en un Hipergrafo se representan además de las relaciones entre variables, un conjunto de relaciones de independencia condicional entre ellas. Este hecho hace que un Hipergrafo represente un papel similar a las redes bayesianas [15] en un formalismo probabilístico, permitiéndonos utilizar técnicas de propagación local, de forma que se obtengan los mismos resultados que se obtienen cuando utilizamos la distribución de posibilidad inicial [2]

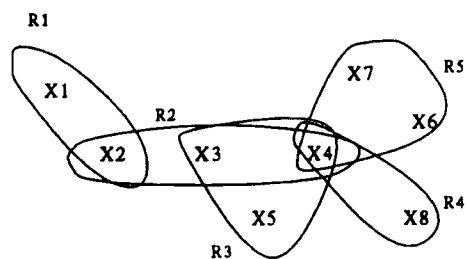


Figura 3: Hipergrafo asociado a T

Por tanto, para obtener un hipergrafo, nos bastará con encontrar el conjunto de nodos hoja en un árbol posibilístico. El algoritmo seguirá el mismo esquema utilizado para construir el árbol posibilístico, esto es:

```

BuscarHojas( S:Conjunto de Nodos )
{
  Buscar subconjuntos disjuntos L, R, C en S con
    I(L | C | R) y L ∪ R ∪ C = S.
  Si no es posible, Entonces
    - Incluir S en la lista de nodos hoja.
    - Asociar a S la distribución marginal πS.
  En caso contrario
    BuscarHojas( R ∪ C )
    BuscarHojas( L ∪ C )
}

```

El siguiente ejemplo nos muestra el comportamiento del anterior algoritmo cuando partimos de una base de casos. En el ejemplo, al partir de una base de casos, hemos considerado que estos no tienen porque ser una muestra exacta de la distribución de posibilidad que sigue el modelo. Por tanto, a la hora de realizar los tests, relajaremos el criterio de independencia. En particular, consideraremos que se satisface la relación $I(X|Z|Y)$ cuando, conocido el valor de Z , el conocer el valor de la variable Y proporciona una **información similar** (sobre los valores de X) a la que teníamos antes de conocer el valor de Y [1]. Más formalmente, diremos que

$$I(X|Z|Y) \Leftrightarrow \pi(xyz) \simeq [\pi(xz) \wedge \pi(zy)]$$

donde $\simeq$ viene definida por:

$$\pi \simeq \pi' \Leftrightarrow \forall x, |\pi(x) - \pi'(x)| \leq \epsilon$$

Ejemplo: Sean X, Y, Z, W variables bivaluadas, y $B1, B2$ dos bases de casos sobre $X \times Y \times Z \times W$, con tamaño 1.000 y 10.000 respectivamente. En la Tabla 1 representamos la frecuencia con la que aparecen cada uno de los distintos casos en $B1$ y $B2$ ¹. Partiendo

Tabla 1: Bases de Casos: $B1$ y $B2$.

YZW	B1		B2	
	$X = x_1$	$X = x_2$	$X = x_1$	$X = x_2$
$y_1 z_1 w_1$	12	11	135	129
$y_1 z_1 w_2$	13	13	139	134
$y_1 z_2 w_1$	53	41	533	401
$y_1 z_2 w_2$	66	42	665	394
$y_2 z_1 w_1$	93	93	938	931
$y_2 z_1 w_2$	108	131	1059	1337
$y_2 z_2 w_1$	81	80	801	791
$y_2 z_2 w_2$	80	82	809	804

de estos datos, utilizando una normalización por el máximo [13] obtenemos las distribuciones de posibilidad asociadas, π_1 y π_2 , que siguen los datos (Tabla 2). Considerando un umbral $\epsilon = 0.02$, obtenemos que $I(X|Y|ZW)$ y $I(Z|Y|W)$. Tras aplicar el algoritmo

¹Nota: El conjunto de datos en $B1$ y $B2$ se han obtenido mediante el siguiente proceso: (1) Partiendo del Hipergrafo de la Figura 4, calculamos $\pi(xyzw)$ mediante

$$\pi(xyzw) = \pi(xy) \wedge \pi(yz) \wedge \pi(yw)$$

con $\pi_{XY}(x_1 y_1, x_1 y_2, x_2 y_1, x_2 y_2) = (0.5, 0.8, 0.3, 1)$, $\pi_{YZ}(y_1 z_1, y_1 z_2, y_2 z_1, y_2 z_2) = (0.1, 0.5, 1, 0.6)$ y $\pi_{YW}(y_1 w_1, y_1 w_2, y_2 w_1, y_2 w_2) = (0.4, 0.5, 0.7, 1)$; (2) La distribución de posibilidad se transforma en una distribución de probabilidad, P , utilizando una normalización por el máximo [13]; (3) A partir de P se generan aleatoriamente las bases de casos.

Tabla 2: Distribuciones de Posibilidad π_1 y π_2 .

YZW	π_1		π_2	
	$X = x_1$	$X = x_2$	$X = x_1$	$X = x_2$
$y_1 z_1 w_1$	0.099	0.083	0.100	0.096
$y_1 z_1 w_2$	0.115	0.099	0.103	0.100
$y_1 z_2 w_1$	0.404	0.312	0.398	0.299
$y_1 z_2 w_2$	0.503	0.320	0.497	0.294
$y_2 z_1 w_1$	0.709	0.709	0.701	0.696
$y_2 z_1 w_2$	0.824	1	0.792	1
$y_2 z_2 w_1$	0.618	0.610	0.599	0.591
$y_2 z_2 w_2$	0.610	0.625	0.605	0.601

BuscarHojas, obtenemos el Hipergrafo de la Figura 4 (esto es, recuperamos el Hipergrafo original), donde las distribuciones de posibilidad (utilizando π_2) para cada hiperlado vienen dadas por:

$$\begin{aligned} \pi_{XY}(x_1 y_1, x_1 y_2, x_2 y_1, x_2 y_2) &= (0.497, 0.792, 0.299, 1), \\ \pi_{YZ}(y_1 z_1, y_1 z_2, y_2 z_1, y_2 z_2) &= (0.103, 0.497, 1, 0.605) \text{ y} \\ \pi_{YW}(y_1 w_1, y_1 w_2, y_2 w_1, y_2 w_2) &= (0.398, 0.497, 0.701, 1) \end{aligned}$$

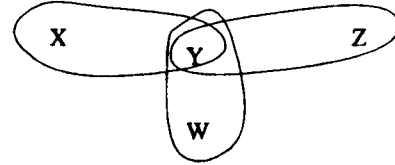


Figura 4: Hipergrafo aprendido

Cuando utilizamos un umbral ϵ igual a cero, obtenemos que el Hipergrafo refleja fielmente la distribución de posibilidad que siguen los datos. Sin embargo, cuando el umbral es estrictamente positivo, el Hipergrafo representará una aproximación de la distribución original. La siguiente proposición nos permite relacionar ambas distribuciones:

Proposición: Sean U un conjunto de variables, B una base de casos sobre U , π_B la distribución de posibilidad que se obtiene a partir de B , $\mathcal{H}$ un Hipergrafo obtenido al aplicar el algoritmo *BuscarHojas* con un umbral $\epsilon > 0$ y $\pi_{\mathcal{H}}$ la distribución de posibilidad representada por el Hipergrafo. Entonces, $\pi_{\mathcal{H}}$ es menos precisa que π_B

$$\forall u \in U, \pi_B(u) \leq \pi_{\mathcal{H}}(u)$$

Demostración: Es suficiente con ver que, para toda instanciación $u \in U$, asignamos a $\pi_{\mathcal{H}}(u)$ un valor mayor o igual que $\pi_B(u)$. En particular, tenemos que $\pi_{\mathcal{H}}(u) = \bigwedge_j \pi_{L_j}(u^{\downarrow L_j})$, con $u^{\downarrow L_j}$ la proyección de la instanciación u sobre las variables en L_j . Considerando que cada $\pi_{L_j}(u^{\downarrow L_j})$ es una marginal de $\pi_B(u)$, esto es, $\pi_{L_j}(u^{\downarrow L_j}) = \max_{U \setminus L_j} \pi_B(u)$, tenemos $\pi_{L_j}(u^{\downarrow L_j}) \geq$

$\pi_B(u)$, luego $\pi_B(u) \leq \pi_H(u)$, quedando demostrada la proposición. $\square$

Esta proposición nos permite asegurar que en el Hipergrafo tenemos representada una distribución de posibilidad menos informativa que la original, pero satisfaciendo las relaciones de independencia testeadas. El principal problema que plantea este algoritmo *BuscarHojas* es su alto coste computacional, pudiendo ser necesario realizar un número exponencial de tests de independencia para encontrar el conjunto C .

Con el fin de obtener algoritmos más eficientes de aprendizaje, nos planteamos una segunda aproximación al problema. En particular, consiste en utilizar, como paso intermedio, una estructura causal, para posteriormente transformar esta estructura en un árbol posibilístico. En este sentido, podemos considerar el concepto de independencia como un concepto abstracto, independiente del formalismo que se utilice para representar la información, y por tanto utilizar algoritmos de aprendizaje de estructuras causales, particularizados para el formalismo posibilístico [12].

Una vez que tenemos la estructura causal G , esto es, tenemos un grafo dirigido acíclico, el siguiente algoritmo nos permite trasformarla en un árbol posibilístico. En el algoritmo utilizamos la siguiente relación de orden sobre el conjunto de variables en G : Si existe el arco $X_i \rightarrow X_j$ en G , entonces $X_i < X_j$ en el orden d . Notaremos como $\text{Pred}_d(X_i)$ al conjunto de predecesores de X_i en el orden d , esto es, $\text{Pred}_d(X_i) = \{X_1, X_2, \dots, X_{i-1}\}$ y notaremos como $\text{Padres}(X_i)$ al conjunto de variables, X_p , tales que existe un arco directo $X_p \rightarrow X_i$ en G . Para realizar la transformación utilizaremos la siguiente propiedad [15]: Sea G una estructura causal y d un orden sobre las variables (obtenido mediante la relación anterior). Entonces, se satisface que

$$I(X_i | \text{Padres}(X_i) | \text{Pred}_d(X_i) \setminus \text{Padres}(X_i)).$$

Por tanto, considerando el conjunto de variables en orden inverso, esto es, $S = \{X_n, X_{n-1}, \dots, X_1\}$, podemos contruir un árbol posibilístico T siguiendo los siguientes pasos:

- Construir un árbol T con raíz $I(X_n | \text{Padres}(X_n) | \text{Pred}_d(X_n) \setminus \text{Padres}(X_n))$.
- Insertar como hijo izquierda de T el conjunto de variables $L = \{X_n\} \cup \text{Padres}(X_n)$, asociado con la distribución de posibilidad marginal $\pi_{\{X_n\} \cup \text{Padres}(X_n)}$.
- Recursivamente, si $\text{Pred}_d(X_n) \setminus \text{Padres}(X_n)$ no es el conjunto vacío, insertar como hijo derecha de T el árbol que se obtiene al repetir el proceso para el conjunto de variables $\{X_{n-1}, X_{n-2}, \dots, X_1\}$.

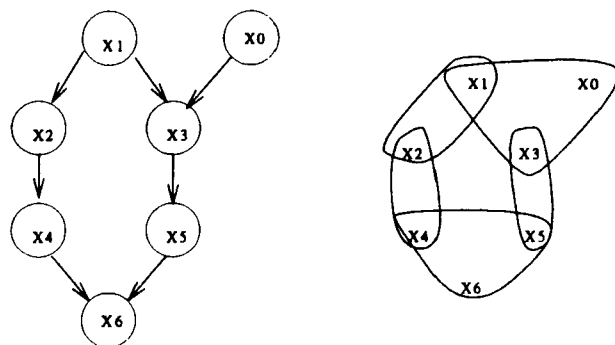


Figura 5: Red Causal e Hipergrafo asociado

Es posible que, como resultado de aplicar el esquema anterior, aparezcan hiperlados anidados en el Hipergrafo salida. En este caso, puesto que operamos con el mínimo, podemos eliminar los hiperlados más internos (con menos variables) sin modificar la distribución de posibilidad representada en el Hipergrafo. Por ejemplo, dada la red causal de la Figura 5 (parte izquierda), el algoritmo recupera el Hipergrafo representado en la parte derecha de la misma figura, incluyendo dos hiperlados más, el formado por la variable x_0 y el formado por la variable x_1 .

5 Conclusiones y Trabajos Futuros.

En este trabajo hemos visto como un Hipergrafo se puede considerar como una representación gráfica de una distribución de posibilidad. Para ello, vemos como el concepto de independencia condicional representa un importante papel. Sin embargo, a la hora de diseñar un algoritmo de aprendizaje que intente recuperar la distribución de forma exacta, el principal problema que se plantea es el alto coste computacional necesario. Por tanto, como posibles trabajos futuros nos plantearemos el estudio de algoritmos aproximados de aprendizaje para este tipo de estructuras. Además, es interesante realizar un estudio axiomático sobre las relaciones de independencia que se representan mediante un Hipergrafo y su relación con la axiomática de independencias dada para modelos causales [15]. Otro punto de interés es el estudio de los hipergrafos cuando consideramos que la medida de posibilidad es un caso particular de las medidas de evidencia [5], en concreto es una medida de plausibilidad consonante.

Referencias

- [1] L.M.de Campos and J.F. Huete. Possibilistic Independence. In *Proceeding of third European Congress on Intelligent Techniques and Soft Computing. EUFIT'95*, pages 69-74, 1995.

- [2] L.M.de Campos and J.F. Huete. How to construct possibility measures from a set of marginals. In *International Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems, IPMU'96*, Julio, 1996.
- [3] J.E. Cano. *Propagación de probabilidades inferiores y superiores en grafos*. Tesis Doctoral. Universidad de Granada, 1992.
- [4] P. Chatalic, D. Dubois, and H. Prade. A system for handling relational dependencies in approximate reasoning. In *Proc. 3 Inter. Expert Systems Conference, London*, pages 495–502, 1987.
- [5] A.P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematics and Statistics*, 38:325–339, 1967.
- [6] D. Dubois. Belief structures, possibility theory, decomposable confidence measures on finite sets. *Computer and Artificial Intelligence*, 5(5):403–417, 1986.
- [7] D. Dubois and H. Prade. *Possibility Theory: An approach to computerized processing of uncertainty*. Plenum Press, 1988.
- [8] D. Dubois and H. Prade. Inference in Possibilistic Hypergraphs. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'90*, pages 250–259. Lecture Notes in Computer Science 521. Eds B. Bouchon-Meunier and R.R. Yager and L.A. Zadeh, 1990.
- [9] J. Gebhardt and R. Kruse. Learning Possibilistic Graphical Models. In *Proceeding of third European Congress on Intelligent Techniques and Soft Computing. EUFIT'95*, pages 57–64. Lecture Notes in Computer Science 747. Eds M. Clarke and R. Kruse and S. Moral, 1993.
- [10] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press., 1980.
- [11] E. Hisdal. Conditional possibilities, independence and noninteraction. *Fuzzy Sets and Systems*, 1:283–297, 1978.
- [12] J.F. Huete. *Aprendizaje de Redes de Creencia mediante la detección de independencias: Modelos no probabilísticos*. Ph.D. Thesis, University of Granada, 1995.
- [13] G. Klir and B. Parviz. Probability-Possibility transformations: A comparison. *International Journal of General Systems*, 21:291–310, 1992.
- [14] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of fuzzy systems*. Wiley, 1994.
- [15] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan and Kaufmann, San Mateo, 1988.
- [16] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, Princeton N.J., 1976.
- [17] G. Shafer and P.P. Shenoy. Local computation in hypertrees. In *Working Paper 201*. School of Business, Univ. of Kansas, 1988.
- [18] L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, (1):3–28, 1978.

SOBRE ALGUNAS PROPIEDADES EN TERNAS DE DE MORGAN

M. Carbonell
dmimch0@ps.uib.es

M. Mas
dmimmg0@ps.uib.es

G. Mayor*
dmigmf0@ps.uib.es

J. Suñer
dmijsl0@ps.uib.es

J. Torrens*
dmijts0@ps.uib.es

Dpt. de Ciències Matemàtiques i Informàtica
Universitat de les Illes Balears

Resumen

Es conocido que las ternas de De Morgan (T, S, N) con T arquimediana, no son nunca distributivas ni modulares. En este trabajo, estudiamos propiedades más débiles en ternas de De Morgan arquimedianas, pero próximas a la distributividad y modularidad. En concreto, presentamos las ternas subdistributivas y submodulares (verificando sólo una desigualdad en la propiedad respectiva), vemos que las subdistributivas son siempre submodulares y caracterizamos estas últimas. Por otra parte, introducimos los grados de distributividad y de modularidad y damos diversos resultados sobre los mismos. Después estudiamos dichos grados en dos casos particulares: las ternas de Lukasiewicz y las aquí llamadas ternas Producto (una clase particular de ternas arquimedianas estrictas). A lo largo del trabajo presentamos una serie de ejemplos ilustrando todos estos conceptos.

Palabras clave: t-norma, t-conorma, negación, terna de De Morgan, distributividad, modularidad, ecuación funcional.

1. Introducción.

En este apartado presentamos los conceptos básicos que utilizaremos a lo largo del presente trabajo.

Definición 1.1. Una *t-norma* es una función $T : [0, 1] \times [0, 1] \rightarrow [0, 1]$ asociativa, conmutativa, con 1 como unidad y no decreciente en cada componente.

*Parcialmente subvencionados por la DGICYT, proyecto n. PB91-0334

Definición 1.2. Una *t-conorma* es una función $T : [0, 1] \times [0, 1] \rightarrow [0, 1]$ asociativa, conmutativa, con 0 como unidad y no decreciente en cada componente.

Definición 1.3. Una *t-norma* T es *arquimediana* si es continua y $T(x, x) < x$ para todo $x \in (0, 1)$.

Definición 1.4. Una *t-norma* T es *estricta* si es continua y estrictamente creciente en $(0, 1]^2$.

Definición 1.5. Una *negación* es una función $N : [0, 1] \rightarrow [0, 1]$ involutiva y decreciente.

Definición 1.6. Dada una *t-norma* T , una *t-conorma* S y una negación N , se dice que T y S son *N-duales* si verifican

$$S(x, y) = N(T(N(x), N(y)))$$

para cualesquiera $x, y \in [0, 1]$.

Definición 1.7. Una terna (T, S, N) se dice que es una *terna de De Morgan continua* (o simplemente una *terna de De Morgan*) si T es una *t-norma* continua, N una negación y S la *t-conorma* N -dual de T .

Existe un teorema de representación para *t-normas* arquimedianas debido a Aczél-Ling ([4], ver también [1] o [6] para la versión que presentamos):

Teorema 1.8. Una función $T : [0, 1] \times [0, 1] \rightarrow [0, 1]$ es una *t-norma* arquimediana si y sólo si existe una función continua estrictamente decreciente $t : [0, 1] \rightarrow [0, +\infty]$ tal que $t(1) = 0$ y

$$T(x, y) = t^{(-1)}(t(x) + t(y))$$

donde $t^{(-1)}$ es la pseudo-inversa de t definida como sigue

$$t^{(-1)}(x) = \begin{cases} t^{-1}(x) & \text{si } 0 \leq x \leq t(0) \\ 0 & \text{si } x > t(0). \end{cases}$$

A esta función t se le llama *generador aditivo* de T y es único salvo una constante multiplicativa. La *t-norma* T es estricta si y sólo si $t(0) = +\infty$ y entonces, $t^{(-1)}$ es la función inversa de t . Para *t-normas* arquimedianas no estrictas existe un único generador

aditivo tal que $t(0) = 1$ al cual se le llama usualmente *generador aditivo normalizado*. Finalmente, si T es una t -norma arquimediana no estricta con generador aditivo normalizado t , entonces la *región cero* de T viene dada por $Z(T) = \{(x, y) \in [0, 1]^2 \mid y \leq N(x)\}$ donde N es la negación definida por $N(x) = t^{-1}(1 - t(x))$ para todo $x \in [0, 1]$ y se denota por N_T .

Existe un teorema de representación análogo para t -conormas arquimedianas y se verifica que si T es una t -norma arquimediana (estricta) con generador aditivo t , entonces la t -conorma N -dual de T es también arquimediana (estricta) con generador aditivo $t \circ N$.

Definición 1.9. Diremos que una terna de De Morgan (T, S, N) es *arquimediana* (o *estricta*) cuando lo es T . Además una terna de De Morgan arquimediana no estricta (T, S, N) , se dice que es una *terna de Łukasiewicz* si $N = N_T$.

2. Subdistributividad y submodularidad.

Como ya se ha comentado, no existen ternas arquimedianas distributivas ni modulares, de hecho, si imponemos la propiedad distributiva o modular en una terna de De Morgan (T, S, N) , obtenemos de manera inmediata $T = \text{Min}$ y $S = \text{Max}$ (véase [7]). Así, nuestro interés radica en "medir" de alguna manera, lo cerca o lejos que está de tener estas propiedades cualquier terna arquimediana (T, S, N) . En este trabajo presentamos dos posibles formas de atacar esta medición. Este apartado lo dedicamos a estudiar cuando una terna arquimediana verifica solamente una de las desigualdades en las propiedades de distributividad (las dos, de T respecto de S y de S respecto de T) y de modularidad. En el siguiente apartado en cambio, introducimos los llamados grados de distributividad (también los dos posibles) y de modularidad.

Para empezar este apartado, notemos que si en una terna de De Morgan imponemos una cualquiera de las siguientes desigualdades:

$$T(x, S(y, z)) \geq S(T(x, y), T(x, z))$$

para cualesquiera $x, y, z \in [0, 1]$, o

$$S(x, T(y, z)) \leq T(S(x, y), S(x, z))$$

para cualesquiera $x, y, z \in [0, 1]$, obtenemos fácilmente $T = \text{Min}$ y $S = \text{Max}$ como única solución. Consecuentemente, las desigualdades contrarias son las que nos llevan a las siguientes definiciones.

Definición 2.1. Diremos que una terna de De Morgan (T, S, N) es (T, S) -*subdistributiva* si verifica

$$T(x, S(y, z)) \leq S(T(x, y), T(x, z))$$

para cualesquiera $x, y, z \in [0, 1]$.

Definición 2.2. Diremos que una terna de De Morgan (T, S, N) es (S, T) -*subdistributiva* si verifica

$$S(x, T(y, z)) \geq T(S(x, y), S(x, z))$$

para cualesquiera $x, y, z \in [0, 1]$.

Análogamente, en el caso de la modularidad definimos

Definición 2.3. Diremos que una terna de De Morgan (T, S, N) es *submodular* si verifica

$$T(x, S(y, z)) \leq S(T(x, y), z)$$

para cualesquiera $x, y, z \in [0, 1]$.

Observación: De nuevo ésta es la única desigualdad a tratar ya que la contraria lleva otra vez a $T = \text{Min}$ y $S = \text{Max}$ como única solución.

Por otra parte, notemos que la condición de modularidad sólo involucra valores $x, z \in [0, 1]$ tales que $z \leq x$. Sin embargo, en la definición de submodularidad podemos incluir también los valores $x < z$, ya que para estos valores se tiene:

$$T(x, S(y, z)) \leq x < z \leq S(T(x, y), z).$$

Proposición 2.4. Dada una terna de De Morgan (T, S, N) , se verifica:

i) (T, S, N) es (T, S) -subdistributiva si y sólo si es (S, T) -subdistributiva.

ii) Si (T, S, N) es subdistributiva (en adelante no utilizaremos los prefijos (T, S) y (S, T) , dada la equivalencia establecida en i)) entonces es submodular.

Proposición 2.5. Si (T, S, N) es una terna arquimediana subdistributiva entonces es estricta.

Ejemplos 2.6. i) Es inmediato comprobar que la terna $(\prod, \prod^*, 1-j)$ es subdistributiva y por tanto submodular.

ii) Por otra parte, se ve fácilmente que la terna $(W, W^*, 1-j)$ es submodular y en cambio no es subdistributiva como prueba la proposición anterior. Más adelante veremos cómo incluso entre las ternas estrictas se pueden encontrar ejemplos de ternas submodulares que no son subdistributivas.

Teorema 2.7. Sea (T, S, N) una terna arquimediana estricta, t un generador aditivo de T y f la involución sobre $[0, +\infty]$ dada por $f = t \circ N \circ t^{-1}$. Sea $F : [0, +\infty]^2 \rightarrow [0, +\infty]$ la función dada por $F(x, y) = f(f(x) + f(y))$. Entonces:

i) (T, S, N) es subdistributiva si y sólo si

$$F(x + y, x + z) - F(y, z) \leq x \quad (1)$$

para cualesquiera $x, y, z \in [0, +\infty]$.

ii) (T, S, N) es submodular si y sólo si

$$F(x + y, z) - F(y, z) \leq x \quad (2)$$

para cualesquiera $x, y, z \in [0, +\infty]$.

Teorema 2.8. Sea (T, S, N) una terna arquimediana no estricta, t un generador aditivo de T y f la involución sobre $[0, 1]$ dada por $f = t \circ N \circ t^{-1}$. Sea T_f la t -norma arquimediana no estricta cuyo generador aditivo es f . Entonces (T, S, N) es submodular si y sólo si T_f verifica (2) para cualesquiera $x, y, z \in [0, t(0)]$.

Corolario 2.9. Sea (T, S, N) una terna arquimediana y t un generador aditivo de T . Entonces (T, S, N) es submodular si y sólo si $f = t \circ N \circ t^{-1}$ es una función convexa.

Observación: Notemos que si (T, S, N) es una terna estricta subdistributiva entonces la función $f = t \circ N \circ t^{-1}$ es necesariamente convexa como consecuencia del corolario anterior. Este resultado está demostrado también en [2] aunque de forma totalmente distinta. En dicha referencia pueden encontrarse más resultados sobre ternas estrictas subdistributivas.

Ejemplos 2.10. i) Si T es una t -norma estricta con generador aditivo t , entonces existe una familia de negaciones (N_k) con $k \neq 0$ dadas por

$$N_k(x) = t^{-1} \left(\frac{k}{t(x)} \right) \quad \text{para todo } x \in [0, 1]$$

de manera que las ternas de De Morgan (T, S_k, N_k) son subdistributivas, ya que en estos casos la función F resultante viene dada por

$$F_k(x, y) = \frac{xy}{x+y}$$

que claramente verifica (1).

ii) Si consideramos la negación $N(x) = t^{-1}(-\ln(1 - e^{-t(x)}))$ y denotamos por S la t -conorma N -dual de T , entonces la terna (T, S, N) también es subdistributiva.

iii) Notemos que en particular, el teorema anterior prueba que todas las ternas de Lukasiewicz son submodulares, ya que en tales casos $f = 1 - j$.

iv) Como ya hemos mencionado, incluso entre las ternas estrictas podemos encontrar ejemplos de ternas submodulares que no son subdistributivas. Concretamente, sea $a < 1$ la única solución real de la ecuación $x = -\ln(x)$ y consideremos la función sobre $[0, +\infty]$ dada por:

$$f(x) = \begin{cases} -\ln(x) & \text{si } x \in [0, a] \\ e^{-x} & \text{si } x > a \end{cases}$$

Es fácil ver que f es convexa y por tanto toda terna estricta (T, S, N) con $tNt^{-1} = f$ será submodular. En cambio, tomando $x = y = 9a/10$ y $z = a/10$ se ve fácilmente que la correspondiente F no verifica (1) y por tanto estas ternas no serán subdistributivas.

Observación: Otra propiedad reticular clásica y estrechamente relacionada con la distributividad y la

modularidad es la "ortomodularidad". Concretamente una terna de De Morgan (T, S, N) se dice que es ortomodular si

$$S(T(x, Ny), y) = x$$

para cualesquiera $x, y \in [0, 1]$ tales que $x \geq y$. En relación a esta propiedad se demuestra en [7] que una terna de De Morgan es ortomodular si y sólo si es una terna de Lukasiewicz. Notemos que el corolario 2.9 prueba que toda terna de De Morgan ortomodular es submodular, pero no al revés.

3. Grados de distributividad y de modularidad.

En esta sección tratamos de determinar cuan distributiva o modular es una terna de De Morgan. Un estudio similar pero sólo para ternas de Lukasiewicz y únicamente para la (T, S) -distributividad puede hallarse en [3].

Definición 3.1. Dada una terna de De Morgan (T, S, N) , sea D_1 el número real no negativo más pequeño tal que (T, S, N) es D_1 -(T, S)-distributiva; es decir, tal que

$$|T(x, S(y, z)) - S(T(x, y), T(x, z))| \leq D_1$$

para cualesquiera $x, y, z \in [0, 1]$.

Diremos entonces que (T, S, N) tiene *grado de (T, S) -distributividad* $1 - D_1$.

Definición 3.2. Dada una terna de De Morgan (T, S, N) , sea D_2 el número real no negativo más pequeño tal que (T, S, N) es D_2 -(S, T)-distributiva; es decir, tal que

$$|S(x, T(y, z)) - T(S(x, y), S(x, z))| \leq D_2$$

para cualesquiera $x, y, z \in [0, 1]$.

Diremos entonces que (T, S, N) tiene *grado de (S, T) -distributividad* $1 - D_2$.

Definición 3.3. Dada una terna de De Morgan (T, S, N) , sea M el número real no negativo más pequeño tal que (T, S, N) es M -modular, es decir, tal que

$$|T(x, S(y, z)) - S(T(x, y), z)| \leq M$$

para cualesquiera $x, y, z \in [0, 1]$ tales que $z \leq x$.

Diremos entonces que (T, S, N) tiene *grado de modularidad* $1 - M$.

A partir de estas definiciones resulta claro que una terna de De Morgan (T, S, N) es (T, S) -distributiva (resp. (S, T) -distributiva, modular) si y sólo si su grado de (T, S) -distributividad (resp. (S, T) -distributividad, modularidad) es uno. Por otra parte, contrariamente a lo que ocurre con la subdistributividad, los dos grados de distributividad definidos en general no coinciden (ver ejemplos 3.12). Sin embargo, hay casos en que sí coinciden como demuestra la siguiente

Proposición 3.4. Para todas las ternas de De Morgan de la forma $(T, S, 1-j)$ ambos grados de distributividad coinciden.

En cambio, en el caso de ternas subdistributivas lo que podemos establecer son las siguientes comparaciones:

Teorema 3.5. Sea (T, S, N) una terna de De Morgan subdistributiva.

i) Si (T, S, N) es ϵ -modular, entonces es ϵ -(T, S)-distributiva (por tanto, $D_1 \leq M$)

ii) Si (T, S, N) es ϵ -modular, entonces es ϵ -(S, T)-distributiva (por tanto, $D_2 \leq M$).

Ejemplos 3.6. Se ve fácilmente que para la terna $(\prod, \prod^*, 1-j)$ se verifica $D_1 = D_2 = M = \frac{1}{4}$ y por lo tanto los dos grados de distributividad y el de modularidad coinciden y valen $\frac{3}{4}$. Análogamente, para la terna $(W, W^*, 1-j)$ también coinciden los tres grados y en este caso valen $\frac{1}{2}$.

Nuestro interés radica ahora en estudiar los tres grados introducidos para dos clases de ternas especiales: las ternas de Lukasiewicz y las ternas producto (una clase especial de ternas estrictas). En cuanto a las ternas de Lukasiewicz:

Proposición 3.7. Sea (T, S, N) una terna de Lukasiewicz, t el generador aditivo de T y s el punto fijo de N . Se verifica:

i) $D_1, D_2, M \geq \max(s, 1-s) \geq 1/2$.

ii) Si existe una constante positiva K tal que

$$|t^{-1}(x) - t^{-1}(y)| \leq K|x - y|$$

para cualesquiera $x, y \in [0, 1]$ entonces $D_1, D_2, M \leq K/2$.

Observación: El resultado anterior nos permite dar ejemplos de ternas de Lukasiewicz con los tres grados iguales a cualquier número real previamente designado entre $(0, 1/2]$. Sea $a \in [0, 1)$ y $f : [0, 1] \rightarrow [0, 1]$ definida por

$$f(x) = \begin{cases} (a-1)x + 1 & \text{si } x \leq 1/2 \\ -(1+a)(x-1) & \text{si } x \geq 1/2 \end{cases}$$

Fácilmente se comprueba que

$$|f(x) - f(y)| \leq 2f(1/2)|x - y|.$$

Ahora, si consideramos la terna de Lukasiewicz (T, S, N) con T aditivamente generada por $t = f^{-1}$, entonces el punto fijo de N será $s = f(1/2) = \frac{a+1}{2}$ y por tanto los tres grados de (T, S, N) son menores o iguales que $1 - \max(s, 1-s) = 1 - s = \frac{1-a}{2}$. Por otra parte, los tres grados han de ser también mayores o iguales que $1 - f(1/2) = \frac{1-a}{2}$. De este modo, los tres grados coinciden con $\frac{1-a}{2} \in (0, 1/2]$.

Para el caso de ternas estrictas recordemos primero que existe un equivalente multiplicativo a los generadores aditivos. Concretamente tenemos el siguiente teorema (véase [5]).

Teorema 3.8. Si T es una t-norma estricta, existe una función continua estrictamente creciente $h : [0, 1] \rightarrow [0, 1]$ tal que $h(0) = 0, h(1) = 1$ y

$$T(x, y) = h^{-1}(h(x) \cdot h(y))$$

para cualesquiera $x, y \in [0, 1]$.

Esta función h se llama *generador multiplicativo* de T . No es único pero si h y h' son dos generadores multiplicativos de una misma t-norma estricta entonces existe una constante real positiva k tal que $h' = h^k$. Más detalles sobre generadores multiplicativos se pueden encontrar en [5], [6].

Definición 3.9. Dada una t-norma estricta T con generador multiplicativo h , sea N_h la negación definida por $N_h(x) = h^{-1}(1 - h(x))$ para todo $x \in [0, 1]$ y sea S_h la t-conorma N_h -dual de T . Llamaremos a las ternas de la forma (T, S_h, N_h) *ternas producto*.

Observación: Notemos que para cada t-norma estricta T existen infinitas ternas producto de la forma (T, S_h, N_h) , una para cada generador multiplicativo. Es fácil comprobar además que todas ellas son subdistributivas.

Ejemplos 3.10. Tomando como h la función identidad en $[0, 1]$ obtenemos la terna producto $(\prod, \prod^*, 1-j)$ a la que nos referiremos como la *terna producto canónica*. Por otra parte, considerando para cada número real positivo k , la función $h_k : [0, 1] \rightarrow [0, 1]$ definida por $h_k(x) = x^k$ obtenemos otras ternas producto $(\prod, S_k, N_k)$, donde N_k denota la negación asociada a h_k y S_k la t-conorma N_k -dual de $\prod$.

Teorema 3.11. Sea (T, S_h, N_h) una terna producto. Entonces

$$D_1 = \max_{\lambda \in (0,1)} \{h^{-1}(2\lambda - \lambda^2) - h^{-1}(\lambda)\}$$

$$D_2 = \max_{\lambda \in (0,1)} \{h^{-1}(\sqrt{\lambda}) - h^{-1}(\lambda)\}$$

y $M = \max\{t_1, t_2, t_3\}$ donde

$$t_1 = \max_{\lambda \in (0,1/4)} \{h^{-1}(\sqrt{\lambda}) - h^{-1}(\lambda)\}$$

$$t_2 = \max_{\lambda \in [1/4, 1/2]} \{h^{-1}(\lambda + 1/4) - h^{-1}(\lambda)\}$$

y $t_3 = \max_{\lambda \in [1/2, 1]} \{h^{-1}(2\lambda - \lambda^2) - h^{-1}(\lambda)\}$.

Ejemplos 3.12. i) Retomando los ejemplos 3.10 donde las correspondientes funciones son $h_k(x) = x^k$ tenemos que todas estas ternas verifican $D_2 = 1/4$. En cambio variando $k \in \mathbb{R}^+$ se puede ver que D_1 toma todos los valores entre $(0, 1)$ mientras que M toma todos los valores entre $[1/4, 1)$. De esta manera, obtenemos ejemplos de ternas de De Morgan con grado de (T, S) -distributividad igual a cualquier valor entre $(0, 1)$ y también ejemplos con grado de modularidad igual a cualquier valor entre $(0, 3/4]$.

ii) Considerando ternas producto (T, S_h, N_h) con generador multiplicativo h de T tal que

$$h^{-1}(x) = \begin{cases} ax & \text{si } x \leq \alpha \\ bx + c & \text{si } x \geq \alpha \end{cases}$$

donde $\alpha \in (0, 1)$, $a \in (0, 1/\alpha)$ y

$$b = \frac{1 - \alpha a}{1 - \alpha} \quad \text{y} \quad c = \frac{\alpha(a - 1)}{1 - \alpha}$$

vemos que D_2 toma todos los valores entre $(1/4, 1)$ y por lo tanto, el grado de (S, T) -distributividad todos los valores entre $(0, 3/4]$.

Por otra parte, $1/4$ no es cota inferior para D_2 ni para M , de hecho con los mismos generadores, tomando $\alpha = 1/16$ y $a = 2$ obtenemos $D_1 = D_2 = M = 7/30 < 1/4$.

Referencias.

1. Aczél, J., Lectures on functional equations and their applications, Academic Press. New York, 1966.
2. Alsina, C., Mayor, G., Tomás, M.S. and Torrens, J., "Subdistributive De Morgan triplets", *Serdica* 19, 258 - 266, 1993.
3. Alsina, C. and Trillas, E., "On almost distributive Łukasiewicz triplets", *Fuzzy Sets and Systems* 50(2), 175 - 178, 1992.
4. Ling, C.H., "Representation of associative functions", *Publ. Math. Debrecen* 12, 189 - 212, 1965.
5. Schweizer, B. and Sklar, A., "Associative functions and statistical triangle inequalities", *Publ. Math. Debrecen* 8, 169 - 186, 1961.
6. Schweizer, B. and Sklar, A., *Probabilistic metric spaces*, North Holland, New York, 1983.
7. Trillas, E. and Valverde, L., "A few remarks on some-lattice type properties on Fuzzy connectives", *Proc. of the XIIth. Int. Symp. on Multiple-valued Logic*, Paris, 228 - 231, 1982.

EJECUCIÓN EFICIENTE DE REGLAS BORROSAS: INFERENCIA PARAMETRIZADA

P. Cariñena, A. Bugarín y S. Barro
Departamento de Electrónica e Computación.
Universidade de Santiago de Compostela.
ellapuri@usc.es

Resumen

En este trabajo se presentan los resultados del estudio realizado sobre el cierre de las operaciones básicas en el proceso de ejecución en el dominio de la verdad de una Base de Conocimiento Borroso, empleando una descripción paramétrica de los valores lingüísticos de la variable verdad. El estudio se centra en las operaciones de conjunción lingüística y función de razonamiento hacia adelante, indicando los operadores para los que ambos procesos son cerrados, siendo de este modo posible trabajar en el dominio de la verdad parametrizado. Además, en ese caso se proporcionan las expresiones simbólicas de cada operación, en las que se comprueba el considerable ahorro en coste computacional obtenido.

Palabras Clave: Inferencia en el dominio de la verdad, Razonamiento Borroso, Parametrización en la inferencia borrosa.

1 INTRODUCCIÓN

Dentro del ámbito de la Lógica Borrosa, una de las áreas de mayor aplicación práctica es la relativa a Sistemas Basados en Reglas. La utilización de este tipo de sistemas para resolver problemas reales, sin embargo, conlleva en ciertos casos el inconveniente del alto coste computacional de su ejecución, principalmente en sistemas donde se requiera un alto número de reglas, la organización del conocimiento se realice a través de variables intermedias, o las reglas presenten un número elevado de antecedentes y/o consecuentes.

En determinadas aplicaciones de control o en Sistemas Expertos Borrosos, en donde puede resultar necesaria

la descripción de estrategias para la ejecución de la BCB, la proyección del proceso sobre una arquitectura *hardware* dedicada no es inmediata, por lo que la búsqueda de la eficiencia debe orientarse hacia estrategias que reduzcan la carga computacional del proceso de ejecución, rebajando el número de operaciones a realizar o simplificándolas. Dos posibles soluciones descritas en la bibliografía son la *compactación* de la base de reglas [8,9] y la *parametrización* o descripción simbólica de las distribuciones de posibilidad involucradas [6]: la compactación consiste en un precompilado de las reglas que forman la Base de Conocimiento Borroso (BCB), de manera que se traslada parte de la carga computacional al momento de la definición de la BCB, reduciendo así el número de operaciones a realizar durante la etapa de ejecución. La parametrización persigue evitar la manipulación directa de distribuciones de posibilidad, trasladando el proceso de razonamiento al dominio de la verdad [7], lo que permite un tratamiento homogéneo para todas las variables, al manipularse únicamente Valores Lingüísticos de Verdad (VLV) (distribuciones de posibilidad definidas en el intervalo $[0,1]$), y planteando una descripción simbólica de los mismos [5,6], lo que permite realizar el proceso de ejecución de la BCB a través de operaciones simples entre valores numéricos sencillos. Para ello es necesario que la parametrización elegida sea cerrada bajo las operaciones que se realizan en el proceso: conjunción lingüística, cálculo de la Función de Razonamiento Hacia Adelante (FRHA), agregación y encadenamiento [2,7].

En este trabajo se aborda parcialmente el estudio de la estrategia de parametrización en el dominio de la verdad, analizando sus condiciones de aplicación y las ventajas que presenta. Para ello, partimos de la descripción más general del proceso de razonamiento en una BCB, según la Regla Composicional de Inferencia (RCI) "SUP-*" de Zadeh [10] (apartado 2). En el apartado 3 se introducen las expresiones que describen el razonamiento en el dominio de la verdad, para pasar seguidamente a definir (apartado 4) la parame-

trización elegida para los valores lingüísticos de la variable verdad, presentando a continuación (apartado 5) los resultados obtenidos en el estudio del cierre de los procesos de conjunción y FRHA para un amplio conjunto de operadores. Por último, el apartado 6 recoge las conclusiones derivadas del análisis realizado.

2 REGLA COMPOSICIONAL DE INFERENCIA

El elemento básico en la representación de conocimiento borroso es la variable lingüística, o variable cuyo valor son palabras o frases del lenguaje natural. Estas variables lingüísticas se estructuran en proposiciones que, al relacionarse entre sí, conforman reglas borrosas. Las Bases de Conocimiento Borroso (BCB) se componen de una serie de reglas que, en conjunto, expresan el saber acerca de un dominio de aplicación concreto. En adelante asumiremos que la expresión general de la regla r -ésima dentro de una BCB será de la forma Múltiple-Antecedente Único-Consecuente, puesto que una regla con más de un consecuente, ligados por el conectivo "y", puede reescribirse sencillamente de esta forma:

$$SI X_1^r ES A_1^r Y \dots Y X_{M_r}^r ES A_{M_r}^r \quad (1) \\ ENTONCES Z^r ES B^r$$

donde M_r representa el número de antecedentes de la regla r -ésima. $X_{m_r}^r$, $m_r=1, \dots, M_r$, y Z^r , son variables lingüísticas, y $A_{m_r}^r$, $m_r=1, \dots, M_r$ y B^r , son valores lingüísticos de las variables correspondientes, los cuales se suponen asociados a una distribución de posibilidad que expresa su significado, y que denotaremos con el mismo nombre que el valor lingüístico al que representan.

Siguiendo con esta notación, denotaremos por $\tilde{A}_{m_r}^r(i_{m_r})$, $m_r=1, \dots, M_r$, $i_{m_r}=1, \dots, I_{m_r}$, las distribuciones de posibilidad asociadas a los valores lingüísticos observados para las correspondientes variables de la parte antecedente de la regla, y por $\tilde{B}^r(j)$, $j=1, \dots, J$, la distribución de posibilidad inferida para la variable de la parte consecuente.

El proceso de ejecución de las reglas se lleva a cabo generalmente mediante la Regla Composicional de Inferencia (RCI) "SUP-*" de Zadeh [10] (donde el operador "*" representa una t-norma o una función generadora de Modus Ponens), que interpreta el proceso de ejecución de la regla 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 propia existencia de la regla, que liga antecedentes y consecuentes (SI-ENTONCES).

La distribución de posibilidad inferida para la variable de la parte consecuente de la regla, se obtiene según la expresión:

$$\tilde{B}^r(j) = \bigvee_{U_1 \times \dots \times U_{M_r}} \sigma(\tilde{A}_1^r(i_1), \dots, \tilde{A}_{M_r}^r(i_{M_r})) * \rho[\sigma(A_1^r(i_1), \dots, A_{M_r}^r(i_{M_r})), B^r(j)] \quad (2)$$

siendo "V" el operador "supremo" (SUP), " σ " una t-norma, " ρ " un operador de implicación y U_{m_r} , $m_r=1, \dots, M_r$, los universos de discurso donde se definen las variables $X_{m_r}^r$, $m_r=1, \dots, M_r$.

Esta estrategia de ejecución de reglas borrosas basada en la RCI presenta un inconveniente a la hora de su aplicación a problemas reales, debido fundamentalmente a que existe una fuerte dependencia con la discretización que se efectúe de los universos de discurso en que se definen cada una de las variables involucradas en el proceso. Esto origina que en aquellos casos en que el número de variables sea elevado o la discretización muy fina el proceso suponga un alto coste computacional. Una manera de soslayar este problema es realizar el proceso de razonamiento en el dominio de la verdad [7], con lo que se consigue que el proceso de inferencia sea independiente de los universos de discurso, realizándose un tratamiento homogéneo de todas las variables. La discretización del universo de verdad (intervalo [0,1]) podrá ser mucho menor en número de puntos a la de un universo U genérico cualquiera, con lo que se gana en eficiencia.

3 RAZONAMIENTO EN EL DOMINIO DE LA VERDAD

Retomamos la expresión general de la regla r -ésima de una BCB, asumiéndose implícitamente el índice r por simplicidad:

$$SI X_1 ES A_1 Y \dots Y X_M ES A_M \\ ENTONCES Z ES B \quad (3)$$

La ejecución de esta regla en el dominio de la verdad [7] se inicia con el cálculo del Grado de Verificación Lingüístico (GDVL) para cada proposición de la parte antecedente de la regla, definido como:

$$\tau_{A_m}(a) = \bigvee_{i/A_m(i)=a} \tilde{A}_m(i), \quad a \in [0, 1] \quad (4)$$

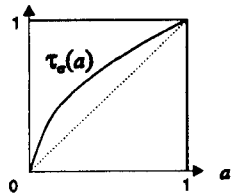
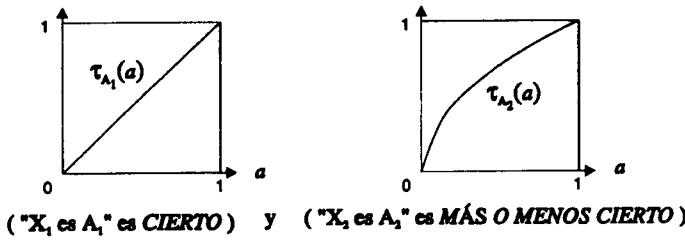
cuyo significado es un Valor Lingüístico de Verdad (VLV) que cuantifica el grado en que se cumple la proposición correspondiente, dada la observación realizada.

En reglas con múltiples antecedentes, se combinan los GDVL de cada proposición en un único GDVL global " τ_σ " para toda la parte antecedente (proceso de conjunción lingüística):

$$\tau_\sigma(a) = \bigvee_{a=\sigma(a_1, \dots, a_M)} \sigma(\tau_{A_1}(a_1), \dots, \tau_{A_M}(a_M))$$

$$a_m \in [0, 1], \forall m = 1, \dots, M, a \in [0, 1] \quad (5)$$

donde " σ " es el operador de conjunción. Este proceso se representa gráficamente en el ejemplo de la Figura 1, en la que partiendo de dos GDVL definidos por: $\tau_{A_1}(a) = a$ y $\tau_{A_2} = a^{1/2}$, se obtiene $\tau_\sigma(a) = a^{1/2}$.



("X₁ es A₁" y "X₂ es A₂") es M\u00c1S O MENOS CIERTO

Figura 1: Ejemplo de conjunción lingüística de dos proposiciones.

El proceso de inferencia propiamente dicho se realiza a través de la Función de Razonamiento Hacia Adelante (FRHA), " τ_B ", definida como:

$$\tau_B(b) = \bigvee_{a \in [0,1]} \tau_\sigma(a) * \rho(a, b), \quad b \in [0, 1] \quad (6)$$

donde " ρ " es la relación de implicación y " $*$ " el operador descrito por la RCI "SUP- $*$ " (t-norma o función generadora de modus ponens). Así, en la Figura 2 se observa como, a partir de un GDVL global para la parte antecedente de la regla dado por $\tau_\sigma(a) = a^2$, se obtiene como resultado de la inferencia la FRHA $\tau_B(b) = b$. En el ejemplo se ha utilizado la relación de implicación de Gödel.

La FRHA se compone con la distribución de posibilidad $B(j)$ asociada al valor lingüístico B en la parte

SI ("X₁ es A₁" y "X₂ es A₂") ENTONCES "Z es B"

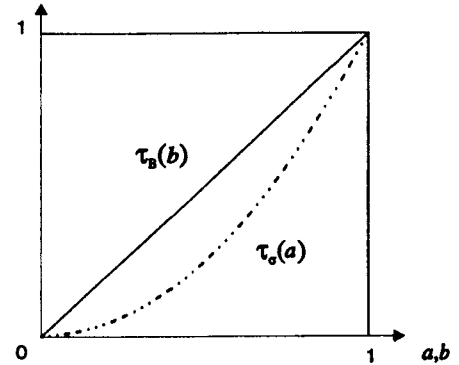


Figura 2: Cálculo de la FRHA τ_B a partir del GDVL τ_σ de la parte antecedente, para la relación de implicación de Gödel (ver apartado 3).

consecuente de la regla, para obtener la distribución de posibilidad inferida:

$$\tilde{B}(j) = \tau_B(B(j)), \quad j = 1, \dots, J \quad (7)$$

La ventaja más importante de este proceso es que, una vez realizada la transformación (4), es posible ejecutar el proceso de conjunción (expresión (5)) y el de inferencia (6) en el dominio de la verdad, independientemente de los universos de discurso en los cuales se definen las variables, ya que todas las operaciones se realizan sobre el intervalo $[0,1]$, consiguiéndose por tanto un tratamiento homogéneo para todas ellas, y una reducción en las necesidades de discretización de los universos de discurso.

De todos modos, el proceso todavía puede ser mejorado en cuanto a eficiencia computacional si se emplean métodos alternativos que eviten el trabajar explícitamente con distribuciones de posibilidad, aunque sea sobre el intervalo $[0,1]$. Uno de los métodos posibles es el realizar una descripción paramétrica de las distribuciones de posibilidad que definen los valores lingüísticos de la variable verdad, de manera que el proceso de inferencia se reduzca a operaciones simples entre parámetros. Esto será posible siempre que la parametrización empleada sea cerrada bajo las operaciones de interés: conjunción lingüística, cálculo de la FRHA. Existen otras operaciones necesarias para la ejecución de una BCB completa en el dominio de la verdad (agregación y encadenamiento), que no hemos descrito por simplicidad y porque no presentan problemas de cierre [2]. En las siguientes secciones describiremos la parametrización propuesta por Baldwin [1] y analizaremos sus propiedades de cierre para un conjunto amplio de operadores. Otros autores [5,6] han propuesto parametrizaciones alternativas, con el mis-

mo propósito de conseguir un ahorro computacional. Hemos seleccionado la de Baldwin por su simplicidad (empleo de un sólo parámetro para la descripción de los VLV) y su uso más extendido en la bibliografía.

4 DESCRIPCIÓN PARAMÉTRICA DE LA VARIABLE LINGÜÍSTICA VERDAD

La definición de los valores de la variable lingüística verdad propuesta por Baldwin [1] se expresa analíticamente como:

$$\tau(x) = x^q, \forall x \in [0, 1], q \in \mathbb{R}^+ \quad (8)$$

Tenemos así una familia parametrizada donde el exponente q identifica de forma unívoca a cada uno de los elementos. Por ejemplo, el exponente $q=1$ se asocia generalmente al significado de *CIERTO*, con $q=2$ se identifica *MUY CIERTO*, y con $q=1/2$, *MÁS O MENOS CIERTO*.

Esta definición permite un tratamiento simbólico de la familia de valores lingüísticos a través del exponente q . Para que la parametrización sea útil a la hora de realizar inferencias en el dominio de la verdad, que es el objetivo perseguido, deberá ser cerrada para las operaciones indicadas anteriormente.

5 ANÁLISIS DEL CIERRE DE OPERACIONES EN EL DOMINIO DE VERDAD PARAMETRIZADO

5.1 CONJUNCIÓN LINGÜÍSTICA

Ya que las situaciones con Múltiple-Antecedente pueden ser reducidas a la composición de casos con sólo dos proposiciones en la parte antecedente, consideraremos que la ecuación (5) se puede reescribir como:

$$\tau_\sigma(a) = \bigvee_{a=\sigma(a_1, a_2)} \sigma(\tau_{A_1}(a_1), \tau_{A_2}(a_2)), a \in [0, 1] \quad (9)$$

donde

$$\tau_{A_1}(a_1) = a_1^{q_1}, \tau_{A_2}(a_2) = a_2^{q_2}, q_1, q_2 \in \mathbb{R}^+ \quad (10)$$

son los GDVL de las proposiciones en la parte antecedente de la regla y " σ " es la t-norma que realiza la operación de conjunción. Este proceso será cerrado cuando se obtenga un GDVL global $\tau_\sigma(a) = a^{q_\sigma}$,

$q_\sigma \in \mathbb{R}^+$, y pueda calcularse q_σ directamente a partir de q_1 y q_2 .

El interés de que esta parametrización sea cerrada para la conjunción reside en la posibilidad de trabajar directamente con los exponentes q_1, q_2 para obtener el exponente del GDVL global q_σ . De esta forma puede eludirse la manipulación directa de los VLV, y el proceso podrá realizarse de una forma considerablemente menos costosa desde el punto de vista de la carga computacional.

En este trabajo hemos analizado el proceso de conjunción lingüística para un conjunto de seis t-normas (*mínimo, producto, producto drástico, producto acotado, cociente y Einstein*). Para los tres primeros casos la parametrización es cerrada con $q_\sigma = q_1 \wedge q_2$. Las otras tres t-normas son cerradas parcialmente, es decir, sólo para determinados valores de los parámetros, lo cual las hace poco útiles desde el punto de vista práctico.

El hecho de que en los tres primeros casos el resultado para q_σ sea idéntico parece hacer innecesario el disponer de un amplio abanico de operadores de conjunción, pues los resultados no varían de uno a otro. Por otra parte, se comprueba la gran ventaja, en cuanto a coste computacional, de trabajar con los valores de verdad parametrizados, comparando el resultado obtenido para q_σ (una simple operación de mínimo entre dos valores numéricos) con la expresión original para el cálculo de " τ_σ " en la expresión (9), en la que el cálculo debe hacerse sobre todo el intervalo $[0,1]$ (este ahorro computacional se analiza más en detalle en la sección 5.3). En aquellos casos en que la parametrización no sea cerrada, deberá implementarse directamente la expresión (9).

5.2 FUNCIÓN DE RAZONAMIENTO HACIA ADELANTE

La función de razonamiento hacia adelante, definida en la ecuación (6), será cerrada para la parametrización de Baldwin si puede ser descrita como $\tau_B(b) = b^{q_B}$, $q_B \in \mathbb{R}^+$, cuando " τ_σ " es de la forma $\tau_\sigma(a) = a^{q_\sigma}$, $q_\sigma \in \mathbb{R}^+$. Al ser la FRHA composición de dos funciones (relación de implicación " ρ " y operador " $**$ "), la condición de cierre vendrá determinada por la combinación " ρ^* " presente en cada caso particular. En la Tabla 1 se resumen los resultados obtenidos en el análisis del cierre de la FRHA para aquellas combinaciones de operador de implicación y " $**$ " (t-norma) cerradas.

Vemos que, a diferencia de la operación de conjunción lingüística, la FRHA no se comporta de manera uniforme en los casos en que es cerrada para la parametrización considerada, sino que se distinguen com-

Tabla 1: Análisis de cierre de la FRHA.

	T-NORMA	T-NORMA	T-NORMA	T-NORMA	T-NORMA	T-NORMA
IMPLICACIÓN	Mínimo	Producto	P. acotado	P. Drástico	Cociente	Einstein
Gödel (R_g)	1	1	1	1	1	1
R_{gs}	1	1	1	1	1	1
R_{gg}	1	1	1	1	1	1
Goguen (R_Δ)	3	1	1	1	-	-
Estándar (R_s)	2	2	2	2	2	2
R_{ss}	2	2	2	2	2	2
R_{sg}	2	2	2	2	2	2
$R_\emptyset$	-	-	2	2	-	-

portamientos diferenciados según el tipo de operador de implicación involucrado. Así, las combinaciones " ρ -" pueden clasificarse en FRHA no cerradas para esta parametrización (con las relaciones de implicación de Lukasiewicz (R_a), Zadeh (R_m), Booleana (R_b), Mizumoto (R_*), $R_\#$, $R_\cap$) o, dentro de las que son cerradas, según el tipo de comportamiento, que puede asociarse con tres tipos distintos de razonamiento (indicados por 1, 2 y 3 en la Tabla 1). De entre estos tres tipos, únicamente los dos primeros tienen interés desde el punto de vista práctico, ya que el tercero, aparte de ser específico para una única combinación " ρ -" concreta, no parece tener un significado intuitivo, que se suponga aplicable a un problema real. En los casos 1 y 2, a partir de un mismo GDVL de conjunción $\tau_\sigma(a) = a^{q_\sigma}$, $a \in [0, 1]$, $q_\sigma \in \mathbb{R}^+$, obtenemos para el parámetro q_B en la FRHA el siguiente valor:

TIPO 1: $q_B = 1 \wedge q_\sigma$

TIPO 2: $q_B = q_\sigma$

Así, en el tipo de razonamiento 2, el GDVL de las proposiciones de la parte consecuente de la regla va a coincidir con el GDVL global de la parte antecedente, y tendremos que:

MUY CIERTO $\Rightarrow$ *MUY CIERTO*

CIERTO $\Rightarrow$ *CIERTO*

MÁS O MENOS CIERTO $\Rightarrow$ *MÁS O MENOS CIERTO*

En el tipo de razonamiento 1, sin embargo, nunca se obtendrá ningún valor de verdad más específico que *CIERTO* para las proposiciones de la parte consecuente de la regla, independientemente de cual sea el GDVL global de la parte antecedente. Así, se tiene:

MUY CIERTO $\Rightarrow$ *CIERTO*

CIERTO $\Rightarrow$ *CIERTO*

MÁS O MENOS CIERTO $\Rightarrow$ *MÁS O MENOS CIERTO*

Este tipo de razonamiento se corresponde con el ejemplo presentado en la Figura 2. En ella se puede ver como, partiendo de un GDVL con $q_\sigma=2$ (correspondiente a *MUYCIERTO*), se obtiene directamente la FRHA " τ_B " sin más que calcular q_B como $1 \wedge q_\sigma = 1$ (es decir, *CIERTO*).

Este tipo de razonamiento es más coherente desde el punto de vista lógico (no se puede inferir nada más preciso para la parte consecuente que lo que dice la regla [4]), mientras que el tipo 2 se corresponde con las situaciones en que la regla expresa más bien una relación del tipo '*el GDVL del consecuente es una función creciente del GDVL del antecedente*'.

Otro aspecto a destacar, a la vista de los resultados presentes en la Tabla 1, es que, a pesar de disponer de un número elevado de combinaciones " ρ -", únicamente se obtienen tres tipos de razonamiento distintos, lo cual lleva a plantearse nuevamente la utilidad de disponer de tan amplia gama de t-normas y operadores de implicación.

5.3 ANÁLISIS COMPARATIVO DE LA COMPLEJIDAD COMPUTACIONAL

Una vez obtenidas las expresiones que describen el proceso de ejecución de reglas en el dominio de la verdad parametrizado, se puede analizar de forma cuantitativa el ahorro en carga computacional que supone trabajar en este dominio de la verdad parametrizado, respecto a hacerlo sin parametrizar. Para ello, estudiaremos seguidamente el orden de complejidad computacional de las expresiones (6) y (9), que describen el razonamiento en el dominio de la verdad.

Por simplicidad, vamos a suponer que el grado de discretización N del intervalo de verdad $[0,1]$ es uniforme en todos los casos. Así, para la conjunción lingüística (expresión (9)), supondremos que se recorren todos los valores $\sigma(a_1, a_2)$, lo que permite fijar una cota superior para el número de operaciones a realizar. Ésta es de la forma $4 \cdot N^2 - 1$, y por tanto, la complejidad computacional es $O(N^2)$, que depende tan solo del grado de discretización del universo de verdad (no intervienen los universos de discurso concretos en que se definen las distintas variables involucradas en las reglas), tal y como cabría esperar. Para la FRHA (expresión (6)) se obtiene que el número de operaciones a realizar es $4 \cdot N^2 \cdot N$, que de nuevo depende sólo de la discretización

del intervalo $[0,1]$, y también supone una complejidad computacional $O(N^2)$. Dado que en el dominio de la verdad parametrizado las operaciones (6) y (9) se reducen cada una a un único cálculo de mínimo entre dos valores numéricos sencillos en el peor de los casos (en otros simplemente se mantiene el valor del parámetro de entrada), el considerable ahorro computacional que se obtiene resulta evidente.

6 CONCLUSIONES

En este trabajo se ha descrito el proceso de inferencia en una BCB, centrando la atención en el análisis de una solución alternativa a la aplicación directa de la RCI. El método descrito es exacto, en el sentido de que únicamente se reordenan de una manera conveniente las operaciones involucradas en la RCI, consiguiéndose un considerable ahorro computacional en dos niveles: paso al dominio de la verdad, y así manipulación de VLV, y parametrización de éstos, lo cual, además, facilita la descripción de distintos algoritmos de ejecución de BCB [3].

Se comprueba que para gran parte de los operadores utilizados habitualmente en el proceso de inferencia (tanto t-normas como operadores de implicación) la parametrización elegida es cerrada, y se indica la correspondiente operación entre parámetros que sustituye al cálculo inicial sobre todo el dominio de la verdad.

Un aspecto a señalar es que, aunque el trabajo se ha basado en la parametrización de Baldwin, se obtienen resultados similares si el análisis se realiza con otro tipo de parametrización [6]. Se ha preferido para presentar este estudio la parametrización de Baldwin por su uso más extendido, además de por ser más sencilla en su descripción: un único parámetro q frente a los dos necesarios en la otra propuesta. Otra ventaja de la parametrización de Baldwin, de interés cuando se trabaja con BCB que presentan encadenamiento entre reglas [2], es que el cálculo del VLV inverso de uno dado es también una operación cerrada. En cualquier caso, puesto que el dominio de aplicación concreto condiciona fuertemente la elección de la familia de VLV, haciendo que éstos deban adecuarse a la expresión humana del conocimiento, en ciertos casos puede resultar más conveniente, desde este punto de vista, utilizar la parametrización descrita en [6].

En lo que respecta al trabajo futuro, se intentará ampliar el número de casos en los que sea aplicable la parametrización de los VLV como solución para reducir el coste computacional, analizando los casos para los que no es cerrada. Se trata de comprobar si, realizando una aproximación "razonable" de los resultados reales que se obtienen, las operaciones podrían considerarse cerradas, con lo cual se incrementará el número

de opciones a la hora de elegir el operador adecuado en cada problema particular.

Agradecimientos

Este trabajo ha sido financiado por la CICyT y por la Xunta de Galicia a través de los proyectos de investigación TIC95-0604 y XUGA20603A95, respectivamente.

Referencias

- [1] J.F. Baldwin (1979). A New Approach to Approximate Reasoning using a Fuzzy Logic. *Fuzzy Sets and Systems* 2, 2:309-325.
- [2] A. Bugarín y S. Barro (1995). Reasoning with Truth Values on Compacted Fuzzy Chained Rules. *Informe Interno USC/DEC/GSI 9-95*.
- [3] A. Bugarín, P. Cariñena, M. Fernández Delgado y S. Barro (1996). Incremental Fuzzy Reasoning on Petri Nets. Aceptado en *Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96)*, Granada, 1-5 Julio 1996.
- [4] D. Dubois y H. Prade (1991). Fuzzy Sets in Approximate Reasoning, Part I: Inference with Possibility Distributions. *Fuzzy Sets and Systems* 40:143-202.
- [5] L. Godó, J. Jacas y L. Valverde (1991). Fuzzy Values in Fuzzy Logic. *International Journal of Intelligent Systems* 6:199-212.
- [6] R. López de Mántaras (1990). *Approximate Reasoning Models*. Ellis Horwood Series in Artificial Intelligence, Chichester, England.
- [7] E. Trillas y L. Valverde (1985). 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: 157-166.
- [8] K. Uehara y M. Fujise (1991). Multi-stage Fuzzy Inference by Propagation of Linguistic Truth Values. *Proc. 4th IFSA World Congress*: 196-199.
- [9] K. Uehara y M. Fujise (1993). Multistage Fuzzy Inference Formulated as Linguistic-Truth-Value Propagation and its Learning Algorithm Based on Back-Propagation Error Information. *IEEE Transactions on Fuzzy Systems* 1, 3:205-221.
- [10] L.A. Zadeh (1973). Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on Systems, Man and Cybernetics* 3, 1:28-44.

APRENDIZAJE INDUCTIVO UTILIZANDO ALGORITMOS GENETICOS

J.L. Castro*

Dpto. Ciencias de la Computación e I.A.
E.T.S.I. Informática. Universidad de Granada
Avenida de Andalucía, 38 18071 Granada (Spain)
castro@robinson.ugr.es

J.J. Castro-Schez

J.M. Zurita

zurita@robinson.ugr.es

Resumen

Este trabajo pretende mostrar un nexo de unión entre los Algoritmos Genéticos y la Lógica Fuzzy para desarrollar un algoritmo de aprendizaje automático que extraiga un conjunto de reglas fuzzy a partir de un conjunto de ejemplos de entrenamiento, que muestran la forma de actuar del sistema a aprender.

Palabras Clave: Algoritmos genéticos, Adquisición del conocimiento, Aprendizaje automático.

1 INTRODUCCION

En la vida cotidiana una de las formas obvias, que se dan de *Adquisición de Conocimiento* (AC) es la de aprendizaje por observación, es decir a partir de ejemplos. Sino, formulemos la siguiente pregunta ¿Cómo un aprendiz se convierte en un maestro? Está claro que mientras el aprendiz ve al maestro, él observa ejemplos de problemas que se producen constantemente, así como los casos difíciles, y toma conciencia de cómo el maestro responde en cada caso. El maestro no educa explícitamente al aprendiz en todas las heurísticas que él aplica durante su actividad de resolución de un problema, el aprendiz comienza a aprender por inducción.

Inducción es el proceso por el cual podremos inferir "leyes" o desarrollar teorías mediante la observación y análisis de eventos o casos en los cuales esa ley es aplicada. Gracias al proceso de inducción vamos a llegar a conclusiones generales usando ejemplos particulares.

La inducción se ha convertido en un método muy eficaz para la AC en aquellos casos, en los que se posee

un conocimiento incompleto o el experto no es capaz de explicar totalmente su forma de actuar, es decir, los procedimientos o heurísticas que lo conducen a la solución del problema.

Debido también, a la tendencia natural de los expertos de suministrar ejemplos de problemas y sus soluciones, es razonable pensar en la automatización del proceso de inducción para automatizar a su vez al proceso de AC.

En los últimos años, la investigación en el campo del aprendizaje automático, utilizando la teoría de la inducción, ha conducido al campo de la lógica fuzzy.

Zadeh propuso un mecanismo de modelado del comportamiento humano, basado en valores lingüísticos fuzzy más que en números, esto nos lleva a la introducción de lo difuso en la teoría de sistemas, y al desarrollo de una nueva clase de sistemas llamados *Sistemas Fuzzy*. Extendiendo la definición clásica de un modelo de un sistema, se llega a la de un modelo fuzzy. Un *Modelo Fuzzy* es definido como una representación de las características esenciales de un sistema por medio de la teoría de conjuntos fuzzy. Una característica de los sistemas fuzzy es que están basados en el concepto de codificación fuzzy de la información. Los sistemas fuzzy operarán con conjuntos fuzzy en lugar de con números. Cada conjunto fuzzy tendrá más poder expresivo que un simple número, ya que los conjuntos fuzzy generalizan la información. En esencia, la representación de la información en sistemas fuzzy imita el mecanismo de razonamiento que utilizamos los humanos.

El modelado fuzzy propuesto por Zadeh, es razonable ya que provee un medio aproximativo y efectivo para la descripción de la conducta o comportamiento de sistemas que son demasiado complejos o están muy mal definidos para admitir el uso de análisis matemáticos precisos.

Los modelos de sistemas fuzzy básicamente caen dentro de dos categorías, que difieren fundamentalmente en su habilidad para representar diferentes tipos de información. La primera incluye los *Mode-*

*This work has been supported by Research Project TIC95-0453.

los *Lingüísticos*, que van a describir el sistema por medio de un conjunto de reglas IF-THEN con predicados vagos; el conjunto de reglas toma la función del usual conjunto de ecuaciones usadas para caracterizar un sistema. Esta categoría es la que nos va a interesar, y en la que nosotros vamos a trabajar.

La idea es extraer un conjunto de reglas fuzzy a partir de un conjunto de ejemplos que muestran el funcionamiento del sistema. Este acercamiento ha sido investigado por diversos autores [1], [3], [4], [5], [6], [11], [12], entre otros.

Cada autor utiliza una metodología diferente para el desarrollo de modelos fuzzy a partir de un conjunto de ejemplos de entrenamiento, así nos encontramos metodologías como: la proyección del clustering difuso, el backpropagation de las redes neuronales, utilización de un ATMS, definición y uso de ventanas sobre el dominio de las variables,...

Nosotros proponemos el uso de los algoritmos genéticos para obtener el conjunto de reglas difusas IF-THEN que identifican un sistema, a partir de un conjunto de ejemplos que muestran el funcionamiento de dicho sistema.

2 PRELIMINARES

El problema será encontrar un conjunto finito de reglas fuzzy capaz de reproducir el comportamiento de un sistema. Las reglas tendrán la siguiente forma:

R_j : If X_1 is A_{1j} and ... and X_n is A_{nj} then Y_1 is B_{1j} and ... and Y_m is B_{mj} .

donde R_j ($j=1, \dots, p$) nota la j -ésima regla, p es el número de reglas fuzzy, A_{ij} y B_{ij} son conjuntos fuzzy, X_i son las variables de entrada, e Y_i son las variables de salida.

Para obtener el conjunto de reglas se parte de un conjunto $\varepsilon = \{E_1, E_2, \dots, E_m\}$ de ejemplos donde $E_i = (e_{i1}, \dots, e_{in}; e_{in+1})$ siendo $(e_{i1}, \dots, e_{in}) \in \mathcal{P}(D_{X_1} \times \dots \times D_{X_n})$ los valores de las variables de entrada y $e_{in+1} \in D_s$ la salida, siendo $\mathcal{P}$ la función partes de un conjunto y D_{X_i} y D_s los dominios de las variables X_i e Y respectivamente.

Nosotros hemos trabajado con un sistema MISO (múltiples entradas, una sola salida) pero el algoritmo es generalizable a un sistema MIMO, realizando algunos cambios.

En este artículo se presenta un acercamiento al aprendizaje automático a partir de ejemplos utilizando las ideas de los algoritmos genéticos y el modelado de sistemas fuzzy.

3 APRENDIENDO REGLAS FUZZY CON LA AYUDA DE LOS ALGORITMOS GENETICOS

En este artículo nosotros proponemos un procedimiento para generar reglas fuzzy que modelan un sistema, a partir de un conjunto de ejemplos que muestran el funcionamiento de dicho sistema. La idea es la siguiente generar un conjunto de reglas iniciales y dejar que evolucionen hasta que se obtengan un conjunto de reglas maximales o definitivas.

Para obtener el conjunto de reglas iniciales, nosotros nos vamos a servir del conjunto de ejemplos de entrenamiento. Cada ejemplo será convertido en una regla en la que el valor de cada variable de entrada será representada por medio de un conjunto fuzzy y la salida será codificada o representada también por medio de un conjunto fuzzy dependiendo del dominio de la variable de salida.

Ejemplo 3.1: Sea $(x_1, x_2, x_3, x_4 : A)$ un ejemplo del conjunto de entrenamiento donde x_1, x_2, x_3 y x_4 son valores concretos de las variables de entrada y A la respuesta del sistema a esa entrada. Este ejemplo será convertido en una regla de la forma $((L_1, L_2, L_3, L_4), y_i)$ que es una forma abreviada de:

If X_1 is L_1 and X_2 is L_2 and X_3 is L_3 and X_4 is L_4
then Y is y_i

donde L_1, L_2, L_3, L_4 son conjuntos fuzzy e y_i puede ser un código o un conjunto fuzzy dependiendo del problema. $\square$

Así pues el conjunto de reglas iniciales será dividido en conjuntos fuzzy, cada conjunto tendrá asociado una función de pertenencia: μ_j . Cuando nosotros construimos las reglas iniciales a partir de los ejemplos de entrenamiento lo que hacemos es ver a qué conjunto fuzzy se adapta mejor cada valor de concreto x_i de la variable de entrada en el ejemplo. Es decir:

$$L_j = \max_j \{\mu_j(x_i)\}$$

Si a la variable de salida se le va a asociar un conjunto fuzzy se actúa de la misma forma.

Una vez que tenemos las reglas iniciales entra en juego la idea de la evolución, es decir, el proceso de desarrollar las reglas iniciales hasta que estas alcancen cierto grado de evolución en unas reglas definitivas y más o menos maximales. Para esto nosotros vamos a utilizar los algoritmos genéticos. La idea básica antes de entrar en la formulación definitiva del procedimiento es la siguiente:

3.1 JUSTIFICACION TEORICA DEL PROCEDIMIENTO

Nosotros vamos a tener tantas especies como valores tenga la variable de salida (ya sean valores codificados

o conjuntos difusos). La supervivencia de cada especie considerada como un todo está garantizada, ya que los individuos que la forman van a producir otros que van a reemplazar a los que vayan desapareciendo. Esto se va a conseguir mediante el proceso reproductor.

La reproducción se realizará siempre entre individuos de la misma especie. Esta idea no es descabellada, ya que podemos pensar en la reproducción en un ecosistema (varias poblaciones conviviendo en el mismo habitat), obviamente la reproducción se produce entre individuos de la misma especie no entre individuos de distintas especies.

La capacidad reproductora de cada individuo va a depender de dos aspectos:

- Su adaptación al medio (se tasará con la función de evaluación).
- La compatibilidad con el individuo con el que se va a reproducir (es decir, el futuro de sus crías, si este enlace va a producir seres débiles por qué permitirlo).

Otro aspecto que vamos a tener en cuenta es el de la renuncia a la reproducción, es decir un individuo descubre que no es un buen miembro de la especie y renuncia a su capacidad reproductora. Esta idea tampoco es original, ya que en Nueva York, un grupo de judíos ortodoxos llamado *Dor Yeshorim* ("generación de los virtuosos") ha estado aplicando pruebas genéticas desde 1983 para encontrar enfermedades hereditarias. De 8.000 hombres sometidos a estas pruebas, 67 renunciaron al matrimonio a causa de los resultados.

Los individuos de cada especie van a ser asexuados, presentando todos los individuos las mismas células sexuales o gametos, es decir y en términos científicos va a existir isogamia. Cuando más adelante entremos en la definición formal de nuestro procedimiento veremos la estructura de los gametos.

Con respecto a las mutaciones, tengamos en cuenta la siguiente cuestión: Es "realmente interesante" que se produzcan mutaciones en nuestra población, cuando una mutación sobre un gen inadecuado puede producir una confusión sobre la pertenencia del individuo mutado a una especie u otra. La respuesta parece obvia: No. Si bien las mutaciones son importantes en el proceso evolutivo, parece más o menos coherente en nuestro problema garantizar la inmutabilidad de algunos genes.

En el mundo real se demuestra que las mutaciones se presentan con escasa frecuencia, si bien esta varía mucho según la especie y el gen de que se trate. Hay pues genes que mutan con relativa facilidad (genes inestables), mientras que otros lo hacen con una frecuencia muy escasa (genes estables).

3.2 RESOLUCION MEDIANTE UN ALGORITMO GENETICO

El problema de la modelización de un sistema mediante un conjunto de reglas fuzzy puede verse como un problema de optimización con restricciones, partimos de un conjunto de reglas iniciales que modelan el sistema (los ejemplos de entrenamiento) y mediante un proceso evolutivo con restricciones pretendemos optimizar el conjunto de reglas iniciales para obtener el conjunto de reglas finales que simulan el comportamiento del sistema.

Para llevar a cabo ese proceso evolutivo nosotros hemos pensado en los algoritmos genéticos, a los cuales nosotros les hemos añadido alguna información adicional (para manejar nuestras restricciones).

3.2.1 ESPECIFICACIONES DE ENTRADA

La estructura de los gametos (a partir de ahora los llamaremos genotipos) de nuestros individuos van a ser una cadena de elementos de alfabeto $A=\{0,1,\#\}$. La longitud de la cadena depende del número de conjuntos difusos que se empleen para dividir el dominio de cada variable de entrada y de la longitud del código utilizado para la codificación de la salida o del número de conjuntos difusos en los que se divide el dominio de la variable de salida. De esta forma:

$$Length(cadena) = \sum_{i=0}^n |L_i| + \begin{cases} length(codigosalida) \\ o \\ |L_s| \end{cases}$$

donde $|L_i|$ es el número de conjuntos difusos en los que se ha dividido el dominio de la variable X_i , $|L_s|$ es el número de conjuntos difusos en los que se ha dividido el dominio de la variable de salida y $length(código de salida)$ es la longitud del código utilizado para codificar la salida.

Ejemplo 3.2: Si tenemos tres variables de entrada y sus dominios están divididos en los siguientes conjuntos difusos: X_0 en cinco conjuntos fuzzy, X_1 y X_2 en siete y la salida es codificada mediante un código de longitud uno, la longitud de la cadena será:

$$Length(cadena) = 5 + 7 + 7 + 1 = 20 \square$$

Como puede observarse en la fórmula antes mostrada nuestros genotipos tienen dos partes netamente distinguible: un **núcleo** que contiene información sobre el individuo y una **cabeza** que contiene información sobre la especie a la que pertenece (la salida).

La cabeza va a ser útil para satisfacer una de nuestras restricciones; "la reproducción va a ser un acto realizado con cabeza", es decir, la reproducción sólo va a ser posible entre elementos con la misma cabeza (elementos de la misma especie).

Nuestro objetivo será conseguir un número de reglas lo más generales y maximales posibles que simulen

el comportamiento del sistema (que es mostrado por el conjunto de ejemplos de entrenamiento). Así pues nuestras reglas se van a tener que optimizar y como consecuencia reproducir, en función de estas dos características.

La primera población de las distintas especies se obtendrá como se indicó al comienzo del apartado 3, introduciendo esa información obtenida en nuestros genotipos (uno por ejemplo).

Ejemplo 3.3: Si tenemos tres variables y sus dominios han sido divididos en los siguientes conjuntos fuzzy:

$$\begin{aligned} X_0 &\rightarrow \mathcal{L}_0 = \{N, Z, P\} \\ X_1 &\rightarrow \mathcal{L}_1 = \{HN, MN, SN, Z, SP, MP, HP\} \\ X_2 &\rightarrow \mathcal{L}_2 = \{HN, SN, Z, SP, HP\} \end{aligned}$$

y la salida es codificada con un número de un dígito. De acuerdo con esta información, nuestros genotipos tendrán la forma que se muestra en la Figura 1.

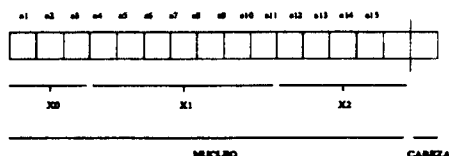


Figure 1: Genotipo.

Si hemos obtenido la regla inicial ((Z,MP,HN),3) como se indicó en el comienzo del apartado 3, esta se codificará como el genotipo que se muestra en la Figura 2. □

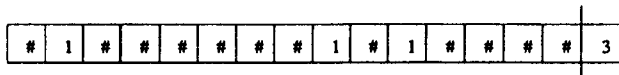


Figure 2: Genotipo del ejemplo.

El resto de valores se ponen a #, para indicar que en principio pueden tomar cualquier valor (0 ó 1).

Las mutaciones en el gen que distingue entre las especies (cabeza) van a ocurrir con probabilidad 0. es decir, es totalmente improbable que un individuo mute de una especie a otra.

Estos genotipos obtenidos son pre-procesados antes de ser ejecutado sobre ellos el proceso evolutivo. El pre-proceso consiste en:

Ver para cada genotipo generados posibles *subsumisiones simples* con miembros de otras especies. Aquellos genes que hagan la subsumisión simple posible se hacen distinguibles.

Definición 3.1. Decimos que entre dos genotipos hay *subsumisión simple* cuando sólo se diferencian en dos genes del núcleo, siendo un gen determinado y el otro libre y sus cabezas son diferentes.

Ejemplo 3.4: En el mismo contexto descrito en el ejemplo 3.3, tenemos en nuestro ecosistema los individuos cuyos genotipos se muestran en la Figura 3.

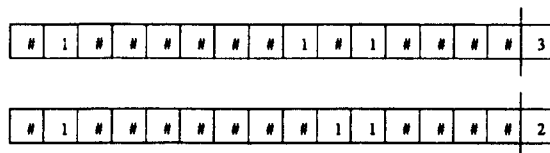


Figure 3: Individuos en los que se produce subsumisión simple.

Como podemos observar se diferencian en los genes 9 y 10 del núcleo. Diferenciamos, y obtenemos los individuos cuyos genotipos aparecen en la Figura 4. □

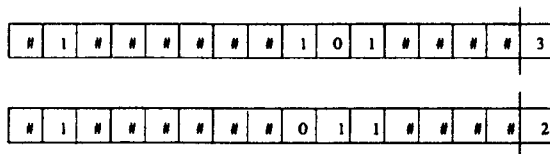


Figure 4: Individuos después de hacer diferenciación.

Se puede observar que la diferenciación consiste en invertir (poner a cero) en cada genotipo el gen libre que produce la subsumisión simple.

3.2.2 ESPECIFICACIONES DE SALIDA

La salida del algoritmo genético propuesto será un conjunto de reglas maximales y generales, que maximizan la función de evaluación. De este conjunto nosotros seleccionaremos el menor número de reglas que cubren el conjunto de ejemplos de entrenamiento. Esta será la salida de nuestro procedimiento de aprendizaje inductivo.

3.2.3 FUNCIONAMIENTO DEL PROGRAMA

El conjunto de genotipos de entrada será el obtenido después del pre-proceso antes mencionado. La función de evaluación que va a asignar un valor a cada genotipo (o individuo) va a medir cómo se adapta ese individuo al ambiente. Nuestro ambiente va venir determinado por el conjunto de ejemplos de entrenamiento. De esta forma una función de evaluación apropiada para el problema es aquella que nos premie las reglas (o individuos que están evolucionando) que mayor número de ejemplos clasifique cometiendo el menor número de errores. Es decir, una función $f(e_p, e_n)$ que sea creciente en cuanto al número de ejemplos que clasifica correctamente (e_p) y decreciente en cuanto al número de ejemplos que clasifica erróneamente (e_n). Una función ejemplo podría ser:

$$\phi = f(e_p, e_n) = \frac{n \text{ total ejemplos que clasifica} - \text{errores cometidos}}{n \text{ total de ejemplos de su clase}}$$

$$\phi = \frac{\text{n total ejemplos que clasifica} - \text{errores cometidos}}{\text{n total de ejemplos de su clase}} \times \lambda + \frac{\text{n de genes determinados} \times (1 - \lambda)}{\text{n de genes totales}}$$

$$p = \frac{\phi(A_i)}{\sum_{k=1}^N \phi(A_k)}$$

87

realidad este gasto no va a ser tan grande puesto que al admitir sólo la reproducción entre individuos de la misma especie, la mayoría de las veces la reproducción va a mejorar a los padres (por la definición del operador propuesto). Este cálculo sólo va a tener sentido en las primeras selecciones para evitar que aquellos ejemplos que introducen ruido, se reproduzcan propagando así el ruido.

Mutación:

Con este operador nosotros vamos a intentar evitar que la población de individuos (o conjunto de genotipos) pierdan un valor específico de un gen.

La mutación se producirá en cualquier gen del cuerpo, no permitiéndose las mutaciones en la cabeza (los cambios de especie por mutación), ni en los genes determinados (serán genes estables).

Para todos los genotipos A de $\beta(t)$ reemplazamos todos los genes con probabilidad de mutación $p_{mut}=0.005$. Cuando se produce una mutación comprobamos si es maligna, es decir va a empeorar la función ϕ del individuo. Si esto ocurre no parece justo que nosotros que podemos evitarla en este contexto no lo hagamos. Así pues, cuando se produzca una mutación maligna, nosotros la corregimos o no la permitimos.

Definición 3.5. Decimos que una mutación de un ser es posible cuando se verifica que:

$$\phi(\text{ser mutado}) \geq \phi(\text{ser no mutado})$$

Cabe hacer aquí la misma observación hecha anteriormente referente al gasto computacional ¿ Por qué calcular la función de evaluación de mutaciones que se van a descartar ? Este gasto es compensable ya que el introducir ruido (mutaciones malignas) en la población va a hacer que introduzcamos inestabilidad en la población haciendo necesario un tiempo adicional para volver a conseguir la estabilidad (una población sin ruido). Las mutaciones malignas no afectan en principio a un problema de búsqueda de una solución, e incluso puede ser beneficioso ya que puede conducir a nuevas zonas de búsqueda. Sin embargo al problema de optimización de un conjunto si le afecta (aumenta el tiempo), debido a esa ruptura de la estabilidad.

3.3 ALGORITMO DE APRENDIZAJE

El algoritmo de aprendizaje propuesto es el siguiente:

Paso 1:

Convertir los ejemplos de entrenamiento en reglas iniciales.

Paso 2:

Pre-procesar esas reglas iniciales.

Paso 3:

Aplicar el AG con la población obtenida en el paso 2 y con el ambiente que establece los ejemplos de entrenamiento.

Paso 4:

De la población final que devuelve el AG, nosotros nos quedamos con el menor número de reglas que cubren todos los ejemplos de entrenamiento.

References

- [1] Abe, S., and M.S. Lan (1995). "A method for fuzzy rules extraction directly from numerical data and its application to pattern classification". *IEEE Transactions on Fuzzy Systems*, 3, 1.
- [2] Bains, W. Ingeniería Genética para todos. Alianza Editorial, 1991.
- [3] Campos, L.M. de, and S. Moral (1993). "Learning rules for a fuzzy inference Model". *Fuzzy Sets and Systems*, 59.
- [4] Castro, J.L, and J.M. Zurita (1995). "Generating the structure of a fuzzy rule under uncertainty". Proceedings of the Eleventh Conference in Uncertainty in Artificial Intelligence, Montreal.
- [5] Delgado, M., A.F. Gomez-Skarmeta y F. Martín (1995). "Un enfoque aproximativo para la generación de reglas mediante Análisis Cluster". Proceedings del Quinto Congreso sobre Tecnología y lógica Fuzzy, Murcia.
- [6] Delgado, M., and A. González (1993). "A inductive learning procedure to identify fuzzy systems". *Fuzzy Sets and Systems*, 55.
- [7] Dualde Perez, V. Biología. E. Lopez Mezquida Editor, Valencia, 1982.
- [8] Gribbin, J. In Search of the Double Helix. Transworld Publishers, Londres, 1985.
- [9] Herrera, F., M. Lozano, J.L. Verdegay (1994). "Algoritmos Genéticos: Fundamentos, Extensiones y Aplicaciones". Technical Report #DECSAI 94105
- [10] Hoagland, M.B. The roots of the life. Houghton Mifflin Company, Boston, 1985.
- [11] Nomura, H., I. Hayashi, and N. Wakami (1992). "A learning method of fuzzy inference rules by descent method". Proceedings of the IEEE International Conference on Fuzzy Systems.
- [12] Wang and Mendel (1992). "Generating Fuzzy Rules by learning from examples". *IEEE Transactions on S.M.C.*, 22, 6.
- [13] Yager R., and D.P. Filew. Essencials of fuzzy modeling and control. John Wiley and sons (eds.), Wiley, New York, 1994.

UN PROCESO EVOLUTIVO PARA EL APRENDIZAJE DE BASES DE CONOCIMIENTO DE CONTROLADORES DIFUSOS *

Oscar Cordon
Dpto. de Ciencias de la
Computación e I.A.
E.T.S. de Ingeniería Informática
Universidad de Granada
18071 - Granada
e-mail: ocordon@robinson.ugr.es

Francisco Herrera
Dpto. de Ciencias de la
Computación e I.A.
E.T.S. de Ingeniería Informática
Universidad de Granada
18071 - Granada
e-mail: herrera@robinson.ugr.es

Resumen

El objetivo de este trabajo es presentar un proceso evolutivo que combina los algoritmos genéticos y las estrategias de evolución en el aprendizaje de bases de conocimiento de controladores difusos a partir de ejemplos.

Palabras clave: Controladores difusos, bases de conocimiento, aprendizaje, algoritmos genéticos, estrategias de evolución.

1 INTRODUCCIÓN

Los controladores difusos (CDs) [6] son considerados como una de las aplicaciones más importantes de la teoría de conjuntos difusos. Hacen uso del conocimiento existente sobre el proceso, expresado en forma de reglas lingüísticas de control recopiladas en una base de conocimiento (BC), para controlar éste. Esta forma de trabajo los hace muy adecuados para manejar sistemas reales con características no lineales o dinámica compleja.

Es necesario llevar a cabo una serie de tareas para diseñar un CD para una aplicación concreta. Una de las más importantes y complejas es la extracción del conocimiento experto disponible sobre el proceso a controlar. Puesto que en muchos casos los operadores humanos no son capaces de expresar su conocimiento en forma de reglas de control, se han desarrollado técnicas automáticas para llevar a cabo esta tarea. Los algoritmos genéticos (AGs) [7] son la base de muchas de ellas en los últimos años [2]. En este trabajo presentamos un proceso evolutivo híbrido que hace uso de los AGs y las estrategias de evolución (EEs) [10] para aprender la BC del CD a partir de ejemplos, que se basa en los tres componentes siguientes, manteniendo la estructura genérica presentada en [9]:

1. Un *proceso genético de generación* que genera reglas difusas de control, compuesto por un método generador basado en algoritmos evolutivos y un método iterativo de cubrimiento del conjunto de ejemplos del comportamiento del sistema. Este proceso permite obtener un conjunto que reglas

que cubran al conjunto de ejemplos de una forma adecuada.

2. Un *proceso genético de simplificación* que simplifica reglas, basado en un AG binario y una medida de rendimiento del CD en el control del sistema que está siendo identificado. Este componente permite eliminar reglas redundantes.
3. Un *proceso genético de ajuste*, basado en un AG con codificación real, que permite obtener la BC final ajustando las funciones de pertenencia de cada regla.

El trabajo se estructura de la siguiente forma. La sección 2 muestra unos preliminares sobre el tipo de BC considerada. La sección 3 presenta en profundidad el proceso de generación mientras que la sección 4 introduce los dos componentes restantes. Finalmente, incluimos un ejemplo de aplicación del método en la sección 5.

2 PRELIMINARES

En este trabajo nos centramos en el modelo de Mamdani para sistemas con varias entradas y una única salida, MISO, en el que la BC del controlador está formada por una serie de reglas difusas de control que describen las acciones de control de la siguiente forma:

R_i : SI x_1 es A_{i1} y ... y x_n es A_{in} ENTONCES y es B

donde $x_1, \dots, x_n$ e y son las variables de entrada y la de salida del proceso, respectivamente; y $A_{i1}, \dots, A_{in}$, B son conjuntos difusos definidos en los universos de discurso $U_1, \dots, U_n, V$; caracterizados por sus funciones de pertenencia

$$A_{ij}(B) : U_j(V) \rightarrow [0, 1], j = 1, \dots, n.$$

En nuestro estudio trabajaremos con funciones de pertenencia triangulares, representados por 3-tuplas de la forma (a_{ij}, b_{ij}, c_{ij}) , (a_i, b_i, c_i) , $j = 1, \dots, n$.

La figura 1 muestra un ejemplo del tipo de partición difusa empleada en la que las funciones de pertenencia intersecan a altura 0.5. El número de conjuntos difusos que forman la partición difusa asociada a cada una de las variables del problema puede ser especificado por el diseñador del CD con objeto de obtener el nivel de granularidad deseado.

Trabajo realizado en el proyecto DGICYT PB92-0933

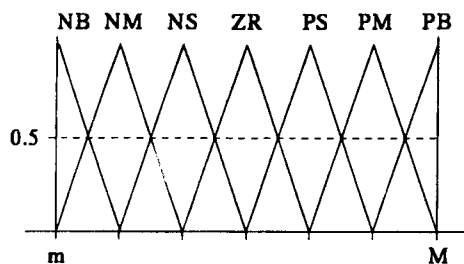


Figura 1: Partición difusa genérica

A cada uno de los tres puntos que definen las funciones de pertenencia $A_t(\cdot)$, (a_t, b_t, c_t) , se le asocia un intervalo de variación basado en la composición de la partición difusa previa para ajustar localmente la forma de la función de pertenencia durante el proceso de generación:

$$\begin{aligned} [a_t^l, a_t^r] &= [a_t - \frac{b_t - a_t}{2}, a_t + \frac{b_t - a_t}{2}] \\ [b_t^l, b_t^r] &= [b_t - \frac{b_t - a_t}{2}, b_t + \frac{c_t - b_t}{2}] \\ [c_t^l, c_t^r] &= [c_t - \frac{c_t - b_t}{2}, c_t + \frac{c_t - b_t}{2}] \end{aligned}$$

De este modo, las reglas difusas de control generadas mantendrán su semántica dentro de los intervalos establecidos por las funciones de pertenencia de la partición inicial. Cada regla presenta así una semántica específica para cada etiqueta lingüística, lo que constituye un enfoque *aproximativo* [2] en contra del habitual enfoque *descriptivo* que caracteriza a las BC de los CDs de tipo Mamdani.

Por otro lado, para generar un conjunto de reglas R que describa el comportamiento de un sistema, es necesario exigir el cubrimiento de todos los posibles pares situación-acción, $e_i \in E_p$, es decir, la *propiedad de completitud* [6]. Dicha propiedad requiere la unión no vacía de los conjuntos difusos $A_i(\cdot), B_i(\cdot)$, $i = 1, \dots, T$, $T = |R|$, y puede ser formalizada haciendo uso de una constante $\tau \in [0, 1]$, mediante las expresiones siguientes:

$$\begin{aligned} C_R(e_l) &= \bigcup_{i=1..T} R_i(e_l) \geq \tau, \quad l = 1, \dots, p \\ R_i(e_l) &= *(A_i(ex^l), B_i(ey^l)) \\ A_i(ex^l) &= *(A_{i1}(ex_1^l), \dots, A_{in}(ex_n^l)) \end{aligned}$$

donde $*$ es una t-norma y $R_i(e_l)$ es el *grado de compatibilidad* entre la regla R_i y el ejemplo e_l .

Dado un conjunto de reglas R , el *grado de cubrimiento* de un ejemplo e_l se define como

$$CV_R(e_l) = \sum_{i=1}^T R_i(e_l)$$

y nuestro proceso requiere la siguiente condición

$$CV_R(e_l) \geq \epsilon \quad l = 1, \dots, p$$

Un buen conjunto de reglas debe satisfacer las dos condiciones presentadas, verificar la propiedad de completitud y poseer un grado de cubrimiento adecuado.

3 EL PROCESO GENÉTICO DE GENERACIÓN

Como hemos indicado en la introducción, el primer proceso está compuesto por los dos métodos siguientes, que serán descritos a continuación:

1. El método generador se basa en un AG que codifica una regla en cada cromosoma. En cada ejecución, el AG encuentra la mejor regla con respecto al estado actual del conjunto de ejemplos, de acuerdo con una serie de criterios incluidos en la función de adaptación. Se hace uso de una EE para ajustar localmente las mejores reglas difusas de control obtenidas en cada interacción de la búsqueda genética.
2. El método de cubrimiento es un proceso iterativo que permite obtener un conjunto de reglas difusas que cubran el conjunto de ejemplos. En cada iteración, obtiene la mejor regla actual a partir del método generador, estudia el cubrimiento relativo que efectúa dicha regla sobre el conjunto de ejemplos y elimina del mismo aquellos ejemplos cubiertos en un grado superior a un valor ϵ especificado por el diseñador del controlador.

3.1 EL MÉTODO GENERADOR DE REGLAS DIFUSAS

3.1.1 REPRESENTACIÓN

Cada cromosoma C que representa una regla candidata está compuesto por dos partes, C_1 and C_2 , codificando dos niveles de información diferentes, cada uno asociado a un componente de la BC. Mientras que C_1 codifica la regla lingüística (perteneciente a la base de reglas), C_2 representa la semántica asociada a cada término lingüístico de la misma (perteneciente a la base de datos).

Con objeto de poder representar la primera parte, es necesario enumerar ordenadamente las etiquetas de cada conjunto de términos. Así, a la variable x_i que toma valores en el conjunto de términos $T(x_i) = \{L_1(x_i), \dots, L_{n_i}(x_i)\}$ se le asocia el conjunto ordenado $T'(x_i) = \{1, \dots, n_i\}$.

Por otro lado, la segunda parte emplea la misma representación usada en el método generador con semántica libre no restringida presentado en [9]. Cada función de pertenencia triangular es codificada mediante su 3-tupla asociada $(a_{L_i(x_j)}, b_{L_i(x_j)}, c_{L_i(x_j)})$.

De este modo, la siguiente regla genérica

$$\begin{aligned} \text{SI } x_1 \text{ es } L_{i_1}(x_1) \dots \text{ y } x_n \text{ es } L_{i_n}(x_n) \\ \text{ENTONCES } y \text{ es } L_{i_{n+1}}(y) \end{aligned}$$

se codifica en un cromosoma C de la forma:

$$\begin{aligned} C_1 &= (i_1, \dots, i_n, i_{n+1}) \\ C_2 &= (a_{L_{i_1}(x_1)}, b_{L_{i_1}(x_1)}, c_{L_{i_1}(x_1)}, \dots, a_{L_{i_n}(x_n)}, \\ &\quad b_{L_{i_n}(x_n)}, c_{L_{i_n}(x_n)}, a_{L_{i_{n+1}}(y)}, b_{L_{i_{n+1}}(y)}, c_{L_{i_{n+1}}(y)}) \\ C &= C_1 C_2 \end{aligned}$$

3.1.2 GENERACIÓN DE LA POBLACIÓN INICIAL

El proceso de generación de la población inicial se lleva a cabo seleccionando aleatoriamente $t = \min\{|E_p|, \frac{M}{3}\}$ ejemplos de E_p (donde M es el tamaño de la población inicial del AG) y según los pasos siguientes:

1. Haciendo uso de las particiones difusas primarias existentes para cada variable, generar t individuos a partir de la regla que mejor cubra los t ejemplos generados aleatoriamente. Inicializar C_1 y C_2 codificando respectivamente las etiquetas y su semántica en el modo arriba descrito.
2. Generar otros t individuos iniciando C_1 al igual que en el paso anterior y calculando aleatoriamente los valores de C_2 , cada uno dentro de su intervalo de variación asociado.
3. Generar los restantes $M - 2 \cdot t$ individuos calculando aleatoriamente los valores de la primera parte, C_1 , y haciendo uso de éstos para generar aleatoriamente los valores de la segunda parte, C_2 , cada uno dentro de su intervalo de variación asociado.

3.1.3 FUNCIÓN DE ADAPTACION

La función de adaptación considerada es una función multiobjetivo basada en varios criterios:

Alta frecuencia [9]: La frecuencia de una regla difusa de control, R_i , sobre un conjunto de ejemplos, E_p , se define de la forma:

$$\Psi_{E_p}(R_i) = \frac{\sum_{l=1}^p R_i(e_l)}{p}$$

Alto grado de cubrimiento medio sobre los ejemplos positivos [9]: El conjunto de ejemplos positivos para la regla R_i con grado de compatibilidad mayor o igual que ω se define de la forma:

$$E_{\omega}^{+}(R_i) = \{e_l \in E_p / R_i(e_l) \geq \omega\}$$

con $n_{\omega}^{+}(R_i)$ igual a $|E_{\omega}^{+}(R_i)|$. El *grado de cubrimiento medio* sobre $E_{\omega}^{+}(R_i)$ puede definirse como:

$$G_{\omega}(R_i) = \sum_{e_l \in E_{\omega}^{+}(R_i)} R_i(e_l) / n_{\omega}^{+}(R_i)$$

Pequeño conjunto de ejemplos negativos [4]: El conjunto de ejemplos negativos para R_i se define como:

$$E^{-}(R_i) = \{e_l \in E_p / R_i(e_l) = 0 \text{ and } A_i(ex^l) > 0\}$$

Un ejemplo se considera negativo para una regla si casa mejor con otra regla que posea el mismo antecedente pero distinto consecuente. Los ejemplos negativos siempre se consideran sobre el conjunto de entrenamiento al completo.

Notando por $n_{R_i}^{-} = |E^{-}(R_i)|$ al número de ejemplos negativos, la *función de penalización sobre el conjunto de ejemplos negativos* viene representada por la expresión:

$$g_n(R_i^{-}) = \begin{cases} 1 & \text{if } n_{R_i}^{-} \leq k \cdot n_{\omega}^{+}(R_i) \\ \frac{1}{n_{R_i}^{-} - k n_{\omega}^{+}(R_i) + \exp(1)} & \text{otherwise} \end{cases}$$

en la que se permite un número de ejemplos negativos por regla menor o igual a un tanto por ciento del número de ejemplos positivos, $k \cdot n_{\omega}^{+}(R_i)$, sin que exista penalización alguna. Este porcentaje viene determinado por el parámetro $k \in [0, 1]$.

Bajo ratio de interacción entre nichos [4]: Sean $N_i = (N_i x, N_i y)$ los centros de las reglas (nichos) generadas hasta el momento ($i = 1, \dots, d$, donde d es el número de iteraciones realizadas del método de cubrimiento) y C un individuo de la población actual, el *ratio de interacción entre nichos* penaliza la adaptación de C del siguiente modo:

$$LNIR(C) = 1 - NIR(C)$$

$$NIR(C) = \text{Max}_i \{h_i\}$$

$$h_i = *(A(N_i x), B(N_i y)), i = 1, \dots, d$$

$$A(N_i x) = *(A_1(N_i x_1), \dots, A_n(N_i x_n))$$

$$C \sim R_i : \text{IF } x_1 \text{ is } A_1 \text{ and } \dots \text{ and } x_n \text{ is } A_n \text{ THEN } y \text{ is } B$$

De este modo, $LNIR(C)$ está definido en $[0, 1]$. Devuelve el valor máximo 1 (no penaliza) cuando la regla codificada en C no interactúa con ninguna de las reglas generadas hasta ahora. El valor mínimo 0 (equivalente a la penalización máxima) se obtiene cuando dicha regla es igual que alguna de las generadas anteriormente.

Así, la combinación del esquema de nichos y el método de cubrimiento permite que el método de generación verifiquen los dos aspectos fundamentales siguientes:

- asegura la obtención de reglas difusas de control en todas las zonas del espacio en las que el problema de control esté definido, es decir, en todas las zonas en las que existan ejemplos. Con ello se verifica la completitud del conjunto de reglas obtenido.
- Asimismo, mantiene una distribución adecuada de las reglas en cada nicho que se defina en el espacio de soluciones. Puesto que el buen funcionamiento de los CDs se debe al razonamiento interpolativo que llevan a cabo y normalmente actúa más de una regla ante un estado concreto del proceso, es necesario que exista un ratio de interacción adecuado entre las reglas vecinas para permitir al CD obtener la mejor acción de control posible para ese estado. Cuando las reglas están demasiado próximas en el espacio de control provocan una excesiva interacción, lo que da lugar a que la acción de control se desplace con respecto a la óptima, mientras que cuando están demasiado lejanas hacen que el CD pierda su capacidad de interpolación, lo que provoca también un mal funcionamiento.

La función de adaptación para la regla R_i se define de la forma:

$$Z(R_i) = \Psi_{E_p}(R_i) \cdot G_w(R_i) \cdot g_n(R_i^-) \cdot LNIR(R_i)$$

con el objetivo de ser maximizada.

3.1.4 OPERADORES GENÉTICOS

Debido a la naturaleza especial de los cromosomas que se manejan en el método generador, es necesario diseñar operadores genéticos específicos para el problema. Para una descripción más detallada de los mismos véase [4].

Mutación: Se emplean dos operadores distintos, cada uno actuando sobre una parte del cromosoma. Puesto que en C_2 se efectúa una codificación real directa de las funciones de pertenencia, tal y como se hacía en [9], emplearemos sobre esta parte de cromosoma el mismo operador de mutación que se usaba en ese trabajo, la mutación no uniforme de Michalewicz. Por otro lado, en C_1 trabajaremos con un operador de mutación similar al de Thrift [11], basado en modificar el valor del gen mutado cambiándolo aleatoriamente por el término lingüístico inmediatamente anterior o posterior en el conjunto de términos. Si el valor que va a ser mutado es el primero o el último del conjunto, se efectúa el único cambio posible. Naturalmente, una mutación en C_1 provoca un cambio en C_2 . Cuando cambia el término asociado a una variable lingüística, los parámetros de la función de pertenencia que define su semántica son automáticamente actualizados en la segunda parte del cromosoma, tomando los valores iniciales definidos por la partición difusa primaria considerada.

Cruce: También son empleados dos operadores de cruce en el proceso de recombinación, los cuales son aplicados dependiendo de la información que codifiquen los dos padres que van a ser cruzados. Si ambos representan la misma regla (con distinta semántica asociada), se efectúa una explotación de esa zona del espacio manteniendo los valores de C_1 en los descendientes y aplicando el operador de cruce max-min-aritmético en C_2 . Cuando los padres codifiquen reglas distintas, la aplicación de este operador no tendría sentido puesto que daría lugar a la obtención de dos descendientes sin sentido. En ese caso, se aplica el operador de cruce estándar en las dos partes del cromosoma, es decir, se genera un punto de cruce aleatorio en C_1 , cp , y se cruzan los dos padres a partir de ese punto en C_1 y a partir del punto correspondiente, $n + 1 + 3 \cdot cp$ (puesto que hay $n + 1$ variables en el problema y 3 puntos por función de pertenencia), en C_2 .

Estrategia de Evolución: El último operador genético a emplear es una $EE-(1+1)$. Integramos esta técnica evolutiva de optimización en el proceso de obtención de la nueva población del AG con objeto de llevar a cabo un ajuste local de los mejores individuos de la población (reglas) en cada generación. En cada iteración del AG, este operador ajusta la segunda parte, C_2 , de un porcentaje α de los mejores individuos (sin tener en cuenta los repetidos) de la población actual. Las bases de la EE empleada se encuentran en [1, 4]. En nuestro caso, el parámetro σ que determina la potencia de la mutación no puede ser un único valor puesto que cada función de pertenencia codificada en C_2 está definida sobre un dominio

distinto. De este modo, siguiendo el modo de trabajo presentado en [1], cada componente del individuo padre x_i que toma valores en el intervalo de variación $[x_i^l, x_i^r]$ tendrá asociado su propio parámetro $\sigma_i = \sigma \cdot s_i$ (aunque únicamente sea adaptado en cada iteración el valor del parámetro σ global), donde $s_i = \frac{x_i^r - x_i^l}{4}$. Cuando el valor mutado $x_i' = x_i + z_i$ se salga del intervalo de actuación por uno de los dos extremos, x_i^l o x_i^r , se asocia el valor de dicho extremo a x_i' .

Selección: El mecanismo de selección se basa en el muestreo universal estocástico de Baker y el esquema de selección elitista.

3.2 EL MÉTODO DE CUBRIMIENTO

El método de cubrimiento fue presentado en [9] y consta de los siguientes pasos:

1. *Inicialización:*
 - (a) Introducir los valores de k , ω y ϵ .
 - (b) Inicializar $CV[l] \leftarrow 0$, $l = 1, \dots, p$.
 - (c) Inicializar el conjunto final de reglas $B^g \leftarrow \emptyset$
2. Aplicar el método generador sobre el conjunto de ejemplos E_p , obteniendo así la mejor regla difusa R_r . Introducir R_r en R^g .
3. Para cada ejemplo $e_l \in E_p$ hacer
 - (a) $CV[l] \leftarrow CV[l] + R_r(e_l)$,
 - (b) Si $CV[l] \geq \epsilon$ entonces eliminar e_l de E_p .
4. Si $E_p = \emptyset$ entonces Parar si no volver al paso 2.

4 LOS PROCESOS GENÉTICOS DE SIMPLIFICACIÓN Y AJUSTE

Debido a la naturaleza iterativa del proceso genético de generación de reglas, puede ocurrir que se produzca un fenómeno de sobreaprendizaje. Esto tendrá lugar cuando algunos ejemplos hayan sido cubiertos en un grado superior al deseado y provocará que la base de reglas obtenida contenga reglas redundantes. Para solucionar este problema, es necesario simplificar el conjunto de reglas obtenido en el proceso anterior con objeto de obtener la base de reglas final.

El proceso de simplificación (descrito en [9]) se basa en un AG que representa el conjunto de reglas obtenido hasta el momento en un cromosoma de modo que un 1 refleja la existencia de esa regla en la base final y un 0, la ausencia de la misma. La función de evaluación emplea una medida de error, el error cuadrático medio, en combinación con un modificador que penaliza la pérdida de la propiedad de completitud en la base codificada.

El proceso genético de ajuste (presentado en [8]) se basa en representar la BC completa en cromosomas con codificación real de tal modo que cada cromosoma codifica la base de reglas obtenida en el proceso anterior con una semántica distinta asociada. El AG actúa sobre dicha semántica, ajustando la forma de las funciones de pertenencia, permitiendo obtener una BC aproximativa con mejor comportamiento.

Tabla 1: Resultados obtenidos en la función F_1

Parámetros			Generación		Simplificación		Ajuste
ϵ	ω	τ	#R	EC	#R	EC	EC
1.0	0.05	0.25	70	1.9924	58	1.6305	0.6986
1.0	0.05	0.5	70	1.9924	63	1.7709	0.6631
1.5	0.05	0.25	98	2.4114	67	1.7791	0.6968
1.5	0.05	0.5	98	2.4114	73	2.1301	1.1182
W	M		49	4.6518			0.4638

Tabla 2: Resultados obtenidos en la función F_2

Parámetros			Generación		Simplificación		Ajuste
ϵ	ω	τ	#R	EC	#R	EC	EC
1.0	0.05	0.25	251	0.2884	190	0.2446	0.2142
1.0	0.05	0.5	251	0.2884	206	0.2667	0.2307
1.5	0.05	0.25	346	0.2680	232	0.2139	0.1952
1.5	0.05	0.5	346	0.2680	253	0.2321	0.2101
W	M		49	2.0940			0.9700

5 EJEMPLO DE APLICACIÓN DEL PROCESO

Con objeto de analizar el comportamiento del proceso presentado, hemos escogido dos funciones matemáticas n -dimensionales para derivar dos superficies de control tridimensionales a partir de ellas. Las funciones y los universos de discurso de las variables considerados se muestran a continuación. El *modelo esférico*, F_1 , es una función unimodal mientras que la *función de Rastrigin generalizada*, F_2 , es fuertemente multimodal, como puede observarse en sus representaciones gráficas (figuras 2 y 4).

$$F_1(x_1, x_2) = x_1^2 + x_2^2, \\ x_1, x_2 \in [-5, 5], F_1(x_1, x_2) \in [0, 50]$$

$$F_2(x_1, x_2) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2), \\ x_1, x_2 \in [-1, 1], F_2(x_1, x_2) \in [2, 3.5231]$$

Emplearemos dos métodos distintos para llevar a cabo el modelado difuso de estas superficies y compararemos los resultados obtenidos por cada uno de ellos. El primero está basado en la derivación de la base de reglas empleando el esquema propuesto por Wang and Mendel's (WM) [12] y la definición de la base de datos empleando el tercer componente del proceso presentado en este trabajo, el método genético de ajuste [8]. El segundo es el propio proceso de aprendizaje inductivo de BCs presentado.

Para cada función se ha generado experimentalmente un conjunto de entrenamiento formado por 1681 datos uniformemente distribuidos en el espacio tridimensional de definición del problema (se han considerado 41 valores equidistantes para cada una de las dos variables de entrada). Hemos generado aleatoriamente otros dos conjuntos de datos, cuyo tamaño es un porcentaje (un diez por ciento, es decir, 168 datos) del correspondiente de entrenamiento, para emplearlos como conjuntos de evaluación, con lo que se evita cualquier posible dependencia del proceso de aprendizaje con respecto a los conjuntos de entrenamiento.

La base de datos empleada en el método generador hace uso de tres particiones difusas primarias como las mostradas la figura 1.

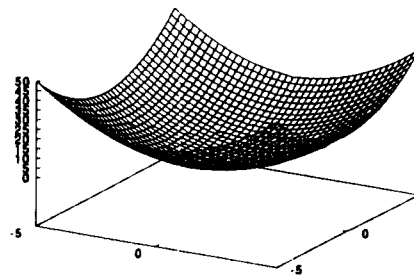


Figura 2: Representación gráfica de F_1

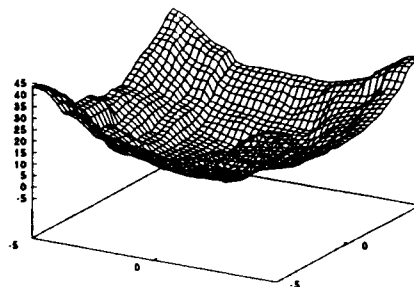


Figura 3: Modelado difuso obtenido de F_1

Se han efectuado distintas ejecuciones del proceso presentado, cuatro por función, empleando para ello valores de los parámetros mostrados en las tablas 1 y 2. La t -norma * empleada es el Mínimo, el método generador se ejecuta durante 50 generaciones, la ES se aplica sobre un porcentaje de individuos de la población, $\alpha = 20\%$, hasta que no se produzca ninguna mejora durante 25 generaciones (el parámetro c de la regla de éxito-1/5 vale 0.9); y los procesos genéticos de simplificación y ajuste se ejecutan durante 500 y 1000 generaciones, respectivamente. En todos los casos, la población está formada por 61 individuos, el valor del parámetro b de la mutación no-uniforme es 5.0, y las probabilidades de cruce y mutación son, respectivamente, $P_c = 0.6$ y $P_m = 0.1$ (esta última por individuo). El parámetro a del cruce max-min-aritmético vale 0.35.

Para finalizar, con respecto al método de razonamiento empleado por el CD, hemos seleccionado la t -norma del Mínimo como operador de implicación y conjunción y el Centro de Gravedad ponderado por el Grado de Emparejamiento como método de defuzzificación [3].

Los resultados obtenidos se muestran en las siguientes tablas en las que #R representa el número de reglas de la BC correspondiente y EC el error cuadrático obtenido por el CD sobre el conjunto de datos de evaluación. La última fila de cada tabla muestra los resultados obtenidos por el proceso basado en WM. Analizando estos resultados, podemos observar el buen comportamiento que presenta nuestro proceso. Todos los modelados obtenidos para la función F_2 superan claramente al basado en el método WM. En lo que respecta a F_1 ,

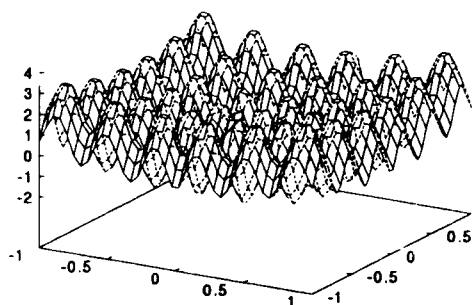


Figura 4: Representación gráfica de F_2

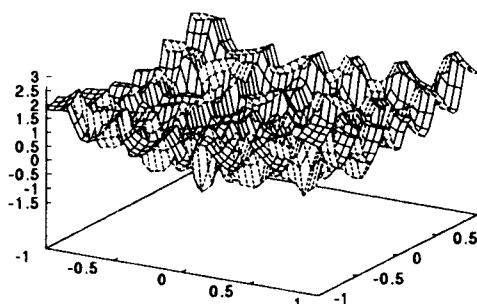


Figura 5: Modelado difuso obtenido de F_2

nuestro proceso mejora claramente los resultados obtenidos por WM cuando no se aplica posteriormente el proceso genético de ajuste, y las BCs obtenidas se comportan sólo un poco peor cuando éste es empleado.

Para ilustrar el comportamiento del proceso propuesto, mostramos dos representaciones gráficas de los modelados difusos obtenidos (figuras 3 y 5). Para dibujarlas, hemos hecho uso de los dos CDs que mejor aproximaban cada una de las funciones, es decir, aquellos que presentaban menor EC en cada una de las funciones.

Referencias

- [1] T. Bäck, H.-P. Schwefel (1995). Evolution strategies I: Variants and their computational implementation. En: J. Periaux, G. Winter, M. Galán, P. Cuesta (Eds.), *Genetic Algorithms in Engineering and Computer Science*. John Wiley and Sons. 111-126.
- [2] O. Cordon, F. Herrera (1995). A general study on genetic fuzzy systems. En: J. Periaux, G. Winter, M. Galán, P. Cuesta (Eds.), *Genetic Algorithms in Engineering and Computer Science*. John Wiley and Sons. 33-57.
- [3] O. Cordon, F. Herrera, A. Peregrín (1996). Applicability of the fuzzy operators in the design of fuzzy logic controllers. *Fuzzy Sets and Systems*. Por aparecer.
- [4] O. Cordon, F. Herrera (1996). A hybrid genetic algorithm-evolution strategy process for learning Fuzzy Logic Control Knowledge Bases. En: F. Herrera, J.L. Verdegay (Eds.), *Genetic Algorithms and Soft Computing*. Physica Verlag. 251-278.
- [5] O. Cordon, F. Herrera (1996). Fuzzy Identification by Means of Genetic Algorithms. *Technical Report DECSAI-96107*. Dpto. de Ciencias de la Computacion e I.A. Universidad de Granada, España.
- [6] D. Driankov, H. Hellendoorn, M. Reinfrank (1993). *An introduction to fuzzy control*. Springer-Verlag.
- [7] D.E. Goldberg (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, New York.
- [8] F. Herrera, M. Lozano, J.L. Verdegay (1995). Tuning fuzzy controllers by genetic algorithms. *International Journal of Approximate Reasoning* 12: 299-315.
- [9] F. Herrera, M. Lozano, J.L. Verdegay (1995). A learning process for fuzzy control rules using genetic algorithms. *Technical Report DECSAI-95108*. Dpto. de Ciencias de la Computacion e I.A. Universidad de Granada, España.
- [10] H.-P. Schwefel (1995). *Evolution and optimum seeking. Sixth-Generation Computer Technology Series*. John Wiley and Sons.
- [11] P. Thrift (1991). Fuzzy logic synthesis with genetic algorithms. *Proc. Fourth International Conference on Genetic Algorithms*. 509-513.
- [12] L.X. Wang, J.M. Mendel (1992). Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics* 22: 1414-1427.

SISTEMA DE CLASIFICACION CON REGLAS DIFUSAS UTILIZANDO ALGORITMOS GENETICOS

Oscar Cordón
Dpto. de Ciencias de la
Computación e I.A.
E.T.S. de Ingeniería Informática
Universidad de Granada
18071 - Granada
e-mail: ocordon@robinson.ugr.es

Maria José del Jesus
Dpto. de Informática
Escuela Politécnica Superior
Universidad de Jaén
23071 - Jaén
e-mail: mjj@apolo.ujaen.es

Francisco Herrera
Dpto. de Ciencias de la
Computación e I.A.
E.T.S. de Ingeniería Informática
Universidad de Granada
18071 - Granada
e-mail: herrera@robinson.ugr.es

Resumen

En este trabajo se propone un método para obtener un sistema de clasificación basado en reglas difusas a partir de ejemplos. El método consta de un proceso iterativo de generación de reglas difusas, un proceso de selección de reglas que actúa sobre conjunto de reglas obtenido en la fase anterior para eliminar reglas redundantes, y un proceso de ajuste de la base de datos (la semántica de las etiquetas). Estos dos últimos se desarrollan utilizando algoritmos genéticos. Se muestra el rendimiento del método propuesto, comparando la capacidad de predicción del sistema de clasificación obtenido para la base de ejemplos Iris, con los resultados obtenidos por los algoritmos C4.5 y CART.

Palabras Clave: Clasificación, Aprendizaje, Algoritmos Genéticos, Reglas difusas.

1 INTRODUCTION

La adquisición de conocimiento es uno de los principales problemas que se encuentran en el desarrollo de sistemas basados en el conocimiento. El aprendizaje inductivo es una de las técnicas automáticas alternativas a la clásica extracción de conocimiento. Este tipo de aprendizaje tiene como objetivo extraer conocimiento de una Base de Datos que contiene ejemplos del comportamiento de un sistema o proceso. Desde esta vertiente, el aprendizaje se puede ver como un proceso de generalización, ya que su finalidad es inducir descripciones generales de objetos a partir de instancias específicas de estos conceptos [14, 16].

Se pueden distinguir dos vertientes principales dentro del aprendizaje inductivo; el aprendizaje a partir de ejemplos (también conocido como adquisición de conceptos) y el aprendizaje a través de la observación o generalización descriptiva.

El presente trabajo se sitúa en el primer área, concretamente en el aprendizaje a partir de ejemplos aplicado a problemas de clasificación. En este tipo de

problemas, partimos de un conjunto de ejemplos de los cuales conocemos la clase a la cual pertenecen. A partir de este conjunto de ejemplos -llamados *ejemplos de entrenamiento*- se diseñará un sistema de clasificación que produce una decisión -clasificación- para cualquier patrón de datos admisible que se le presente. Como característica adicional, hay que señalar que el aprendizaje se desarrolla en ambiente difuso. En muchos casos las variables que describen las características de los ejemplos son continuas y una división de los dominios de los atributos en clases disjuntas puede plantear problemas, y por tanto, parece adecuado que una forma de discretizar su dominio sea mediante la división de éste en subconjuntos difusos. Por otro lado, con la discretización en subconjuntos difusos se puede asociar a éstos etiquetas lingüísticas, y de este modo se puede describir lingüísticamente el sistema. Distintas aproximaciones se han hecho a este enfoque, una metodología genérica se pueden encontrar [4].

En este trabajo desarrollamos un método de diseño de un sistema de clasificación en tres etapas, un proceso de derivación de reglas iterativo, un proceso de simplificación o selección que elimina reglas redundantes, y un proceso de ajuste de la semántica de las etiquetas lingüísticas que definen las reglas. Estos dos últimos se desarrollan utilizando algoritmos genéticos (AGs).

Los AGs son algoritmos de búsqueda basados en la evolución natural. Trabajan con una población de individuos (soluciones del problema), los cuales se combinan para obtener una nueva población mediante procedimientos basados en la genética, como son la reproducción de individuos y los operadores genéticos (cruce y mutación) [3]. En los últimos años, los AGs se han aplicado se han aplicado como herramienta de búsqueda ne procedimientos de aprendizaje, [10], y en particular, en el aprendizaje de reglas difusas. En [1] se puede encontrar un estudio sobre la aplicación de los AGs en sistemas basados en reglas difusas y en [2] se pueden encontrar un gran número de referencias en esta línea (sección 3.12 en clasificación y sección 3.13 en control).

La organización de la contribución es como sigue. En la sección 2 se describe el método propuesto. En la sección 3 se exponen los experimentos realizados y los

resultados obtenidos, comparándolos con los obtenidos sobre los mismos conjuntos de ejemplos por los algoritmos C4.5 y CART. Por último, se extraen algunas conclusiones y se destacan algunos aspectos de interés.

2 METODO DE CLASIFICACION

El método propuesto está formado por tres etapas:

1. Un proceso de *generación* de una base de reglas capaz de incluir todo el conocimiento que proporciona el conjunto de ejemplos. Esta base de reglas se genera de forma iterativa a partir de una semántica prefijada para las etiquetas de las variables.
2. Un proceso de *selección* de reglas utilizando AGs, que trata de eliminar del conjunto obtenido en el paso anterior, reglas innecesarias o redundantes.
3. Un proceso de *ajuste* de la base de datos usada para describir la semántica, de forma que se mejore en lo posible la exactitud de la base de reglas obtenida.

2.1 GENERACION DE LA BASE DE REGLAS

Esta primera etapa consiste en un proceso de generación de la Base de Reglas basado en una exploración iterativa del espacio de búsqueda del problema.

Para ello, además del conjunto de ejemplos de entrenamiento, se considera una Base de Datos previamente definida, constituida por particiones difusas uniformes con funciones de pertenencia triangulares. Se considera el número de términos lingüísticos que permita obtener el nivel de granularidad deseado.

Utilizamos reglas $R_i(A_{i1}, \dots, A_{im})$ con la siguiente estructura:

Si X_1 es A_{i1} y ... y X_m es A_{im} , entonces Y es B_i

donde $X_1, \dots, X_m$ son las variables del antecedente y B_i es una de las clases posibles.

Notaremos nuestro conjunto de ejemplos de entrenamiento como $E_p = \{e_1, e_2, \dots, e_p\}$, donde

$$e_i = (ex_i^1, ex_i^2, \dots, ex_i^m, ey_i)$$

La Base de Reglas se va construyendo seleccionando la regla más adecuada en cada paso del algoritmo. Una vez obtenida la regla, se contabiliza su cubrimiento sobre el conjunto de ejemplos de entrenamiento. Los ejemplos cubiertos en un grado mayor que un valor ϵ especificado se eliminan del conjunto. De esta forma, es el cubrimiento iterativo de los ejemplos, lo que guía la búsqueda hacia zonas prometedoras del espacio de búsqueda en cada paso. La eliminación del conjunto de entrenamiento de los ejemplos cubiertos, descarta la posibilidad de que algunos ejemplos dominen todo el proceso de búsqueda.

En este proceso de generación, cada vez que se selecciona la mejor regla, se mide la adecuación de las reglas candidatas mediante una función que recoge un

conjunto de criterios que permitan asegurar la completitud y consistencia de la base de reglas finalmente generada. Los tres criterios considerados se muestran a continuación.

- *Un valor de frecuencia alto.*

La frecuencia de una regla R_i , a través de un conjunto de ejemplos, E_p , se define como:

$$\psi_{E_p}(R_i) = \frac{\sum_{l=1}^p R_i(e_l)}{p}$$

siendo $R_i(e_l)$ el *grado de compatibilidad* entre la regla " $R_i = \text{Si } X_i \text{ es } A_{i1} \text{ y } \dots \text{ y } X_{im} \text{ es } A_{in} \text{ entonces clase } B_i$," y el ejemplo e_l definido como:

$$R_i(e_l) = \begin{cases} A_i(ex_l) & \text{si } ey_l = B_i \\ 0 & \text{en otro caso} \end{cases}$$

con

$$A_i(ex_l) = *(A_{i1}(ex_l^1), \dots, A_{im}(ex_l^m))$$

el grado de emparejamiento de las variables del antecedente de la regla con los valores para esas variables. Donde $*$ es un operador de conjunción, nosotros utilizaremos la t-norma del mínimo.

- *Alto grado de cubrimiento de los ejemplos positivos.*

Se define el conjunto de *ejemplos positivos* de la regla R_i , con un grado de compatibilidad mayor o igual a ω como:

$$E_\omega^+(R_i) = \{e_l \in E_p / R_i(e_l) \geq \omega\}$$

Si $n_\omega^+(R_i) = |E_\omega^+(R_i)|$, el grado medio de cubrimiento de $E_\omega^+(R_i)$ puede ser definido como:

$$G_\omega(R_i) = \sum_{e_l \in E_\omega^+(R_i)} R_i(e_l) / n_\omega^+(R_i)$$

- *Un conjunto de ejemplos negativos pequeño.*

El conjunto de *ejemplos negativos* para una regla R_i se define como:

$$R_i^-(R_i) = \{e_l \in E_p / R_i(e_l) = 0 \text{ y } A_i(ex_l) > 0\}$$

Un ejemplo es considerado negativo para una regla cuando empareja con el antecedente pero pertenece a una clase diferente a la que indica el consecuente de la regla. Los ejemplos negativos son considerados siempre sobre el conjunto de entrenamiento completo.

Si $n_{R_i}^- = |E^-(R_i)|$ es el número de ejemplos negativos, la función de penalización del conjunto de ejemplos negativos es:

$$g_n(R_i^-) = \begin{cases} 1 & \text{si } n_{R_i}^- \leq k \cdot n_\omega^+(R_i) \\ \frac{1}{n_{R_i}^- - k n_\omega^+(R_i) + \exp(1)} & \text{en otro caso} \end{cases}$$

De esta forma, permitimos un porcentaje del número de ejemplos positivos, $k \cdot n_p^+(R_i)$ de ejemplos negativos por regla sin penalización. Este porcentaje está determinado por el parámetro $k \in [0, 1]$.

Estos tres criterios son combinados en una función de evaluación de la siguiente forma:

$$F(R_i) = \psi_{E_p}(R_i) \cdot G_w(R_i) \cdot g_n(R_i^-)$$

Las reglas que tienen un mayor valor de esta función son las más adecuadas.

El método de generación consta de las siguientes etapas:

1. Inicialización:
 - (a) Introducir los valores de los parámetros k, ω , y ε .
 - (b) Fijar el grado de cubrimiento de los ejemplos a cero ($CV[l] \leftarrow 0, l = 1, \dots, p$).
 - (c) Inicializar la base de reglas final, B^g al conjunto vacío.
2. Inicializar el conjunto de reglas difusas candidatas, B^c , al conjunto vacío.
3. Para cada $e_l \in E_p$, generar la regla difusa R_c que mejor grado de compatibilidad tenga con el ejemplo e_l . Si $R_c \notin B^c$ añadirla a B^c .
4. Evaluar todas las reglas contenidas en B^c y seleccionar la regla R_r que tenga un valor más alto de la función de evaluación.
5. Introducir R_r en el conjunto de reglas B^g .
6. Para cada $e_l \in E_p$ hacer
 - (a) $CV[l] \leftarrow CV[l] + R_r(e_l)$.
 - (b) Si $CV[l] \geq \varepsilon$ entonces quitar el ejemplo e_l de E_p .
7. Si $E_p = 0$ entonces parar, si no, volver al paso 2.

2.2 METODO DE INFERENCIA

Una vez obtenido el conjunto de reglas que constituyen el sistema de clasificación, el método que se utiliza para inferir la clase a la cual pertenece un nuevo ejemplo e_l es el siguiente: Siendo $A_k(ex_l)$ el grado de emparejamiento del antecedente de la regla R_k con los valores del ejemplo, e_l , sea R_j la regla que verifica que:

$$A_j(ex_l) = \max_{i \in S} A_i(e_l)$$

donde S es el conjunto de índices asociados a las reglas, y la regla R_j viene dada por la expresión

$$R_j : \text{Si } X_1 \text{ es } A_{j1} \text{ y } \dots \text{ y } X_m \text{ es } A_{jm} \text{ entonces } B_j$$

entonces se dirá que el ejemplo pertenece a la clase B_j .

El ejemplo se asocia a la clase del consecuente de la regla que tiene mayor grado de compatibilidad con el ejemplo.

2.3 SELECCION DE LA BASE DE REGLAS

Como consecuencia de la naturaleza iterativa del proceso de generación de reglas, puede aparecer un fenómeno de sobreaprendizaje, en el sentido de que existan reglas innecesarias o redundantes lo que puede provocar que algunos ejemplos estén cubiertos en un grado mayor que el deseado, y esto haga que la base de reglas obtenida clasifique peor. Para resolver este problema y mejorar el rendimiento, se propone un proceso de simplificación y combinación de las reglas obtenidas en el proceso anterior, para derivar una base de reglas que nos permita clasificar correctamente.

El proceso de selección o simplificación fue presentado en [7], y está basado en un Algoritmo Genético con codificación binaria, en el cual la selección de individuos se realiza con el procedimiento de muestreo estocástico universal, la recombinación mediante el operador de cruce clásico binario multipunto (realizado en dos puntos) y de mutación uniforme. Además, se aplica el esquema de selección elitista.

El esquema de codificación genera individuos de longitud fija. Si consideramos que en la fase anterior se ha obtenido un conjunto de m reglas, una cadena de r -bits $C = (c_1, \dots, c_r)$ representa un subconjunto de reglas candidato a formar la base de reglas obtenida finalmente como salida de esta etapa, B^s , con el siguiente significado,

$$\text{Si } c_i = 1 \text{ entonces } R_i \in B^s \text{ si no } R_i \notin B^s$$

La población inicial se genera introduciendo un cromosoma que representa el conjunto de reglas que se obtuvo en la fase anterior, (B^g), al completo. El resto de los cromosomas se seleccionan aleatoriamente.

En este trabajo, hemos utilizado dos funciones de evaluación:

1. Una función de evaluación basada en el error de clasificación sobre el conjunto de ejemplos de entrenamiento. Esta función, $E_1()$, para un individuo C_i toma como valor el porcentaje de ejemplos que son mal clasificados con el conjunto de reglas que representa el cromosoma C_i .

Para mantener la propiedad de completitud que se consideró en la etapa anterior, requerimos que todo ejemplo contenido en la base de ejemplos de entrenamiento sea cubierto por la base de reglas codificada en un grado mayor o igual a τ ,

$$C_{R(C_j)}(e_l) = \bigcup_{j=1..T} R_j(e_l) \geq \tau \\ \forall e_l \in E_p \text{ y } R_j \in R(C_j)$$

donde τ representa el grado mínimo de completitud aceptado en el proceso de simplificación y $R(C_j)$ el conjunto de reglas codificado en el cromosoma C_j .

Se define el mínimo grado de completitud sobre el conjunto de entrenamiento para el conjunto

de reglas codificadas en el cromosoma C_j de la siguiente forma,

$$TSCD(R(C_j), E_p) = \bigcap_{e_i \in E_p} C_{R(C_j)}(e_i)$$

la función de evaluación final que penaliza la pérdida de la propiedad de completitud queda así:

$$F_1(C_j) = \begin{cases} E_1(C_j) & \text{si } TSCD(R(C_j), E_p) \geq \tau \\ 100 & \text{en otro caso} \end{cases}$$

2. Una función de evaluación basada en el enfoque que propone Ishibuchi ([12]) en su proceso de selección de reglas difusas. La función que utilizamos, penaliza los cromosomas con mayor número de reglas, pero además, penaliza la pérdida de completitud expuesta anteriormente. De esta forma, se intenta simplificar el conjunto de reglas generadas, maximizando el número de ejemplos clasificados correctamente, minimizando el número de reglas difusas. La función de evaluación tiene la siguiente expresión:

$$E_2(C_j) = |R(C_j)| \cdot W_R - W_{NCP} \cdot NCP(R(C_j))$$

$$F_2(C_j) = \begin{cases} E_2(C_j) & \text{si } TSCD(R(C_j), E_p) \geq \tau \\ |R(C_j)| \cdot W_R & \text{en otro caso} \end{cases}$$

donde $|R(C_j)|$ es el número de reglas presentes en el conjunto codificado por C_j , $NCP(R(C_j))$ es el número de ejemplos de entrenamiento correctamente clasificados con ese conjunto de reglas, y W_R y W_{NCP} son constantes positivas que verifican que $0 < W_R \ll W_{NCP}$. Esta relación se justifica con el hecho de que, en general, el poder de clasificación de un sistema es más importante que el hecho de que éste sea compacto.

2.4 AJUSTE DE LA BASE DE DATOS

En este proceso, cada uno de los cromosomas que constituyen la población genética, codifica una definición completa de la base de datos que será combinada con la base de reglas simplificada -obtenida en la fase anterior- para evaluar la adaptación de los individuos.

El Algoritmo Genético diseñado para el proceso de ajuste utiliza codificación real [9], como procedimiento de selección el muestreo estocástico universal, y como operadores de recombinación, el operador de mutación no uniforme y el operador de cruce Max-Min-Aritmético, [8].

En las dos fases previas, hemos considerado que las etiquetas lingüísticas de las variables tenían asociadas funciones de pertenencia triangulares. Como cada función de pertenencia tiene asociado una representación paramétrica basada en un triple de valores reales, una partición difusa primaria se puede representar mediante un array compuesto por $3N$ valores

reales, siendo N el número de términos que constituyen el conjunto de etiquetas lingüísticas de la variable. De esta forma, la definición completa de la base de datos para un problema de clasificación que involucre s variables lingüísticas, se codifica en un cromosoma C_r con codificación fija, construido mediante la unión de la representación de la partición difusa de cada una de estas variables de la siguiente forma:

$$C_{r,i} = (a_{i1}, b_{i1}, c_{i1}, \dots, a_{iN_i}, b_{iN_i}, c_{iN_i})$$

$$C_r = C_{r,1} C_{r,2} \dots C_{r,s}$$

La población inicial se crea haciendo uso de la definición de la Base de Datos inicial. Esta Base de Datos se codifica directamente en el primer cromosoma C_1 . El resto de los individuos se generan asociando un intervalo de actuación $[c_h^l, c_h^r]$, a cada gen c_h de C_1 , $h = 1 \dots \sum_{i=1}^s N_i \times 3$. Cada intervalo de actuación representa el intervalo de ajuste para el correspondiente gen, $c_h \in [c_h^l, c_h^r]$. Si $(t \bmod 3 = 1)$ entonces, c_t es el valor izquierdo del soporte de un número difuso. El número difuso está definido mediante los tres parámetros (c_t, c_{t+1}, c_{t+2}) , y los intervalos de actuación vienen dados por las siguientes expresiones:

$$c_t \in [c_t^l, c_t^r] = [c_t - \frac{c_{t+1}-c_t}{2}, c_t + \frac{c_{t+1}-c_t}{2}]$$

$$c_{t+1} \in [c_{t+1}^l, c_{t+1}^r] = [c_{t+1} - \frac{c_{t+2}-c_{t+1}}{2}, c_{t+1} + \frac{c_{t+2}-c_{t+1}}{2}]$$

$$c_{t+2} \in [c_{t+2}^l, c_{t+2}^r] = [c_{t+2} - \frac{c_{t+3}-c_{t+2}}{2}, c_{t+2} + \frac{c_{t+3}-c_{t+2}}{2}]$$

Así, se crea una población de cromosomas que contiene a C_1 como primer individuo y el resto inicializados aleatoriamente, con cada gen dentro de su respectivo intervalo de actuación.

Como función de evaluación se utiliza la función $E_1(\cdot)$ definida en la subsección anterior.

Un estudio completo del proceso de ajuste se puede encontrar en [6].

3 EXPERIMENTACION

Se ha ejecutado el método para desarrollar un sistema de clasificación para la conocida base de ejemplos IRIS, creada por R.A. Fisher [13].

El conjunto de ejemplos tiene tres clases, cuatro atributos (variables) y cincuenta ejemplos de cada clase. Para cada uno de los atributos se han considerado particiones difusas de tipo uniforme con cinco etiquetas lingüísticas. El conjunto de ejemplos se ha dividido de cinco formas distintas, en conjuntos de entrenamiento y prueba (70% y 30% respectivamente), y los resultados que aparecen en las tablas son medias del porcentaje de clasificación correcta en cada caso. El sistema de clasificación se ha construido, en cada caso, con toda la base de entrenamiento.

Los valores que se han utilizado para los parámetros son los siguientes:

ω	0.5 y 0.05
τ	0.5 y 0.25
Fitness (simplificación)	F_1 y F_2
k	0.1
t-norma	Mínimo
P_c	0.6
P_m	0.1
N. individuos	100
N. generaciones	500
b (Mutación no uniforme)	5.0
a (Cruce Max-Min-Aritmético)	0.35

Los resultados obtenidos en el proceso de generación se muestran en la Tabla 1.

Table 1: Datos obtenidos en la fase de generación

ω	Reglas	Entrenamiento	Prueba
0.5	61.6	96.92804	93.76868
0.05	62	95.30392	94.2565

Estos varían ligeramente en función del nivel mínimo de cubrimiento que se les exija a los ejemplos positivos ($\omega = 0.5$ y $\omega = 0.05$). La diferencia entre un caso y otro está en que al estar más ajustadas las reglas a la base de entrenamiento en el primer caso que en el segundo, el porcentaje de clasificación sobre datos test es mayor en el segundo experimento y el de clasificación sobre datos de entrenamiento es mayor en el primer experimento. De cualquier forma, las diferencias aún no se pueden considerar importantes.

En la fase de simplificación al eliminarse reglas que podrían introducir ruido, el porcentaje de clasificación para datos de entrenamiento aumenta en todos los casos -con las dos funciones de evaluación y los dos valores de τ - significativamente. Los experimentos realizados no nos permiten establecer diferencias homogéneas entre los dos valores diferentes para el parámetro τ .

Table 2: Datos obtenidos en la fase de selección

F	ω	τ	Reglas	Entren.	Prueba
1	0.5	0.5	44.2	97.47952	94.74394
1	0.5	0.25	40.4	97.74952	93.6294
1	0.05	0.5	45	97.663	93.86912
1	0.05	0.25	41.2	97.663	92.69228
2	0.5	0.5	23.8	97.663	92.46798
2	0.5	0.25	17.4	97.46884	90.76518
2	0.05	0.5	23.2	97.47952	93.24238
2	0.05	0.25	17	97.29604	90.82744

Respecto a las funciones de evaluación, se observa una disminución del número de reglas con el uso de F_2 , en torno al 50%. El porcentaje de clasificación disminuye también, aunque no significativamente.

El ajuste de la semántica de las variables produce un nuevo incremento en la capacidad de predicción del

conjunto de reglas, tanto sobre datos de entrenamiento como sobre datos test, como se muestra en la Tabla 3.

Table 3: Datos obtenidos en la fase de ajuste

F	ω	τ	Reglas	Entren.	Prueba
1	0.5	0.5	44.2	98.7648	94.39544
1	0.5	0.25	40.4	97.47.952	93.6294
1	0.05	0.5	45	99.28204	92.83122
1	0.05	0.25	41.2	99.10964	92.83122
2	0.5	0.5	23.8	99.10964	91.96682
2	0.5	0.25	17.4	98.7648	92.64088
2	0.05	0.5	23.2	98.93722	93.28088
2	0.05	0.25	17	98.74266	91.35374

Si comparamos los resultados obtenidos, con los obtenidos por los conocidos algoritmos C4.5 y CART sobre la misma base de ejemplos, considerando las mismas particiones entrenamiento-prueba, vemos que tienen, en media, un parecido porcentaje de clasificación.

Table 4: Datos obtenidos por C4.5 y CART

Sistema	Reglas	Entrenamiento	Prueba
C4.5	5.2	99.16	93.18
CART		97.38	95.65

4 CONCLUSIONES

El método propuesto, consigue extraer de una base de ejemplos, un conjunto de reglas con un poder de predicción, en media del 96%.

El número de reglas que se obtienen con él es superior al que obtienen otros métodos. Algunos autores como Ishibuchi ([11]) insisten en la reducción del número de reglas justificando que un número pequeño de reglas puede ser examinado cuidadosamente para ser mejorado y puede representar más fielmente el conjunto de reglas que utilizaría un experto en el tema para clasificar. Realmente, en un sistema de clasificación automatizado, lo más importante es su poder de predicción independientemente del número de reglas internas de que conste. De hecho, tal y como señala Weiss [16], un sistema de aprendizaje es un programa de ordenador que toma decisiones basándose en la experiencia acumulada en casos resueltos correctamente. A diferencia de un sistema experto que resuelve los problemas utilizando un modelo computacional del razonamiento experto humano, un sistema de aprendizaje puro puede emplear diferentes técnicas para explotar el poder computacional de un ordenador aunque no tengan relación con los procesos de conocimiento humanos.

Los resultados obtenidos parecen indicar que el método de inferencia utilizado no aprovecha toda la potencia del razonamiento aproximado por cuanto se

queda con la clase correspondiente a la regla de mayor grado de compatibilidad no teniendo en cuenta la información proporcionada por el resto de reglas. En próximos trabajos se intentará trasladar la filosofía del razonamiento de aproximado a un método de inferencia en clasificación.

También parece interesante trabajar con semánticas no prefijadas, es decir, diseñar un método que trabaje con el enfoque aproximativo, modificando las funciones de pertenencia de las etiquetas en la fase de generación, o aprendiendo directamente la semántica de las reglas.

Referencias

- [1] O. Cordon, F. Herrera (1995). A General Study on Genetic Fuzzy Systems. J. Periaux, G. Winter, M. Galan, and P. Cuesta (Eds.), *Genetic Algorithms in Engineering and Computer Science*. John Wiley and Sons, 33-57.
- [2] O. Cordon, F. Herrera, M. Lozano (1995). A Classified Review on the Combination Fuzzy Logic-Genetic Algorithms Bibliography. Technical Report DECSAI-95129, Dept. of Computer Science and A.I., University of Granada, Spain (Last version, February 1996). Available at <http://decsai.ugr.es/~herrera/fl-ga.html>
- [3] D.E. Goldberg (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, New York .
- [4] A. González (1995). A. A Learning Methodology in Uncertain and Imprecise Environments. *International Journal of Intelligent Systems* 10: 357-371.
- [5] A. González, R. Pérez (1995). Completeness and Consistency Conditions for Learning Fuzzy Rules. Technical Report DECSAI-95103, Dept. of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain.
- [6] F. Herrera, M. Lozano, J.L. Verdegay (1995). Tuning Fuzzy Controllers by Genetic Algorithms. *International Journal of Approximate Reasoning* 12: 299-315.
- [7] F. Herrera, M. Lozano, J.L. Verdegay (1995). A Learning Process for Fuzzy Control Rules Using Genetic Algorithms. Technical Report DECSAI-95108, Dept. of Computer Science and A.I., University of Granada, Spain.
- [8] F. Herrera, M. Lozano, J.L. Verdegay (1995). The Use of Fuzzy Connectives to Design real-Coded Genetic Algorithms. *Mathware & Soft Computing* 1: 239-251.
- [9] F. Herrera, M. Lozano, J.L. Verdegay, J.L. (1996). Tackling Real Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. *Artificial Intelligence Review*. Por aparecer.
- [10] J.J. Grefenstette (1995). *Genetic Algorithms for Machine Learning*. Kluwer Academic Publishers.
- [11] H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka (1994): Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms. *Fuzzy Sets and Systems* 65: 238-253.
- [12] H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka, (1995). Selecting Fuzzy If Then Rules for Classification Problems Using Genetic Algorithms. *IEEE Transactions on Fuzzy Systems* 3: 260-270.
- [13] R.A. Fisher (1950). The use of multiple measurements in taxonomic problems. *Annual Eugenics* 7, Part II. (1933). 179-188; also in *Contributions to Mathematical Statistics*, John Wiley, NY, (1950).
- [14] R. Michalski (1983). A theory and methodology of inductive learning. *Artificial Intelligence* 20: 111-161.
- [15] J.R. Quinlan (1993). *C4.5 Programs for Machine Learning*. Morgan Kauffman.
- [16] S.M. Weiss, C.A. Kulikowski (1991). *Computer Systems that learn*. Morgan Kaufmann.

Árboles difusos de clasificación y regresión.

M. Delgado
Universidad de Granada
Dep. CC. Computación e I.A.
mdelgado@ugr.es

A. F. Gómez-Skarmeta
Universidad de Murcia
Dep. Informática
skarmeta@dif.um.es

L. J. Linares
Universidad de Castilla-La Mancha
Dep. Informática
ljimenez@inf-cr.uclm.es

Resumen

Presentamos aquí un algoritmo para tratar el problema de la regresión (en el sentido general de aproximación de funciones) mediante técnicas difusas. La idea es inducir un árbol binario difuso a partir del cual obtendremos un conjunto de reglas difusas que configuraran una base de conocimiento con la que se realizará la inferencia. El mecanismo de generación del árbol que aquí proponemos es una extensión natural del algoritmo CART dentro de un ambiente difuso y por ello lo denominaremos FCART.

1 Introducción

Sea $\delta = \{s_1, \dots, s_n\}$ un conjunto de datos, correspondientes a valores de ciertas variables $X = \{X^1, \dots, X^d, Y\}$, $s_i = \{x_i^1, \dots, x_i^d, y_i\}$ con $x_i^j \in D^j$, $y_i \in O$ donde D^j y O denotan los dominios de X^j para $j = 1 \dots d$ e Y respectivamente. Supongamos que existe una función $F : D^1 \times \dots \times D^d \rightarrow O$ que es conocida sólo en los puntos de S y tal que $F(x_i^1, \dots, x_i^d) = y_i$.

El objetivo de la regresión es determinar explícitamente una nueva función $F' : D^1 \times \dots \times D^d \rightarrow O$, de tal forma que haga mínima distancia entre los valores y_i y los predichos por F' $y'_i = F'(x_i^1, \dots, x_i^d)$. La función distancia más frecuentemente empleada es el error cuadrático medio:

$$\frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2 \quad (1)$$

La metodología clásica presenta dos dificultades básicas:

1. Hay que asumir algún conocimiento previo sobre el comportamiento de la función F para

poder optar por un modelo concreto de función paramétrica.

2. El proceso de minimización de la ecuación (1) puede llegar a ser muy complejo, especialmente si el número de variables que intervienen es grande.

La primera de ellas es la más difícil de abordar, ya que nos obliga a suponer un conocimiento sobre el comportamiento de la función, sin que realmente dispongamos de él.

Un enfoque de la regresión distinto al clásico, que no presenta esta dificultad, consiste en delimitar regiones en el dominio de definición de la función F , y asignar valores a F' para cada una de ellas. Estas regiones podrían describirse mediante un conjunto de restricciones que debiesen verificar las variables de entrada de la función F .

Sólo hemos de suponer, con esta metodología, que la función F se comporta uniformemente dentro de una región, esto es que los valores de F dentro de una misma región han de ser "similares".

Este enfoque conlleva obviamente las siguientes tareas:

1. Delimitación de las regiones.
2. Asignación del valor de F' en cada región.

En estas condiciones, el mecanismo de inferencia a partir de F' comenzaría por determinar a qué región pertenecen los nuevos valores de las variables de entrada y considerar como valor el asignado a F' en dicha región.

Ambas tareas, la definición de las regiones y la asignación de los valores de F' , han de llevarse a cabo, basándose sólo en el conocimiento que se tenga de la función F que en definitiva se reduce al conjunto de datos S . Para ello pueden emplearse técnicas muy distintas como clustering, redes neuronales, métodos difusos, etc. En este trabajo abordaremos la regresión

mediante la metodología de los árboles de regresión, aplicando técnicas difusas.

En la sección 2 describiremos algunas ideas básicas sobre los árboles de regresión generalizándolos para considerar regiones difusas, y poder utilizarlos en el proceso de inferencia difusa. En la sección 3 presentaremos un método para realizar una partición difusa de un conjunto difuso, mecanismo que utilizaremos para la división de los nodos de los árboles de regresión. En la siguiente sección 4 indicaremos como inferir nuevos valores con la información obtenida por el árbol de regresión difuso, para finalizar en la secciones 5 y 6 viendo un ejemplo y presentado algunas conclusiones

2 Árboles de regresión y FCART

2.1 Nociones básicas sobre árboles de regresión

El problema de la regresión difiere del problema de clasificación en que las variables de salida pueden ser continuas, mientras que en la clasificación han de ser estrictamente discretas (categorías o clases). Desde esta perspectiva, la clasificación se puede observar como un caso particular de regresión. Métodos de división recursiva del espacio de definición como la inducción de árboles de decisión ID3 [4] han sido aplicados, como métodos de regresión mediante restricciones, en el programa CART [3].

El programa CART (árbol de regresión y clasificación) esta basado en la construcción de una estructura arbórea, donde las regiones se caracterizan como posibles respuestas a un conjunto de preguntas sobre valores de las variables que definen el problema.

Sea T un nodo del árbol que representa un conjunto de datos S cuya proyección sobre $D^1 \times \dots \times D^d$ notaremos S_X . De acuerdo con nuestro enfoque la idea es asociar un valor único a F' sobre S_X que notaremos $F'(T)$ (salida inducida en el nodo T) y tomar $F'(x^1, \dots, x^d) = F'(T)$ para cada $(x^1, \dots, x^d) \in S_X$. Una elección inmediata es:

$$F'(T) = \frac{\sum_{(x^1, \dots, x^d, y) \in S} y_i}{|S|} \quad (2)$$

Con error estimado según (1)

$$\text{Err}(T) = \frac{\sum_{(x^1, \dots, x^d, y) \in S} (y_i - F'(T))^2}{|S|} \quad (3)$$

Ahora podemos dividir T realizando alguna pregunta relativa a una variable, y por cada posibles respuestas crearemos un nodo hijo de T .

En CART el tipo de pregunta más frecuente es si el valor de una variable pertenece a un conjunto dado de valores. Por ejemplo, una pregunta podría ser ¿ $x^j < c$?, donde c es una constante en el dominio de X^j . Esta cuestión tiene dos respuestas posibles: si o no. Esto puede formularse como $x^j \in \{x \in X^j | x < c\}$ o $x^j \in \{x \in X^j | x \geq c\}$, dividiendo el nodo T en dos subnodos T_1 y T_2 asociados a $S_1 = \{(x_i^1, \dots, x_i^d, y_i) \in T | x < c\}$ y $S_2 = \{(x_i^1, \dots, x_i^d, y_i) \in T | x \geq c\}$ respectivamente.

La calidad de la partición asociada a una pregunta p puede estimarse mediante

$$C(T, p) = \text{Err}(T_1) * P(T_1) + \text{Err}(T_2) * P(T_2) \quad (4)$$

donde $P(T_1)$ y $P(T_2)$ son respectivamente, los tamaños relativos de S_1 y S_2 con respecto a S . Estos dos valores han de verificar

$$P(T_1) + P(T_2) = 1 \quad (5)$$

Sea $Q(T) = \{q_1, \dots, q_m\}$ el conjunto de todas las posibles preguntas que podemos realizar en el nodo T . Seleccionaremos la pregunta $q^* \in Q(T)$ tal que

$$q^* = \min_{q \in Q(T)} C(T, q) \quad (6)$$

El proceso de división anterior se repite para cada nodo obtenido, hasta satisfacer algún criterio de parada. De esta forma, se construye un árbol binario que define las regiones que nosotros buscamos mediante el camino recorrido por un valor desde la raíz del árbol hasta alguna de sus hojas.

El algoritmo CART crea un conjunto de regiones $SR = \{r_1, \dots, r_p\}$ que verifican $(\cup_{i=1 \dots p \text{ and } i \neq j} r_i) \cap r_j = \emptyset$ y $\cup_{i=1 \dots p} r_i = D^1 \times \dots \times D^d$. En otras palabras, SR es una partición del dominio de definición del problema, lo que está motivado por el tipo de preguntas que nosotros hemos formulado. Con el objetivo de generalizar este modelo, introducimos la teoría de los conjuntos difuso de Zadeh [7] estableciendo el algoritmo FCART, árboles de regresión y clasificación difusos.

2.2 FCART

Sea T un nodo arbitrario y supongamos que hay asociado a dicho nodo un conjunto difuso sobre el conjunto de datos δ con función de pertenencia $\mu_T : \delta \rightarrow [0, 1]$. Admitiremos como condición inicial que si T es el nodo raíz entonces $\mu_{raiz} : \delta \rightarrow 1$. Definimos ahora un valor de salida para el nodo T $F''(T)$, calculado como la media ponderada

$$F''(T) = \frac{\sum_{i=1}^n \mu_T(s_i)^m * y_i}{\sum_{i=1}^n \mu_T(s_i)^m} \quad (7)$$

asociándole el error ¹

$$\text{Err}(T) = \frac{\sum_{i=1}^n (F''(T) - y_i)^2 * \mu_T(s_i)^m}{\sum_{i=1}^n \mu_T(s_i)^m} \quad (8)$$

Ahora, nuestro problema es crear el conjunto de preguntas a realizar en el nodo T. Como en el caso anterior, las cuestiones se formulan para cada variable, obteniendo una partición binaria difusa de cada conjunto difuso en el mismo sentido que Dimitrescu [2], teniendo así que $|Q(T)| \leq d$.

Supongamos que por algún procedimiento (ver sección 3) se ha construido una partición binaria difusa del nodo T mediante el conjunto difuso μ_T^j de la variable j asociado al nodo T, esto es $Q_T^j = \{A(x), B(x)\}$. Esta partición origina dos nuevos nodos y dos nuevos conjuntos difusos asociados a cada uno de ellos

$$\begin{aligned} \mu_{T_1}(s_i) &= \mu_T(s_i) \otimes A(x_i^j) \\ \mu_{T_2}(s_i) &= \mu_T(s_i) \otimes B(x_i^j) \end{aligned}$$

donde $\otimes$ es una t-norm, por ejemplo el mínimo

Siguiendo el algoritmo CART, definimos las proporciones para A(x) y B(x) relativas al conjunto difuso μ_T^j como

$$P(A) = \frac{\sum_{i=1}^n A(x_i^j)}{\sum_{i=1}^n \mu_T^j(x_i^j)} \quad (9)$$

Este enfoque de creación de las cuestiones origina una partición jerárquica difusa para cada una de las variables del problema. El criterio de selección de una determinada pregunta en el nodo T, será el mismo que en el algoritmo CART.

El proceso de división parará cuando el mayor de los errores estimados sea menor que un valor preestablecido.

$$\text{ERROR} = \max_{T \in \bar{T}} \text{Err}(T) \leq \epsilon \quad (10)$$

donde $\bar{T}$ es el conjunto de nodos hoja del árbol, cada uno de los cuales representa una región de nuestra solución. En estas condiciones construiremos la función de regresión a partir de los valores $F''(T)$, $T \in \bar{T}$, de

¹m es un parámetro real tal que $1 \leq m < \infty$

tal forma que para cada entrada $x = (x^1, \dots, x^d) \in D^1 \times \dots \times D^d$ le haremos corresponder

$\text{input}(s_i) = \{x_i^1, \dots, x_i^d\}$ será

$$F'(x^1, \dots, x^d) = \frac{\sum_{T \in \bar{T}} \mu_T(x^1, \dots, x^d) * F''(T)}{\sum_{T \in \bar{T}} \mu_T(x^1, \dots, x^d)} = y' \quad (11)$$

3 Partición difusa de un conjunto difuso

En esta sección vamos a describir un método para obtener la partición difusa de cada nodo, necesaria para el algoritmo FCART tal como se describe en la sección anterior.

Igual que propone Dimitrescu en [2], nosotros estamos interesados en obtener una partición difusa del conjunto difuso μ_T^j sobre la variable j, para establecer las preguntas de cada nodo. Sea $U = \{u_1, \dots, u_c\}$ un conjunto de conjuntos difusos definidos sobre el dominio de la variable X, y μ_T otro conjunto difuso también definido sobre el mismo referencial X. Diremos que U es una partición difusa de μ_T si verifica

$$\sum_{i=1}^c u_i(x) = \mu_T(x) \quad \forall x \in X \quad (12)$$

Generalizaremos el algoritmo de clustering difuso FCM [1], para obtener el que denominaremos FFCM con el objetivo de establecer una partición difusa de μ_T . Siguiendo la notación clásica de FCM, nuestra tarea es minimizar

$$J(U, V) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^m (d_{ik})^2 \quad (13)$$

sujeto a la restricción

$$\sum_{i=1}^c u_{ik} = \mu_{T_k} \quad k = 1, \dots, n \quad (14)$$

Donde u_{ik} nota a $u_i(s_k)$ y d_{ik}^2 es

$$\begin{aligned} d_{ik}^2 &= (v_i - s_k)^2 \\ v_i &= \frac{\sum_{k=1}^n (u_{ik})^m s_k}{\sum_{k=1}^n (u_{ik})^m} \end{aligned} \quad (15)$$

Resolviendo esta minimización mediante multiplicadores de Lagrange, tenemos que

$$F_k(\lambda, u_k) = \sum_{i=1}^c (u_{ik})^m d_{ik}^2 + \lambda \left(\sum_{i=1}^c u_{ik} - \mu_{T_k} \right) \quad (16)$$

$$\frac{\partial F_k(\lambda, u_k)}{\partial u_{st}} = md_{st}^2 u_{st}^{m-1} - \lambda = 0$$

$$\frac{\partial F_k(\lambda, u_k)}{\partial \lambda} = \sum_{i=1}^c u_{ik} - \mu_{T_k} = 0$$

de donde se obtiene:

$$u_{st} = \frac{\mu_{T_k}}{\sum_{i=1}^c \left(\frac{d_{st}}{d_{it}}\right)^{\frac{2}{m-1}}} \quad (17)$$

solución que engloba a la obtenida en GFI [2] sin más que hacer $m=2$. Finalmente el algoritmo FFCM sería

FFCM Algorithm:

1. Fijar c , $2 \leq c < n$; Fijar m ; $1 \leq m < \infty$ e inicializar la primera partición difusa $U^{(0)}$.
2. Calcular los c prototipos $\{v_i^{(l)}\}$ mediante la ecuación(15) y $U^{(l)}$.
3. Actualizar $U^{(l)}$ usando: Si $I_k = \emptyset$ entonces ecuación(17) si no $u_{ik} = 0 \forall i \in \bar{I}_k$ and $u_{ik} = \mu_{T_k} \forall i \in I$ where $I_k = \{i | 1 \leq i \leq c \text{ and } d_{ik} = 0\}$ y $\bar{I}_k = \{1, \dots, c\} - I_k$.
4. Comparar $U^{(l)}$ con $U^{(l-1)}$ mediante una norma de matrices apropiada: Si $\|U^{(l)} - U^{(l-1)}\| \leq \epsilon_l$ entonces Parar en el caso contrario volver a 2.

4 Base de conocimiento difuso

Una vez obtenido el árbol binario difuso, podemos construir partiendo de él un conjunto de reglas difusas que configuraran nuestra base de conocimiento con la que podremos realizar el proceso de inferencia. Esta tarea tiene dos etapas:

1. la obtención para cada nodo hoja de su regla asociada(ver [5]) creando la base de conocimiento.
2. Una vez creada, establecer un mecanismo de inferencia sobre dicha base de conocimiento.

De igual forma que en ID3, la regla asociada a cada hoja del árbol, se establece mediante el camino entre la raíz y dicha hoja. Si hemos alcanzado un nodo hoja a través de un camino, significa que se han debido de satisfacer todas las condiciones difusas que caracterizan dicho camino. Estas condiciones se definen mediante el grado de pertenencia del valor de una variable de entrada al conjunto difuso que originó el nodo. Todo el conjunto de condiciones establecidas en un camino son reunidas mediante alguna t-norma, para formar la parte antecedente de la regla asociada a dicho nodo

hoja. La parte consecuente de la regla se toma como el valor de salida asignado al nodo hoja.

$$\text{Regla}(T) : \text{IF } A_T^1(x_1^1) \text{ y } \dots \text{ y } A_T^d(x_1^d) \text{ ENTONCES } F''(T) \quad (18)$$

De esta forma, obtenemos una base de conocimiento difuso formada por $|\bar{T}|$ reglas correspondientes a cada una de las hojas del árbol. Observemos que la reglas construidas tienen un consecuente formado por un valor nítido sobre el dominio O . Así pues el método de inferencia más adecuado para este caso será el de Takagi y Sugeno [6]. Asignando en el nodo hoja T , el valor de verdad de su regla asociada

$$w_T = A_T^1(x_1^1) \otimes \dots \otimes A_T^d(x_1^d) \quad (19)$$

y su valor de salida inferido será

$$F'(x^1, \dots, x^d) = \frac{\sum_{T \in \bar{T}} w_T F''(T)}{\sum_{T \in \bar{T}} w_T} \quad (20)$$

5 Ejemplo

Hemos aplicado nuestro algoritmo FCART a un conjunto de 50 puntos, obtenidos simulando la función de dos variables $\text{seno}(u) * \cos(v)$, y distribuida uniformemente sobre $[0,6.5] \times [0,6.5]$. Fijando $m=1.75$ y admitiendo un error máximo de cada regla ϵ de 0.05, obtenemos el conjunto siguiente de 19 reglas.

Regla	u in	v in	F"
R_{00}	(2.56,3.68,6.21,6.21)	(2.48,6.13,6.13,6.13)	0.599
R_{01}	(2.56,3.68,6.21,6.21)	(2.65,3.31,3.63,4.06)	0.789
R_{02}	(0.07,0.07,2.56,3.68)	(0.15,0.15,1.50,1.83)	0.196
R_{03}	(0.07,0.07,2.56,3.68)	(3.63,4.06,4.38,5.01)	-0.358
R_{04}	(0.07,0.07,2.56,3.68)	(1.98,2.27,2.48,2.85)	-0.449
R_{05}	(0.07,0.07,2.56,3.68)	(2.65,3.31,3.63,4.06)	-0.716
R_{06}	(0.07,0.07,2.56,3.68)	(4.38,5.01,6.13,6.13)	0.434
R_{07}	(2.56,3.68,6.21,6.21)	(2.39,2.46,2.48,2.85)	0.080
R_{08}	(2.56,3.68,6.21,6.21)	(0.60,1.01,1.50,1.83)	-0.170
R_{09}	(0.07,0.07,1.32,1.91)	(2.48,6.13,6.13,6.13)	-0.335
R_{10}	(2.56,3.68,6.21,6.21)	(1.50,1.83,1.98,2.27)	0.292
R_{11}	(2.56,3.68,6.21,6.21)	(4.38,5.01,6.13,6.13)	-0.543
R_{12}	(0.07,0.07,2.56,3.68)	(1.50,1.83,1.98,2.27)	-0.139
R_{13}	(2.27,2.44,2.56,3.68)	(2.48,6.13,6.13,6.13)	-0.163
R_{14}	(1.32,1.91,2.27,2.44)	(2.48,6.13,6.13,6.13)	-0.929
R_{15}	(2.56,3.68,6.21,6.21)	(0.15,0.15,0.60,1.01)	-0.946
R_{16}	(2.56,3.68,6.21,6.21)	(3.63,4.06,4.38,5.01)	0.028
R_{17}	(2.56,3.68,6.21,6.21)	(1.98,2.27,2.32,2.39)	-0.048
R_{18}	(2.56,3.68,6.21,6.21)	(2.32,2.39,2.39,2.46)	0.663

6 Conclusión

En este trabajo, hemos establecido un mecanismo para generar una base de conocimiento difuso que utilizamos

para resolver un problema de regresión mediante una técnica de restricciones. Todos los procesos de definición son controlados por los datos (conocimiento real) y no siendo necesaria ninguna suposición sobre su comportamiento. Proponemos dos algoritmos nuevos FCART y FFCM que generalizan los algoritmos clásicos FCM y CART. La estructura de árbol obtenida podría servir para simplificar o mejorar la inferencia de nuevos valores.

Referencias

- [1] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
- [2] D. Dimitrescu. Hierarchical pattern classification. *Fuzzy and Systems*, 28:145-162, 1988.
- [3] Jerome H. Friedman Leo Breiam and Charles J. Stone. *Classification and Regression Trees*. Wadsworth & Brooks, 1984.
- [4] J. R. Quilan. Induction of decision tree. *Machine Learning*, 1:81-106, 1986.
- [5] J. R. Quilan. Simplifying decision trees. *Int. T. Man-Machines Studies*, 27:221-234, 1987.
- [6] T. Takagi and M. Sugeno. Fuzzy identification of system and its applications to modeling and control. *IEEE Transaction on system, Man and Cybernetics*, smc-15:116-132, January 1985.
- [7] L. A. Zadeh. Fuzzy set. *Information and control*, pages 338-353, 1965.

Optimización de operadores reticulares en un entorno de síntesis automática de controladores borrosos

Jean-Pierre Deschamps
Dpto. Ingeniería Informática
Facultad de Ciencias Físicas
Univ. Autónoma de Madrid
jimt@eucmax.sim.ucm.es

José Ignacio Martínez Torre
Dpto. Informática y Automática
Facultad de Ciencias Físicas
Univ. Complutense de Madrid
jimt@eucmax.sim.ucm.es

Rafael Moreno Vozmediano
Dpto. Informática y Automática
Facultad de Ciencias Físicas
Univ. Complutense de Madrid
rmoreno@eucmax.sim.ucm.es

Resumen

Los algoritmos de control borroso se prestan muy bien a la aplicación de métodos de síntesis de alto nivel (SAN); sin embargo, una materialización directa de la base de reglas representante del problema puede conllevar un número muy alto de operadores reticulares.

En este artículo, se propone un algoritmo de minimización de expresiones reticulares válido para ser integrado en un entorno de desarrollo y síntesis automática de controladores borrosos. Además se presentan resultados experimentales de dicha optimización.

Palabras clave: Expresión reticular, síntesis de alto nivel, algoritmo borroso, optimización, ciclo de diseño.

1 INTRODUCCIÓN

Los requisitos de una buena parte de las aplicaciones borrosas pueden satisfacerse mediante procesadores o microcontroladores de propósito general [1-3]. Sin embargo, las aplicaciones borrosas con restricciones muy exigentes, como el procesamiento en tiempo real de imagen o sonido, necesitan la existencia de *hardware* borroso dedicado. La aparición tanto de herramientas de ayuda al diseño electrónico, cada vez más sofisticadas y más eficientes, como de dispositivos programables, cada vez más complejos y más rápidos, facilita enormemente la tarea de reducir el tiempo de diseño del *hardware* dedicado.

Las peculiaridades propias del procesamiento borroso pueden conllevar la aparición de herramientas específicas. Dentro del conjunto de herramientas de síntesis propuestas [4] para el diseño de *hardware* borroso dedicado, las de síntesis de alto nivel ocupan un papel preponderante (Figura 1). Se parte de la base de que la aplicación borrosa que queremos materializar está perfectamente fijada y verificada, es decir que disponemos de su especificación correcta y completa

Este trabajo ha sido subvencionado por el programa HCM de la CEE con número de contrato CHRX-C794-0459.

(precisiones, número y forma de las funciones de pertenencia de entradas y salidas, los métodos de inferencia, la base de reglas, etc.).

En un entorno de síntesis de alto nivel, la primera tarea a realizar es la planificación del flujo de datos que implica el algoritmo en cuestión. En esa planificación influyen de manera decisiva cuáles son los operadores de que se dispone, cuál es su coste computacional tanto en área como en tiempo, y cuáles son los requisitos de la aplicación.

Considerando como operadores de inferencia y agregación al mínimo y al máximo, se puede hacer una traducción directa de la base de reglas de la especificación borrosa a un conjunto de asignaciones como las que se presentan en [5].

Tomándolas como ejemplo, se puede comprobar a simple vista que sin más que aplicar una serie de axiomas (distributividad, absorción, etc.) se puede simplificar convenientemente el conjunto de asignaciones requeridas. Por tanto, se impone como tarea previa a la planificación una optimización del número de operadores.

En el segundo apartado, se presentan los fundamentos teóricos en que se basa la optimización de operadores reticulares. A continuación en el tercer apartado, se describe el algoritmo de cálculo propuesto. Como cuarto apartado, se presentan algunos resultados de dicho algoritmo de optimización. Por último, se presentan las conclusiones y futuros desarrollos previstos.

2 FUNDAMENTOS TEÓRICOS

Tras aplicar los métodos de inferencia y agregación sobre la base de reglas se genera un conjunto de expresiones reticulares w_i que informan sobre cuál es el grado de certeza de cada una de las funciones de pertenencia de las salidas. Es decir, utilizamos el operador $\min$ para saber cuál es, como mínimo, la fuerza de aplicación de la parte antecedente de todas las reglas y finalmente, mediante el operador $\max$, agregamos todas las partes antecedentes de las reglas que poseen como parte consecuente la misma combinación salida-etiqueta de salida. En la ecuación (1) se muestra el aspecto de una cualquiera de dichas expresiones.

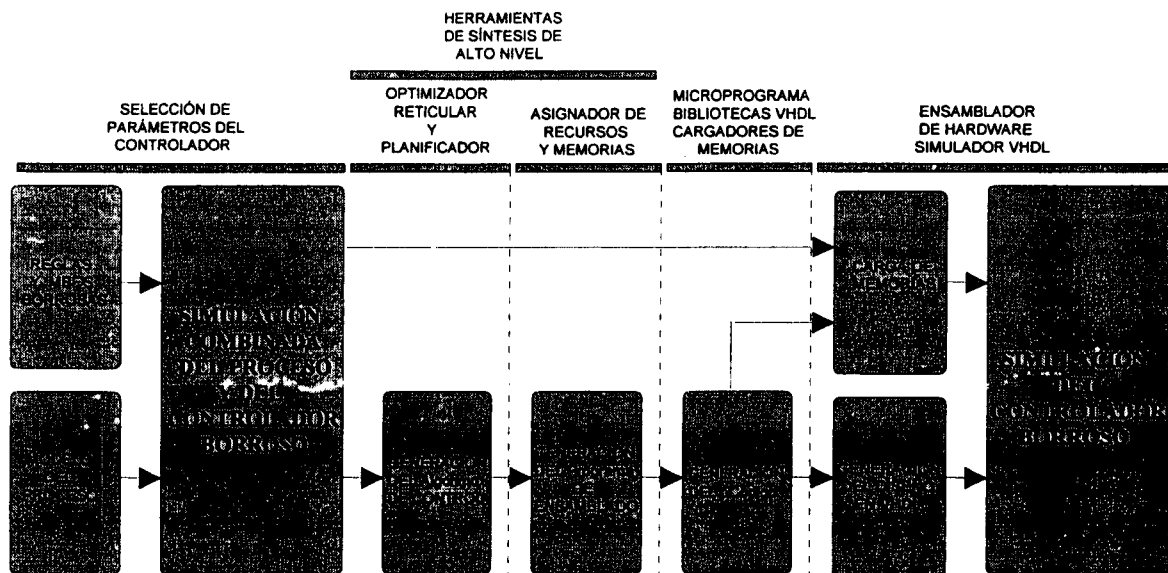


Figura 1: Flujo de diseño propuesto.

$$w_i = \max_{\substack{\forall \text{ reglas} \\ \text{con } y_j \neq 0}} (y_j) \text{ donde } y_j = \min (A_k) \quad (1)$$

siendo A_k los antecedentes de cada regla.

Por tanto, dada una serie de expresiones reticulares w_i , vamos a tratar de reducir el número de recursos *hardware* min y max de dos operandos al mínimo necesario para materializarlas; además, se van a reutilizar todas aquellas expresiones intermedias que sirvan para distintas etiquetas lingüísticas de salida.

2.1 RETÍCULOS DISTRIBUTIVOS

El conjunto L , con las operaciones suma y producto $(\vee, \cdot)$ es un retículo distributivo si, para todo a, b, c perteneciente a L , se cumplen los axiomas de idempotencia (2), conmutatividad (3), absorción (4) y distributividad (5):

$$a \vee a = a \quad a \cdot a = a \quad (2)$$

$$a \vee b = b \vee a \quad a \cdot b = b \cdot a \quad (3)$$

$$a \vee a \cdot b = a \quad a \cdot (a \vee b) = a \quad (4)$$

$$a \cdot (b \vee c) = a \cdot b \vee a \cdot c \quad a \vee b \cdot c = (a \vee b) \cdot (a \vee c) \quad (5)$$

El intervalo $[0,1]$ con las operaciones max y min, para la suma y el producto respectivamente, es un retículo distributivo [6].

2.2 EXPRESIONES RETICULARES CANÓNICAS

Las expresiones reticulares se definen por inducción:

- Las variables simples, $x_0, x_1, \dots, x_{n-1}$, son expresiones reticulares.

- Si E_1 y E_2 son expresiones reticulares, entonces $E_1 \vee E_2$ y $E_1 \cdot E_2$ son también expresiones reticulares.

Dos expresiones que incluyan a las variables $x_0, x_1, \dots, x_{n-1}$ se dice que son equivalentes si representan a la misma función $w: [0,1]^n \rightarrow [0,1]$.

A una expresión tipo suma de productos de la cual no se puede quitar ningún producto por aplicación de los axiomas se la denomina *expresión canónica disyuntiva*. A una expresión tipo producto de sumas de la cual no se puede quitar ninguna suma por aplicación de los axiomas se la denomina *expresión canónica conjuntiva*.

A continuación se exponen dos propiedades útiles para el desarrollo posterior:

- A toda expresión reticular le corresponde una expresión canónica disyuntiva (conjuntiva) equivalente. Basta aplicar reiteradamente los axiomas.
- Dos expresiones canónicas disyuntivas (conjuntivas) son equivalentes sí y sólo sí, después de reordenar los productos (las sumas) y las variables dentro de los productos (las sumas), son idénticas. Basta suponer que una función w tiene dos expresiones canónicas disyuntivas (conjuntivas) diferentes para llegar a una contradicción.

2.3 FUNCIONES RETICULARES

Toda función $w: [0,1]^n \rightarrow [0,1]$ que puede ser definida por una expresión reticular es una función reticular. El número de funciones reticulares de n variables es igual al número de expresiones canónicas disyuntivas (o conjuntivas) de n variables (considerando como iguales a las expresiones equivalentes).

2.4 ESQUEMA DE CÁLCULO

Toda función reticular puede ser definida mediante un esquema de cálculo basado en las operaciones min y max de dos operandos. Por ejemplo, para la función w (6) el esquema de cálculo directamente traducido es (7):

$$w = x_1 \cdot x_2 \cdot x_5 \vee x_1 \cdot x_2 \cdot x_4 \vee x_3 \cdot x_5 \vee x_3 \cdot x_4 \quad (6)$$

$$\begin{array}{ll} r_1 = x_1 \cdot x_2 & \downarrow \\ r_2 = r_1 \cdot x_5 & r_6 = x_3 \cdot x_4 \\ r_3 = x_1 \cdot x_2 & r_7 = r_2 \vee r_4 \\ r_4 = r_3 \cdot x_4 & r_8 = r_5 \vee r_6 \\ r_5 = x_3 \cdot x_5 & w = r_7 \vee r_8 \end{array} \quad (7)$$

A primera vista se comprueba que no es una solución óptima ya que se puede reducir la expresión original sin más que aplicar los axiomas y además se repiten expresiones reticulares intermedias, por ejemplo $x_1 \cdot x_2$.

A continuación se presenta una función w equivalente (8) calculada aplicando los axiomas y el esquema de cálculo correspondiente (9), donde ya se ha reducido el número de operadores y además no se repiten expresiones reticulares intermedias.

$$w = (x_1 \cdot x_2 \vee x_3) \cdot (x_4 \vee x_5) \quad (8)$$

$$\begin{array}{ll} r_1 = x_1 \cdot x_2 & \\ r_2 = r_1 \vee x_3 & \\ r_3 = x_4 \vee x_5 & \\ w = r_2 \cdot r_3 & \end{array} \quad (9)$$

Se observa en ambos esquemas que el número de operaciones es igual al número de letras (variables) de la expresión menos uno ($9 = 10 - 1$ y $4 = 5 - 1$). Esto es cierto para todo esquema de cálculo en el que los valores intermedios (r_i) aparezcan una sola vez en la parte derecha de las ecuaciones (estructura en forma de árbol). Esto se puede demostrar por inducción sabiendo que toda expresión E o bien es una variable o bien se puede descomponer en forma de suma ($E_1 \vee E_2$) o en forma de producto ($E_1 \cdot E_2$).

Si denotamos el número de letras o coste de la expresión E mediante $C(E)$, podemos decir que:

- si la expresión contiene una sola letra, el esquema carece de operaciones.
- si la expresión se descompone como $E_1 \cdot E_2$, la parte del esquema que corresponde a E_1 contiene $C(E_1) - 1$ operaciones y la parte del esquema que corresponde a E_2 contiene $C(E_2) - 1$ operaciones. En total, el esquema de cálculo de $E_1 \cdot E_2$ contiene el siguiente número de operaciones:

$$C(E_1) - 1 + C(E_2) - 1 + 1 = C(E_1) + C(E_2) - 1 = C(E_1 \cdot E_2) - 1$$

- si la expresión se descompone como $E_1 \vee E_2$ se aplica el punto anterior.

3 ALGORITMO DE OPTIMIZACIÓN PROPUESTO

En este apartado se presenta un método heurístico para la generación de expresiones reticulares casi-óptimas. Un método válido de optimización podría basarse en un algoritmo de inteligencia artificial que fuese capaz de aplicar los diferentes axiomas para reducir la expresión original.

Sin embargo, el método que se presenta aquí explota la existencia de dos expresiones canónicas para cada expresión w cuyo esquema de cálculo se desea optimizar. El algoritmo consiste en la generación de una serie de listas de expresiones reticulares ($L(0)$, $L(1)$, etc.) a través de una serie de etapas sucesivas. Su objetivo es la generación de un esquema de cálculo para la materialización de las funciones reticulares (w_1 , w_2 , etc.) definidas por una serie de expresiones (E_1 , E_2 , etc.).

3.1 TAREAS DEL ALGORITMO

Seguidamente se presentan las tareas a realizar durante las diferentes etapas del algoritmo:

• etapa inicial

- $L(0) = \emptyset$
- Para cada expresión E_j :
 - ◊ Se calculan las expresiones canónicas disyuntiva y conjuntiva equivalentes a E_j .
 - ◊ Se denota por E_{j0} a la expresión canónica de menor coste.
- Se eliminan de la lista las expresiones repetidas.

• etapa $i+1$

Se suponen generadas las listas $L(0)$, $L(1)$, ..., $L(i)$, es decir, en particular $L(i) = \{E_{1i}, E_{2i}, \dots\}$.

- $L(i+1) = \emptyset$
- Para cada expresión E_{ji} :
 - ◊ Si $E_{ji} = q_1 \cdot q_2 \cdot \dots \cdot q_r$ (expresión conjuntiva):
 - * Se calculan las $2^{r-1} - 1$ descomposiciones $E_{ji} = Fc \cdot Hc$, donde Fc es el producto de parte de las sumas q_k y donde Hc el producto de las restantes sumas.
 - * Para cada descomposición $Fc \cdot Hc$:
 - ⇒ Se calculan las expresiones canónicas disyuntivas Fd y Hd equivalentes a Fc y Hc .
 - ⇒ Se elige la descomposición $F \cdot H$ de menor coste $C(F) + C(H)$ global, donde $F = Fc$ ó Fd y $H = Hc$ ó Hd .

- * De entre las 2^{r-1} descomposiciones generadas anteriormente se elige la de menor coste efectivo; el coste efectivo de una descomposición $F \cdot H$ es $C(F)+C(H)$, donde $C(F)$ y $C(H)$ son los números de letras de F y H , a no ser que F ó H se encuentren ya a alguna de las listas $L(0)$, $L(1)$, ..., $L(i)$, $L(i+1)$, en cuyo caso el coste correspondiente $C(F)$ ó $C(H)$ es cero.
- * Si la descomposición elegida finalmente es $E_{ji} = F_{ji} \cdot H_{ji}$ se añaden ambas expresiones F_{ji} y H_{ji} a la lista $L(i+1)$, a no ser que ya estén incluidas en una de las listas $L(1)$, ..., $L(i)$, $L(i+1)$.
- * Si $F_{ji} = E_{st}$ (expresión número s de la lista $L(t)$ con $t \leq i+1$) y $H_{ji} = E_{uv}$ (expresión número u de la lista $L(v)$ con $v \leq i+1$) se añade al esquema de cálculo la ecuación: $e_{ji} = e_{st} \cdot e_{uv}$.
Si e_{st} (ó e_{uv}) es una variable, se sustituye e_{st} (ó e_{uv}) por dicha variable.
- ◊ Si $E_{ji} = q_1 \vee q_2 \vee \dots \vee q_r$ (expresión disyuntiva), se repiten los mismos pasos descritos anteriormente sin más que intercambiar los operadores $\cdot$ y $\vee$.
- ◊ Si E_{ji} es una variable no se hace nada.

El algoritmo alcanza su final cuando ninguna de las expresiones de la última lista obtenida se pueden descomponer, es decir son variables terminales y no expresiones complejas.

3.2 ENCUADRE DEL ALGORITMO

Como se ha comentado anteriormente, el algoritmo de optimización es un proceso que se va a ejecutar tras procesar la especificación cerrada de un problema de control borroso (Figura 2) y antes de la ejecución del algoritmo de planificación de nuestra herramienta de síntesis.

El optimizador de expresiones reticulares se ha integrado dentro del compilador que traduce las especificaciones borrosas a las estructuras de datos que requieren el resto de pasos de la síntesis.

En la creación de dicho compilador se han empleado las herramientas estándar de UNIX (Lex y Yacc) y en el optimizador en sí, gnu C como lenguaje.

El planificador empleado [7] realiza simultáneamente planificación y asignación basándose en un algoritmo de búsqueda, tipo *branch and bound*, acotada por heurísticas y guía basadas en un análisis global previo del grafo de flujo de datos del problema y de la librería de módulos disponibles para la materialización final.

SPECIFICATION SpecificationName

INPUTS

```
IdI1 IdLab1 IdLab2 ... IdLabI1 UOFD LeftVal RightVal PREC
ValNum DownVal UpVal PREC ValNum PATH MbfsPathI1
IdI2 IdLab1 IdLab2 ... IdLabI2 UOFD LeftVal RightVal PREC
ValNum DownVal UpVal PREC ValNum PATH MbfsPathI2
...
```

```
IdIn IdLab1 IdLab2 ... IdLabIn UOFD LeftVal RightVal PREC
ValNum DownVal UpVal PREC ValNum PATH MbfsPathIn
```

OUTPUTS

```
IdO1 IdLab1 IdLab2 ... IdLabO1 UOFD LeftVal RightVal PREC
ValNum DownVal UpVal PREC ValNum PATH PathO1
IdO2 IdLab1 IdLab2 ... IdLabO2 UOFD LeftVal RightVal PREC
ValNum DownVal UpVal PREC ValNum PATH PathO2
...
```

```
IdOn IdLab1 IdLab2 ... IdLabOn UOFD LeftVal RightVal PREC
ValNum DownVal UpVal PREC ValNum PATH PathOn
```

RULES

```
IF IdIna IS IdLabaa AND IdInb IS IdLabbb ...OR IdInc IS IdLabcc ...
THEN IdOuta IS IdLabaa AND IdOutb IS IdLabbb ...
...
```

```
IF IdIns IS IdLabss AND IdInd IS IdLabdd ...OR IdIna IS IdLabaa ...
THEN IdOutc IS IdLabee AND IdOuta IS IdLabaa ...
```

FUZZIFICATION

IdFuz

INFERENCE

IdInf

DEFUZZIFICATION

IdDef

Figura 2: Especificación del problema.

4 RESULTADOS EXPERIMENTALES

Para mostrar la bondad del algoritmo de optimización propuesto se presentan los resultados obtenidos sobre el ejemplo sugerido en [8].

En dicho ejemplo se modela la trayectoria de un camión que aparca en un muelle de carga marcha atrás. Del sistema se observan dos variables de entrada (x , ϕ), con cinco y siete etiquetas lingüísticas cada una, y se genera una salida de control (ψ) descrita con siete etiquetas lingüísticas. La base de reglas que modela el comportamiento del sistema se representa en la Figura 3.

Si se materializan directamente los operadores min y max inferidos de la especificación del problema se generan hasta un 26'98 % más de operadores de los estrictamente necesarios. Por tanto, una optimización previa del número de operadores reticulares revierte en una notable disminución del número de recursos hardware y, por ende, del número de interconexiones. A partir de aquí, entran en juego el resto de las herramientas propuestas en la Figura 1.

APPROXIMATE-BASED MULTI-MODAL SYSTEMS AND POSSIBILISTIC LOGIC: A COMPARATIVE STUDY

F. Esteva

Institut d'Investigació en Intel·ligència Artificial (CSIC)
Campus UAB, 08193 Bellaterra
e-mail: esteva@iia.csic.es

P. Garcia

Institut d'Investigació en Intel·ligència Artificial (CSIC)
Campus UAB, 08193 Bellaterra
e-mail: pere@iia.csic.es

L. Godo

Institut d'Investigació en Intel·ligència Artificial (CSIC)
Campus UAB, 08193 Bellaterra
e-mail: godo@iia.csic.es

R. Rodriguez

Departamento de Ciencias de la Computación
Univ. Nacional de Buenos Aires, Argentina
e-mail: ricardo@iia.csic.es

Resumen

El presente trabajo trata sobre los modelos de razonamiento aproximado basados en similitudes. A partir de una relación de similitud entre las interpretaciones del lenguaje, se habían definido previamente operadores consecuencia y un sistema multi-modal en el cual tenían una interpretación. En este trabajo se estudia de forma axiomática la noción de aplicación aproximación para representar "aproximadamente p". Se demuestra que viene definida por una relación fuzzy, se define un sistema multi-modal, generalización del definido para similitudes y se estudia la relación entre las nociones de consecuencia en lógica posibilística y en el sistema multi-modal definido.

Palabras clave: aproximación superior, similitud, razonamiento aproximado, sistema multi-modal, lógica posibilística.

1. INTRODUCTION

One of the goals of a variety of approximate reasoning models is to cope with inference patterns more flexible than those of classical reasoning. For instance, a pattern like this one,

*if A approximately entails B, and we observe A',
then it is plausible, at some extent, to conclude B
whenever A' is close enough to A,*

would be a generalization of the well-known modus ponens rule. This kind of patterns have been the focus of a huge amount of research in the field of fuzzy logic, where, in general, the statement "if A approximately

entails B" has been modelled as a fuzzy if-then rule whereas A, B and A' are modelled as fuzzy facts (see for instance [14]). However, terms like "approximately" or "close", although fuzzy, denote notions of resemblance or proximity among propositions which must not be necessarily fuzzy. One way of proceeding is to equip the set of interpretations or possible worlds Ω with a similarity relation S, that is, a reflexive, symmetric and t-norm transitive fuzzy relation (Introduced in [13] and generalized in [10]). This kind of approach was started out by Ruspini [9] by proposing a similarity-based semantics for fuzzy logic, trying to capture inference patterns like the so-called *generalized modus ponens*. Given a similarity relation S on the set Ω of interpretations of a boolean language L, Ruspini proposes two measures, the implication and consistency measures, to account for the degree with which a proposition B is an approximate consequence from, or is consistent with, another proposition A. Based on such measures, Ruspini provides then a formalization of the generalized modus ponens in fuzzy logic. This framework has been recently extended in [4] and compared to the possibilistic approach in [4], [3] and [5]. See also [8] for another approach to similarity-based reasoning.

From a logical point of view, several formalisms can be envisaged to capture a notion of similarity-based reasoning system. A basic reference is [1] where the authors consider different types of graded consequence relations which make sense in similarity-based reasoning. They also turn their attention to a non-standard fuzzy logic approach, and also the frameworks of sphere semantics and multi-modal logics are examined at the semantic level, in a similar way Fariñas and Herzig did it for possibility theory [7]. Finally in [5] and [6] the authors study some multi-modal systems related to implication measures and prove some completeness results.

In this paper we focus on weaker relations than t-norm transitive similarity relations. Namely we introduce an axiomatic definition of what we call approximation mappings which are proved to be obtained through the so-called approximation relations, i.e., fuzzy reflexive relations on the set of interpretations. Axioms characterizing symmetric, t-norm transitive and separating approximations relations are also provided. Following Ruspini we generalize implication and consistency measures for approximation relations and we present semantical definitions of the multi-modal system which arise from these generalized implication measures. Kripke structures are defined as pairs (Ω, R) where Ω is the set of interpretations and R is a fuzzy relation $R: \Omega \times \Omega \rightarrow [0,1]$. Logical consequence relations with respect to the three classes of models defined by the set of approximation relations, the set of similarity relations and the set of separating similarity relations respectively are studied and compared. Finally, in a restricted language, relationships between these different consequence relations and corresponding possibilistic consequence relation are also provided.

2. APPROXIMATION MAPPINGS

In this section we give an axiomatic approach to upper approximation mappings of classical propositions and prove to be in direct relation with a kind of fuzzy relations for which we generalize the notions of implication and consistency measures.

Let L be an atomic boolean algebra of classical propositions (with connectives $\vee, \wedge, \rightarrow, \neg$) and let Ω be the set of classical interpretations of L . We will denote by $[p]$ the subset of interpretations that makes the proposition p true and by $\delta_p: \Omega \rightarrow \{0,1\}$ its characteristic function. It is well known that the mapping between propositions and subsets of interpretations given by $p \mapsto [p]$ is one-to-one and so we will sometimes identify, if no confusion is possible, a interpretation w with the proposition p_w such that $[p_w] = \{w\}$ ⁽¹⁾. On the other hand we will denote by $PF(\Omega)$ the set of fuzzy sets on Ω with values on $[0,1]$, by μ_{p^*} the characteristic function of a fuzzy set $p^* \in P^*(\Omega)$ and by $[p^*]_\alpha$ the α -cut of p^* ([12]). Finally we will use $\otimes$ to denote a t-norm on the set $[0,1]$.

Definition 1.- A mapping $*$: $L \rightarrow PF(\Omega)$, $(p \rightarrow p^*)$, is said to be an upper approximation mapping if the following properties hold:

(1) This is always possible because of our assumption of atomicity of L .

- (i) $\mu_{p^*} \geq \delta_p$ ($p^* \supseteq [p]$) for every proposition p ,
- (ii) $(\bigvee_{i \in I} p_i)^* = \bigcup_{i \in I} p_i^*$ where $\bigcup$ denotes the fuzzy set union defined by the supremum, i.e., $\mu_{\bigcup_{i \in I} p_i^*}(w) = \sup_{i \in I} \mu_{p_i^*}(w)$.

Moreover,

- (iii) A mapping $*$ is said to be symmetric if for every $p, q \in L$,
 $[p^*]_\alpha \cap [q] \neq \emptyset$ iff $[q^*]_\alpha \cap [p] \neq \emptyset$ for every $\alpha \in [0,1]$.
- (iv) A mapping $*$ is said to be $\otimes$ -transitive if for every $p, q, r \in L$,
 $[p] \subseteq [q^*]_\alpha$ and $[q] \subseteq [r^*]_\beta$ imply $[p] \subseteq [r^*]_{\alpha \otimes \beta}$ for every $\alpha, \beta \in [0,1]$.
- (v) A mapping $*$ is said to be separating if it satisfies:
 $[p^*]_1 = [p]$, i.e., $\mu_{p^*}(w) = 1$ iff $w \in [p]$.

Definition 2.- A fuzzy relation R on Ω is said to be,

- an approximation relation if it is reflexive,
- a $\otimes$ -similarity relation if it is reflexive, symmetric and $\otimes$ -transitive and
- a separating $\otimes$ -similarity if it is $\otimes$ -similarity and separating ($R(w, w') = 1$ iff $w = w'$).

Proposition 1.- The following relationships between approximation relations and upper approximation mappings hold:

- (i) Given an upper approximation mapping
 $*$: $L \rightarrow PF(\Omega)$,
the relation R defined by
 $R(w, w') = \mu_{w^*}(w')$ for every $w, w' \in \Omega$,
is an approximation relation.
- (ii) Given an approximation relation R ,
the mapping $*$: $L \rightarrow PF(\Omega)$ defined by:
 $p \rightarrow p^*$ with $\mu_{p^*}(w) = \sup_{w' \models p} R(w, w')$,
is an upper approximation mapping.

The proof is an easy computation.

Proposition 2.- If R is the approximation relation associated to the upper approximation mapping $*$, then the following equivalences hold:

- a) The mapping $*$ is symmetric iff R is also symmetric
- b) The mapping $*$ is $\otimes$ -transitive iff R is also $\otimes$ -transitive
- c) The mapping $*$ is separating iff R is separating.

The proof is an obvious computation.

In this framework, given an approximation relation R we can define its corresponding implication and consistency measures of pairs of propositions p and q following Ruspini, by:

$$I_R(p|q) = \inf_{w' \models q} \sup_{w \models p} R(w, w') = \inf_{w' \models q} \mu_{p^*}(w')$$

$$C_R(p|q) = \sup_{w' \models q} \sup_{w \models p} R(w, w') = \sup_{w' \models q} \mu_{p^*}(w')$$

It is obvious that if q is a proposition such that $[q]$ is a singleton $\{w\}$, then, identifying q and w , we have

$$I_R(p|w) = C_R(p|w) = \mu_{p^*}(w).$$

Moreover, as in the case of a similarity relation, given a proposition q , the mapping

$C_R(_|q): L \rightarrow [0,1]$, defined by $(C_R(_|q)(p) = C_R(plq))$,

is a possibility measure while $I_R(_|q)$ is not a necessity measure. This is the reason that the similarity-based logical approaches we have studied are based in the implication measures. In [3] and [5] the relation between similarity and possibilistic logics are studied. In this paper we will complete this study and we will also extend it to the relationships between multi-modal systems based on approximation relations and possibilistic logic.

3. MULTI-MODAL SYSTEM BASED ON APPROXIMATION RELATIONS

Language. - Formulas of the new language $\mathcal{L}$ are built over the formulas of L adding the family of unary modal connectives $\Diamond_{\alpha}$ and $\Diamond_{\alpha}$ for every $\alpha \in [0,1]$.

Kripke Models. - A Kripke model $\Sigma_R = (\Omega, R)$ is a pair where Ω is the set of interpretations of L and R is an approximation relation.

Satisfiability. - It is defined by:

- a) If p is a non modal proposition, then satisfiability is defined as usual,
- b) $(w, \Sigma_R) \models \Diamond_{\alpha} A$ if $I_R(A|w) \geq \alpha$,
- c) $(w, \Sigma_R) \models \Diamond_{\alpha} A$ if $I_R(A|w) > \alpha$.

Note that this notion of satisfiability needs a definition of implication measures for modal formulas while the definition given before is only valid for non modal formulas. Nevertheless the implication measure for modal formulas A is defined as a natural extension in the following way,

$$I_R(A|w) = \text{Sup } (w, \Sigma_R) \models A \text{ } R(w, w')$$

On the other hand we define the family of dual modal operators $\Box_{\alpha} = \neg \Diamond_{\alpha} \neg$ and $\Box_{\alpha} = \neg \Diamond_{\alpha} \neg$ whose satisfiability relations are defined by,

- d) $(w, R) \models \Box_{\alpha} A$ if $I_R(\neg A|w) < \alpha$,
- e) $(w, R) \models \Box_{\alpha} A$ if $I_R(\neg A|w) \leq \alpha$.

Of course, these definitions can be extended in the usual way to satisfiability with respect to a model Σ_R (noted by $\Sigma_R \models \varphi$) and to a class of models Σ (noted by $\Sigma \models \varphi$). The notion of logical consequence can be defined as well but we will consider three different definitions in this paper. Given a set of formulas Γ and a formula φ , then we define:

- 1.- Logical consequence in a model Σ_R (noted by $\models_R$) :

$\Gamma \models_R \varphi$ iff $(\forall w \in \Omega) [(w, \Sigma_R) \models \psi \text{ for every } \psi \in \Gamma \text{ imply } (w, \Sigma_R) \models \varphi]$.

- 2.- Local logical consequence in a family of models Σ (noted by $\models_{\Sigma}$) :

$\Gamma \models_{\Sigma} \varphi$ iff $(\forall \Sigma_R \in \Sigma) (\Gamma \models_R \varphi)$

- 3.- Global logical consequence in a family of models Σ (noted by $\models_{\Sigma}$) :

$\Gamma \models_{\Sigma} \varphi$ iff $(\forall \Sigma_R \in \Sigma) (\Sigma_R \models \Gamma \text{ implies } \Sigma_R \models \varphi)$

Notice that the local logical consequence is stronger than the global one, i.e., $\Gamma \models_{\Sigma} \varphi$ implies $\Gamma \models_{\Sigma} \varphi$ but the converse is not true in general.

In [6] we study the corresponding multi-modal system in the case that R is a separating similarity relation. In this paper we are interested in the classes of models $\Sigma = \{(\Omega, R) \mid R \text{ is an approximation relation}\}$, $\Sigma^* = \{(\Omega, S) \mid S \text{ is a similarity relation}\}$ and $\Sigma^{**} = \{(\Omega, S) \mid S \text{ is a separating similarity relation}\}$. As it is well known in modal logic it is interesting to note that these three classes of models have different sets of valid formulas. Namely, the following relations hold for valid formulas with respect to the three different classes of models noted by V_{Σ} , V_{Σ^*} and $V_{\Sigma^{**}}$,

$$V_{\Sigma} \subset V_{\Sigma^*} \subset V_{\Sigma^{**}}$$

but the converse inclusions are not true in general.

An easy computation shows that

$$\Diamond_{\alpha} \Diamond_{\beta} A \rightarrow \Diamond_{\alpha \otimes \beta} A$$

belongs to V_{Σ^*} and does not belong to V_{Σ} and

$$\Diamond_{c1} w \rightarrow \Box_{c1} w$$

belongs to $V_{\Sigma^{**}}$ and does not belong to V_{Σ^*} .

Observe also that it is possible to define a multi-modal system taking as accessibility relations the open and closed α -cuts of a fuzzy relation. Namely, the satisfiability of the family of modal operators obtained in this way, and noted by $\langle \rangle_{\alpha}$ and $\langle \rangle_{\alpha}$, are defined by:

$(w, \Sigma_R) \models \langle A \rangle_{\alpha}$ if there exists w' such that $R(w, w') \geq \alpha$

and $w' \models A$

$(w, \Sigma_R) \models \langle A \rangle_{\alpha}$ if there exists w' such that $R(w, w') > \alpha$ and $w' \models A$.

Like in [6] it is easy to show that $\langle \rangle_{\alpha} = \Diamond_{\alpha}$ but this is not the case for $\langle \rangle_{\alpha}$ and $\Diamond_{\alpha}$ which collapse only in some particular cases, for example, when Ω is a finite set. This observation makes clear that, while $\Diamond_{\alpha}$ can be obtained from an accessibility relation, this is not the case for $\Diamond_{\alpha}$ in general.

4. RELATIONSHIP WITH POSSIBILISTIC LOGIC

In this section we study the relation between multi-modal logical consequences defined in the previous section and the possibilistic logical consequences.

Recall that possibilistic formulas are pairs of classical propositions and lower bounds of necessity or possibility measures, that is, the possibilistic language is $L_{PL} = \{(p, \Pi_\alpha), (p, N_\alpha) \mid p \in L \text{ and } \alpha \in [0,1]\}$. Models are possibility distributions $\pi: \Omega \rightarrow [0,1]$ and the possibilistic satisfiability relation is defined by:

$$\begin{aligned} \pi \models_{PL} (p, \Pi_\alpha) & \text{ iff } \text{Poss}_\pi(p) = \text{Sup} \{ \pi(w) \mid w \models p \} \geq \alpha, \\ \pi \models_{PL} (p, N_\alpha) & \text{ iff } \text{Nec}_\pi(p) = 1 - \text{Poss}_\pi(\neg p) \geq \alpha. \end{aligned}$$

Thus, the possibilistic logical consequences between a family of possibilistic formulas Γ and a formula ϕ is defined, as usual, by:

$$\Gamma \models_{PL} \phi \text{ iff for every possibility distribution } \pi \text{ such that } \pi \models_{PL} \Gamma, \text{ then } \pi \models_{PL} \phi.$$

The natural translation $T: L_{PL} \rightarrow \mathcal{L}$ from possibilistic to multi-modal formulas we will study is given by:

$$\begin{aligned} T(p, \Pi_\alpha) &= \diamond_{c\alpha} p, \\ T(p, N_{1-\alpha}) &= \square_{o\alpha} p. \end{aligned}$$

It is interesting to remark that T is one-to-one but $T(L_{PL})$ are formulas with only one modal operator applied to a classical propositions. Thus the relationship is defined between possibilistic logical consequence $\models_{PL}$ on formulas of L_{PL} and the different multi-modal logical consequences defined on the previous section restricted to the subset of formulas

$$\mathcal{L}' = T(L_{PL}) = \{ \diamond_{c\alpha} p, \square_{o\alpha} p \mid p \text{ is a classical proposition} \}.$$

First of all we present results about valid formulas. Let V_{PL} be the valid formulas in possibilistic logic and let V'_Σ, V'_{Σ^*} and $V'_{\Sigma^{**}}$ be the formulas of $\mathcal{L}'$ valid in the classes of models Σ, Σ^* and Σ^{**} respectively. Then the following equalities hold,

$$T(V_{PL}) = V'_\Sigma = V'_{\Sigma^*} = V'_{\Sigma^{**}}$$

The proof is as follows:

It is easy to show that if $\alpha \neq 0$, (p, Π_α) is valid in possibilistic logic and $\diamond_{c\alpha} p$ is valid in any of the classes Σ, Σ^* or Σ^{**} if, and only if, p is a classical tautology. Analogous result is valid for (p, N_α) and $\square_{o1-\alpha} p$. Obviously, (p, Π_0) and (p, N_0) is always valid (for any proposition p) in possibilistic logic as well as $\diamond_{c0} p$ and $\square_{o1} p$ in any of the classes Σ, Σ^* or Σ^{**} .

In the following we study the relations between the corresponding logical consequences.

Proposition 3.- Given an approximation relation R on Ω , for every $w \in \Omega$ let π_w be the possibility distribution defined by $\pi_w(w') = R(w, w')$. Then, if p is a non modal formula the following equivalences hold:

$$\begin{aligned} (w, R) \models \diamond_{c\alpha} p & \text{ iff } \pi_w \models (p, \Pi_\alpha) \\ (w, R) \models \square_{o\alpha} p & \text{ iff } \pi_w \models (p, N_{1-\alpha}) \end{aligned}$$

Proposition 4.- Given a possibility distribution π , there exists an approximation relation R such that

$$\begin{aligned} \pi \models (p, \Pi_\alpha) & \text{ iff for all } w, \sum_{R\pi} \models \diamond_{c\alpha} p \\ \pi \models (p, N_{1-\alpha}) & \text{ iff for all } w, \sum_{R\pi} \models \square_{o\alpha} p \end{aligned}$$

The proofs of these propositions are obvious. The only interesting point is the definition of R_π in proposition 4. Of course, the definition is given by $R_\pi(w, w') = \pi(w')$.

Theorem 1.- The possibilistic logical consequence is equivalent to the local logical consequence respect to the class of models Σ for the corresponding formulas in the multi-modal system, i.e., the following equivalences hold:

$$\begin{aligned} \Gamma \models_{LP} (p, \Pi_\alpha) & \text{ iff } T(\Gamma) \models_{\Sigma} \diamond_{c\alpha} p \\ \Gamma \models_{LP} (p, N_{1-\alpha}) & \text{ iff } T(\Gamma) \models_{\Sigma} \square_{o\alpha} p. \end{aligned}$$

The proof is a consequence of the notion of local consequence and propositions 3 and 4.

The theorem does not hold if the global consequence notion is used in the multi-modal system as the following counterexample shows:

Counterexample 1.-

Let p be a classical proposition such that p and $\neg p$ are satisfiable and let $\alpha \in (0,1)$. Then the following global consequence holds for every proposition q ,

$$\square_{o\alpha} p \models_{g\Sigma} \square_{o\alpha} p \wedge q$$

because there exist no approximation relation R satisfying the left part of the consequence relation. But this consequence relation is obviously not true in the local sense. Perhaps it is more evident that the translated consequence relation is not true in the possibilistic logic, that is, the following consequence relation does not hold,

$$(p, N_{1-\alpha}) \models_{LP} (p \wedge q, N_{1-\alpha}).$$

But the result of the theorem is also valid if we restrict the class of models to the ones defined by symmetric and $\otimes$ -transitive approximation relations, that is, to Σ^* . First of all we need some previous results.

In [4] we studied the relation between similarity relations and possibility distributions π . Here we recall the main results:

1.- Given a similarity relation S , to define a possibility distribution, we need a subset A of Ω . Then, we may define the following mapping:
 $(S, A) \rightarrow \pi_{S,A}$ where $\pi_{S,A}(w) = \text{Sup}\{S(w, w') \mid w' \in A\}$.
 Note that A takes the role of the evidential set in Ruspini's approach ([9]), i.e., the set of prototypes models corresponding to the information.

2.- Given a possibility distribution π , we may build the corresponding similarity relation, defined using Valverde's representation theorem [11], as the similarity relation S_π generated by π :
 $\pi \rightarrow S_\pi$ where
 $S_\pi(w, w') = \text{Min}(\pi(w) \otimes \rightarrow \pi(w'), \pi(w') \otimes \rightarrow \pi(w))$
 being $\otimes \rightarrow$ the residuated implication defined by $\otimes$.

Proposition 5.- The following relations hold:

- (i) If $S = S_\pi$ and $A \subseteq \text{Core}(\pi) = \{w \mid \pi(w) = 1\}$, then $\pi_{S,A} = \pi$.
- (ii) If S' is the similarity relation defined from $\pi_{S,A}$ being A a subset of Ω , then $S' \geq S$ and $S'(w, w') = 1$ for every $w, w' \in \Omega$.

The following theorem proves the equivalence between $\models_{PL}$ and $\models_{\Sigma^*}$. Before to prove it we need a previous lemma.

Lemma.- Given a similarity Kripke model $\Sigma_S = (\Omega, S)$ and a non-modal formula $p \in L$, then, for all $w \in \Omega$, the following equivalences hold,

$$\begin{aligned} (w, S) \models \Diamond_{c\alpha} p & \text{ iff } \pi_{S,w} \models (p, \Pi_\alpha) \\ (w, S) \models \Box_{o\alpha} p & \text{ iff } \pi_{S,w} \models (p, N_{1-\alpha}). \end{aligned}$$

Theorem 2.- The possibilistic logical consequence is equivalent to the local logical consequence of the corresponding formulas in the multi-modal system in the class of models Σ^* , i.e., the following equivalences hold:

$$\begin{aligned} \Gamma \models_{PL} (p, N_{1-\alpha}) & \text{ iff } T(\Gamma) \models_{\Sigma^*} \Box_{o\alpha} p \\ \Gamma \models_{PL} (p, \Pi_\alpha) & \text{ iff } T(\Gamma) \models_{\Sigma^*} \Diamond_{c\alpha} p. \end{aligned}$$

In this case the equivalence does also not hold if we take the global logical consequence relation as the following counterexample shows:

Counterexample 2.-

Let p and q be two different maximal elementary conjunctive propositions, i.e., $[p] = \{w'\}$ and $[q] = \{w''\}$ and let $\alpha \in (0, 1]$ such that $\alpha \otimes \alpha \neq 0$. Then, the following global logical consequence holds,

$$\Diamond_{c\alpha} w' \models_{\Sigma^*} \Diamond_{c\alpha \otimes \alpha} w''.$$

The proof is as follows. If a similarity relation S satisfies $I_S(w'lw) = S(w, w') \geq \alpha$ for every $w \in \Omega$, then $I_S(w''lw) = S(w, w'') \geq S(w, w') \otimes S(w', w'') \geq \alpha \otimes \alpha$. But this logical consequence is not true in the local sense because if S is the minimal classical equivalence relation

$(S(w, w') = 1 \text{ if } w = w' \text{ and } S(w, w') = 0 \text{ otherwise})$, it is obvious that

$$(w', S) \models_{\Sigma^*} \Diamond_{c\alpha} p \text{ and } (S, w') \not\models_{\Sigma^*} \Diamond_{c\alpha \otimes \alpha} q.$$

Perhaps it is more evident that the translated consequence relation is not true in possibilistic logic, that is,

$$(p, \Pi_\alpha) \not\models_{LP} (q, \Pi_{\alpha \otimes \alpha}).$$

An easy calculation show that the possibility distribution π defined by $\pi(w) = 1$ if $w = w'$ and $\pi(w) = 0$ otherwise, satisfies the left hand side but not the right hand side.

This example also proves that if we restrict the class of models to the ones defined by separating similarity relations, i.e. to Σ^{**} , possibilistic consequence relation does not coincide with global consequence relation respect to Σ^{**} .

Moreover possibilistic logical consequence and local logical consequence in Σ^{**} are not equivalent as the following counterexample shows:

Counterexample 3.- For any different interpretations w, w' and w'' , the following logical consequence holds:

$$\{\Diamond_{c1} w, \Diamond_{c1} w'\} \models_{\Sigma^{**}} \Diamond_{c1} w''$$

But it is not the case if we substitute Σ^{**} by Σ^* , i.e.,

$$\{\Diamond_{c1} w, \Diamond_{c1} w'\} \not\models_{\Sigma^*} \Diamond_{c1} w''.$$

The proof is an easy consequence of the separating property of the elements of Σ^{**} because this implies that no model of Σ^{**} satisfies the left hand side and so the logical consequence is valid in Σ^{**} but there are models of Σ^* satisfying the left hand side and not the right hand side. It suffices to take a model (S, w) defined by a similarity S such that $S(w, w') = 1 \neq S(w, w'')$.

5. CONCLUSIONS

The following schema shows the relations studied in the last section. In it Γ and ϕ denote a family of possibilistic formulas and a possibilistic formula respectively and $T(\Gamma)$ and $T(\phi)$ denote the corresponding formulas of $\mathcal{L}'$.

$$(\Gamma \models_{PL} \phi) \Leftrightarrow (T(\Gamma) \models_{\Sigma} T(\phi)) \Leftrightarrow (T(\Gamma) \models_{\Sigma^*} T(\phi)), \text{ and}$$

$$(T(\Gamma) \models_{\Sigma} T(\phi)) \Leftrightarrow (T(\Gamma) \models_{\Sigma^*} T(\phi)) \Rightarrow_1 (T(\Gamma) \models_{\Sigma^{**}} T(\phi))$$

$$\Downarrow_2$$

$$\Downarrow_3$$

$$\Downarrow_4$$

$$(T(\Gamma) \models_{\Sigma} T(\phi)) \Rightarrow_5 (T(\Gamma) \models_{\Sigma^*} T(\phi)) \Rightarrow_6 (T(\Gamma) \models_{\Sigma^{**}} T(\phi))$$

In this schema a subindexed arrow means that the converse arrow does not hold. Counterexample 1 proves that the converse of arrow 2 does not hold, counterexample 2 proves that the converse of arrows 3 and 5 do not hold and counterexample 3 proves that the converse of arrows 1, 4 and 6 do not hold.

Obviously, in this schema it is necessary to remark that the multi-modal system is restricted to the formulas of

$\mathcal{L}'$ and so, the general result is that possibilistic logical consequence is only a restriction of local logical consequence in the classes Σ and Σ^* .

Acknowledgement

This research has been partially supported by the UE-Copernicus 10053 project MUM and by the CICYT, TIC96-1038-C04-01 project SMASH.

References

- [1] Dubois D., Esteva F., Garcia P., Godo L., Prade H. (1995) Similarity-based consequence relations. In: Symbolic and Quantitative Approaches to Reasoning and Uncertainty (C. Froidevaux, J. Kohlas, eds.), Lecture Notes in Artificial Intelligence, Vol. 946, Springer Verlag, Berlin, 171-179. Extended version: Tech. Report IIIA 96/7, IIIA-CSIC.
- [2] Dubois D., Lang J., Prade H. (1994) Possibilistic logic. In: Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3 (D.M. Gabbay, C.J. Hogger, J.A. Robinson, D. Nute, eds.), Oxford University Press, 439-513
- [3] Dubois D., Prade H. (1995) Comparison of two fuzzy set-based logics: similarity logic and possibilistic logic. Proc. FUZZIEEE'95, Yokohama, Japan, 1219-1226
- [4] Esteva F, Garcia P, Godo L. (1994) Relating and extending semantical approaches to possibilistic reasoning. Int. J. of Approximate Reasoning, 10, 311-344.
- [5] Esteva F, Garcia P, Godo L. (1995) Un análisis modal de la Lógica de la Similitud, in: Anales de Informática Vol.1. Tecnologías y Lógica Fuzzy: ESTYLF'95 (R.Marin et al. eds) Universidad de Murcia, 153, 159.
- [6] Esteva F., Garcia P., Godo L., Rodriguez, O., (1996) A Modal Account of Similarity-based Reasoning, Tech. Report IIIA 96/8, IIIA-CSIC.
- [7] 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.
- [8] Klawonn F., Kruse R. (1993) Equality relations as a basis for fuzzy control. Fuzzy Sets and Systems, 54, 147-156.
- [9] Ruspini E. (1991) On the semantics of fuzzy logic. Int. J. of Approximate Reasoning, 5, 45-88.
- [10] Trillas E., Valverde L. (1984) An inquiry on T-indistinguishability operators. In: Aspects of Vagueness (H. Skala et al., eds.), Reidel, Dordrecht, 231-256.
- [11] Valverde L. (1985) On the structure of F-indistinguishability operators. Fuzzy Sets and Systems, 17, 313-328.
- [12] Zadeh L.A. (1965) Fuzzy sets. Information and Control, 8, 338-353.
- [13] Zadeh L.A. (1971) Similarity relations and fuzzy orderings. Information Sciences, 3, 177-200.
- [14] Zadeh L.A. (1979) A theory of approximate reasoning. Machine Intelligence, Vol. 9, Elsevier, New York, 149-194.

ALGUNOS TIPOS PARTICULARES DE ECUACIONES EN RELACIONES DIFUSAS

M.J. Fernández - F. Suárez
Departamento de Matemáticas-
Universidad de Oviedo
Calvo Sotelo s/n. 33007 Oviedo
e-mail: mjfg@pinon.ccu.uniovi.es

Resumen

En este trabajo estudiamos algunos tipos particulares de ecuaciones difusas. Más concretamente, nos limitamos a investigar las ecuaciones en las que intervienen dos subconjuntos difusos de un mismo referencial y una relación difusa definida sobre subconjuntos difusos.

1. INTRODUCCION

Presentamos en esta sección los conceptos básicos sobre los que se sustenta nuestro trabajo.

Definición 1 [9, 18]. Sea $I = [0,1]$ y sea $T : I \times I \rightarrow I$ tal que :

- a) $T(x,1) = T(1,x) = x$
- a') $T(x,0) = T(0,x) = x$
- b) Si $x_1 \leq x_2$ e $y_1 \leq y_2 \Rightarrow T(x_1, y_1) \leq T(x_2, y_2)$
- c) $T(x,y) = T(y,x)$
- d) $T[T(x,y), z] = T[x, T(y,z)]$

Si T verifica las propiedades a), b), c) y d) se dice que es una norma triangular (t-norma). Si satisface las propiedades a'), b), c) y d) se dice que es una conorma triangular (t-conorma). En adelante denotaremos T para una t-

norma y S para una t-conorma.

Dada una t-norma T la correspondiente t-conorma S , dual de T , está definida por:

$$S(x,y) = 1 - T(1-x, 1-y) \\ \forall x,y \in I$$

En el estudio de las ecuaciones en relaciones difusas tiene gran importancia el operador ϕ , asociado con una t-norma, cuya definición recordamos a continuación:

Definición 2 [13]. Un operador $\phi : I \times I \rightarrow I$ está asociado con una t-norma T si verifica:

- a) $\phi(a, \max\{b,c\}) = \max\{\phi(a,b), \phi(a,c)\}$
- b) $T(a, \phi(a,b)) \leq b \quad \forall a,b,c \in I$
- c) $\phi(a, T(a,b)) \geq b$

Si T es la t-norma del mínimo este operador es el introducido por Sanchez en [16].

Teorema 1 [6]. Un operador ϕ está asociado con una t-norma T si, y sólo si, T es semicontinua inferiormente

Teorema 2 [2,3,10,11]. Sea T una t-norma semicontinua inferiormente y ϕ el operador asociado con T . Se verifica:

$$\begin{aligned} \phi(a,b_1) &\leq \phi(a,b_2) \quad \text{si } b_1 \leq b_2 \\ \phi(a_1,b) &\geq \phi(a_2,b) \quad \text{si } a_1 \leq a_2 \\ \phi(\phi(a),b) &\geq a \\ \phi(a,b) &= \sup \{c \in I / T(a,c) \leq b\} \end{aligned}$$

Denotemos por $F(X)$ la familia de todos los subconjuntos difusos de X .

Definición 3 [12, 14]. Si $A \in F(X)$ y $R \in F(X \times Y)$ se define la sup-T composición de A y R como el subconjunto difuso $B \in F(Y)$ definido por:

$$B(y) = (ATR)(y) = \sup_{x \in X} [T(A(x), R(x, y))] \quad \forall y \in Y$$

Teorema 3 [12, 14]

Sean $B \in F(Y)$, $R \in F(X \times Y)$ y T una t-norma semicontinua inferiormente. Si la ecuación $ATR = B$ tiene soluciones $A \in F(X)$, entonces

$(R \phi B) \in F(X)$ definido por

$$(R \phi B)(x) = \inf_{y \in Y} [T(R(x, y), B(y))] \quad \forall x \in X$$

es la mayor solución.

Sea X un conjunto inicial y $A, B \in F(X)$. Usaremos la definición de producto cartesiano propuesta por Zadeh [20]: $A \times B$ es el subconjunto difuso de $X \times X$ definido por: $(A \times B)(x, y) = \min(A(x), B(y)) \quad \forall x, y \in X$

Definición 4 [1, 20]. Sea $R \in F(A \times B)$, es decir, $R(x, y) \leq \min(A(x), B(y)) \quad \forall x, y \in X$. Entonces R es una relación difusa de A en B .

2. COMPOSICION DE RELACIONES DIFUSAS

La composición max-min (sup-min) de dos relaciones difusas está fuertemente relacionada con el principio de extensión [21] como indica Sanchez en [17] y Pedrycz en [15]. En [25] hemos propuesto una generalización de este tipo de composición, utilizando las t-normas, en el mismo sentido con el que Dubois y Prade en [4] extienden las

operaciones difusas sobre números difusos.

Debido a la asociatividad de las t-conormas en [7] se define:

$$S\{x_1, x_2, \dots, x_n\} = S\{S\{x_1, x_2, \dots, x_{n-1}\}, x_n\} \quad n \geq 2$$

Así mismo, teniendo en cuenta la monotonía de las t-conormas y que el intervalo I es cerrado y acotado, se pueden definir [8] sobre un conjunto numerable:

$$S_{\infty N}\{x_i\} = \lim_{n \rightarrow \infty} S\{x_1, x_2, \dots, x_n\}$$

Sean A, B y C tres subconjuntos difusos de X , T una t-norma y S una t-conorma dadas. Si el referencial X es finito ó infinito numerable, podemos definir la composición de relaciones difusas como sigue:

Definición 5. Sean $Q \in F(A \times B)$ y $R \in F(B \times C)$. Definimos $Q_{ST}R$ como el subconjunto difuso cuya función de pertenencia viene dada por:

$$(Q_{ST}R)(x, y) = S_{z \in X} [T(Q(x, z), R(z, y))] \quad \forall x, y \in X$$

donde T y S son una t-norma y una t-conorma respectivamente, no necesariamente duales.

En el caso de que $S = \max$ y $T = \min$, estamos en la definición clásica de max-min composición de relaciones difusas.

Teorema 4 [5]. Para toda t-norma T , $Q_{ST}R$ será una relación difusa sobre subconjuntos difusos, para todo par de relaciones difusas $Q \in F(A \times B)$ y $R \in F(B \times C)$ si, y sólo si, $S = \max$.

Sabemos, por el teorema anterior, que dados tres subconjuntos difusos A, B y $C \in F(X)$ y dos relaciones difusas $Q \in F(A \times B)$ y $R \in F(B \times C)$, la relación difusa $Q \circ R$ definida por:

$$(Q \circ R)(x, y) = \sup_{z \in X} [T(Q(x, z), R(z, y))] \quad \forall x, y \in X$$

verifica: $Q \circ R \in F(A \times C)$

3. COMPOSICION DE UN SUBCONJUNTO DIFUSO Y UNA RELACION DIFUSA

Consideremos ecuaciones de relaciones difusas sobre subconjuntos difusos de la forma:

$A \circ R = B$ donde $A \in F(X)$, $R \in F(A \times B)$ y $B \in F(X)$

$A \circ R = B$ donde $A \in F(X)$, $R \in F(B \times B)$ y $B \in F(X)$

y nos planteamos los siguientes problemas:

- p1) Conocidos A y B , determinar R en la primera ecuación
- p2) Conocidos B y R , determinar A en la segunda ecuación.

Para el estudio de p1) hacemos uso del siguiente resultado:

Lema. Dados dos subconjuntos difusos $A, B \in F(X)$, se verifica:

$$A \circ (A \times B) \subset B$$

Demostración.

$$\begin{aligned} [A \circ (A \times B)](y) &= \sup_{x \in X} [T(A(x), (A \times B)(x, y))] = \\ &= \sup_{x \in X} [T(A(x), \min(A(x), B(y)))] \leq \\ &= \sup_{x \in X} [\min(A(x), \min(A(x), B(y)))] \\ &= \sup_{x \in X} [\min(A(x), B(y))] \leq B(y) \end{aligned}$$

$$\forall y \in X$$

c.q.d.

$$\text{Sea } \mathbb{R} = \{ R \subset A \times B / A \circ R = B \}$$

Teorema 5. Sean $A, B \in F(X)$. Si $\mathbb{R} \neq \emptyset$ se verifica que $A \times B$ es el mayor elemento de $\mathbb{R}$.

En efecto: Sea $R \in \mathbb{R}$. Entonces:

$$B = A \circ R \subset A \circ (A \times B) \subset B,$$

es decir, $A \circ (A \times B) = B$

Teorema 6. Si $T = \min$ y $\sup_{x \in X} A(x) \geq B(y) \quad \forall y \in X$, se verifica que:

$$A \times B \in \mathbb{R}$$

En efecto:

$$\begin{aligned} [A \circ (A \times B)](y) &= \sup_{x \in X} [\min(A(x), (A \times B)(x, y))] = \\ &= \sup_{x \in X} [\min(A(x), B(y))] = \\ &= \min(\sup_{x \in X} A(x), B(y)) = B(y) \end{aligned}$$

Además, en el caso particular de que $A(x) > 0$ y $B(x) = 0$ para todo x , Sessa [19] ha probado que la relación difusa $R(x, y) = \min(A(x), B(y)) = 0$ es la única solución de la ecuación $A \circ R = B$.

Si $T = \min$ y $A = B$, se verifica asimismo como corolario del teorema anterior que la relación difusa $A \times A$ definida por:

$$(A \times A)(x, y) = \min(A(x), A(y))$$

es el mayor elemento de $\mathbb{R}$

Respecto al problema p2) se verifica:

Teorema 7. Sean $B \in F(X)$ y R una relación difusa tal que $R \in F(B \times B)$. Si la ecuación $A \circ R = B$, siendo T una t -norma semicontinua inferiormente, tiene soluciones $A \in F(X)$ entonces,

$A(x) = 1$ es la mayor solución.

Demostración.

Sea ϕ el operador asociado a la t-norma T . Aplicando el teorema 3 se tiene que el subconjunto difuso $R \phi B \in F(X)$ definido por:

$$(R \phi B)(x) = \inf_{y \in X} [\phi(R(x,y), B(y))] \quad \forall x \in X$$

es la mayor solución.

Por otra parte $R \in F(B \times B)$, es decir,

$$R(x,y) \leq \min[B(x), B(y)] \leq B(y)$$

y teniendo en cuenta que:

$$\phi(a,b) = \sup \{ c \in [0,1] / T(a,c) \leq b \} = 1 \quad \text{si } a \leq b$$

se verifica:

$$(R \phi B)(x) = 1 \quad \forall x \in X \quad \text{c.q.d.}$$

El teorema anterior nos proporciona un método indirecto para comprobar cuándo una ecuación del tipo p2) no tiene soluciones; eso ocurrirá cuando $A \equiv 1$ no sea solución de la ecuación.

Ejemplo.

Sea $B = (0.7, 0.4)$

$$B \times B = \begin{bmatrix} 0.7 & 0.4 \\ 0.4 & 0.4 \end{bmatrix} \quad R = \begin{bmatrix} 0.6 & 0.3 \\ 0.1 & 0.2 \end{bmatrix}$$

$$(1, 1) T \begin{bmatrix} 0.6 & 0.3 \\ 0.1 & 0.2 \end{bmatrix} = (0.6, 0.3) \neq B$$

En este ejemplo, la ecuación $A T R = B$ no tiene soluciones $A \in F(X)$ siendo T cualquier t-norma semicontinua inferiormente.

Teorema 8. Sean $B \in F(X)$ y R una relación difusa tal que $R \in F(B \times B)$. Si llamamos

$$A = \{ A \in F(X) / A T R = B \},$$

se verifica:

$$A \neq \phi \quad \text{si, y sólo si,}$$

$$B(y) = \sup_{x \in X} R(x,y) \quad \forall y \in X$$

Demostración.

$$\text{Si } \forall y \in X$$

$$B(y) = \sup_{x \in X} R(x,y) \\ = \sup_{x \in X} [T(1, R(x,y))]$$

tomamos $A(x) = 1 \quad \forall x \in X$. Recíprocamente, si $A \neq \phi$ por el teorema anterior, $A(x) = 1$ es solución, es decir, $B(y) = \sup_{x \in X} [T(1, R(x,y))] = \sup_{x \in X} R(x,y)$.

c.q.d.

Bibliografía

- [1] M.K. Chakraborty and M. Das, Studies in fuzzy relations over fuzzy subsets, *Fuzzy Sets and Systems* 9 (1983), 79-89.
- [2] A. Di Nola, S. Sessa, W. Pedrycz and E. Sanchez, *Fuzzy Relation Equations and their Applications to Knowledge Engineering* (Kluwer Academic Publishers, Dordrecht-Boston-London, 1989).
- [3] J. Drewniak, Note on fuzzy relation equations, *Busefal* 12 (1982), 50-51.
- [4] D. Dubois and H. Prade, Additions of interactive fuzzy numbers, *IEEE Trans. Auto. Contr.* AC-26 (1981), 926-936.
- [5] M.J. Fernández, F. Suárez and P. Gil, Equations of fuzzy relation defined on fuzzy subsets, *Fuzzy Sets and Systems* 52 (1992), 319-336.
- [6] S. Gottwald, Fuzzy set theory with T-norms and ϕ -operators, in Topics in the Mathematics of Fuzzy Systems (A.

- Di Nola and A. Ventre, Eds.), Verlag TÜV Rheinland, Köln (1986), 143-196.
- [7] C. Kimberling, On a class of associative functions, *Public. Math. Debreces* **20** (1973) 21-39.
 - [8] E.P. Klement, Construction of fuzzy σ -algebras using triangular norms, *J. Math. Anal. Appl.* **85** (1982), 543-565.
 - [9] K. Menger, Statistical metric Spaces, *Proc. Nat. Acad. Sci. USA* **28** (1942), 535-537.
 - [10] M. Miyakoshi and M. Shimbo, Composite fuzzy relations with T-norms, *Trans. IECE. Japan* **J 67-D(4)** (1984), 391-398.
 - [11] M. Miyakoshi and M. Shimbo, Solutions of composite fuzzy relational equations with triangular norms, *Fuzzy Sets and Systems* **16** (1985), 53-63.
 - [12] W. Pedrycz, Fuzzy relational equations with triangular norms and their resolutions, *Busefal* **11** (1982), 24-32.
 - [13] W. Pedrycz, Some aspects of fuzzy decision-making, *Kybernetes* **11** (1982), 297-301.
 - [14] W. Pedrycz, Fuzzy relational equations with generalized connectives and their applications, *Fuzzy Sets and Systems* **10** (1983), 185-201.
 - [15] W. Pedrycz, *Fuzzy Control and fuzzy Systems*. (Wiley / Research Studies Press, London - New York, 1989).
 - [16] E. Sanchez, Resolution of composite fuzzy relation equations, *Inform. and Control* **30** (1976), 38-48.
 - [17] E. Sanchez, Compositions of fuzzy relations, in : M.M. Gupta, R.K. Ragade and R.R. Yager, Eds., *Advances in Fuzzy Set Theory and Applications* (North-Holland, Amsterdam, 1979) 421-433.
 - [18] B. Schweizer and A. Sklar, *Probabilistic Metric Spaces*. (North-Holland, Amsterdam, 1983).
 - [19] S. Sessa, Some results in the setting of fuzzy relation equation theory, *Fuzzy Sets and Systems* **14** (1984), 281-297.
 - [20] L.A. Zadeh, Similarity relations and fuzzy orderings, *Inform. Sciences* **3** (1971), 177-200.
 - [21] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, *Information Sciences*, Part I, **8**, (1975) 199-249 ; Part II, **8**, (1975) 301-357 ; Part III, **9**, (1975) 43-80.

DIAGNOSIS DE EQUIPOS MECÁNICOS BASADA EN LÓGICA DIFUSA

Ramón Ferreiro García

Dep. de Electrónica e Sistemas. Universidade da Coruña.
E.S. Marina Civil. Paseo de Ronda 51, 15011. A Coruña.
e-mail: ferreiro@sol.des.fi.udc.es

Xoán Carlos Pardo Martínez

Dep. de Electrónica e Sistemas. Universidade da Coruña.
Campus de Elviña, s/n. 15071. A Coruña.
e-mail: pardo@sol.des.fi.udc.es

Resumen

En este artículo se recoge el estudio y aplicación de un procedimiento difuso basado en computador para el diagnóstico de sistemas mecánicos utilizando una aproximación de reconocimiento de patrones. El tipo de sistemas de este trabajo es el de los mecanismos rotativos acoplados por ejes. El método de diagnóstico utiliza un conjunto de bases de reglas, determinísticas y difusas para detectar defectos de alineación y equilibrado. La diagnosis está basada en dos modelos de información: (a) información difusa (fuzzy) del estado actual de la máquina, (b) información de los patrones de fallos. Mediante el estudio propuesto se pretende identificar si uno o más patrones del tipo (b) están contenidos en la información del tipo (a) captada con un analizador de vibraciones.

Palabras clave: Diagnosis de Fallos, Lógica Difusa, Reconocimiento de Patrones.

1. INTRODUCCIÓN

El diagnóstico de fallos mecánicos basado en la utilización de computadores es una de las técnicas más populares en el mantenimiento global de equipos mecánicos. En esta aproximación se propone un método para diagnosticar defectos de alineación y equilibrado de ejes en sistemas mecánicos mediante el análisis del espectro que se obtiene a partir de los datos capturados mediante el equipo de medición de vibraciones adecuado [4].

Cortina [1] y Collacot [3] entre otros examinaron la posibilidad de aplicar las técnicas de clasificación de señal del Procesador de Bayes al diseño de sistemas de detección automática de fallos optimizados estadísticamente utilizando datos aprendidos de las pruebas. La técnica de decisión del Procesador de Bayes está basada en un

procedimiento estadístico Bayesiano [2], y ha sido aplicada durante mucho tiempo al diagnóstico de fallos.

La arquitectura general del procedimiento de diagnóstico basado en el reconocimiento de patrones está compuesto de las etapas siguientes [6]:

- Captación de información mediante un conjunto de sensores.
- Etapa de procesamiento formada por los pasos siguientes: parametrización de los datos, formación de las clases de patrones, selección de características y diseño del clasificador.
- Etapa de diagnóstico On-line en la que las nuevas observaciones son asignadas por el clasificador a las clases preespecificadas o rechazadas para su posterior análisis.

Estos pasos se resumen en la descripción siguiente: dado un patrón de fallo, la estrategia de resolución del problema consiste en la aplicación de un método estructurado para determinar si ese patrón está incluido en la información captada. La selección de la estrategia propuesta depende en parte del conocimiento a priori acerca del mecanismo analizado, la cual es esencial para la deducción del consecuente.

El conocimiento básico afecta a los siguientes aspectos: tipo de máquina, características dinámicas y estáticas, tipo de sensor, localización y posición, etc. y la diagnosis mediante procedimientos basados en reglas difusas pueden utilizarse con éxito como alternativa a los métodos estadísticos.

La secuencia a procesar se compone de las tareas siguientes:

- Obtención de la respuesta temporal de la máquina en estudio para hallar el espectro de las frecuencias de excitación mediante el cálculo de la FFT de los datos recogidos.
- Toma de decisiones/ diagnóstico basados en el análisis del espectro.

Este trabajo se centra en esta última tarea: cada tipo de fallo tiene un espectro individual y particular. Por lo tanto,

la búsqueda de un fallo particular implica reconocer y discriminar únicamente ese espectro. La técnica consiste en determinar si el espectro de un fallo particular está incluido en el espectro general. Así, comparando el espectro del patrón del fallo con el general y excluyendo las frecuencias no relevantes se decide si realmente existe el fallo. Para encontrar la información necesaria sobre un fallo se utiliza un espectro patrón de un fallo particular: por medio de estimaciones empíricas se calculan los conjuntos de la función de pertenencia para la base de reglas difusas. El universo de discurso es obtenido del espectro patrón. Eso implica que para cada fallo debe haber un espectro patrón como conocimiento a priori.

2. PROCEDIMIENTO GENERAL

El procedimiento general para realizar una tarea de diagnóstico basada en una estrategia de reconocimiento de patrones, en la que está involucrada información con incertidumbre, comprende una secuencia de tareas que se describen a continuación y que serán descritas más adelante:

- Medición de las vibraciones
- Cálculo de la FFT
- Clasificación de las mediciones mediante lógica booleana basada en un patrón de fallo dado.
- Cálculo del conjunto de la función de pertenencia.
- Deborrosificación de una base de reglas difusas.
- Procedimiento de corrección de acuerdo con los resultados de la deborrosificación.

Esta metodología es clara y bien estructurada para aquellas máquinas cuyos patrones de fallos son bien conocidos y las frecuencias que definen los fallos están claramente separadas de aquellas que pertenecen al funcionamiento normal de la máquina.

3. APLICACIÓN

En esta aplicación se propone diagnosticar una máquina rotativa industrial. Consideremos una máquina rotativa constituida por un motor, un eje y una bomba centrífuga, cuyas características son las siguientes:

- Potencia del motor: 75 kW
- rps = 24.92 Hz = 1495 rpm
- Localización del medidor = radial y horizontal, en el cojinete del motor en el lado de acoplamiento.

La figura 1 muestra la configuración del motor de

empuje de la bomba, puede observarse la colocación del medidor cerca de la unión del eje para detectar la familia de fallos cuyo patrón pertenece a los de pérdida de alineación o equilibrio.

La figura 2 muestra el espectro obtenido por el cálculo de la FFT de la respuesta temporal captada por el equipo de medición adecuado. Los síntomas de vibración típicos mostrados en la figura 2, son picos en 2 y 3 veces su frecuencia nominal (NF) de rotación.

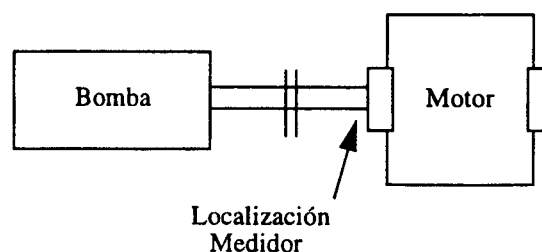


Figura 1: Configuración de la máquina y localización del equipo de medición.

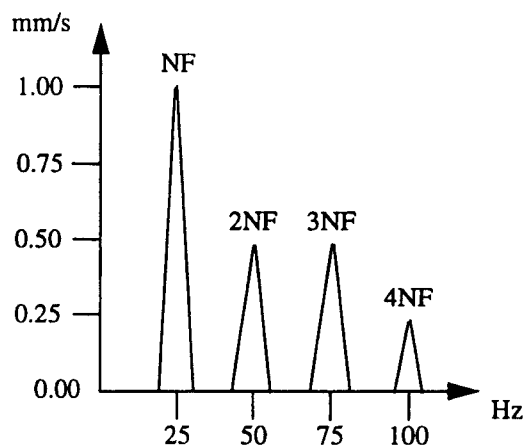


Figura 2: Espectro para la diagnosis de defectos de alineación y equilibrado.

4. CARACTERÍSTICAS DE LA BASE DE REGLAS

Es necesaria una base de reglas para cada familia de fallos. Por lo tanto, como estamos diagnosticando fallos de pérdida de alineación y equilibrio (una familia de fallos relacionada por su espectro patrón) entonces necesitaremos una base de reglas para ambos tipos de fallos.

La información más relevante y útil para diagnosticar defectos de alineación y equilibrado en la máquina rotativa, y que entrará en la base de reglas, es el significado que tiene la presencia de armónicos (Ha) en el espectro y de la relación entre ellos. Esta es la información que un

observador humano utilizaria para concluir el estado de la máquina.

El reconocimiento de los patrones de ese tipo particular de fallos va a ser identificado por la siguiente base de reglas, que comprenderá dos variables difusas de entrada y dos variables de salida. Las variables difusas de entrada y sus conjuntos de las funciones de pertenencia son los siguientes:

- La media de los armónicos (Ha) definida como:

$$Ha = [Mag(NF) + Mag(xNF)] / 2$$

con

$$\forall y, (y < x) \Rightarrow Mag(yNF) = 0$$

y

$$x \in \{2, \dots, n\}$$

y los conjuntos de la función de pertenencia definidos como:

Muy Bajo (VL)	$0 < Ha < 0.1$
Bajo (L)	$0.1 < Ha < 0.2$
Medio (M)	$0.2 < Ha < 0.4$
Alto (H)	$0.4 < Ha < 0.6$
Muy Alto (VH)	$0.6 < Ha$

- La relación de armónicos (Hr) definida como:

$$Hr = Mag(NF) / Mag(xNF)$$

con

$$x \in \{2, \dots, n\}$$

y

$$\forall y, (y < x) \Rightarrow Mag(yNF) = 0$$

y los conjuntos de la función de pertenencia:

Alto (H)	$3.3 < Hr$
Medio (M)	$1.5 < Hr < 3.3$
Bajo (L)	$0.5 < Hr < 1.5$

Las variables difusas de salida de la base de reglas con sus conjuntos de la función de pertenencia son definidos como:

- Estado de la máquina (MC) que se compone de cuatro conjuntos: Bueno (G), Utilizable (U), Aceptable (S) y No Aceptable (N).
- Corrección del fallo (FC) que se compone también de cuatro conjuntos: Desequilibrado (Un), Desequilibrado / Desalineado (Un/Mi), Desalineado /

Desequilibrado (Mi/Un) y Desalineado (Mi).

Las figuras 3 y 4 muestran las estructuras de las bases de reglas para los defectos de alineación y equilibrado.

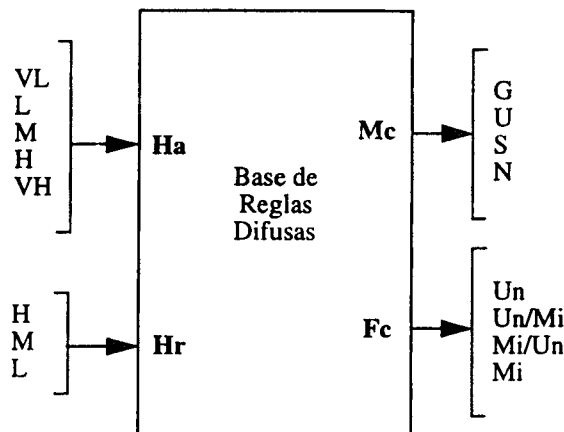


Figura 3: Estructura de la base de reglas de la aplicación.

	H	M	L
VL	G	G	G
L	S	U	U
M	S	S	S
H	N	S	N
VH	N	N	N

(a)

	H	M	L
VL			
L	Un		
M	Un	Mi/Un	
H	Un/Mi	Mi/Un	Mi
VH	Un/Mi	Mi/Un	Mi

(b)

Figura 4: Contenido de la base de reglas de la aplicación.

a) Estado de la máquina. b) Corrección del fallo.

La clasificación de la información por discriminación de los datos no relevantes es realizada mediante lógica booleana de la manera siguiente:

```

IF NF and 2NF then process Ha1 and Hr1
else IF NF and 3NF then process Ha2 and Hr2
else IF NF and 4NF then process Ha3 and Hr3
end IF function

```

donde

$$Ha1 = (Mag(NF) + Mag(2NF))/2$$

$$Hr1 = Mag(NF)/Mag(2NF)$$

$$Ha2 = (Mag(NF) + Mag(3NF))/2$$

$$Hr2 = Mag(NF)/Mag(3NF)$$

$$Ha3 = (Mag(NF) + Mag(4NF))/2$$

$$Hr3 = Mag(NF)/Mag(4NF)$$

El resultado del procesamiento de las reglas en la función IF asociado con la consiguiente expresión matemática, dará el valor de pertenencia de las variables difusas de entrada Ha y Hr. De acuerdo a esos valores se procede a la deborrosificación, obteniendo un estado de la maquina en el rango (G, U, S o N) y Un, Un/Mi, Mi/Un y Mi como se explicó anteriormente.

5. RESULTADOS Y CONCLUSIONES

El espectro mostrado en la figura 5 ilustra el aspecto del patrón de un fallo, que es el patrón de una familia obtenida a partir de la pérdida de alineación y/o equilibrio. Los resultados del procesamiento de la base de reglas con $Ha = M$, $Hr = M$ da, como puede verse en la figura 5, S y Mi/Un, lo que significa que el estado de la bomba todavia es aceptable, aunque la perdida de alineación y/o equilibrio está presente, y se recomienda un orden de corrección igual al de escritura. En la figura 6 se observa el espectro del motor de empuje de la bomba una vez realizada la corrección de la pérdida de equilibrio. En la figura puede verse que para $Ha = L$, $Hr = H$, la base de reglas da S y Un, lo que significa que se recomienda la corrección por pérdida de equilibrio aunque sigue siendo posible el funcionamiento de la bomba y el estado de la misma es claramente mejor.

En este trabajo hemos mostrado que el estado de monitorización puede realizarse con ayuda de un sistema experto difuso sencillo integrado en el equipo de monitorización. Como en todo proceso difuso hay un monton de incertidumbres, o los resultados no son deterministas, lo que requiere de la supervisión de un experto humano por inspección visual del espectro. Esta estrategia puede mejorarse a través del procesamiento de más información.

Agradecimientos

Esta investigación está financiada en parte por el CICYT español bajo el código TIC92-0267-C02-02.

Referencias

- [1] Andersdon, T. W., (1958), An introduction to multivariate statistical analysis, John Wiley & Sons Inc, New York.
- [2] Cortina W. E., Engel H. L. y Scott W. K., (1970), Pattern recognition techniques applied to diagnosis, SAE Paper 700497, Mayo.
- [3] Collacott R. A., (1977), Mechanical fault diagnosis and condition monitoring, Ed. Chapman and Hall, Great Britain.
- [4] Barschdorff D. Kramer S., (1993), Preprocessing for gearbox monitoring with a DSP-based diagnostic system. International Conference on Fault Diagnosis. Tooldiag'93, CERT ONERA, Toulouse, France.
- [5] Rolf Isermann y Mikaela Uliaru, (1993), Integrated fault detection and diagnosis. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics.
- [6] G. Morout, S. Bousghiri, J. Ragot, (1993), Pattern Recognition for diagnosis of technological systems; A review. IEEE. SMC. Conference Proceedings. Le Touquet. France 1993 Vol 5, pp 275-281

METRITTEK-DIV. MANTENIMIENTO
Machine : ALI-1.1
Date & Time : 20.12.94 17:58:57
Operator : MAN
Comment : AUSRICHTFEHLER

Window : Hanning
Envelope : No Envelope
Average : No average

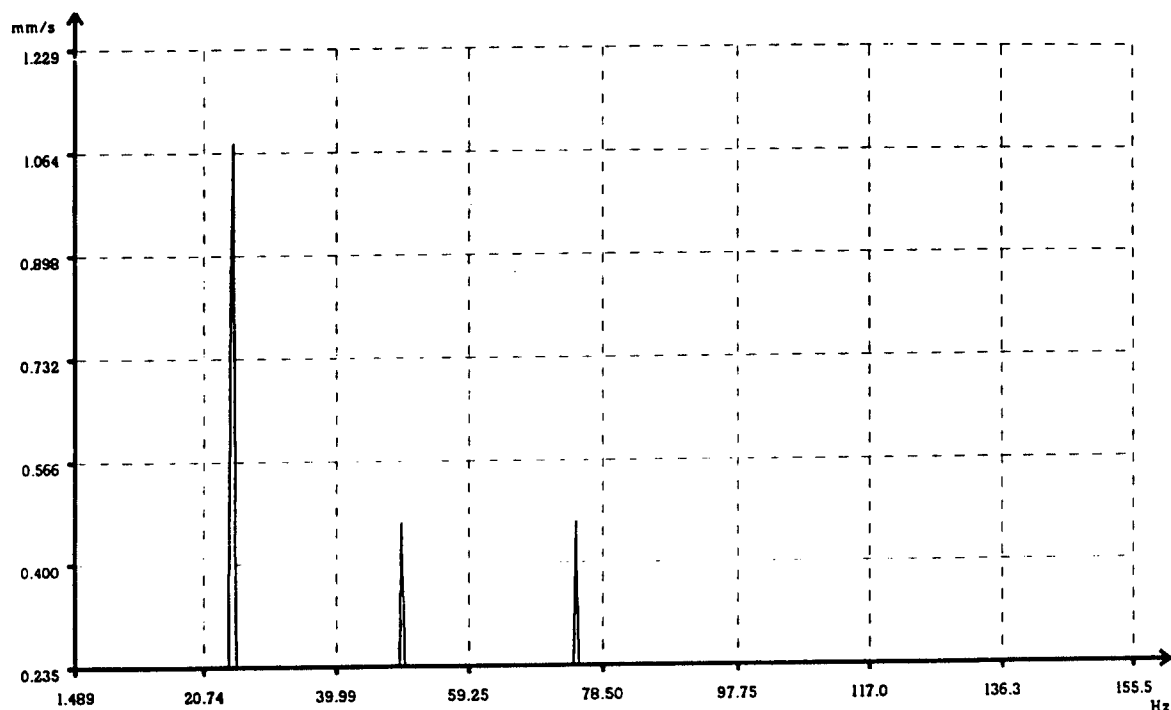


Table of maximum amplitudes and their frequencies:

Hz	mm/s
25.00	1.079
49.61	0.468
74.61	0.468
78.12	0.071
83.98	0.066
100.0	0.066
76.95	0.061
44.53	0.056
45.31	0.056
43.75	0.051

Figura 5

METRITEX-DIV. MANTENIMIENTO
Machine : ALI-1.2
Date & Time : 20.12.94 17:59:45
Operator : MAN
Comment : KEIN AUSRICHTFEHLER

Window : Hanning
Envelope : No Envelope
Average : No average

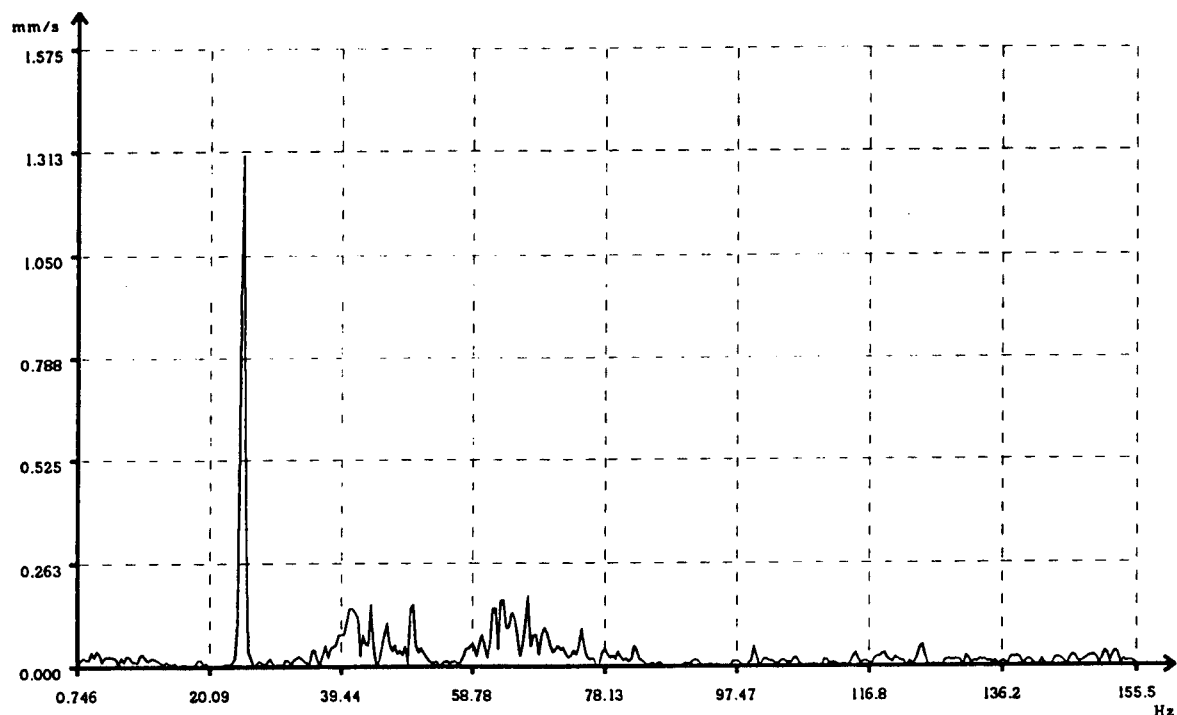


Table of maximum amplitudes and their frequencies:

Hz	mm/s
25.00	1.305
66.80	0.181
62.89	0.170
63.28	0.170
43.75	0.160
50.00	0.160
40.62	0.149
41.02	0.149
61.72	0.149
62.11	0.149

Figura 6

Un algoritmo para el refinamiento de reglas difusas de control *

Antonio GONZALEZ

Dpto. Ciencias de la Computación e I.A.
E.T.S.I. Informática
Universidad de Granada
18071-Granada (Spain)
gonzalez@decsai.ugr.es

Raúl PEREZ

Dpto. Ciencias de la Computación e I.A.
E.T.S.I. Informática
Universidad de Granada
18071-Granada (Spain)
fgr@decsai.ugr.es

Resumen

El aprendizaje automático de bases de reglas de control es una componente muy importante en el desarrollo de los controladores basados en lógica difusa. Recientemente, hemos desarrollado un algoritmo de aprendizaje, llamado SLAVE, que mediante la inducción aprende las reglas difusas más relevantes de un sistema. Asociado a SLAVE, propusimos un algoritmo de refinamiento que fue diseñado tanto para mejorar la capacidad de predicción como para mejorar la comprensión de las reglas. Los procesos básicos del algoritmo de refinamiento son la generalización y la especificación. Originalmente, este algoritmo se definió para problemas de aprendizaje de conceptos, en este trabajo, extendemos su aplicación a tipos más generales de reglas y exploramos su utilidad para problemas de control.

Palabras clave: aprendizaje automático, lógica difusa, reglas difusas, algoritmo de refinamiento, problemas de control.

1 Introducción

Los sistemas de control basados en implicaciones difusas constituyen uno de los campos de investigación más importantes en la teoría de sistemas difusos [1]. Como es conocido, un controlador basado en lógica difusa consta de cinco componentes fundamentales [5]: una base de datos que describe la semántica de las reglas a través de las funciones de pertenencia de los dominios de las distintas variables, una base de reglas de control que describen el comportamiento del sistema, un

proceso de inferencia que dada una entrada y teniendo en cuenta la base de datos y la base de reglas devuelve una salida, un módulo de difuminación (fuzzification) que transforma la entrada no difusa al controlador en una entrada difusa y un módulo de clarificación (defuzzification) que transforma la salida del proceso de inferencia en una salida no difusa.

Un tema de interés dentro de esta línea es el estudio de algoritmos que sean capaces de aprender o la base de reglas [7], o la base de datos del controlador, o ambas simultáneamente [6]. En [2] propusimos un algoritmo de aprendizaje de reglas, llamado SLAVE, que inicialmente se aplicó a problemas de aprendizaje de conceptos y posteriormente fue extendido para resolver problemas de control [3]. SLAVE utiliza una semántica fija sobre las etiquetas difusas y aprende cada regla de forma independiente. Este último hecho ocasiona que sea necesario realizar un proceso posterior de reajuste y refinamiento de las reglas adquiridas. En [4] propusimos un algoritmo de refinamiento para reglas con consecuente nominal. En este trabajo proponemos una versión modificada del anterior algoritmo capaz de trabajar sobre reglas con consecuente difuso, y por tanto, aplicable a problemas de control.

Los algoritmos existentes en la bibliografía, tratan de mejorar el comportamiento del controlador, alterando la semántica de las reglas (es decir, modificando la base de datos) o cambiando la base de reglas o modificando simultáneamente ambas componentes. El algoritmo de refinamiento que proponemos en este trabajo, trata de mejorar el comportamiento del controlador modificando exclusivamente la base de reglas sin alterar la base de datos.

Una componente fundamental del algoritmo de refinamiento es el modelo de regla difusa empleado. Este modelo de regla, que ya fue utilizado por SLAVE, representa en su parte antecedente una conjunción de las variables de estado, permitiendo disyunciones internas de valores en cada una de estas variables. Este modelo es importante porque permite determinar las variables

* Este trabajo ha sido parcialmente financiado por la CICYT dentro del proyecto TIC95-0453

relevantes para cada regla, de manera, que si una variable toma como valor todo su dominio de definición, esto indica que dicha variable no es necesaria para describir la regla. Utilizando este modelo de regla, el algoritmo de refinamiento se compone de procesos de generalización (añadir nuevos valores a una variable) y de especificación (eliminar valores de una variable).

En la siguiente sección describimos la extensión del algoritmo de refinamiento al problema del control, así como un ejemplo de su funcionamiento. En la sección 3 evaluamos el comportamiento de SLAVE junto con el algoritmo de refinamiento propuesto sobre el conocido problema del péndulo. Llamaremos SLAVE+R al algoritmo que se obtiene después de aplicar el algoritmo de refinamiento a las reglas obtenidas por SLAVE. Por último, mostramos las conclusiones de este trabajo.

2 El algoritmo de refinamiento para problemas de control

Inicialmente el algoritmo de refinamiento fué propuesto como una componente más del sistema de aprendizaje SLAVE en [4]. Este algoritmo fué diseñado tanto para mejorar la capacidad de predicción como para mejorar la comprensión de las reglas devueltas por SLAVE. Realmente, el algoritmo puede ser formulado de forma independiente a SLAVE y únicamente necesita partir de una base de reglas obtenidas por un experto o por cualquier sistema de aprendizaje que devuelva reglas compatibles con la descripción anterior. El proceso de refinamiento de reglas consta de un algoritmo heurístico que intenta minimizar el error de las reglas. El algoritmo usa una estrategia de ascensión de colinas y está compuesto por cuatro pasos que se repiten hasta que el error global y el número de reglas no cambien en dos iteraciones consecutivas. Antes de describir los cuatro pasos es necesario definir el concepto de corrección.

Cuando trabajamos con problemas en donde la variable consecuente toma un conjunto discreto de valores, podemos definir la corrección de una regla como la relación existente entre el número de ejemplos clasificados correctamente y el número de ejemplos que son cubiertos por esta regla. Este criterio se utilizó en [4]. En el caso, en el que la variable consecuente es continua, se debe extender el concepto de corrección, porque la salida del proceso predictivo depende de la interpolación entre todas las reglas que son aplicables. Definimos la corrección de una regla, en el caso continuo, como la media de los errores producidos en las salidas en las cuales esta regla se ha usado. Por extensión, definimos la corrección de un conjunto de reglas como la relación entre la media de los errores ocasionados al disparar todas las reglas, y el número de

ejemplos del conjunto de entrenamiento.

Una vez explicado el concepto de corrección, describiremos brevemente los pasos del algoritmo de refinamiento.

1. **Mejorar la corrección de las reglas:** Este paso consiste en un proceso de especificación de las reglas, donde se eliminan los valores irrelevantes de las variables antecedente. Consideramos que un valor de una variable antecedente es irrelevante, cuando al eliminar este caso de la descripción de la regla la corrección del conjunto de reglas mejora o se mantiene.

Este proceso de especificación puede entenderse como un proceso que disminuye el número de reglas que son aplicables a cada ejemplo del conjunto de entrenamiento, de manera, que si un valor de una variable puede especificarse para alguna regla, significa que en la base de reglas existen otras reglas cuya interpolación provoca un error menor, si la regla disminuye su grado de influencia. Por tanto, este paso realiza dos tareas interesantes:

- a) Fortalece las reglas con un alto grado de corrección.
- b) Debilita las reglas con un bajo grado de corrección hasta incluso llegar a eliminarlas. En este contexto, una regla desaparece del conjunto de reglas, si todos los valores posibles de una variable son irrelevantes. Por tanto, la eliminación de reglas se incluye como un caso especial de especificación.

2. **Aumentar el grado de influencia de las reglas en la interpolación de la salida:** En este paso, al contrario que en el anterior, tratamos de incrementar la interacción entre las distintas reglas mediante un proceso de generalización. El proceso consiste en añadir un nuevo valor a una variable antecedente siempre y cuando la corrección global es estrictamente mejorada. De tal forma, que sólo se permite que una regla forme parte del conjunto de reglas que influyen en la interpolación para un subconjunto de ejemplos, si su participación provoca una disminución del error global.

3. **Introducir nuevas reglas:** Supongamos que mediante el proceso anterior la interacción entre las distintas reglas no puede mejorarse generalizando algunas de las reglas existentes. En este caso, es posible que se necesiten nuevas reglas para decrementar los errores en las salidas. Este paso, es el encargado de introducir estas nuevas reglas para reducir el error, cuando el paso anterior, no es capaz de realizarlo.

El problema que surge en este paso es el siguiente: ¿Qué regla es la que debemos añadir al conjunto de reglas?

Responder a esta pregunta implica establecer un criterio para decidir entre todas las posibles reglas que se pueden introducir. Puesto que nuestro objetivo consiste en encontrar el conjunto de reglas que produzcan, en media, el menor error posible sobre el conjunto de entrenamiento. Para obtener este objetivo, establecemos un criterio basado en la elección de la regla más específica (es decir, la que menor distorsión provocará en el conjunto de reglas) que reduce el error del ejemplo peor representado del conjunto de entrenamiento, en otras palabras, construimos a partir del ejemplo que posee el máximo error individual, una regla que reduzca el mismo. En el caso de que exista más de un ejemplo que tenga el máximo error individual se tomará aleatoriamente cualquiera de ellos. La nueva regla se construye de la siguiente manera:

- como parte condición: el antecedente más específico que mejor cubre a dicho ejemplo.
- como parte consecuente: el valor del dominio que menor error global proporciona, al ser incluida esta regla en el conjunto de reglas ya existentes.

La idea subyacente en este criterio es que la introducción de una regla que mejore el máximo error individual, también mejorará el error medio global.

Este paso va introduciendo reglas hasta que se encuentra con un ejemplo que no mejora ni el máximo error individual del ejemplo, ni el error medio global. La regla que satisface esta condición no es introducida en la base de reglas.

4. Determinar las variables relevantes para cada regla: Este es el último paso del algoritmo de refinamiento que trata de determinar el conjunto de variables que son relevante para cada regla. Diremos que una variable es irrelevante, cuando no incrementa el error global al tomar esta variable como valor todo su dominio. Este proceso puede verse como un proceso especial de generalización.

El algoritmo de refinamiento empieza con un conjunto de reglas ordenado mediante un criterio de relevancia. Dadas dos reglas, consideraremos la regla más relevante aquella que cubre más ejemplos en el conjunto de entrenamiento. Este orden es uno de los componentes del proceso heurístico y decide la secuencia en la cual las reglas se modificarán. El orden en el cual

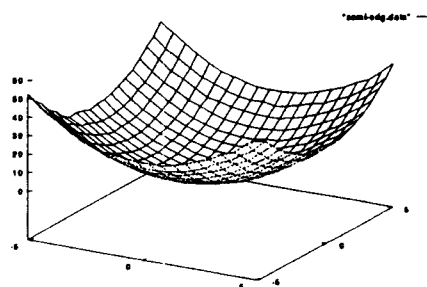
las variables y sus valores se modifican viene dado por el orden en el que estos aparecen en la descripción de la regla.

Otros dos componentes importantes en este algoritmo son el proceso de inferencia y la medida del error. El proceso predictivo utilizado es el sistema de inferencia max-min y como módulo de clarificación la media de los centros gravedad ponderados por la adaptación a la regla al ejemplo y el operador mínimo como t-norma. Basandonos en el sistema de inferencia, definimos una medida del error para establecer la precisión del conjunto de reglas. Esta medida de error es una modificación sobre el error cuadrático medio, y se define como:

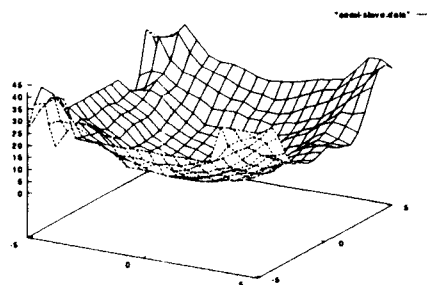
$$Error = \sum_{i=1}^n \frac{(y_i - y'_i)^2}{2n}$$

donde y_i es la salida correcta, y'_i es la salida del proceso de inferencia y n el número de ejemplos.

2.1 Un ejemplo del funcionamiento del algoritmo de refinamiento



(a) Superficie de control original

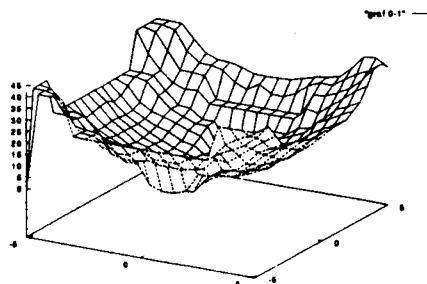


(b) Superficie de control inicial

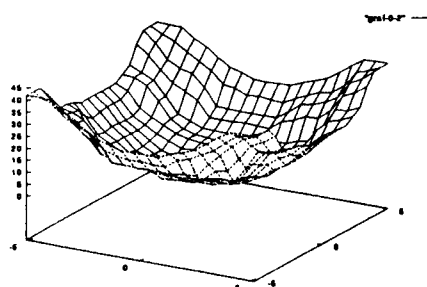
Figura 1: Superficie original y la obtenida por SLAVE

Supongamos que la figura 1(a) representa la superficie para controlar un determinado sistema. La figura 1(b) representa la superficie aprendida usando el algoritmo

SLAVE a través de un conjunto de ejemplos del sistema original. Podemos observar que las reglas propuestas por SLAVE aproximan bastante correctamente la zona central de la superficie, pero no así los extremos de la gráfica.



(c) Superficie después del paso 1



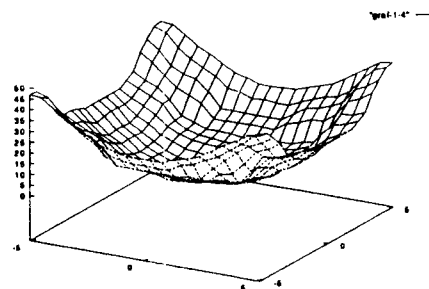
(d) Superficie después del paso 2

Figura 2: Los dos primeros pasos del ejemplo

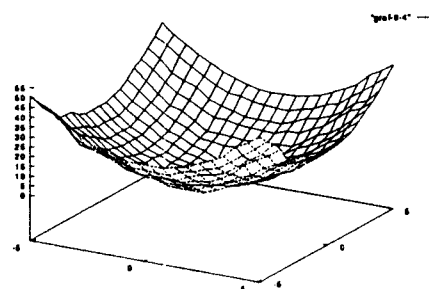
El algoritmo de refinamiento intenta corregir la superficie de control, disminuyendo el error. En la figura 2(c) se muestra la superficie después del primer paso. Aparentemente, esta nueva superficie presenta una aproximación peor que la inicial aunque el error es mejor. Este aspecto es debido a la influencia negativa o insuficiente de algunas reglas en el proceso de interpolación. En las siguientes iteraciones del algoritmo, se va suavizando este comportamiento que inicialmente es brusco.

Después de este paso, podemos ver la evolución del algoritmo de refinamiento como un proceso de aproximación suave a la superficie original. En la figura 2(d) podemos observar que la superficie experimenta una sensible mejora después del proceso de generalización. Esto es debido al aumento de las reglas que influyen positivamente en la interpolación sobre los subconjuntos de ejemplos mal representados.

La figura 3(e) muestra el comportamiento del conjunto de reglas después del paso 3. En esta gráfica podemos observar que las reglas existentes no eran suficientes para aproximar los extremos de la gráfica, pero



(e) Superficie después del paso 3



(f) Superficie de control estimada

Figura 3: Paso 3 y superficie estimada del ejemplo

después de incluir las nuevas reglas esta aproximación se mejora.

En este ejemplo, la gráfica después del cuarto paso no se muestra debido a que en este problema no existen variables irrelevantes, y por lo tanto, este paso devuelve la misma superficie que se obtuvo en el paso 3.

Para este ejemplo, el algoritmo de refinamiento se repitió ocho veces hasta que la verificó la condición de terminación. En la figura 3(f) podemos ver la superficie de control resultante después de la ejecución del algoritmo. Se puede observar que es bastante similar a la superficie original, y que mejora notablemente las superficies de partida (figura 1(b)).

3 Aplicación del algoritmo de refinamiento a un problema de control

Un problema tradicional de control, utilizado para comprobar el comportamiento de los nuevos controladores difusos, es el problema del péndulo. El problema del péndulo consiste en, dados un conjunto de ejemplos que describen el comportamiento del sistema, aprender el conjunto de reglas que permiten controlarlo. Si asumimos que el valor de $|\theta| \ll 1$ (radianes)

la ecuación diferencial no lineal que describe el comportamiento del péndulo es

$$m \frac{l^2 d^2 \theta}{3 dt^2} = \frac{l}{2} (-f + m g \sin \theta - k \frac{d\theta}{dt})$$

donde $k \frac{d\theta}{dt}$ es una aproximación de la fuerza de fricción.

Por tanto, el sistema se puede describir a través de dos variables de estado θ (ángulo) y ω (velocidad angular) y la variable de control f (fuerza). Para la simulación real se ha considerado un péndulo de 5 kg de peso y 5 m de largo, al que se han aplicado fuerzas sobre el centro de gravedad cada 10 ms. Con estos parámetros, el universo de discurso de las variables son los siguientes:

$$\theta \in [-0.277, 0.277]$$

$$\omega \in [-0.458, 0.458]$$

$$f \in [-1593, 1593]$$

De forma experimental, hemos obtenido dos conjuntos de ejemplos bajo las anteriores condiciones, el primero de ellos tiene 213 ejemplos y se ha usado para aprender (conjunto de entrenamiento) y el segundo contiene 125 ejemplos y se ha utilizado para comprobar el comportamiento de las reglas aprendidas (conjunto de test). Estos conjuntos de ejemplos se han obtenido de las ecuaciones anteriores teniendo en cuenta dos condiciones iniciales diferentes:

a) $\theta = -0.277$ y $\omega = 0$.

b) $\theta = 0.277$ y $\omega = 0$

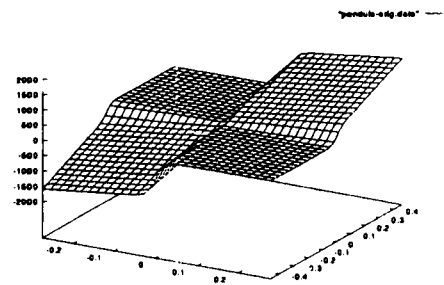
La figura 4(a) representa la superficie de control sobre las anteriores condiciones.

La tabla 1, muestra el número de reglas y el error obtenido usando el conjunto de ejemplos de test y otras bases de reglas que se han propuesto en la literatura para este problema. En todos los casos, se ha usado el sistema de inferencia max-min, como método de defuzzificación la media del centro de gravedad ponderado por la adaptación y como t-norma el mínimo.

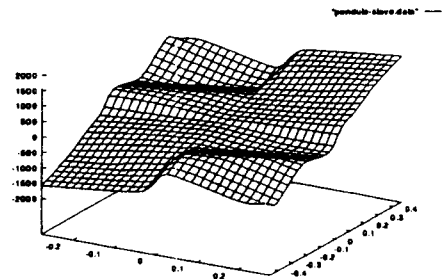
Tabla 1: Tabla de resultados

PROPUESTA	E^2 Medio	REGLAS
SLAVE	492.16	6
Wang y Mendel	7.45	9
SLAVE+R	1.21	5

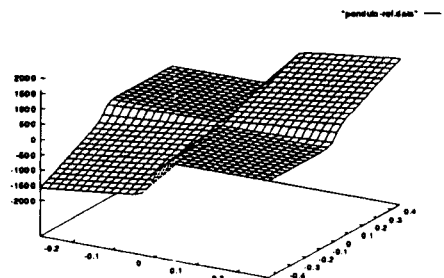
La figura 4(b) representa la superficie de control obtenida por SLAVE y la figura 4(c) por el algoritmo de refinamiento, tomando como conjunto de reglas de partida, las obtenidas usando SLAVE (SLAVE+R).



(a) Superficie de control original



(b) Superficie de control SLAVE



(c) Superficie de control SLAVE+R

Figura 4: Superficies de control

Los resultados de Wang y Mendel se han obtenido a partir del algoritmo de aprendizaje propuesto en [7] y usando la misma discretización difusa para los tres algoritmos.

4 Conclusiones y trabajos futuros

En este trabajo se ha propuesto un algoritmo para el refinamiento de la base de reglas de un controlador sin alterar la base de datos. Este algoritmo de refinamiento parte de un conjunto de reglas que describen un determinado sistema y mediante procesos de generalización y especificación, guiados por una heurística, trata de mejorar el comportamiento del controlador. El conjunto inicial de reglas puede ser obtenido a partir de un algoritmo de aprendizaje automático, o bien haber

sido propuesta por un experto.

En la parte experimental, el algoritmo de refinamiento aplicado a las reglas que devuelve el algoritmo de aprendizaje SLAVE (SLAVE+R), ha demostrado mejorar los resultados obtenidos por el propio SLAVE y el algoritmo de aprendizaje propuesto por Wang y Mendel.

Una línea futura de estudio consiste en usar el algoritmo de refinamiento en combinación con algoritmos de ajuste que sean capaces de modificar la semántica de las reglas.

Referencias

- [1] Bonissone, P.P. (1994), Fuzzy Logic Controllers: An Industrial Reality. In Computational Intelligence: *Imitating Life*. IEEE Press pp.316-327.
- [2] González, A., Pérez, R., Verdegay, J.L. (1993), Learning the structure of a fuzzy rule: a genetic approach, *Proc. First European Congress on Fuzzy and Intelligent Technologies*, Vol. 2, pp. 814-819.
- [3] González, A., Pérez, R., (1996) , A learning system of fuzzy control rules *Genetic algorithms and soft computing*. Eds. F. Herrera and J.L. Verdegay, Springer-Verlag, pp. 202-225.
- [4] González, A., Pérez, R., (1996) A refinement algorithm of fuzzy rules for classification problems, *por aparecer en los Proceedings of the IPMU'96*. (1996)
- [5] Lee, C.C., Fuzzy Logic in Control Systems: Fuzzy Logic Controller (1990)- Parts I and II. *IEEE Transactions on Systems, Man and Cybernetics*, 20, pp. 404-435 (1990)
- [6] Sugeno, M., Tanaka, K., (1991) Successive identification of a fuzzy model and its applications to prediction of a complex system, *Fuzzy Sets and Systems* 42, pp. 315-334.
- [7] Wang, L., Mendel, M., (1992) Generating fuzzy rules by learning from examples, *IEEE Transactions on systems, man and cybernetics*, vol.22 n.6, pp.1414-1427 (1992)

CONJUNTOS, PUNTOS Y NÚMEROS GRADUALES

José A. Herencia

Dep. Matemáticas, Fac. de Ciencias, Univ. de Córdoba

Av. San Alberto Magno, 14071-CÓRDOBA

E-mail: malhegoj@uco.es

Resumen

Consideramos los “conjuntos graduales”, cuya definición está sugerida por las familias de α -cortes de conjuntos difusos, así como las correspondientes extensiones de los conceptos de punto y de número real (también análogas a las definiciones efectuadas en el caso difuso). De esta forma se solventan los inconvenientes señalados por Gerla (para los puntos difusos) y por Kaleva y Seikkala (para la convergencia de números difusos).

Palabras clave: α -corte, punto difuso, número difuso.

1 INTRODUCCIÓN

En los más diversos aspectos de la Matemática Difusa nos encontramos con una gran variedad de criterios y definiciones, como pone de manifiesto Kerre [15]. Frente a esta heterogeneidad (que suele darse al manejar funciones de pertenencia), el uso de α -cortes aporta un método unificador [19] que ha resultado fructífero en distintos campos, tanto teóricos como aplicados, especialmente en la Teoría de la Decisión y Optimización en Ambiente Difuso [1, 2, 21, 22]... Pero las familias de α -cortes de conjuntos difusos están sujetas a las condiciones establecidas en el conocido Teorema de representación de Negoita y Ralescu, las cuales resultan restrictivas en determinados casos. Por ejemplo, al operar números difusos mediante el Principio de extensión de Zadeh [17] o al calcular el límite por niveles de sucesiones de números difusos [13].

Con objeto de aprovechar la utilidad de las familias de α -cortes, pero evitando a la vez los inconvenientes derivados de las condiciones a que están sujetas, el autor introduce los denominados “conjuntos graduales” [8]. Se trata de familias de subconjuntos

sujetas a la única condición de decrecimiento (que es más débil que las condiciones establecidas por Negoita y Ralescu). Esto permite asimismo definir muchos “conceptos graduales” por analogía a los correspondientes conceptos difusos [10]. En particular, consideramos aquí las definiciones de “punto gradual” [8] y de “número gradual” [9], debido a las interesantes propiedades que poseen (según veremos más adelante).

La condición de decrecimiento exigida a los conjuntos graduales hace que éstos mantengan una estrecha relación con los conjuntos difusos. Concretamente, dado cualquier conjunto gradual ψ , se demuestra la existencia de un único conjunto difuso μ_ψ (que denominamos su “conjunto difuso asociado”) tal que ψ está comprendido entre la familia de α -cortes fuertes y la familia de α -cortes débiles de μ_ψ . Recíprocamente, dado cualquier conjunto difuso μ , es evidente que cualquier familia de subconjuntos comprendidos entre sus dos tipos de α -cortes es decreciente (por lo que denominaremos a estas familias “conjuntos graduales asociados” al conjunto difuso μ).

El resultado anterior sugiere representar los conjuntos difusos mediante cualesquiera de sus conjuntos graduales asociados (no necesariamente iguales a sus α -cortes). De esta forma resulta un método de representación que convierte a los conjuntos graduales en una herramienta útil en la Matemática Difusa. Aquí consideraremos este método para deducir propiedades de los números difusos a partir de los números graduales. Concretamente establecemos relaciones entre los números difusos de Zadeh y los de Hutton (que se estudian de forma inconexa en la Matemática Difusa) y definimos la “convergencia gradual” de sucesiones de números difusos. Este nuevo criterio extiende la convergencia usual de $\mathbb{R}$ (lo que no ocurre con la convergencia puntual de funciones de pertenencia) y asegura la existencia del límite de cualquier sucesión monótona y acotada (propiedad que no cumple la convergencia por niveles).

El orden que seguimos a continuación es el siguiente.

En primer lugar (en la sección 2) fijamos las definiciones y notaciones correspondientes a los distintos conceptos difusos usados. Las secciones 3, 4 y 5 contienen las definiciones y resultados fundamentales sobre conjuntos, puntos y números graduales, respectivamente. Finalmente, en la sección 6, se enumeran algunos de los resultados sobre números difusos que se deducen a partir de los resultados análogos previamente obtenidos para los números graduales.

2 CONJUNTOS, PUNTOS Y NÚMEROS DIFUSOS

En lo sucesivo notamos por $\mathbf{I}$ al intervalo unidad $[0, 1]$, por $\mathbf{R}$ a la recta real y por $\mathbf{N}$ al conjunto de números enteros positivos. Dado cualquier conjunto X , notamos por $\mathcal{P}(X)$ al conjunto de partes [ordinarias o "crisp"] de X y por $\mathcal{F}(X) := \{\mu : X \rightarrow \mathbf{I}\}$ al conjunto de partes difusas (o borrosas) de X . Consideramos que $\mathcal{P}(X) \subset \mathcal{F}(X)$, mediante la identificación $A \equiv \chi_A$, siendo χ_A la función característica del subconjunto crisp A . Para cualquier $\alpha \in \mathbf{I}$ y cualquier conjunto difuso $\mu \in \mathcal{F}(X)$, tenemos su α -corte [débil] $\mu^{(\alpha)} := \mu^{-1}[\alpha, 1]$ y su α -corte fuerte $\mu^{(\alpha)} := \mu^{-1}(\alpha, 1]$.

La extensión de aplicaciones arbitrarias $f : X \rightarrow Y$ (resp. $f : X^2 \rightarrow Y$) a $f : \mathcal{P}(X) \rightarrow \mathcal{P}(Y)$ (resp. $\mathcal{P}(X)^2 \rightarrow \mathcal{P}(Y)$) la efectuamos en la forma usual, mientras que la extensión $f : \mathcal{F}(X) \rightarrow \mathcal{F}(Y)$ (resp. $\mathcal{F}(X)^2 \rightarrow \mathcal{F}(Y)$) se obtiene mediante el Principio de extensión de Zadeh.

Se llama **punto** (o "singleton") **difuso de soporte** $x \in X$ y **valor** $\lambda \in (0, 1]$ al subconjunto difuso $\lambda_x := \lambda \chi_{\{x\}}$. Respecto a la pertenencia de puntos difusos a conjuntos difusos, citamos tres definiciones distintas aparecidas en la literatura:

- Según Kerre [14],

$$\lambda_x \in \mu \Leftrightarrow \lambda_x \subseteq \mu \text{ (lo que equivale a } \lambda \leq \mu(x)).$$

- Debido a Wong [23] (que aportó la definición original) y a Gottwald [7] (quien la mejoró, evitando ciertas contradicciones), tenemos este otro criterio de pertenencia, según el cual no se admite el valor $\lambda = 1$ en la definición de punto difuso:

$$\lambda_x \in \mu \Leftrightarrow \lambda < \mu(x).$$

- Pu y Liu [18] definen la **casi-coincidencia** de dos conjuntos difusos $\nu, \mu \in \mathcal{F}(X)$ como la existencia de un $x \in X$ tal que $\nu(x) + \mu(x) > 1$. Aplicando esta definición al caso $\nu = \lambda_x$ resulta una relación

de pertenencia (equivalente a la no pertenencia, según el criterio de Kerre, de λ_x al pseudocomplementario de μ). Se tiene así el siguiente criterio:

$$\lambda_x \in \mu \Leftrightarrow \lambda + \mu(x) > 1.$$

Usando la definición de pertenencia de Kerre no se cumple, en general (si I es un conjunto de índices infinito), la propiedad:

$$\lambda_x \in \bigcup_{i \in I} \mu_i \Leftrightarrow \exists i \in I, \lambda_x \in \mu_i.$$

Cuando se usa la definición de Wong tenemos asegurado el cumplimiento de la propiedad anterior. Precisamente Wong motiva su definición en base al cumplimiento de esta propiedad. Pero no considera que, al adoptar su criterio de pertenencia, deja de verificarse la propiedad siguiente, que sí se cumple con la definición de Kerre:

$$\lambda_x \in \bigcap_{i \in I} \mu_i \Leftrightarrow \forall i \in I, \lambda_x \in \mu_i.$$

La definición de casi-coincidencia de Pu y Liu (que es menos usada que las otras dos), se comporta como la definición de Wong respecto al cumplimiento de las dos propiedades anteriores.

Por otro lado, la definición de Wong, al no admitir el valor $\lambda = 1$ en los puntos difusos, excluye entre estos a los puntos ordinarios o "crisp".

Ante esta situación, Gerla [5, 6] considera distintas propiedades con objeto de proporcionar definiciones axiomáticas de punto difuso y de su pertenencia a conjuntos difusos (simultáneamente). En particular, demuestra que es imposible definir una pareja (X', ϵ') , de manera que X' sea un conjunto de puntos difusos y ϵ' una relación (no difusa) de su pertenencia a conjuntos difusos tales que verifiquen simultáneamente las siguientes propiedades ($\forall P, Q \in X', \forall \mu, \mu_i, \nu, \nu_i \in \mathcal{F}(X)$):

- $P \epsilon' \bigcup_{i \in I} \mu_i \Leftrightarrow \exists i \in I, P \epsilon' \mu_i.$
- $P \epsilon' \bigcap_{i \in I} \mu_i \Leftrightarrow \forall i \in I, P \epsilon' \mu_i.$
- $\{P : P \epsilon' 0_X\} = \emptyset, \quad \{P : P \epsilon' \chi_X\} = X'.$
- $\mu = \nu \Leftrightarrow \{P : P \epsilon' \mu\} = \{P : P \epsilon' \nu\}.$
- $P = Q \Leftrightarrow \{\mu : P \epsilon' \mu\} = \{\mu : Q \epsilon' \mu\}.$

El concepto de "número difuso" ha experimentado, fundamentalmente, dos desarrollos distintos. Por un lado, tenemos distintas definiciones muy usadas en Inteligencia Artificial, que usan funciones de pertenencia que crecen de 0 a 1 para posteriormente decrecer de 1 a 0, y que podemos englobar

bajo la terminología común de “números difusos de Zadeh” (en base a la definición original introducida en [24]). Por otro lado, en la Topología Difusa se emplean los denominados “números difusos de Hutton” [12, 4] con función de pertenencia decreciente (o, de forma similar, tenemos los “números difusos de Höhle” [11], cuya función de pertenencia es una función de distribución de probabilidad). Además, los números de Hutton están sujetos a una relación de equivalencia, que puede obviarse eligiendo representantes semicontinuos superiormente [16]. Aquí usamos las definiciones y notaciones que concretamos a continuación:

Definición 1 Denominamos números [difusos] de Zadeh, de Hutton y de Höhle, respectivamente, a los elementos de los siguientes conjuntos:

$$\begin{aligned}\mathcal{F}_Z(\mathbf{R}) &:= \{\mu \in \mathcal{F}(\mathbf{R}) : \mu \text{ es convexo, normal} \\ &\quad \text{y s.c.s., y } \mu^{(0)} \text{ está acotado}\}, \\ \mathcal{F}_H(\mathbf{R}) &:= \{\mu \in \mathcal{F}(\mathbf{R}) : \mu \text{ es decreciente, normal} \\ &\quad \text{y s.c.s., y } \mu^{(0)} \text{ está acotado supte.}\}, \\ \mathcal{F}_D(\mathbf{R}) &:= \{\mu \in \mathcal{F}(\mathbf{R}) : \mu \text{ es creciente, normal} \\ &\quad \text{y s.c.s., y } \mu^{(0)} \text{ está acotado infte.}\}.\end{aligned}$$

(Donde “s.c.s.” significa “semicontinuo superiormente”)

Cuando hablemos simplemente de “números difusos” nos referimos a cualesquiera de los elementos del conjunto:

$$\mathcal{FN}(\mathbf{R}) := \mathcal{F}_Z(\mathbf{R}) \cup \mathcal{F}_H(\mathbf{R}) \cup \mathcal{F}_D(\mathbf{R}).$$

3 CONJUNTOS GRADUALES

Definición 2 ([8], Definiciones 3.1, 3.2)

Dado cualquier conjunto X , denominamos **subconjunto gradual** o **parte gradual** de X (o simplemente **conjunto gradual**) a cualquier aplicación $\psi : \mathbf{I} \rightarrow \mathcal{P}(X)$ que verifique la siguiente condición:

$$\forall \alpha, \beta \in \mathbf{I}, [\alpha < \beta \Rightarrow \psi(\alpha) \supseteq \psi(\beta)].$$

Decimos que ψ es **normal** cuando $\psi(\alpha) \neq \emptyset, \forall \alpha \in \mathbf{I}$. Notamos por $\mathcal{G}(X)$ al conjunto de partes graduales de X . La inclusión $\mathcal{P}(X) \subset \mathcal{G}(X)$ está determinada por: $A \mapsto (A)_{\mathbf{I}}$, siendo $(A)_{\mathbf{I}}(\alpha) := A, \forall \alpha \in \mathbf{I}$.

Siempre que se pueda, usamos el siguiente criterio general para extender a conjuntos graduales propiedades o conceptos “C” definidos para conjuntos crisp:

$$\psi \text{ es “C”} \Leftrightarrow \psi(\alpha) \text{ es “C”}, \forall \alpha \in \mathbf{I}.$$

En particular, definimos $(\forall \psi_i, \psi, \phi \in \mathcal{G}(X))$:

$$\bullet \psi = \bigcap_{i \in I} \psi_i \Leftrightarrow \psi(\alpha) = \bigcap_{i \in I} \psi_i(\alpha), \forall \alpha \in \mathbf{I}.$$

- $\psi = \bigcup_{i \in I} \psi_i \Leftrightarrow \psi(\alpha) = \bigcup_{i \in I} \psi_i(\alpha), \forall \alpha \in \mathbf{I}$.
- $\psi \subseteq \phi \Leftrightarrow \psi(\alpha) \subseteq \phi(\alpha), \forall \alpha \in \mathbf{I}$.
- Cualquier aplicación $f : X \rightarrow Y$ se extiende a $f : \mathcal{G}(X) \rightarrow \mathcal{G}(Y)$ definida por:

$$f(\psi)(\alpha) := f(\psi(\alpha)), \forall \alpha \in \mathbf{I}.$$

- Cualquier aplicación $f : X^2 \rightarrow Y$ se extiende a $f : \mathcal{G}(X)^2 \rightarrow \mathcal{G}(Y)$ definida por:

$$f(\psi, \phi)(\alpha) := f(\psi(\alpha), \phi(\alpha)), \forall \alpha \in \mathbf{I}.$$

- Para el caso $X = \mathbf{R}$, decimos que ψ es **convexo** (resp. **cerrado**, **acotado inferiormente** o **acotado superiormente**) cuando, $\forall \alpha \in \mathbf{I}, \psi(\alpha)$ es convexo (resp. cerrado, acotado inferior o superiormente).

Obviamente, ψ es normal sii $\psi(1) \neq \emptyset$ y ψ está acotado (inferior y/o superiormente) sii lo está $\psi(0)$.

El principal resultado respecto a la relación entre conjuntos graduales y conjuntos difusos es el siguiente (donde, en lo sucesivo sobreentendemos que $\mu^{[0]} = X$ y $\mu^{(1)} = \emptyset, \forall \mu \in \mathcal{F}(X)$):

Teorema 1 ([8], Teorema 4.3) Para cualquier conjunto X y cualquier aplicación $\psi : \mathbf{I} \rightarrow \mathcal{P}(X)$, se cumple:

1. ψ es un conjunto gradual si y sólo si existe un único conjunto difuso μ_ψ verificando: $\mu_\psi^{(\alpha)} \subseteq \psi(\alpha) \subseteq \mu_\psi^{[\alpha]}, \forall \alpha \in \mathbf{I}$.
2. El conjunto difuso μ_ψ está determinado por:

$$\mu_\psi(x) = \sup\{\alpha \in \mathbf{I} : x \in \psi(\alpha)\}, \forall x \in X,$$

$$\text{y sus } \alpha\text{-cortes son: } \mu_\psi^{[\alpha]} = \bigcap\{\psi(\beta) : 0 \leq \beta < \alpha\}, \quad \mu_\psi^{(\alpha)} = \bigcup\{\psi(\beta) : \alpha < \beta \leq 1\}, \forall \alpha \in \mathbf{I}.$$

En las condiciones del teorema anterior, decimos que μ_ψ es el **conjunto difuso asociado al conjunto gradual** ψ y que ψ es un **conjunto gradual asociado a** μ_ψ . En particular, la familia de α -cortes de cualquier conjunto difuso μ es un caso particular de conjunto gradual asociado a μ (ocurriendo lo mismo con la familia de α -cortes fuertes). Considerando este caso particular en el teorema anterior se obtienen como corolarios el Teorema de representación de Negoita y Ralescu (estableciendo las condiciones que caracterizan a los α -cortes) y el denominado Principio de resolución (obteniendo el conjunto difuso a partir de sus α -cortes).

En [8] se demuestra que las operaciones y relaciones antes definidas para conjuntos graduales se corresponden con las definidas por Zadeh para sus conjuntos difusos asociados en el siguiente sentido:

Proposición 1 Para conjuntos arbitrarios X, Y y cualesquiera $\psi_i, \psi, \phi \in \mathcal{G}(X)$, se cumple:

1. $\mu_{\bigcap_{i \in I} \psi_i} = \bigcap_{i \in I} \mu_{\psi_i}$.
2. $\mu_{\bigcup_{i \in I} \psi_i} = \bigcup_{i \in I} \mu_{\psi_i}$.
3. $\psi \subseteq \phi \Rightarrow \mu_\psi \subseteq \mu_\phi$.
4. Las extensiones $\mathcal{G}(X) \rightarrow \mathcal{G}(Y)$ y $\mathcal{F}(X) \rightarrow \mathcal{F}(Y)$ de cualquier aplicación $f: X \rightarrow Y$ verifican:

$$\mu_{f(\psi)} = f(\mu_\psi).$$

5. Las extensiones $\mathcal{G}(X)^2 \rightarrow \mathcal{G}(Y)$ y $\mathcal{F}(X)^2 \rightarrow \mathcal{F}(Y)$ de cualquier aplicación $f: X^2 \rightarrow Y$ verifican:

$$\mu_{f(\psi, \phi)} = f(\mu_\psi, \mu_\phi).$$

Corolario 1 Para cualquier conjunto X y cualesquiera $\mu, \nu \in \mathcal{F}(X)$, se tiene la inclusión $\mu \subseteq \nu$ si y solamente si existen dos conjuntos graduales, ψ asociado a μ y ϕ asociado a ν , tales que $\psi \subseteq \phi$.

Estos resultados son los que permiten extender la representación de conjuntos difusos por α -cortes mediante el uso adicional de otros conjuntos graduales asociados. Este "método extendido" se usa en [10] en el estudio de diversos temas sobre conjuntos, relaciones binarias, espacios topológicos y varias estructuras algebraicas. Aquí lo usaremos (en la sección 6, considerando el caso $X = Y = \mathbf{R}$) para traspasar propiedades de los números graduales a los números difusos.

4 PUNTOS GRADUALES

Definición 3 Denominamos punto gradual de soporte $x \in X$ y valor $\lambda \in (0, 1]$ al subconjunto gradual x_λ de X dado por:

$$x_\lambda(\alpha) := \{x\}, \quad \text{si } \alpha \leq \lambda, \\ := \emptyset, \quad \text{si } \alpha > \lambda.$$

Diremos que el punto gradual x_λ pertenece al conjunto gradual $\psi \in \mathcal{G}(X)$ cuando se tiene la inclusión $x_\lambda \subseteq \psi$ como conjuntos graduales. Es evidente que podemos escribir esta definición de la siguiente forma:

$$x_\lambda \in \psi \Leftrightarrow x \in \psi(\lambda). \quad (1)$$

Veamos que con esta definición se cumplen todas las condiciones consideradas por Gerla en [5, 6]:

Proposición 2 El conjunto de puntos graduales de X , $X_G := \{x_\lambda : x \in X, \lambda \in (0, 1]\}$ junto con la definición dada de pertenencia, verifican las cinco condiciones siguientes:

1. $x_\lambda \in \bigcup_{i \in I} \psi_i \Leftrightarrow \exists i \in I, x_\lambda \in \psi_i$.
2. $x_\lambda \in \bigcap_{i \in I} \psi_i \Leftrightarrow \forall i \in I, x_\lambda \in \psi_i$.
3. $\{x_\lambda : x_\lambda \in \emptyset\} = \emptyset, \quad \{x_\lambda : x_\lambda \in X\} = X_G$.
4. $\psi = \phi \Leftrightarrow \{x_\lambda : x_\lambda \in \psi\} = \{x_\lambda : x_\lambda \in \phi\}$.
5. $x_\lambda = y_\kappa \Leftrightarrow \{\psi : x_\lambda \in \psi\} = \{\psi : y_\kappa \in \psi\}$.

Demostración: Los apartados 1 y 2 resultan inmediatamente de (1) y las correspondientes propiedades clásicas. El apartado 3 es trivial, así como la parte " $\Rightarrow$ " de los dos restantes. Comprobemos la otra implicación en ambos:

4. Suponiendo que $\psi \neq \phi$, debe existir $\alpha \in \mathbf{I}$ tal que $\psi(\alpha) \neq \phi(\alpha)$. Entonces $\exists \alpha \in \mathbf{I}, \exists x \in X$ tal que $x \in \psi(\alpha) \setminus \phi(\alpha)$ (o bien $x \in \phi(\alpha) \setminus \psi(\alpha)$), con lo que $\exists \alpha_x \in X_G$ tal que $\alpha_x \in \psi$ y $\alpha_x \notin \phi$ (o viceversa), resultando que $\{x_\lambda : x_\lambda \in \psi\} \neq \{x_\lambda : x_\lambda \in \phi\}$.

5. Supongamos que $x_\lambda \neq y_\kappa$ y que $\lambda \leq \kappa$ (se razonaría de forma análoga en caso de ser $\lambda \geq \kappa$). Consideramos entonces el conjunto gradual $\psi = x_\lambda$, con lo que es obvio que $x_\lambda \in \psi$. Sin embargo $y_\kappa \notin \psi$, ya que debe ser $x \neq y$ o $\kappa > \lambda$ (o ambas cosas) por la hipótesis de partida. Por tanto, $\{\psi : x_\lambda \in \psi\} \neq \{\psi : y_\kappa \in \psi\}$. ■

5 NÚMEROS GRADUALES

Definición 4 Considerando las siguientes clases de intervalos:

$$C := \{[a, b] : a, b \in \mathbf{R}, a \leq b\}, \\ C_{-\infty} := \{(-\infty, a] : a \in \mathbf{R}\}, \\ C_{+\infty} := \{[a, +\infty) : a \in \mathbf{R}\},$$

definimos los conjuntos $\mathcal{G}_Z(\mathbf{R})$ (resp. $\mathcal{G}_H(\mathbf{R})$, $\mathcal{G}_D(\mathbf{R})$) de números graduales de Zadeh (resp. de Hutton y de Höhle) como sigue:

$$\mathcal{G}_Z(\mathbf{R}) := \{\psi \in \mathcal{G}(\mathbf{R}) : \forall \alpha \in \mathbf{I}, \psi(\alpha) \in C\}, \\ \mathcal{G}_H(\mathbf{R}) := \{\psi \in \mathcal{G}(\mathbf{R}) : \forall \alpha \in \mathbf{I}, \psi(\alpha) \in C_{-\infty}\}, \\ \mathcal{G}_D(\mathbf{R}) := \{\psi \in \mathcal{G}(\mathbf{R}) : \forall \alpha \in \mathbf{I}, \psi(\alpha) \in C_{+\infty}\}.$$

Y denominamos números graduales a los elementos del conjunto:

$$\mathcal{GN}(\mathbf{R}) := \mathcal{G}_Z(\mathbf{R}) \cup \mathcal{G}_H(\mathbf{R}) \cup \mathcal{G}_D(\mathbf{R}).$$

Usando la definición 2 para definir operaciones y un orden parcial entre los números graduales, se obtienen con facilidad diversos resultados para los mismos, que aparecen en [9, 10] (donde consideramos también otras definiciones alternativas que imponen menos condiciones, tanto para los números difusos como para los números graduales). Respecto a las operaciones unarias y binarias se establecen condiciones que aseguran que se tratan de operaciones internas, viendo entonces que los números (tanto graduales como difusos) de Zadeh admiten más operaciones que los de Hutton y los de Höhle. Esta consideración puede añadirse a otras dadas en [3] respecto a la comparación entre tales tipos de números difusos. Aquí vamos a centrarnos en el orden y la convergencia de sucesiones. Para ambos objetivos usamos el siguiente resultado (fácil de obtener a partir de las igualdades: $[a, b] = [a, +\infty) \cap (-\infty, b]$ y $-[a, +\infty) = (-\infty, -a]$).

Proposición 3 Las siguientes aplicaciones son biyectivas:

1. $\{(\psi_D, \psi_H) \in \mathcal{G}_D(\mathbf{R}) \times \mathcal{G}_H(\mathbf{R}) : \psi_D \cap \psi_H \text{ es normal}\} \rightarrow \mathcal{G}_Z(\mathbf{R}) : (\psi_D, \psi_H) \rightarrow \psi_D \cap \psi_H$.
2. $\mathcal{G}_H(\mathbf{R}) \rightarrow \mathcal{G}_D(\mathbf{R}) : \psi \rightarrow -\psi$ (extensión de $\mathbf{R} \rightarrow \mathbf{R} : x \rightarrow -x$).
3. $\{(\psi_1, \psi_2) \in \mathcal{G}_H(\mathbf{R})^2 : -\psi_1 \cap \psi_2 \text{ es normal}\} \rightarrow \mathcal{G}_Z(\mathbf{R}) : (\psi_1, \psi_2) \rightarrow -\psi_1 \cap \psi_2$.

En lo sucesivo usamos la notación dada en 3.1 para escribir cada número gradual de Zadeh ψ como la intersección de un número gradual de Hutton ψ_H con uno de Höhle ψ_D . Además, extendemos esta notación como sigue:

$$\begin{aligned} \forall \psi \in \mathcal{G}_H(\mathbf{R}), \quad \psi_H &:= \psi \quad \text{y} \quad \psi_D := (\mathbf{R})_I; \\ \forall \psi \in \mathcal{G}_D(\mathbf{R}), \quad \psi_D &:= \psi \quad \text{y} \quad \psi_H := (\mathbf{R})_I. \end{aligned} \quad (2)$$

Definición 5 En $\mathcal{GN}(\mathbf{R})$ usamos la notación anterior definiendo:

- El orden parcial: $\psi \leq \phi : \Leftrightarrow \psi_H \subseteq \phi_H \text{ y } \psi_D \supseteq \phi_D$.
- La convergencia gradual de sucesiones en $\mathcal{GN}(\mathbf{R})$, diciendo que $\{\psi_n\}_{n \in \mathbf{N}} \xrightarrow{g} \psi$ cuando se verifica:

$$\begin{aligned} \forall \alpha \in \mathbf{I}, \{ \inf \psi_{nD}(\alpha) \} &\rightarrow \inf \psi_D(\alpha) \\ \text{y } \{ \sup \psi_{nH}(\alpha) \} &\rightarrow \sup \psi_H(\alpha). \end{aligned}$$

donde $\rightarrow$ denota la convergencia usual en $\mathbf{R} \cup \{-\infty, +\infty\}$.

Con esta última definición se cumplen las condiciones consideradas por Kaleva y Seikkala:

Proposición 4 ([9], Proposición 3.13) En cada uno de los tres conjuntos de números graduales $\mathcal{G}_X(\mathbf{R})$, correspondientes a los casos $X = Z, H, D$, se verifica:

1. La convergencia gradual definida en $\mathcal{G}_X(\mathbf{R})$ extiende la convergencia usual de $\mathbf{R}$.
2. Si $\{\psi_n\} \subset \mathcal{G}_X(\mathbf{R})$ es una sucesión monótona y acotada entonces posee límite en $\mathcal{G}_X(\mathbf{R})$ según la convergencia gradual.

6 PROPIEDADES DERIVADAS PARA LOS NÚMEROS DIFUSOS

La analogía entre números graduales y números difusos se concreta de la siguiente forma:

Teorema 2 ([9], Teorema 4.3) Para cualquier $\mu \in \mathcal{F}(\mathbf{R})$ y cualquiera de los tres tipos de números correspondientes a los casos $X = Z, H, D$, se cumple:

$$\mu \in \mathcal{F}_X(\mathbf{R}) \Leftrightarrow \exists \psi \in \mathcal{G}_X(\mathbf{R}) \text{ asociado a } \mu.$$

Las operaciones en $\mathcal{F}_Z(\mathbf{R})$ se efectúan mediante el Principio de extensión de Zadeh. Lowen [16] prueba que podemos hacer lo mismo en $\mathcal{F}_H(\mathbf{R})$, para efectuar las operaciones previamente definidas por Rodabaugh [20] en la recta real difusa. En base al teorema anterior obtenemos la biyección:

$$\begin{aligned} \{(\mu_D, \mu_H) \in \mathcal{F}_D(\mathbf{R}) \times \mathcal{F}_H(\mathbf{R}) : \mu_D \cap \mu_H \text{ es normal}\} \\ \rightarrow \mathcal{F}_Z(\mathbf{R}) : (\mu_D, \mu_H) \rightarrow \mu_D \cap \mu_H, \end{aligned}$$

que da lugar a la notación $\mu = \mu_D \cap \mu_H$, extendida por:

$$\begin{aligned} \forall \mu \in \mathcal{F}_H(\mathbf{R}), \quad \mu_H &:= \mu \quad \text{y} \quad \mu_D := \chi_{\mathbf{R}}; \\ \forall \mu \in \mathcal{F}_D(\mathbf{R}), \quad \mu_D &:= \mu \quad \text{y} \quad \mu_H := \chi_{\mathbf{R}}. \end{aligned}$$

Entonces, los órdenes parciales considerados por Zadeh [24] y por Hutton [12] quedan englobados en el siguiente ($\forall \mu, \nu \in \mathcal{FN}(\mathbf{R})$):

$$\mu \leq \nu : \Leftrightarrow \mu_D \supseteq \nu_D \text{ y } \mu_H \subseteq \nu_H,$$

cumpléndose que:

Proposición 5 En cualquiera de los tres casos $X = Z, H, D$, se tiene ($\forall \mu, \nu \in \mathcal{F}_X(\mathbf{R})$):

$$\begin{aligned} \mu \leq \nu &\Leftrightarrow \exists \psi \in \mathcal{G}_X(\mathbf{R}) \text{ asociado a } \mu \text{ y } \exists \phi \\ &\in \mathcal{G}_X(\mathbf{R}) \text{ asociado a } \nu, \text{ tales que } \psi \leq \phi. \end{aligned}$$

Todo esto permite trasladar a los números difusos las distintas propiedades obtenidas para los números graduales en [9, 10] respecto a las operaciones unarias y

binarias y el orden parcial considerado. Por último, se obtiene un nuevo criterio de convergencia para sucesiones de números difusos que extiende la convergencia usual de $\mathbf{R}$ (lo que no ocurre con la convergencia puntual de funciones de pertenencia) y que asegura la existencia del límite de cualquier sucesión monótona y acotada (propiedad que no se verifica con la "convergencia por niveles"). Es decir, se superan los dos inconvenientes señalados por Kaleva y Seikkala [13] con el siguiente criterio:

Definición 6 En $\mathcal{FN}(\mathbf{R})$, definimos la **convergencia gradual**:

$$\{\mu_n\} \xrightarrow{g} \mu : \Leftrightarrow \forall n, \exists \psi_n \text{ asociado a } \mu_n$$

$$\text{y } \exists \psi \text{ asociado a } \mu \text{ tales que } \{\psi_n\} \xrightarrow{g} \psi.$$

Referencias

- [1] M. Delgado, J. Kacprzyk, S.A. Orlovski, J.L. Verdegay y M.A. Vila (1990), A survey of fuzzy optimization and fuzzy mathematical programming, in: M. Fedrizzi, J. Kacprzyk y M. Roubens (eds.), *Interactive Fuzzy Optimization and Mathematical Programming* (Springer Verlag, Berlin).
- [2] M. Delgado, J.L. Verdegay y M.A. Vila (1989), A General Model for Fuzzy Linear Programming, *Fuzzy Sets and Systems* **29**: 21-29.
- [3] M. Delgado, J.L. Verdegay y M.A. Vila (1994), Fuzzy Numbers, Definitions and Properties, *Mathware & Soft Computing* **1**: 31-43.
- [4] T.E. Gantner, R.C. Steinlage y R.H. Warren (1978), Compactness in fuzzy topological spaces, *J. Math. Anal. Appl.* **62**: 547-562.
- [5] G. Gerla (1986), On the concept of fuzzy point, *Fuzzy Sets and Systems* **18**: 159-172.
- [6] G. Gerla (1986), Generalized Fuzzy Points, *J. Math. Anal. Appl.* **120**: 761-768.
- [7] S. Gottwald (1981), Fuzzy points and local properties of fuzzy topological spaces, *Fuzzy Sets and Systems* **5**: 199-201.
- [8] J.A. Herencia (1996), Graded Sets and Points: A stratified approach to Fuzzy Sets and Points, *Fuzzy Sets and Systems* **77**: 191-202.
- [9] J.A. Herencia (1996), Graded numbers and graded convergence of fuzzy numbers, *Fuzzy Sets and Systems*, to appear.
- [10] J.A. Herencia (1996), *Origen y uso de los conjuntos graduales en la Matemática Difusa*, Tesis Doctoral (Universidad de Granada).
- [11] U. Höhle (1978), Probabilistische Topologien, *Manuscripta Math.* **26**: 223-245.
- [12] B. Hutton (1975), Normality in fuzzy topological spaces, *J. Math. Anal. Appl.* **50**: 74-79.
- [13] O. Kaleva y S. Seikkala (1984), On fuzzy metric spaces, *Fuzzy Sets and Systems* **12**: 215-229.
- [14] E.E. Kerre (1980), Fuzzy Sierpinski space and its generalizations, *J. Math. Anal. Appl.* **74**: 318-324.
- [15] E.E. Kerre (1989), A call for crispness in Fuzzy Set Theory, *Fuzzy Sets and Systems* **29**: 57-65.
- [16] R. Lowen (1983), On $(\mathbf{R}(L), \oplus)$, *Fuzzy Sets and Systems* **10**: 203-209.
- [17] H.T. Nguyen (1978), A Note on the Extension Principle for Fuzzy Sets, *J. Math. Anal. Appl.* **64**: 369-380.
- [18] Pu PaoMing y Liu YingMing (1980), Fuzzy topology. Part 1: Neighborhood structure of fuzzy point and Moore-Smidt convergence, *J. Math. Anal. Appl.* **76**: 571-599.
- [19] D. Ralescu (1979), A survey of the representation of fuzzy concepts and its applications, in: M.M. Gupta, R.K. Ragade y R.R. Yager, Eds., *Advances in Fuzzy Sets Theory and Applications* (North-Holland, Amsterdam) 77-91.
- [20] S.E. Rodabaugh (1982), Fuzzy addition in the L-fuzzy real line, *Fuzzy Sets and Systems* **8**: 39-52.
- [21] J.L. Verdegay (1980), *Problemas de Decisión en ambiente difuso*, Tesis Doctoral (Universidad de Granada).
- [22] J.L. Verdegay (1982), Fuzzy Mathematical Programming, in: M.M. Gupta y E. Sanchez (eds.), *Fuzzy Information and Decision Processes* (North-Holland, Amsterdam) 231-237.
- [23] C.K. Wong (1974), Fuzzy points and local properties of fuzzy topology, *J. Math. Anal. Appl.* **46**: 316-328.
- [24] L.A. Zadeh (1975), The concept of a linguistic variable and its applications to Approximate Reasoning, *Inform. Sci.* **8**: 199-249 (part I), 301-357 (part II), **9** (1976): 43-80 (part III).

UNA GENERALIZACION DE LA ENTROPIA A TRAVES DE LA TEORIA DE DEMPSTER-SHAFER *

J.A.Herencia

Dpto de Matemáticas

Facultad de Ciencias.

14071-Córdoba

E-mail: malhegoj@uco.es

M.T.Lamata

Dpto de Ciencias de la Computación

e Inteligencia Artificial

18071-Granada

E-mail: mlamata@goliat.ugr.es

Abstract

El ánimo de este trabajo es el de proporcionar un marco que englobe las medidas de entropía ya conocidas así como todas aquellas que resulten de particularizar el operador que proponemos a las diferentes medidas difusas **Key-word:** incertidumbre, entropía, medidas difusas, asignación básica de probabilidad, creencia, plausibilidad.

1 Introducción

El concepto del término incertidumbre ha ido cambiando con el tiempo, C.Shannon [15] consideró la teoría de información (incertidumbre) como una rama de la probabilidad, posteriormente con la aparición y estudio de las medidas difusas [10,19], el término incertidumbre se disoció en dos conceptos, vaguedad y ambigüedad [6], nosotros en este trabajo nos restringiremos al estudio de la ambigüedad.

Cuando trabajamos con un tipo particular de medidas difusas como son las evidencias, varias causas de incertidumbre han de ser tenidas en cuenta; la primera, está asociada al tamaño de los subconjuntos que forman los elementos focales y que se denomina **imprecisión** o no-especificidad; también hay que tener en cuenta el **conflicto** o disonancia que se produce cuando los elementos focales son disjuntos ya que la evidencia que se asigna a un subconjunto entra en conflicto con la asignada a otro subconjunto supuesto que estos elementos tienen intersección sin elementos comunes; y por último, el tercer tipo de incertidumbre nos lleva a la fuente de **confusión** y lo relacionamos con el número de conjuntos focales sobre los que tengamos alguna evidencia.

Dentro del campo de la Teoría de Dempster-Shafer se han estudiado varias medidas de incertidum-

bre y de entropía que generalizan la clásica medida de entropía de Shannon.

Nosotros pretendemos crear un marco general para estudiar la entropía asociada a una medida difusa y en el que las ya estudiadas medidas de incertidumbre y entropía relativas a la teoría de la Evidencia se obtengan como casos particulares.

El presente trabajo se compone de los siguientes apartados; Empezamos recordando las definiciones básicas que vamos a usar a lo largo del trabajo. En el apartado tercero proponemos dos funciones h_I y h_A relacionadas con los conceptos de imprecisión y aleatoriedad, respectivamente y que formarán parte del operador entropía total que proponemos.

Son varias las medidas de entropía conocidas y que resultan como casos particulares del operador entropía que proponemos y que se demuestra en el apartado cuarto.

Hemos de hacer constar que por las características del trabajo hemos dejado a un lado el estudio de las propiedades más importantes, que van a depender de la medida que estemos utilizando.

2 Preliminares

En todo lo que sigue supondremos fijado un universo de discurso finito $X = \{x_1, x_2, \dots, x_n\}$. Notaremos por $P(X)$ el conjunto de partes (o subconjuntos) de X , por $|C|$ el cardinal de cualquier conjunto C .

Definición

Se llama **medida difusa** sobre $P(X)$, siendo X finito a toda aplicación $g : P(X) \rightarrow [0, 1]$ que verifique las tres condiciones siguientes:

- $g(\emptyset) = 0$.
- $A \subset B \Rightarrow g(A) \leq g(B)$.

*Este trabajo ha sido financiado por la DGICYT bajo el proyecto PB92-0959

- $g(X) = 1$.

Al conjunto de todas las medidas difusas lo notaremos por $\mathcal{G}$.

Definición

Una **asignación básica de probabilidad** (ABP) sobre X es una aplicación

$$m : P(X) \rightarrow [0, 1]$$

que verifica:

- $m(\emptyset) = 0$.
- $\sum_{A \subseteq X} m(A) = 1$.

Notaremos por $\mathcal{M}$ al conjunto de las asignaciones básicas de probabilidad sobre X .

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:

$$\begin{aligned} Bel(A) &:= \sum_{B \subseteq A} m(B) \\ Pl(A) &:= \sum_{B \cap A \neq \emptyset} m(B), \quad \forall A \subseteq X \end{aligned}$$

Definición

Yager [17] introdujo el concepto de **bag de cardinal k** sobre cierto conjunto C , como cualquier combinación de k elementos de C que pueden estar o no repetidos y sin que importe el orden en el que aparecen.

El bag formado por los elementos $c_1, c_2, \dots, c_k$ es notado por $\langle c_1, c_2, \dots, c_k \rangle$. El bag vacío es el que no contiene ningún elemento $\emptyset = \langle \rangle$.

Notaremos por $\mathcal{U}^C$ al conjunto de todos los bags contruidos sobre C y por $\mathcal{U}_k^C$ a los de cardinal k .

Definición

Dados dos conjuntos cualesquiera C y D , se llama **bag mapping** de C en D a toda aplicación

$$f : \mathcal{U}^C \longrightarrow D$$

que sea simétrica en todos sus argumentos.

Definición

Yager [18] introdujo el concepto de **operador TOT**, definido como cualquier bag mapping

$$T : \mathcal{U}^{[0,1]} \longrightarrow R$$

que cumpla las siguientes condiciones:

- $A \geq B \implies T(A) \geq T(B) \quad \forall A, B \in \mathcal{U}^{[0,1]}$
- $T(A \oplus B) \geq T(A) \quad \forall A, B \in \mathcal{U}^{[0,1]}$
- $T(A \oplus \langle 0 \rangle) = T(A) \quad \forall A \in \mathcal{U}^{[0,1]}$
- $T(\emptyset) = 0$

Como ejemplos de operadores TOT tenemos los siguientes:

$$Sum \langle a_1, a_2, \dots, a_k \rangle = a_1 + a_2 + \dots + a_k$$

$Sum_p \langle a_1, a_2, \dots, a_k \rangle = a_1^p + a_2^p + \dots + a_k^p$ para cualquier $p \geq 0$.

$L_p \langle a_1, a_2, \dots, a_k \rangle = (Sum_p \langle a_1, a_2, \dots, a_k \rangle)^{1/p}$ para cualquier $p \geq 0$.

Evidentemente $L_1 = Sum_1 = Sum$.

Observemos que tanto la definición como los ejemplos anteriores tienen perfecto sentido cuando extendemos $[0, 1]$ a $[0, \infty)$.

También es evidente que un operador TOT puede restringirse al conjunto de bags de cardinal fijo.

Teniendo en cuenta esto, en el presente trabajo usaremos operadores TOT sobre $\mathcal{U}_2^{[0, \infty)}$.

3 Medidas de Entropía

La Teoría de la Evidencia o Teoría de Dempster-Shafer [14] es una de las muchas teorías capaz de explicar la incertidumbre asociada a una variable aleatoria como se ha puesto de manifiesto a través de los muchos trabajos de investigación que han ido apareciendo en las últimas décadas.[7,8,19].

Hace tiempo que nos planteamos [9] dar una medida total de incertidumbre que recogiera las distintas causas de incertidumbre de estas medidas; ahora tratamos de unificar mediante la siguiente expresión

$$h_T(B, f) = t < h_f(|B|), h_A(B, f) >$$

todas estas medidas.

Consideraremos que este operador t es tal que $t < h, 0 \geq h$ para cualquier valor de h .

Con ello vamos a ser capaces de obtener las medidas de imprecisión y aleatoriedad, así como la incertidumbre suma de estas dos.

3.1 Medidas de Imprecisión

Supongamos una ABP m con elementos focales B . Definimos una función

$$h_I : N \longrightarrow R$$

que sea creciente en $[1, \infty)$ de forma que para cualquier elemento focal $B \subset X$, $h_I(|B|)$ nos va a medir la imprecisión. Como ejemplo de este tipo de función consideraremos la función logarítmica.

- Siendo $h_I(k) = \log_2 k$, se obtiene la **medida de información de Hartey** para la no-especificidad de un conjunto:

$$h_I(|B|) = \log_2 |B|$$

- También con $h_I(k) = \log_2 k$, resulta la medida de **imprecisión de Dubois-Prade** [1] para una ABP:

$$H_I(m) = I(m) = \sum_{B \in Fm} m(B) \log_2 |B|$$

que se puede considerar como el valor medio de la información de Hartley cuando la información disponible sobre el verdadero estado es de naturaleza evidencial m .

Puesto que la imprecisión está relacionada con el cardinal de los conjuntos focales, esto implicará que cuanto mayor sea $|B|$ mayor deberá ser esta medida de imprecisión.

Por ello, proponemos la siguiente

Proposición

1. $\min I(m) = 0$ cuando la evidencia es probabilística, es decir, $m_i = (\{p(x_i)\})$.
2. $\max I(m) = \log |X|$ que se alcanza en el supuesto de ignorancia, es decir, cuando $m(X) = 1$.

$I(m)$ generaliza la **U-uncertainty de Higashi y Klir** [4], correspondiente al caso en que la evidencia aportada por m sea una posibilidad.

- Tomando $h_I(k) = 1 - 1/k$, obtenemos la **medida de no-especificidad de Yager** [16], para una ABP:

$$H_I(m) = J(m) = 1 - \sum_{B \in Fm} m(B)/|B| = 1 - S(m)$$

donde $S(m)$ nos mide la precisión de la información, por tanto, $J(m)$ será una medida de la imprecisión.

$S(m)$ será pequeño si m se concentra en subconjuntos $B \subset X$ con gran número de elementos, en este caso $J(m)$ será grande.

3.2 Medida de aleatoriedad

Sea

$$h_A : \mathcal{P}(X) \times (\mathcal{M} \cup \mathcal{G}) \longrightarrow R$$

$$(B, f) \longrightarrow \begin{cases} -\log f(B) & \text{si } f(B) > 0 \\ 0 & \text{si } f(B) = 0 \end{cases}$$

donde $h_A(B, f)$ mide la sorpresa o información según f aportada por la aparición de B , B es un elemento focal sobre el que tenemos alguna evidencia y $f \in (\mathcal{M} \cup \mathcal{G})$ será una medida difusa o bien una ABP.

Si para la función $h_A = -\log f(B)$ consideramos el caso de las evidencias los valores posibles de f van a ser: $(f \in \{m, bel, Pl, \})$ y las respectivas medidas de entropía ya estudiadas son las que a continuación se exponen.

- Tomando $f = m$, se tiene la **entropía** de la variable aleatoria sobre el conjunto de valores $B_i \subset X$ dotada de una asignación básica de probabilidad m .

$$H_A(m) = G(m) = -\sum_{B \in Fm} m(B) \log m(B)$$

Esta medida ha sido considerada desde distintos puntos de vista por Nguyen [11] y por Pal, Bezdek y Hemasinha [13].

- Siendo $f = Pl$, resulta la **medida de disonancia de Yager** [16]:

$$H_A(m) = E(m) = -\sum_{B \in Fm} m(B) \log Pl(B)$$

Esta medida de entropía nos mide el conflicto existente entre los subconjuntos $B_i \subset X$ para los que $m(B) > 0$.

Para que esta medida se anule, es conocido que los elementos focales han de tener intersección distinta de vacío, lo que supone que no existe contradicción en m .

Mientras que el $\max E(m) = \log |X|$ se alcanza para las distribuciones de probabilidad uniformemente distribuidas sobre $|X|$

- La **medida de confusión de Höhle** [5], corresponde a la elección $f = Bel$:

$$H_A(m) = C(m) = -\sum_{B \in Fm} m(B) \log Bel(B)$$

El valor mínimo de esta medida se consigue cuando $Bel(B) = 1$, es decir, si únicamente tenemos un elemento focal.

- Consideremos ahora una evidencia probabilística:

$$X = (\{x_1\}, \{x_2\}, \dots, \{x_n\}); \quad m(\{x_i\}) = p_i$$

Estas medidas de entropía nos miden el conflicto existente entre los distintos subconjuntos de X sobre los que tenemos alguna evidencia.

Si cualquiera de las cuatro situaciones anteriores donde $(f \in \{m, bel, Pl, \})$ las particularizamos al caso probabilístico $f(\{x_i\}) = p_i$, nos resulta la **entropía de Shannon** de una distribución de probabilidad [15]:

$$H_A(m) = -\sum_i p_i \log p_i$$

3.3 Entropía total

Elijamos un operador TOT

$$t : \mathcal{U}_2^{[0,\infty)} \longrightarrow R$$

que cumpla la condición de que $t < h, 0 \geq h \quad \forall h \geq 0$. Este operador engloba las dos medidas antes consideradas, sin alterar el valor de una de ellas en caso de ser la única que se presente.

De esta manera, a cualquier elemento focal B se le asocia el valor:

$$h_T(B, f) = t < h_I(|B|), h_A(B, f) >$$

que mide la incertidumbre total originada por B .

Si consideramos que la entropía total de una ABP m viene dada por la esperanza de las incertidumbres originadas por los distintos elementos focales, su expresión quedará como:

$$H_T(m) = \sum_{B \in F} m(B) h_T(B, f)$$

donde la función $f \in \mathcal{M} \cup \mathcal{G}$ será cualquier medida relacionada con la evidencia, por ello $f \in \{m, Bel, Pl, Pl_p\}$ y consideraremos que H_T depende únicamente de m .

Si únicamente consideramos un tipo de incertidumbre, la entropía correspondiente a la evidencia m , resultaría

$$H_I(m) = \sum_{B \in F} m(B) h_I(|B|)$$

cuando solo consideramos la imprecisión. Y

$$H_A(m) = -\sum_{B \in F} m(B) \log f(B)$$

si solo consideramos la aleatoriedad.

Si como operador TOT consideramos otro distinto del operador suma o las funciones (h_I, h_A) las extendemos a cualquier medida difusa [14] obtendremos diversas medidas de incertidumbre asociadas a las medidas difusas no solo a las medidas de evidencia.

Por la propia definición de entropía como valor medio de las incertidumbres originadas por los distintos elementos focales, y considerando que la incertidumbre asociada a una evidencia es la suma de una medida de imprecisión y de otra de contradicción no tendremos nada más que hacer todas las combinaciones posibles para obtener las posibles medidas de entropía.

De esta forma somos capaces de obtener las siguientes medidas de entropía que ya han sido consideradas en la literatura.

- Si $h_I(k) = \log_2 k$, $f = m$ y $t = Sum$, entonces se obtiene la **entropía global superior de Lamata, Moral** [9], que posteriormente

Pal, Bezdek y Hemasinha [13], denominaron **average total uncertainty**

$$H_T(m) = H^*(m) = I(m) + G(m)$$

- Con $h_I(k) = \log_2 k$, $f = Pl$ y $t = Sum$, resulta la **entropía global inferior de Lamata, Moral** [9]:

$$H_T(m) = H_*(m) = I(m) + E(m)$$

Estas dos últimas medidas son tales que

$H_*(m) = H^*(m) = 0 \iff \exists a \in X$ tal que $m(\{a\}) = 1$ es decir, la entropía superior e inferior se anula cuando tenemos certeza sobre el elemento cuestionado.

Además, es fácil comprobar [9] que estas dos medidas son tales que acotan superior e inferiormente a la entropía de Shannon:

$$H_*(m) \leq H(p) \leq H^*(m)$$

- Tomando $h_I(k) = \log_2 k$, $f = Pl_p$ y $t = Sum$, obtenemos la **incertidumbre total de Klir y Ramer** [7]:

$$H_T(m) = T(m) = I(m) + D(m)$$

4 Conclusiones

Si tenemos en cuenta que el medir la incertidumbre o la cantidad de información que una medida es capaz de proporcionar es algo necesario en muchas aplicaciones, nosotros proporcionamos un operador que engloba las dos fuentes de incertidumbre, imprecisión and "innate contradiction", que están presentes en las medidas difusas.

Hemos demostrado como con la función

$$h_T(B, f) = t < h_I(|B|), h_A(B, f) >$$

somos capaces de obtener varias de las medidas de incertidumbre estudiadas hasta ahora por diversos investigadores.

Como consecuencia podemos afirmar que el mayor aporte de este trabajo es el de proponer para el caso de medidas de Dempster-Shafer una expresión general para la entropía que sería interesante estudiar en un futuro con objeto de encontrar nuevas medidas de entropía.

Además, queda demostrado que medidas de entropía ya conocidas quedan como casos particulares de la expresión que aquí proponemos.

References

- [1] D. Dubois and H. Prade, A note on measures of specificity for fuzzy sets, *BUSEFAL* 19 (1984) 83-89.

- [2] D. Dubois and H. Prade, *Fuzzy Sets and Systems, Theory and Applications*. Academic Press, New York, (1980)
- [3] D. Harmanec and G.J. Klir, Measuring total uncertainty in Dempster-Shafer theory: A novel approach. *Int. J. Gen. Syst.* 22 (1994) 405-419.
- [4] M. Higashi and G.J. Klir, Measures of uncertainty and information based on possibility distributions, *Int. J. Gen. Syst.* 9 (1983) 43-58.
- [5] U. Hohle, Entropy with respect to plausibility measures, *Proceedings of the 12th IEEE International Symposium on Multiple Valued Logic*, Paris, (1982) 167-169.
- [6] G.J. Klir, Where do we stand on measures of uncertainty, ambiguity, fuzziness, and the like?, *Fuzzy Sets and Systems* 24 (1987) 141-160.
- [7] G.J. Klir and A. Ramer, Uncertainty in Dempster-Shafer theory: A critical re-examination, *Int. J. Gen. Syst.* 18 (1990) 155-166.
- [8] M. Lamata and S. Moral, Measures of entropy in the theory of evidence, *Int. J. Gen. Syst.* 14 (1987) 297-305.
- [9] M. Lamata and S. Moral, Medidas de entropía en la teoría de la Evidencia. *Trabajos de Estadística.* 3 (1988) 15-31.
- [10] M. Lamata and S. Moral, Classification of fuzzy measures, *Fuzzy Sets and Systems* 33 (1989) 243-253.
- [11] H.T. Nguyen, On entropy of random sets and possibility distributions, in: J.C. Bezdek, Ed., *The analysis of Fuzzy Information* (CRC Press, Boca Raton, Florida, 1985).
- [12] N.R. Pal, J.C. Bezdek and R. Hemasinha, Uncertainty Measures for Evidential Reasoning I: A Review, *Int. J. Approxi. Reason* 7 (1992) 165-183.
- [13] N.R. Pal, J.C. Bezdek and R. Hemasinha, Uncertainty Measures for Evidential Reasoning II: A New Measure of Total Uncertainty, *Int. J. Approxi. Reason* 8 (1993) 1-16.
- [14] G. Shafer, *"A mathematical theory of Evidence"*. Princenton University Press, Princenton, New Jersey, 1976.
- [15] C.E. Shannon, The mathematical theory of communication, *The Bell System Technical J.* 27 (1948) 379-423, 623-656.
- [16] R.R. Yager, Entropy and specificity in a Mathematical Theory of Evidence, *Internat. J. Gen. Systems* 9 (1983) 249-260.
- [17] R.R. Yager, "On the theory of bags". *Int. J. Gen. Syst* 13, (1986), 23-37.
- [18] R.R. Yager, "A Unified Approach to Aggregation Based upon MOM and MAM Operators", Tech Report MII-1316 (1993).
- [19] L.A. Zadeh, Fuzzy sets, *Information and Control* 8 (1965) 338-353.

Distancias e Índices Parciales de Medidas Difusas

Luis Daniel Hernández Molinero
Dpto. Informática y Sistemas
Universidad de Murcia
e-mail: ldaniel@dif.um.es

Antonio Salmerón Cerdán
Dpto. Estadística y Matemática Aplicada
Universidad de Almería
e-mail: asc@stat.ualm.es

Resumen

Un elemento fundamental en la teoría de las medidas difusas es el estudio de índices que determinen la calidad de ésta (i.e. medidas de disonancia, confusión o discordancia). Dichos estudios se centran principalmente en al teoría de la posibilidad y de la evidencia, y todos ellos consideran toda la información que codifica la medida. En este trabajo proponemos dos técnicas para determinar la probabilidad más cercana a una medida difusa general así como el establecimiento de un conjunto de índices de (in)certidumbre asociados a estas medidas. En su desarrollo establecemos el concepto de distancia entre medidas difusas considerando sólo una parte de la información que éstas suministran.

1 Introducción

La pérdida de aditividad, al ampliar el campo de las medidas de probabilidad a las medidas difusas en general, dota de una especial relevancia al análisis de la medida de subconjuntos no unitarios del referencial [11, 7, 10]. En efecto, una medida de probabilidad queda caracterizada completamente por la medida de los conjuntos de cardinal uno, lo que no siempre ocurre con las medidas no aditivas, en las que, para su determinación, generalmente es necesario conocer la medida de todos los subconjuntos. Precisamente el estudio de la relación entre la medida de subconjuntos de distinto cardinal se convierte en elemento clave para la comprensión de las propiedades de una medida difusa.

Un elemento fundamental en la teoría de las medidas difusas es la búsqueda de índices que caractericen determinadas propiedades de la(s) medida(s) en estudio y nos permitan comparar su grado de cumplimiento; entre tales índices destacan los de entropía, especificidad, no aditividad o similitud entre medidas [12, 5, 6, 8, 9]. Algunas de estas propiedades pueden es-

tudiarse mediante extensiones directas o conceptuales de los conocidos para medidas de probabilidad; sin embargo otras características de las medidas difusas requieren enfoques nuevos, entre los que cabe destacar el propuesto por De Campos [2, 3] que se basa en el establecimiento de una distancia entre dos medidas difusas, y del que se derivan conceptos como el de probabilidad más próxima, índice de no aditividad, índice de certidumbre e incertidumbre, etc...

Todos los conceptos e índices propuestos hasta el momento atienden a la globalidad de la medida difusa (ver p.e. [1]). Nos proponemos aquí la definición de índices que consideren de forma diferenciada la medida de los subconjuntos de distinto cardinal, en un intento de captar mejor las características de una medida difusa sobre un referencial finito. En este sentido, y utilizando el enfoque de las distancias antes mencionado definiremos la probabilidad más próxima a la familia de subconjuntos de un cardinal determinado, así como índices de certidumbre e incertidumbre de la medida difusa en esas familias.

2 Definiciones Generales

Consideraremos siempre un conjunto finito $X = \{x_1, \dots, x_n\}$ y el álgebra $\mathcal{A} = \mathcal{P}(X)$.

Definición 1 Sea $g : \mathcal{A} \rightarrow [0, 1]$. Se dice que g es una medida difusa sobre $(X, \mathcal{A})$ si

1. $g(\emptyset) = 0, g(X) = 1$
2. Si $A, B \in \mathcal{A}$ y $A \subseteq B$, entonces $g(A) \leq g(B)$

Definición 2 Sea g una medida difusa. Se define la medida dual de g , y la notaremos por g^* , a una función $g^* : \mathcal{A} \rightarrow [0, 1]$ definida por: $g^*(A) = 1 - g(\bar{A}) \quad \forall A \subseteq X$

Una clase especialmente importante de medidas difusas son las generadas por una evidencia. El concepto fundamental es el de asignación básica de probabilidad (a.b.p.).

Definición 3 Una aplicación $m : \mathcal{P}(X) \rightarrow [0, 1]$ es una a.b.p. sii $m(\emptyset) = 0$ y $\sum_{A \subseteq X} m(A) = 1$

Sin embargo las a.b.p. no son medidas difusas ya que no son funciones de conjunto monótonas sino tan sólo funciones que asignan "masas de creencia" sobre determinados conjuntos. Sin embargo, dada una a.b.p. podemos construir dos medidas difusas duales.

Definición 4 La medida de creencia asociada a una a.b.p. m , es una aplicación $Bel : \mathcal{P}(X) \rightarrow [0, 1]$ definida por: $Bel(A) = \sum_{B \subseteq A} m(B)$

Su medida dual recibe el nombre de medida de plausibilidad y viene dada por la expresión: $Pl(A) = \sum_{B \cap A \neq \emptyset} m(B) = 1 - Bel(\bar{A})$

Definición 5 Llamaremos probabilidades asociadas a una medida difusa g , a las $n!$ probabilidades P_σ dadas por:

$$P_\sigma(x_{\sigma(1)}) = g(\{x_{\sigma(1)}\})$$

...

$$P_\sigma(x_{\sigma(i)}) = g(\{x_{\sigma(1)}, \dots, x_{\sigma(i)}\}) - g(\{x_{\sigma(1)}, \dots, x_{\sigma(i-1)}\})$$

...

$$P_\sigma(x_{\sigma(n)}) = 1 - g(\{x_{\sigma(1)}, \dots, x_{\sigma(n-1)}\})$$

Para cada $\sigma = (\sigma(1), \dots, \sigma(n)) \in S_n$ con S_n el conjunto de las permutaciones de n elementos.

Es claro que dada una medida difusa g sobre un referencial X , ésta define un único conjunto de medidas de probabilidad. Sin embargo, el conocimiento de las probabilidades asociadas a g , no siempre permite reconstruir la medida difusa g ya que distintas medidas pueden generar el mismo conjunto de probabilidades asociadas. Sin embargo, si conocemos exactamente de qué permutación procede cada probabilidad asociada entonces sí es posible reconstruir la medida difusa [2, 3]. Así, una medida difusa puede identificarse con $n!$ puntos en el espacio n -dimensional:

$$g = \{(P_\sigma(x_1), \dots, P_\sigma(x_n)); \sigma \in S_n\}$$

Al existir una relación biunívoca entre los puntos correspondientes a dos medidas difusas, dada en términos de la permutación correspondiente, De Campos propone definir la distancia entre dos medidas difusas mediante el establecimiento de métricas que vengan determinadas por las distancias que se definan entre las probabilidades asociadas de las dos medidas. Para definir una distancia entre probabilidades, se consideran que éstas son puntos de $\mathbb{R}^n$ y por tanto se puede emplear cualquier distancia usual de este espacio. Más concretamente:

Definición 6 Se define la distancia entre dos medidas difusas g y g' sobre X , con probabilidades asociadas P_σ y P'_σ ($\sigma \in \mathcal{R}^n$) respectivamente, como:

$$S_q(g, g') = \sqrt[q]{\frac{1}{2 \times n!} \sum_{\sigma \in S_n} \sum_{i=1}^n |p_{\sigma_i} - p'_{\sigma_i}|^q} \quad (1)$$

$$S_0(g, g') = \frac{1}{2 \times n!} \sum_{\sigma \in S_n} \sum_{i=1}^n |p_{\sigma_i} - p'_{\sigma_i}| \quad (2)$$

$$S_m(g, g') = \max_{\sigma \in S_n} \max_{1 \leq i \leq n} |p_{\sigma_i} - p'_{\sigma_i}| \quad (3)$$

donde $p_{\sigma_i} = P_\sigma(x_i)$ y $p'_{\sigma_i} = P'_\sigma(x_i) \forall \sigma \in S_n$

Obsérvese cómo las expresiones (1), (2) y (3) responden a las distancias de Minkowski, valor absoluto y del máximo respectivamente salvo constantes; constantes que se introducen con objeto de que $S(g, g') \leq 1$.

Nótese también que el cálculo de las distancias está basado en las diferencias $|p_i - p'_i|$. Puesto que trabajamos con $\mathbb{R}^n$ y estamos considerando todas las posibles permutaciones de S_n , en general serán necesarias calcular $n \times n!$ diferencias del tipo $|p_i - p'_i|$, cálculo excesivo con n no necesariamente muy grande (p.e. si $n = 10$, $n \times n! = 36.288.000$). Además, para el cálculo de distancias, se utiliza un tipo muy particular de medidas difusas, las probabilidades, independientemente del tipo o "status" que tengan las medidas que se consideran (que pueden ser probabilidades, evidencias, medidas representables, capacidades, etc).

Nuestro método alternativo para calcular la distancia entre dos medidas difusas usa tan sólo los $2^n - 2$ valores que definen a cualquier medida sobre $\mathcal{P}(X)$, sin necesidad de recurrir a ningún otro tipo de "medida difusa auxiliar", lo que permite obviar los dos inconvenientes citados.

3 Distancias Basadas en Subconjuntos

Para nuestro desarrollo usaremos la siguiente notación. Dado un referencial finito $X = \{x_1, x_2, \dots, x_n\}$, el conjunto de todos los subconjuntos de X con cardinal m será denotado por A_m . Ya que existen $k = \binom{n}{m}$

subconjuntos de cardinal m , A_m viene dado por la familia $\{A_m^p\}_{p=1}^k$ donde A_m^p denota al p -ésimo subconjunto de X con cardinal m . En lo que sigue, supondremos que, si $A_m^p = \{x_{p_1}, x_{p_2}, \dots, x_{p_m}\}$ entonces $p_i < p_j$ si $i < j$, lo que nos permite construir el siguiente orden entre los subconjuntos del referencial X .

Dado un conjunto A_m^p consideremos el vector que definen los subíndices de los elementos que lo componen, esto es, si $A_m^p = \{x_{p_1}, x_{p_2}, \dots, x_{p_m}\}$ denotaremos por a_m^p al vector $(p_1, p_2, \dots, p_m) \in \mathbb{R}^m$. Con esta notación definimos las siguientes relaciones:

Definición 7

Dados dos vectores $a_m^p = (p_1, p_2, \dots, p_m) \in \mathbb{R}^m$ y $a_l^{p'} = (p'_1, p'_2, \dots, p'_l) \in \mathbb{R}^l$ con $p_i < p_j$ y $p'_i < p'_j$ para todo $i < j$; definimos la siguiente relación:

$$a_m^p <^* a_l^{p'} \iff \begin{cases} m < l \\ \text{si } m = l \implies a_m^p < a_m^{p'} \\ \text{Según el orden lexicográfico} \end{cases}$$

Definición 8 Diremos que $A_m^p \prec A_m^{p'} \iff a_m^p \prec^* a_m^{p'}$

Ejemplo 1 Sea $X = \{x_1, x_2, x_3\}$, entonces:

$$\begin{aligned} A_0^1 &= \emptyset & A_1^1 &= \{x_1\} & A_2^1 &= \{x_1, x_2\} & A_3^1 &= \{x_1, x_2, x_3\} \\ A_1^2 &= \{x_2\} & A_2^2 &= \{x_1, x_3\} & & & & \\ A_1^3 &= \{x_3\} & A_2^3 &= \{x_2, x_3\} & & & & \end{aligned}$$

y según la relación anterior, se tiene que:

$$A_0^1 \prec A_1^1 \prec A_1^2 \prec A_1^3 \prec A_2^1 \prec A_2^2 \prec A_2^3 \prec A_3^1$$

3.1 Distancias basadas en la medida de todos los subconjuntos

Una vez establecido un orden en los elementos de $\mathcal{P}(X)$ y teniendo en cuenta que siempre $g(\emptyset) = 0$ y $g(X) = 1$, una medida difusa podemos identificarla como un punto de $\mathbb{R}^{2^n-2}$. Así, si dos medidas difusas vienen determinadas por:

$$\begin{aligned} g &\equiv (g(A_1^1), \dots, g(A_i^1), \dots, g(A_{k_i}^1), \dots, g(A_{n-1}^{k_i-1})) \\ g' &\equiv (g'(A_1^1), \dots, g'(A_i^1), \dots, g'(A_{k_i}^1), \dots, g'(A_{n-1}^{k_i-1})) \end{aligned}$$

con $k_i = \binom{n}{i}$ y $\sum_{i=1}^{n-1} k_i = 2^n - 2 = s$; el criterio más sencillo para establecer una distancia entre g y g' sería considerar simplemente la distancia entre los dos puntos que definen en $\mathbb{R}^s$, es decir, usar la comparación de los valores de g y g' en cada subconjunto A_m^p de X .

Definición 9 Definimos la distancia entre dos medidas difusas g y g' como:

$$\bar{d}_q(g, g') = \sqrt[q]{\frac{1}{s} \sum_{A \in \mathcal{P}(X)} |g(A) - g'(A)|^q} \quad [q \geq 2] \quad (4)$$

$$\bar{d}_1(g, g') = \frac{1}{s} \sum_{A \in \mathcal{P}(X)} |g(A) - g'(A)| \quad (5)$$

$$\bar{d}_m(g, g') = \max_{A \in \mathcal{P}(X)} |g(A) - g'(A)| \quad (6)$$

Obsérvese que dichas definiciones no son más que la distancias de Minkowski, del valor absoluto y del máximo, respectivamente, en el espacio s -dimensional. Las tres distancias oscilan entre 0 y 1. Obsérvese también que tan sólo se realizan $2^n - 2$ diferencias del tipo $|g - g'|$ frente a las $n \times n!$ diferencias necesarias si se utilizan las probabilidades asociadas.

3.2 Distancias basadas en la medida de familias de subconjuntos

Si bien las distancias que acabamos de definir usan directamente la información que suministran las medidas, pensamos que no son "buenas definiciones" en el sentido de que no establecen distinción alguna entre cada una de las coordenadas de g y g' en $\mathbb{R}^s$. Más

concretamente, si las medidas difusas fuesen en particular medidas de probabilidad, éstas vienen determinadas por la medida sobre los conjuntos unitarios ya que, por la propiedad de aditividad, se puede conocer la medida sobre los conjuntos de cardinal mayor; es decir, estos últimos no suministran nueva información. Esto repercute en el cálculo de las diferencias $|g - g'|$. En efecto, es fácil comprobar que

$$|P(A) - P'(A)| = |P(\bar{A}) - P'(\bar{A})|$$

Es decir, la diferencia de los conjuntos de cardinal 1 coincide con el de los conjuntos de cardinal $n - 1$, la diferencia de los conjuntos de cardinal 2 coincide con los de cardinal $n - 2$, etc... y puesto que ambas diferencias (las de los conjuntos de cardinal m y las de los conjuntos de cardinal $n - m$) proceden de las diferencias de las de cardinal 1, por la aditividad, tan sólo tienen interés las diferencias en los elementos atómicos. En consecuencia cabría plantearse que sentido tiene el calcular todas las diferencias $|g - g'|$ en el caso probabilístico.

Por otro lado, si consideramos dos medidas difusas g y g' cualesquiera, es necesario distinguir los conjuntos de distinto cardinal, pues la información sobre singletons no predice la información de los de cardinal superior (lo único que se verifica es la monotonía). Por ejemplo, si decimos que una medida g la interpretamos como el grado subjetivo de ocurrencia de un suceso y es tal que $g(x) = .9$ y que $g(x, y) = .901$, estamos reflejando el hecho de que tenemos un gran optimismo en que se dé $\{x\}$ y $\{x, y\}$, pero ello no quiere decir que necesariamente seamos pesimistas en la ocurrencia de $\{y\}$ (una probabilidad establecería que $g(y) = .001$). Es por ello, que tampoco nos basta considerar que una medida g es tan sólo un vector de $\mathbb{R}^s$ en el que todas sus componentes tienen la misma importancia, sino como un vector formado por distintas componentes donde cada una trabaja sobre conjuntos del mismo cardinal (i.e. son de la misma categoría), que pueden considerarse "comparables". El siguiente ejemplo nos permite aclarar esta idea.

Ejemplo 2 Consideremos las medidas g y g' definidas por:

B_i	g	g'	$ g - g' $
$\{x_1\}$	0	.9	.9
$\{x_2\}$	.1	.6	.5
$\{x_3\}$	.2	.7	.5
$\{x_1, x_2\}$	.9	.9	0
$\{x_1, x_3\}$	.9	.9	0
$\{x_2, x_3\}$	.7	.7	0

De un modo intuitivo, cabría establecer un criterio en el que se reflejara que las medidas son muy distintas en conjuntos de cardinal 1, ya que se diferencian casi en el máximo valor posible (i.e. uno) en la coordenada del x_1 y en la mitad de este valor en el resto; es decir, g se

diferencia de g' en "la dirección" de $\{x_1\}$ en 0.9, y en las "direcciones" de $\{x_2\}$ y $\{x_3\}$ en 0.5. Sin embargo, ambas medidas verifican que $g = g'$ para conjuntos de cardinal 2, por lo que considerando conjuntos de este tamaño ambas son indistinguibles.

Con la notación introducida, es claro que $\mathcal{P}(X) = \{A_m^p / p = 1, \dots, k; m = 0, \dots, n\}$ y que para cada posible cardinal m , una medida difusa define un único punto en $\mathbb{R}^k$, con $k = \binom{n}{m}$, dado por:

$$g(A_m) = (g(A_m^1), g(A_m^2), \dots, g(A_m^p), \dots, g(A_m^k)) \quad (7)$$

Notar que con el orden definido anteriormente impone el orden de las componentes de $g(A_m)$.

En estas condiciones, vamos a definir distancias entre dos medidas difusas, en primer lugar, en términos de cada subconjunto de un cardinal determinado, para, posteriormente, definir distancias globales entre ellas considerando los distintos cardinales.

Definición 10 Consideremos dos medidas difusas g y g' definidas sobre el mismo referencial finito X . Definimos la distancia de g a g' en la dirección de A_m^p como $d_{A_m^p}(g, g') = |g(A_m^p) - g'(A_m^p)|$ donde $|\cdot|$ denota el valor absoluto en $\mathbb{R}$.

Si consideramos ahora todos los conjuntos de un mismo cardinal m , $A_m = (A_m^1, A_m^2, \dots, A_m^p, \dots, A_m^k)$ podemos extender la definición anterior a **distancias entre vectores de conjuntos** del mismo cardinal:

Definición 11 Consideremos dos medidas difusas g y g' definidas sobre el mismo referencial finito X de cardinal n . Definimos la distancia entre g y g' para el cardinal m (o a nivel m) como:

$$d_{A_m}(g, g') = \|g(A_m) - g'(A_m)\|$$

donde $g(A_m)$ está definida según (7) y $\|\cdot\|$ denota cualquiera de las normas de $\mathbb{R}^k$.

Por último, como $\mathcal{P}(X) = \{A_0 = \emptyset, A_1, \dots, A_{n-1}, A_n = X\}$, se puede calcular la distancia entre

$$\begin{aligned} g &\equiv (g(A_0), g(A_1), \dots, g(A_{n-1}), g(A_n)) \\ g' &\equiv (g'(A_0), g'(A_1), \dots, g'(A_{n-1}), g'(A_n)) \end{aligned}$$

donde $g(A_m)$ está definida según (7); es decir, calcular la distancia global de g y g' .

En este punto las definiciones de esta última distancia son múltiples, atendiendo a la importancia o preferencia que uno estime entre $g(A_i)$ y $g'(A_i)$. Por ejemplo, podemos identificar cada vector $g(A_i)$ con algún valor promedio α_i y establecer la distancia entre g y g' como la distancia en $\mathbb{R}^n$ entre los puntos $(\alpha_1, \dots, \alpha_n)$ y $(\alpha'_1, \dots, \alpha'_n)$. Particularmente, creemos que lo mejor sería establecer algún criterio de media ponderada sobre cada una de las componentes, es decir, considerar las distancias respecto a cada uno de los posibles

cardinales y establecer una media. Así, por ejemplo, se pueden establecer las distancias:

$$d_1(g, g') = \frac{1}{n} \sum_{m=1}^n d_{A_m}(g, g')$$

$$d_2(g, g') = \sum_{m=1}^n \frac{1}{m} d_{A_m}(g, g')$$

4 Probabilidad Más Próxima a una Medida Difusa

Puesto que disponemos ya de una forma de medir el parecido entre medidas difusas (por medio de las distancias d_{A_m}), y una probabilidad es un caso particular de ellas, podemos plantearnos hallar la probabilidad más próxima a una medida difusa dada, lo que permite: (1) Si la aproximación de una medida difusa a su probabilidad más cercana es suficientemente buena, se podría trabajar con dicha probabilidad en lugar de con la medida difusa, lo que en general será más eficiente. (2) Establecer índices de aditividad. Si evaluamos la distancia entre una medida difusa y la probabilidad más cercana, disponemos de un valor que cuantifica la desviación de esa medida respecto de un comportamiento aditivo.

Si definimos una distancia única, podemos plantearnos el conocer la probabilidad más cercana a cierta medida g y en consecuencia medir en cierto modo la aditividad de la medida [5, 2]. Sin embargo, con el planteamiento expuesto, no tiene sentido el cálculo de la probabilidad más cercana a la medida difusa considerada, pero sí el de las probabilidades más cercanas en términos de conjuntos de distinto cardinal. Presentamos para esta tarea dos planteamientos:

• **Primer Planteamiento:** Exigencia de aditividad según un cardinal considerado.

Consideremos una medida difusa g , y supongamos que la suma de la medida en los conjuntos de cardinal m es t_m , es decir:

$$g(A_m^1) + \dots + g(A_m^{p_m}) + \dots + g(A_m^{k_m}) = t_m \quad (8)$$

donde k_m es el número de subconjuntos de cardinal m . Si la medida fuese una probabilidad, entonces, por su comportamiento aditivo se verificaría:

$$P(A_m^1) + \dots + P(A_m^{p_m}) + \dots + P(A_m^{k_m}) = \binom{n-1}{m-1} = r_m \quad (9)$$

Por tanto, si consideramos los subconjuntos de cardinal m , y el vector asociado:

$$g(A_m) = (g(A_m^1), g(A_m^2), \dots, g(A_m^{p_m}), \dots, g(A_m^{k_m}))$$

y si g fuese una probabilidad debería de verificarse que

$t_m = r_m = \frac{(n-1)!}{(m-1)!(n-m)!}$ [ver (8) y (9)]. Puesto que en general dicha igualdad no se verifica, si "exigimos" un comportamiento aditivo en la medida, nos motiva la siguiente definición:

Definición 12 Definimos la probabilidad más cercana a la medida difusa g , y la notamos por P_g , como aquella que viene determinada por el sistema de ecuaciones:

$$\begin{cases} \sum_{x \in A_m^1} p(x) = P_g(A_m^1) = g(A_m^1) \frac{r_m}{t_m} \\ \dots \dots \dots \\ \sum_{x \in A_m^{k_m}} p(x) = P_g(A_m^{k_m}) = g(A_m^{k_m}) \frac{r_m}{t_m} \\ \sum_{x \in X} p(x) = 1 \end{cases}$$

Cabe llamar la atención al caso particular en que $m = 1$ (i.e. conjuntos unitarios), ya que si $t_1 = 1$ la probabilidad más cercana a g es ella misma pues en este caso g es una probabilidad (en los conjuntos unitarios) y por tanto: $P_g(x_i) = g(\{x_i\}) = g(A_1^1)$. En el caso de que $t_1 \neq 1$ la probabilidad más cercana es aquella que obtendríamos al realizar una normalización usual:

$$P_g(x_i) = \frac{g(\{x_i\})}{\sum_i g(\{x_i\})}$$

• **Segundo Planteamiento:** Determinar la probabilidad más próxima a partir del convexo asociado a la medida (según una distancia considerada).

Puesto que en general, para un cierto cardinal m , $\sum_{l=1}^{k_m} g(A_m^l) = t_m \neq r_m$, podemos conseguir que esta igualdad se cumpla añadiendo o restando la cantidad necesaria en cada una de las componentes, $g(A_m^l)$, de $g(A_m)$; con ello conseguimos k_m probabilidades asociadas a la medida g , $\mathcal{P} = \{P_1, \dots, P_{k_m}\}$, y definidas por: $P_l \equiv (g(A_m^1), \dots, p_l, \dots, g(A_m^{k_m}))$ con $p_l \in \mathcal{R}$ tal que

$$g(A_m^1) + \dots + p_l + \dots + g(A_m^{k_m}) = r_m \quad (10)$$

Como en el planteamiento anterior, la idea responde a considerar un cardinal m , para el cual si g fuese aditiva verificaría que el punto $g(A_m) \in \mathcal{R}^{k_m}$ se encontraría en el hiperplano $\sum g(A_m^l) = r_m$. Si no lo está entonces fijamos todas las coordenadas menos una (p.e. $g(A_m^1)$) y ésta la "prolongamos" hasta llegar al hiperplano dado por (10). Al repetir este proceso para todas las coordenadas de $g(A_m)$, se obtiene el conjunto de probabilidades $\mathcal{P}$ que constituyen los puntos extremos del convexo que definen.

Determinado este convexo podemos considerar como probabilidad más cercana P aquella que minimice la distancia d_{A_m} con g (para los conjuntos de cardinal considerado). Es decir, P viene dada por la resolución del siguiente programa:

$$\begin{aligned} & \min_{P \in \mathcal{P}} d_{A_m}(g, P) \\ \text{s.a.} \end{aligned}$$

$$\begin{aligned} & \min_{P_l \in \mathcal{P}} P_l(A_m^1) \leq P(A_m^1) \leq \max_{P_l \in \mathcal{P}} P_l(A_m^1) \\ & \sum_{l=1}^{k_m} P(A_m^l) = r_m \end{aligned}$$

5 Indices de Certidumbre e Incertidumbre

La determinación del grado de (in)certidumbre de una medida puede ser de gran ayuda en determinados problemas. Por ejemplo, en sistemas de razonamiento bajo incertidumbre, es frecuente la imposibilidad de manejar toda la información disponible sobre ciertas variables en el proceso de inferencia. En estos casos, se suele recurrir a métodos que elijan aquella información que es realmente mas relevante. (p.e. ver [4] para probabilidades).

Existen dos tipos de índices clásicos que permite determinar la calidad de la medida: la entropía y la especificidad [12, 10]. La combinación de ambos índices proporciona una idea de la calidad de la medida. La principal limitación es que sólo sirve para evidencias y no se puede extender a medidas difusas en general. De Campos [2] presenta otras alternativas totalmente diferentes para definir índices de incertidumbre para medidas difusas, que se basan en la distancia entre medidas. Ello es posible si se tiene en cuenta que las medidas que proporcionan más incertidumbre son las de ignorancia:

$$Bel_0(A) = \begin{cases} 0 & \text{si } A \neq X \\ 1 & \text{si } A = X \end{cases} \quad Pl_0(A) = \begin{cases} 1 & \text{si } A \neq \emptyset \\ 0 & \text{si } A = \emptyset \end{cases} \quad (11)$$

y las que proporcionan mayor certidumbre son las medidas de Dirac.

Considerando las distancias definidas en este trabajo, podemos establecer índices atendiendo al cardinal del tipo de subconjuntos en el que estemos interesados. Así se puede definir un índice de certidumbre en términos de conjuntos de distinto cardinal midiendo la distancia entre la medida g y una medida de ignorancia: a mayor distancia, menos incertidumbre genera g .

Puesto que existen dos codificaciones de la ignorancia, Bel_0 y Pl_0 , que, en términos muy generales, podemos clasificar como pesimista y optimista, el índice de incertidumbre estará determinado por aquella medida de ignorancia que más se aproxime a g .

Definición 13 Sea g una medida difusa sobre el referencial $X = \{x_1, \dots, x_n\}$. Definimos el índice de certidumbre de g a nivel m como:

$$C_m(g) = \min(d_{A_m}(g, Bel_0), d_{A_m}(g, Pl_0))$$

con d_{A_m} según la definición 11.

De forma análoga podemos definir un índice de incertidumbre en términos de conjuntos de distinto cardinal

midiendo la distancia entre la medida g y las probabilidades degeneradas $P_i(x_j) = \delta_{i,j}$, $i = 1, \dots, n$ ya que éstas son las que menos incertidumbre provocan: a mayor distancia, más incertidumbre proporciona g . Como disponemos de n medidas de Dirac sobre un referencial de cardinal n , parece lógico establecer como índice de incertidumbre la menor de las distancias entre g y P_j .

Definición 14 Sea g una medida difusa sobre el referencial X . Definimos el índice de incertidumbre a nivel m como: $I_m(g) = \min_{1 \leq j \leq n} d_{A_m}(g, P_j)$

Teniendo en cuenta que es equivalente codificar una información por una medida difusa g que por su dual g^* , cabe esperar que el índice de (in)certidumbre definido mida el mismo grado independientemente de la medida utilizada. En efecto, es fácil demostrar que:

Proposición 1 Dada una pareja de medidas difusas (g, g^*) sobre un referencial de cardinal n se verifica:

$$\begin{aligned} d_{A_m}(g, Pl_0) &= d_{A_{n-m}}(g^*, Bel_0) \\ d_{A_m}(g, P_j) &= d_{A_{n-m}}(g^*, P_j) \end{aligned}$$

donde Pl_0 y Bel_0 vienen dadas por (11) y P_j es la distribución de probabilidad que vale 1 en x_j y cero en el resto. Y, en consecuencia:

1. $C_m(g) = \min(d_{A_m}(g, Bel_0), d_{A_m}(g, Pl_0)) = \min(d_{A_{n-m}}(g^*, Pl_0), d_{A_{n-m}}(g^*, Bel_0)) = C_{n-m}(g^*)$
2. $I_m(g) = \min_{1 \leq j \leq n} d_{A_m}(g, P_j) = \min_{1 \leq j \leq n} d_{A_{n-m}}(g^*, P_j) = I_{n-m}(g^*)$

Es decir, fijado cierto nivel m , el grado de (in)certidumbre que establece la medida g coincide con con el grado de (in)certidumbre que establece la medida g^* a nivel $n - m$. De esta forma, utilizando los índices de certidumbre o incertidumbre puede medirse, de forma parcial y detallada, la información que sobre la verificación de un predicado nos dé la medida difusa.

6 Conclusiones y estudios futuros

En este trabajo hemos introducido un nuevo enfoque para la caracterización de medidas difusas. La idea principal consiste en considerar que el comportamiento de una medida difusa puede ser radicalmente distinto dependiendo de los subconjuntos del referencial sobre los que se esté evaluando. En ese sentido, hemos definido nuevas medidas de distancia entre medidas difusas atendiendo al comportamiento de las medidas en los distintos subconjuntos del referencial. Usando estas distancias, definimos dos índices de certidumbre e incertidumbre, de los cuales se estudian algunas propiedades. Así mismo, actualizamos el concepto de probabilidad más cercana a una medida difusa. Pensamos que el presente trabajo puede servir

para una mejor comprensión del comportamiento de las medidas difusas.

Como futuros trabajos, podría estudiarse el comportamiento de las distancias e índices definidos para clases concretas de medidas difusas. Este estudio puede ser tanto teórico como experimental, seleccionando una batería de medidas y calculando sobre ellas las distancias y los índices de (in)certidumbre.

Referencias

- [1] Machine intelligence and pattern recognition. In B.M. Ayyub, M.M. Gupta, L.N. Kanan, eds. *Analysis and Management: Theory and Applications*, vol. 13. North Holland, 1992.
- [2] L.M. de Campos. *Caracterización y estudio de medidas e integrales difusas a partir de probabilidades*. PhD thesis, Dpto. Ciencias de la Computación e I.A. U. de Granada, 1987.
- [3] L.M. de Campos, M.T. Lamata, S. Moral. Distances between fuzzy measures through associated probabilities: some applications. *Fuzzy Sets and Systems*, (35):57-68, 1990.
- [4] J.E. Cano, L.D. Hernández, S. Moral. Importance sampling algorithms for belief networks. Admitido en *International J. of App. Reasoning*, 1996.
- [5] D. Dubois, H. Prade. Unfair coins and necessity measures: towards a possibilistic interpretation of histograms. *Fuzzy Sets and Syst.*, 10:15-20, 1983.
- [6] D. Dubois, H. Prade. A note on measures of specificity for fuzzy sets. *Busefal*, 19:83-89, 1984.
- [7] G.J. Klir. Where do we stand on measures of uncertainty, ambiguity, fuzziness, and the like?. *Fuzzy Sets and Systems*, (24):141-160, 1987.
- [8] G.J. Klir. The role of uncertainty principles in inductive systems modelling. *Kybernetes*, 17(2):23-34, 1988.
- [9] G.J. Klir. Probabilistic versus possibilistic conceptualization of uncertainty. En *Analysis and Management of uncertainty: Theory and Applications*, p.13-25. Elsevier Science Pub., 1992.
- [10] M.T. Lamata, S. Moral. Classification of fuzzy measures. *Fuzzy Sets and Syst.*, 33:243-253, 1989.
- [11] M. Sugeno. *Theory of fuzzy integrals and its applications*. PhD. Thesis, Tokio Ins. of Tech., 1974.
- [12] R.R. Yager. Entropy and specificity in a mathematical theory of evidence. *Int. J. of general systems*, 9:249-260, 1983.

MEDIDAS LINGÜÍSTICAS DE RACIONALIDAD PARA ALCANZAR EL CONSENSO EN GRUPO

F. Herrera
Dpto. de Ciencias de la
Computación e I.A.
E.T.S de Ingeniería Informática
Universidad de Granada
herrera@decsai.ugr.es

E. Herrera-Viedma
Dpto. de Ciencias de la
Computación e I.A.
E.T.S de Ingeniería Informática
Universidad de Granada
viedma@decsai.ugr.es

J.L. Verdegay
Dpto. de Ciencias de la
Computación e I.A.
E.T.S de Ingeniería Informática
Universidad de Granada
verdegay@decsai.ugr.es

Resumen

En este trabajo se proponen varias medidas lingüísticas de racionalidad para problemas de Toma de Decisiones en Grupo, en ambiente lingüístico, con objeto de eliminar las inconsistencias existentes en las opiniones de los individuos, y de este modo conseguir soluciones más coherentes. Estas medidas se pueden usar conjuntamente con las medidas de consenso para guiar los procesos de consenso en grupo.

Palabras Clave: Modelos Lingüísticos, Consenso, Racionalidad, Decisión en Grupo.

1 INTRODUCCIÓN

Un problema de *Toma de Decisiones en Grupo (TDG)* se establece en situaciones 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. A veces, existe una persona, llamada *moderador*, encargada de dirigir todo el proceso de resolución del problema de TDG hasta que los individuos logran un acuerdo sobre la solución a escoger.

En los problemas de TDG 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 alternativas solución. El segundo, hace referencia a cómo obtener el conjunto de alternativas 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 mediante una *medida de consenso*. 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 TDG 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, momento en el cual se obtiene la solución de consenso mediante la aplicación del proceso de selección.

En toda situación de TDG, es normal que los individuos además de presentar opiniones enfrentadas o divergentes, presenten inconsistencias en sus opiniones, es decir, incoherencias en sus preferencias sobre el conjunto de alternativas. Esto puede ocasionar que el proceso de TDG desemboque en soluciones consensuadas poco selectivas (que incorporen todas las alternativas) e incluso distorsionadas (que no incorporen las mejores alternativas). Por tanto, sería bueno intentar eliminar esas inconsistencias de sus preferencias al mismo tiempo que se lleva a cabo el proceso de consenso. Para lograrlo, nosotros presentamos en [6] un *Modelo de Consenso Racional*, mostrado en la figura 1. Este modelo es guiado por dos tipos de medidas: (i) *medidas de consenso* y (ii) *medidas de consistencia*. Ambas dan al moderador toda la información necesaria para coordinar el proceso de consenso con vistas a lograr disminuir las divergencias e inconsistencias en las opiniones de los individuos, respectivamente. En [3, 4], asumiendo que los individuos expresan sus opiniones mediante relaciones lingüísticas de preferencia, presentamos varias medidas lingüísticas de consenso para alcanzar el consenso en problemas de TDG.

En este trabajo, asumiendo un ambiente lingüístico similar, proponemos varias *medidas lingüísticas de racionalidad*. Estas medidas tratan de resolver dos de las cuestiones que plantea el problema de consistencia en TDG: (i) cuándo un individuo, considerado individualmente, se dice que es racional en sus preferencias, y (ii) cuándo un grupo de individuos se dice racional. Tenemos dos niveles de evaluación del estado de consistencia: *nivel del individuo* y *nivel del conjunto de individuos*, y por ello, se presentan dos tipos de medidas de consistencia, dependiendo del nivel de evaluación:

1. **Medidas Lingüísticas Individuales de Consistencia:** miden el grado de consistencia existente en las preferencias de cada individuo.
2. **Medidas Lingüísticas Colectivas de Con-**

2. *Complejidad*, en el siguiente sentido,

$$p_{ij} \geq NEG(p_{ji}), \forall (z_i, x_j).$$

Esta propiedad se requiere para asegurar que todos los individuos consideran factible y comprensible al conjunto de alternativas.

3 MEDIDAS LINGÜÍSTICAS DE RACIONALIDAD

Intentar dar una formalización matemática concreta del concepto general de *racionalidad* resulta complejo. Sin embargo, si el problema de definición de la racionalidad se enfoca desde el punto de vista de las opiniones de los individuos, es decir, si se analiza en base a las preferencias de los individuos, parece más o menos posible dar una caracterización matemática del concepto de racionalidad [2]. En este sentido, la racionalidad se define como una combinación de la *consistencia explícita e implícita* de las preferencias de los individuos [2]. La consistencia explícita hace referencia a la ausencia de contradicciones explícitas, y la consistencia implícita se relaciona con los juicios de valor (y su uso) en base a los que se expresan las opiniones.

En un contexto crisp, donde cada individuo expresa su opinión sobre los pares de alternativas de X mediante una relación binaria crisp de preferencia, R , el concepto de *consistencia* se ha definido tradicionalmente en términos de *acíclicidad* [9], es decir, de ausencia de secuencias como $x_1, x_2, \dots, x_k$ ($x_{k+1} = x_1$) con $x_j R x_{j+1} \forall j = 1, \dots, k$, en las relaciones binarias o *ausencia de ciclos*. Por otro lado, en un contexto difuso, donde cada individuo expresa sus opiniones mediante relaciones difusas de preferencia, P , una forma bien conocida de caracterizar la consistencia es a través de la *MAX-MIN transitividad* [12]. Por tanto, en ambos contextos, un individuo se considera o no consistente si su respectiva relación de preferencia es o no es *acíclica* (*MAX-MIN transitiva*, respectivamente), y por tanto, en este sentido, la *consistencia* se entiende como una propiedad crisp. Sin embargo, de acuerdo con Montero [7], asumimos que la consistencia de los individuos es claramente un concepto difuso, ya que las opiniones de un individuo se pueden considerar más consistentes que las de otro en cierto grado. Bajo esta perspectiva, la consistencia se puede ver como un conjunto difuso caracterizado por una apropiada función de pertenencia, llamada *medida difusa de racionalidad*, que asigna a cada individuo un valor de consistencia (grado) entre 0 (inconsistencia absoluta) y 1 (consistencia absoluta), y de este modo, establece una clasificación difusa de los individuos. En este sentido, en [7] se propone una medida de racionalidad difusa que se basa en una suma ponderada concreta de todos los caminos acíclicos y en [2] una definición axiomática que toda medida de racionalidad difusa consistente explícitamente debe satisfacer. Aquí, dado que trabajamos con relaciones lingüísticas de preferencia, proponemos varias *medidas lingüísticas de consistencia*. Concretamente, presentamos dos tipos de medidas de consistencia: (i) *medidas lingüísticas individuales de consistencia* para medir la consistencia de un individuo, y (ii) *medidas lingüísticas colectivas de consistencia*, para medir la consistencia del grupo de individuos. Sus definiciones se basan en la *idea de acíclicidad*

de Sen [9] y, siguiendo la definición de *racionalidad difusa* dada en [7], pero asumiendo un contexto de trabajo lingüístico. Como se muestra en la figura 2, que representa el esquema del modelo lingüístico de consenso racional propuesto en [6], estas medidas se calculan en dos procesos:

1. **Proceso de Detección**, a partir de las relaciones lingüísticas de preferencia se detectan los conjuntos de inconsistencias (*conjuntos de ciclos de preferencia inconsistentes de cada relación*).
2. **Proceso de Cálculo**, a partir de las inconsistencias detectadas se obtienen las medidas lingüísticas de consistencia.

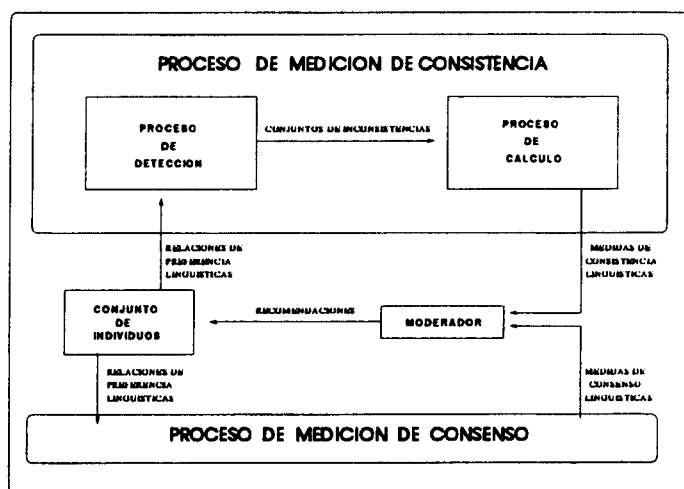


Figure 2: Modelo lingüístico de consenso racional.

A continuación, estudiamos cada uno de ellos.

3.1 PROCESO DE DETECCIÓN

Dado que nuestras medidas de consistencia se basan en los *ciclos de preferencia inconsistentes* existentes en las relaciones de preferencia de los individuos, el principal objetivo de este proceso es la detección del conjunto de ciclos de preferencia inconsistentes de cada relación. Para conseguirlo, los ciclos no se detectan directamente en las relaciones iniciales, sino en las respectivas *relaciones estrictas de preferencia*, ya que en éstas se observan claramente los valores de preferencia reales existentes entre las alternativas. La relación estricta de una dada se obtiene en el sentido propuesto por Orlovski [8], como sigue:

Sea P una relación lingüística de preferencia completa y recíproca débil de $X = \{x_1, \dots, x_n\}$. Entonces, $P^e = (p_{ij}^e)$ es una *relación lingüística estricta de preferencia* evaluada en el conjunto de etiquetas, $S' = S \cup \{\emptyset\}$,

$$\mu_{P^e} : X \times X \rightarrow S',$$

donde $\mu_{P^e}(x_i, x_j) = p_{ij}^e$, tal que, $p_{ij}^e = \emptyset$ si $p_{ij} < p_{ji}$, o $p_{ij}^e = s_k \in S$ si $p_{ij} \geq p_{ji}$ con $p_{ij} = s_l$, $p_{ji} = s_l$ y $l = t + k$.

Por tanto, trabajando con relaciones estrictas de preferencia, un par de alternativas, (x_i, x_j) , puede presentar cualquiera de estas tres relaciones básicas:

1. *Relación de Preferencia (R)*: x_i se prefiere a x_j , y por tanto, $x_i R x_j \iff p_{ij}^s > s_0$.
2. *Relación de No Preferencia (RN)*: x_i no se prefiere a x_j , y por tanto, $x_i RN x_j \iff p_{ij}^s = \emptyset$.
3. *Relación de Indiferencia (I)*: x_i es indiferente a x_j , y por tanto, $x_i I x_j \iff p_{ij}^s = s_0$.

Las relaciones inversas de cada relación (R, RN, I) se definen como: (i) $R^{-1} = RN$, (ii) $RN^{-1} = R$, (iii) $I^{-1} = I$.

Observando P^e , es claro que dada una cadena de alternativas distintas, $x_1 - x_2 - \dots - x_k - x_1$ ($k \geq 3$), será un ciclo de preferencia inconsistente, si y sólo si $x_1 R_1 x_2 R_2 \dots x_k R_k x_1$, de modo que o:

caso 1: $R_h \in \{R, I\} \forall h = 1, 2, \dots, k$ y $R \in \{R_h : h = 1, 2, \dots, k\}$, llamado "ciclo de preferencia inconsistente positivo"; o

caso 2: $R_h \in \{RN, I\} \forall h = 1, 2, \dots, k$ and $RN \in \{R_h : h = 1, 2, \dots, k\}$, llamado "ciclo de preferencia inconsistente negativo".

En otras palabras, una cadena de alternativas será un ciclo de preferencia inconsistente en cada caso, si R_h es o R (RN respectivamente) o I , pero tomando al menos una vez el valor R (RN respectivamente).

Lema. Sea P^e una relación lingüística estricta de preferencia asociada a una relación lingüística de preferencia, P , si $x_1 R_1 x_2 R_2 \dots x_k R_k x_1$ es un ciclo de preferencia inconsistente positivo (negativo) de P^e , entonces $x_1 R_k^{-1} x_k \dots R_2^{-1} x_2 R_1^{-1} x_1$ es un ciclo de preferencia inconsistente negativo (positivo) de P^e .

Su demostración es evidente.

Teorema. Sea P^e una relación lingüística estricta de preferencia asociada a una relación lingüística de preferencia, P , entonces cualquier ciclo de preferencia inconsistente positivo (negativo) de alternativas diferentes, $G = x_1 R_1 x_2 R_2 \dots x_k R_k x_1$ ($k \geq 4$), implica al menos un ciclo de preferencia inconsistente de 3 alternativas diferentes.

Su demostración se puede encontrar en [6].

Por tanto, en base a los anteriores lema y teorema, decidimos usar para definir nuestras medidas de consistencia sólo los ciclos de preferencia inconsistentes positivos de 3 alternativas diferentes, notados como C^k , $\forall e_k \in E$.

3.2 PROCESO DE CÁLCULO

En este proceso, definimos varias medidas lingüísticas de consistencia a partir de los conjuntos de ciclos de preferencia inconsistentes, C^k . Como ya anticipábamos en la introducción, presentamos dos tipos de medidas lingüísticas de consistencia, distinguiendo entre dos niveles de evaluación: (i) *nivel de individuo*, y (ii) *nivel de grupo de individuos*. Además, ambos tipos de medidas de consistencia se pueden obtener en base a dos aspectos:

1. *Aspectos cualitativos* de las preferencias, es decir, de acuerdo a la intensidad o gravedad de la naturaleza de las inconsistencias detectadas.

2. *Aspectos cuantitativos* de las preferencias, es decir, de acuerdo al número o cantidad de inconsistencias detectadas.

Por tanto, trabajamos con cuatro medidas de consistencia como se muestra en la figura 3: *medida de consistencia lingüística individual cualitativa*, *medida de consistencia lingüística colectiva cualitativa*, *medida de consistencia lingüística individual cuantitativa*, *medida de consistencia lingüística colectiva cuantitativa*. Antes de definir las expresiones que nos permiten cal-

MEDIDAS DE CONSISTENCIA			
		PERSPECTIVA CUALITATIVA	PERSPECTIVA CUANTITATIVA
NIVEL DE EVALUACION	NIVEL 1	MEDIDA DE CONSISTENCIA LINGÜISTICA INDIVIDUAL CUALITATIVA	MEDIDA DE CONSISTENCIA LINGÜISTICA INDIVIDUAL CUANTITATIVA
	NIVEL 2	MEDIDA DE CONSISTENCIA LINGÜISTICA COLECTIVA CUALITATIVA	MEDIDA DE CONSISTENCIA LINGÜISTICA COLECTIVA CUANTITATIVA

Figure 3: Medidas lingüísticas de consistencia.

cular estas medidas, introduciremos la definiciones de *cuantificador lingüístico difuso* y del operador de agregación de información lingüística *LOWA*.

1. **Cuantificador Lingüístico.** Los cuantificadores lingüísticos difusos fueron presentados por Zadeh en 1983 [11]. Un cuantificador lingüístico es un conjunto difuso definido sobre $[0, 1]$ que representa una propiedad lingüísticamente cuantificada. Zadeh distinguió dos tipos de cuantificadores, los absolutos como "alrededor de 7" y los proporcionales o relativos como "muchos". Utilizamos los cuantificadores proporcionales crecientes.

Si Q es un cuantificador proporcional, entonces Q puede representarse como un subconjunto difuso de $[0, 1]$ tal que para cada $r \in [0, 1]$, $Q(r)$ indica el grado en el que la proporción r de objetos satisface el concepto representado por Q .

La función de pertenencia de un cuantificador proporcional, Q , puede valorarse numéricamente, $Q : [0, 1] \rightarrow [0, 1]$, o lingüísticamente, $Q : [0, 1] \rightarrow L$, siendo $L = \{l_i\}$, $i \in J = \{0, \dots, U\}$, un conjunto de etiquetas. En el primer caso, lo notaremos, Q^1 , y su función de pertenencia vendrá dada por la siguiente expresión [11]:

$$Q^1(r) = \begin{cases} 0 & \text{si } r < a \\ \frac{r-a}{b-a} & \text{si } a \leq r \leq b \\ 1 & \text{si } r > b \end{cases}$$

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 [4]:

$$Q^2(r) = \begin{cases} l_0 & \text{si } r < a \\ l_i & \text{si } a \leq r \leq b \\ l_U & \text{si } r > b \end{cases}$$

siendo l_0 y l_U las etiquetas mínima y máxima de L , respectivamente y, $l_i = \text{Sup}_{l_q \in M} \{l_q\}$, con

$$M = \{l_q \in L : \mu_{l_q}(r) = \text{Sup}_{i \in J} \{\mu_{l_i}(\frac{r-a}{b-a})\}\}.$$

Algunos ejemplos de cuantificadores son: "La mayoría", "Al menos la mitad", "Tantos como sea posible" con los pares (a,b): (0.3,0.8), (0,0.5), (0.5,1), respectivamente.

2. **Operador LOWA.** El operador LOWA es un operador de agregación de información lingüística por computación directa sobre las etiquetas. Se define de acuerdo a la siguiente expresión:

$$\phi(a_1, \dots, a_m) = W \cdot B^T =$$

$$= C\{w_k, b_k, k = 1, \dots, m\} =$$

$$= w_1 \odot b_1 \oplus (1 - w_1) \odot C\{\beta_h, b_h, h = 2, \dots, m\},$$

donde $C\{\dots\}$ simboliza el operador de combinación convexa de etiquetas lingüísticas; $W = [w_1, \dots, w_m]$, el vector de ponderación, tal que, a) $w_i \in [0, 1]$, b) $\sum_i w_i = 1$; $\beta_h = w_h / \sum_{k=2}^m w_k$, $h = 2, \dots, m$; 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 $(a_1, \dots, a_m)$. 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 $k = \text{MIN}\{T, i + \text{round}(w_1 \cdot (j - i))\}$, 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 $C\{w_i, p_i, i = 1, \dots, m\} = p_j$. Los pesos w_i para la agregación se pueden calcular a partir de la función que describe a un cuantificador proporcional creciente Q mediante la siguiente expresión [10]:

$$w_i = Q(i/m) - Q((i-1)/m), i = 1, \dots, m.$$

En este sentido, cuando usemos un cuantificador Q para hallar los pesos, el operador LOWA será simbolizado por ϕ_Q . Sus propiedades se pueden encontrar en [5].

3.2.1 MEDIDA DE CONSISTENCIA LINGÜÍSTICA INDIVIDUAL CUALITATIVA

Esta medida proporciona una perspectiva cualitativa de la situación de consistencia existente en la relación lingüística de preferencia de cada individuo. Evalúa la calidad (gravedad) de la consistencia de los individuos de acuerdo al tipo de relaciones existentes entre las alternativas en los ciclos de preferencia inconsistentes detectados en sus respectivas relaciones lingüísticas estrictas de preferencia. Esta medida se basa en la medida difusa de racionalidad de Montero [7]. La medida difusa de racionalidad de Montero se basa en el *grado de aciclicidad* de una relación de preferencia, y se calcula usando ponderaciones numéricas que se obtienen a partir de las relaciones existentes entre las alternativas de todos los posibles *ciclos de preferencia consistentes* de la relación de preferencia. Nuestra medida

se basa en el *grado de no aciclicidad*, y se calcula usando sólo los *pesos lingüísticos* obtenidos a partir de las relaciones existentes entre las alternativas de los *ciclos de preferencia inconsistentes positivos* de 3 alternativas diferentes de la relación lingüística estricta de preferencia. Por tanto, esta medida es una medida basada en *pesos lingüísticos*. Se calcula para cada individuo, e_k , y se nota como CI_a^k .

Cualquier ciclo de preferencia inconsistente positivo, $c_i^k \in C^k$, presenta la siguiente estructura: $x_i - x_v - x_w - x_i$, $\forall i, v, w \in \{1, 2, \dots, n\}$. Entonces, para cada ciclo c_i^k , definimos su peso lingüístico, z_i^k , como $z_i^k = \text{MIN}\{p_{iv}^k, p_{vw}^k, p_{wi}^k\}$. Por tanto, CI_a^k se obtiene de acuerdo a la siguiente expresión,

$$CI_a^k = \begin{cases} \text{NEG}(NA^k) & \text{si } C^k \neq \emptyset \\ s_T & \text{en otro caso} \end{cases}$$

donde $NA^k = \text{MAX}_i \{z_i^k, i = 1, \dots, \#(C^k)\}$ es el grado de no aciclicidad de la relación P^k .

3.2.2 MEDIDA DE CONSISTENCIA LINGÜÍSTICA INDIVIDUAL CUANTITATIVA

Esta medida cualitativa proporciona una perspectiva cuantitativa de la situación de consistencia existente en la relación lingüística de preferencia de cada individuo. Evalúa la cantidad de consistencia de los individuos, expresada en número de ciclos consistentes, a partir de los ciclos de preferencia inconsistentes positivos de 3 alternativas diferentes detectados. Por tanto, esta medida es una medida basada en *ciclos inconsistentes*. Se nota, CI_b^k , y se obtiene usando el concepto de mayoría difusa representado por un cuantificador lingüístico difuso, Q^2 .

Sea, c_i , el cardinal del conjunto de todos los posibles ciclos de 3 alternativas diferentes detectados en un conjunto de n alternativas. Entonces, c_i se determina por combinación de n elementos tomados de 3 en 3, de acuerdo a la siguiente expresión, $c_i = \binom{n}{3}$.

Un individuo, e_k , presenta la siguiente proporción de ciclos inconsistentes de 3 alternativas diferentes, r^k , $r^k = \#(C^k)/c_i$, y la medida de consistencia lingüística individual cuantitativa para él se obtiene como,

$$CI_b^k = Q^2(1 - r^k).$$

Por tanto, esta medida obliga a esta restricción $S = L$.

Estas dos medidas de consistencia individual, claramente, adoptan distintas concepciones del concepto de consistencia, que a veces, puede ocasionar situaciones contradictorias. Por ejemplo, puede existir un individuo, e_k , con sólo un ciclo de preferencia inconsistente positivo de 3 alternativas diferentes con un peso lingüístico de valor máximo, s_T . Entonces CI_a^k alcanzaría un valor muy alto, y sin embargo, CI_b^k , sería muy bajo. Por tanto, hemos de intentar lograr un equilibrio entre ambas medidas antes de aconsejar a los individuos.

3.2.3 MEDIDAS DE CONSISTENCIA LINGÜÍSTICAS COLECTIVAS

Estas medidas se definen fácilmente a partir de las medidas individuales. Por tanto tenemos dos medidas colectivas: (i) *Medida de Consistencia Lingüística*

Colectiva Cualitativa (CC_a), (ii) *Medida de Consistencia Lingüística Colectiva Cuantitativa* (CC_b). Se obtienen a partir de las medidas de consistencia individuales, considerando los grados de importancias de los individuos, $\mu_E(k)$, mediante el operador de agregación lingüística LOWA guiado por cuantificadores, Q^1 , representando el concepto de mayoría difusa de individuos. Sus respectivas expresiones de definición son:

$$CC_a = \phi_{Q^1}((CI_a^1 \wedge \mu_E(1)), \dots, (CI_a^m \wedge \mu_E(m))),$$

$$CC_b = \phi_{Q^1}((CI_b^1 \wedge \mu_E(1)), \dots, (CI_b^m \wedge \mu_E(m))).$$

Estas definiciones obligan a la restricción $S = G$.

Estas medidas se pueden usar como un criterio para validar la solución final que se obtenga al problema de TDG. Valores de las medidas de consistencia colectiva próximos a s_T indican una solución consensuada racional mejor, y valores alejados de s_T indican una peor solución.

4 EJEMPLO

Supongamos que tenemos cuatro alternativas y cuatro individuos cuyas preferencias lingüísticas usando el anterior conjunto de 9 etiquetas, S , son:

$$P^1 = \begin{bmatrix} - & PP & EP & MPP \\ BP & - & MP & EP \\ PP & PP & - & MPP \\ EP & P & MP & - \end{bmatrix}$$

$$P^2 = \begin{bmatrix} - & BP & P & MPP \\ P & - & MP & IM \\ P & PP & - & MPP \\ MP & BP & EP & - \end{bmatrix}$$

$$P^3 = \begin{bmatrix} - & EP & C & I \\ P & - & BP & PP \\ EI & P & - & MPP \\ C & EP & MP & - \end{bmatrix}$$

$$P^4 = \begin{bmatrix} - & PP & BP & PP \\ EP & - & P & PP \\ P & MP & - & MPP \\ C & BP & C & - \end{bmatrix}$$

y cuyos grados de importancia son:

$$\mu_E(1) = EP, \mu_E(2) = C, \mu_E(3) = PP, \mu_E(4) = EI.$$

Entonces, por ejemplo, la relación estricta de preferencia del individuo, e_4 , es:

$$P^{4,e} = \begin{bmatrix} - & \emptyset & EI & \emptyset \\ P & - & \emptyset & MPP \\ \emptyset & MPP & - & \emptyset \\ BP & \emptyset & BP & - \end{bmatrix}.$$

y presenta este conjunto de ciclos inconsistentes:

$$C^4 = \{(x_1, x_3, x_2, x_1), (x_2, x_4, x_3, x_2)\},$$

siendo $\{EI, MPP\}$, sus respectivos pesos lingüísticos asociados, donde, por ejemplo, $EI = \min\{p_{13}^{4,e}(EI), p_{32}^{4,e}(MPP), p_{21}^{4,e}(P)\}$. Su proporción de ciclos asociada es $r^4 = 2/4 = 0.5$. Entonces su medida de consistencia lingüística individual cualitativa es $CI_a^4 = EP$, y la cuantitativa $CI_b^4 = EP$, usando

el cuantificador lingüístico "Al menos la mitad" con parámetros (0.0, 0.5).

En base al mismo cuantificador, obtenemos las siguientes medidas de consistencia colectivas:

$$CC_a = \phi_{Q^1}((MC \wedge EP), (C \wedge C), (C \wedge PP), (EL \wedge EU)) = EP.$$

$$CC_b = \phi_{Q^1}((C \wedge EP), (C \wedge C), (C \wedge PP), (C \wedge EI)) = C.$$

De estos resultados, podemos concluir que: (i) el individuo, e_4 , es más consistente desde un punto de vista cuantitativo que cualitativo, e igualmente sucede para el conjunto de individuos; (ii) desde el punto de vista cuantitativo todos los individuos son absolutamente consistentes, sin embargo, cualitativamente son consistentes sólo los individuos e_2 y e_3 .

Referencias

- [1] P.P. Bonissone and K.S. Decker (1986). 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.
- [2] V. Cutello and J. Montero (1994). Fuzzy Rationality Measures. *Fuzzy Sets and Systems* 62:39-54.
- [3] F. Herrera, E. Herrera-Viedma and J.L. Verdegay (1995). Medidas Lingüísticas para el Consenso en Grupo. *Proc. del V Congreso Español sobre Tecnologías y Lógica Fuzzy*, Murcia, 309-314.
- [4] F. Herrera, E. Herrera-Viedma and J.L. Verdegay (1996). A Model of Consensus in Group Decision Making under Linguistic Assessments. *Fuzzy Sets and Systems* 78:73-87.
- [5] F. Herrera, E. Herrera-Viedma and J.L. Verdegay (1996). Direct Approach Processes in Group Decision Making Using Linguistic OWA Operators. *Fuzzy Sets and Systems* 79:175-190.
- [6] F. Herrera, E. Herrera-Viedma and J.L. Verdegay (1996). A Rational Consensus Model in Group Decision Making under Linguistic Assessments. Aceptado en *Fuzzy Sets and Systems*.
- [7] J. Montero (1987). Arrow's Theorem under Fuzzy Rationality. *Behavioural Science* 32:267-273.
- [8] S.A. Orlovsky (1978). Decision Making with a Fuzzy Preference Relation. *Fuzzy Sets and Systems* 1:155-167.
- [9] A.K. Sen (1977). Social Choice Theory: A Re-Examination. *Econometrica* 45:53-89.
- [10] R.R. Yager (1988). On Ordered Weighted Averaging Aggregation operators in Multicriteria Decision Making. *IEEE Transactions on Systems, Man, and Cybernetics* 18:183-190.
- [11] L. Zadeh (1983). A Computational Approach to Fuzzy Quantifiers in Natural Language. *Comps & Maths. with Appls.* 9:149-184.
- [12] H. J. Zimmermann (1985). *Fuzzy Sets Theory and Its Applications* (Kluwer-Nijhoff).

Subconjuntos difusos de un anillo. Ideales difusos

Jorge Jiménez
Dpto. de Matemáticas
Universidad de Oviedo
e-mail: meana@etsig.uniovi.es

Susana Montes
Dpto. de Matemáticas
Universidad de Oviedo
e-mail: smr@pinon.ccu.uniovi.es

Fermín Suárez
Dpto. de Matemáticas
Universidad de Oviedo
e-mail: fermin@etsig.uniovi.es

Resumen

El propósito de esta comunicación es estudiar los L -ideales difusos, generalización de los ideales difusos obtenida sustituyendo el intervalo $[0, 1]$ por un retículo como conjunto final en la definición de subconjunto difuso, línea iniciada por Goguen [6], es decir,

Un subconjunto difuso A de X es cualquier aplicación $A: X \rightarrow L$.

Como consecuencia de esta generalización, se estudian y relacionan los L -ideales primos, L -ideales fuertemente primos, L -ideales máximos y L -ideales finitos valorados. Por último se estudian dichos conjuntos con distintos referenciales. (Anillos locales, noetherianos, artinianos,...)

1 INTRODUCCIÓN

Desde que Zadeh [12] introdujo la definición de conjunto difuso, numerosos autores [1] [3], [4], [8], [10], [11], ..., han estudiado las estructuras algebraicas que pueden encontrarse dentro del referencial X . En este trabajo, se intenta profundizar en el estudio de las subestructuras difusas de estructuras nítidas, como serán los ideales difusos, definidos ya por Kumbhojkar y Bapat [9].

El trabajo está organizado como sigue:

La mayor parte de la segunda sección es una recopilación de resultados conocidos, necesarios para las siguientes secciones, que se pueden consultar en [4], [9].

En la tercera sección definimos y estudiamos los L -ideales difusos máximos, así como la caracterización de los anillos locales.

En las secciones cuarta y quinta, se consideran los L -ideales difusos fuertemente primos y los L -ideales

difusos finitos valorados para terminar el trabajo con el estudio de los L -ideales difusos sobre anillos artinianos y noetherianos, caracterizando los anillos artinianos a partir de sus L -ideales difusos.

2 L -IDEALES DIFUSOS Y L -IDEALES DIFUSOS PRIMOS

A lo largo de todo el trabajo consideraremos que A es un anillo conmutativo unitario con las operaciones binarias $+$ y $\cdot$, y L es un retículo con las operaciones binarias $\vee$ y $\wedge$ que denotan el supremo y el ínfimo respectivamente.

Definición 2.1 Sea L un retículo y sea A un anillo con unidad, se llama ideal difuso sobre el retículo L ó L -ideal difuso, a una aplicación $\tilde{I}: A \rightarrow L$ que verifica las siguientes condiciones:

$$\tilde{I}(x + y) \geq \tilde{I}(x) \wedge \tilde{I}(y) \quad (1)$$

$$\tilde{I}(-x) = \tilde{I}(x) \quad (2)$$

$$\tilde{I}(x \cdot y) \geq \tilde{I}(x) \vee \tilde{I}(y) \quad (3)$$

para cualesquiera x e y en A .

Evidentemente, todos los ideales nítidos del anillo A cumplen las tres condiciones anteriores, por lo que son un L -ideal difuso, y en consecuencia la definición 2.1. no es más que una extensión del concepto clásico.

La siguiente proposición proporciona una forma alternativa de comprobar que un L -conjunto difuso es un L -ideal difuso y por tanto una definición equivalente.

Proposición 2.2 Un subconjunto L -difuso $\tilde{I}$ del anillo A es un ideal difuso si y sólo si verifica las dos condiciones siguientes:

$$\tilde{I}(x - y) \geq \tilde{I}(x) \wedge \tilde{I}(y) \quad (4)$$

$$\tilde{I}(x \cdot y) \geq \tilde{I}(x) \vee \tilde{I}(y) \quad (5)$$

para cualesquiera x e y en A .

Se ven ahora dos propiedades que verifica todo ideal difuso sobre L .

Proposición 2.3 Sea $\tilde{I} : A \rightarrow L$ un L -subconjunto difuso. Si $\tilde{I}$ es un L -ideal difuso y x e y elementos de A , entonces se verifican las siguientes propiedades:

$$\tilde{I}(0) \geq \tilde{I}(x) \geq \tilde{I}(1) \quad (6)$$

$$\text{Si } \tilde{I}(x - y) = \tilde{I}(0), \text{ entonces } \tilde{I}(x) = \tilde{I}(y) \quad (7)$$

$$\text{Si } x \text{ es un invertible, entonces } \tilde{I}(x) = \tilde{I}(1) \quad (8)$$

Como consecuencia de esta proposición, se obtiene el siguiente

Corolario 2.4 Dado un L -ideal difuso $\tilde{I}$ en el anillo A , el completado del conjunto nítido $\text{Im}(\tilde{I})$ es un subretículo de L con máximo y mínimo, siendo éstos $\tilde{I}(0)$ e $\tilde{I}(1)$ respectivamente.

Demostración

Basta tener en cuenta que el completado de $\text{Im}(\tilde{I})$ es

$$\widetilde{\text{Im}(\tilde{I})} = \bigcap_{\text{Im}(\tilde{I}) \subseteq L_i} L_i \text{ donde } L_i \text{ son subretículos de } L.$$

Unos conjuntos nítidos, obtenidos a partir de un subconjunto difuso, especialmente importantes son los α -cortes [7], definidos por:

α -corte del conjunto difuso $\tilde{J}$ de A : $\tilde{J}_\alpha = \{x \in A / \tilde{J}(x) \geq \alpha\}$

α -corte fuerte del conjunto difuso $\tilde{J}$ de A : $\tilde{J}_\alpha = \{x \in A / \tilde{J}(x) > \alpha\}$

Al considerar los α -cortes de un L -ideal difuso se obtiene:

Proposición 2.5 Sea $\tilde{I}$ un L -ideal difuso del anillo A .

i) El α -corte que denotaremos por $\tilde{I}^\alpha$ es un ideal nítido de A .

ii) Si $\text{Im}(\tilde{I})$ es un conjunto totalmente ordenado, el α -corte fuerte que denotaremos por $\tilde{I}^\alpha$ es un ideal nítido de A .

Una vez que se han definido los ideales difusos sobre un retículo, y se han estudiado algunas de sus propiedades, se van a definir ciertos ideales difusos a partir de uno, las clases difusas de equivalencia.

Definición 2.6 Sea $\tilde{I}$ un L -ideal difuso del anillo A . Para cualquier $x \in A$, se define la clase difusa de equivalencia x de $\tilde{I}$, como el L -conjunto difuso $x + \tilde{I} : A \rightarrow L$ definido por

$$(x + \tilde{I})(y) = \tilde{I}(y - x), \forall y \in A$$

Así se forma el conjunto de todas las clases difusas de equivalencia de un L -ideal difuso $\tilde{I}$ del anillo A :

$$A/\tilde{I} = \{(x + \tilde{I}) / x \in A\}$$

Se ha denotado a este conjunto como $A/\tilde{I}$, por analogía con la notación algebraica de anillo cocientes, esto tiene total sentido si se considera la siguiente

Proposición 2.7 Dado un L -ideal difuso $\tilde{I}$ del anillo A , el conjunto $A/\tilde{I}$ formado por todas las clases de equivalencia de $\tilde{I}$, con las operaciones $\oplus$ y $\odot$ definidas como

$$(x + \tilde{I}) \oplus (y + \tilde{I}) = ((x + y) + \tilde{I})$$

$$(x + \tilde{I}) \odot (y + \tilde{I}) = ((x \cdot y) + \tilde{I})$$

para cualesquiera $x, y \in A$, es un anillo.

Como se ha visto en la proposición 2.5, todo α -corte de un L -ideal difuso es un ideal nítido, por lo que tendría sentido considerar el anillo cociente $A/\tilde{I}^\alpha$. En el caso en el que α tome el valor $\tilde{I}(0)$, se obtiene un α -corte especialmente importante, y al que se le da la notación diferenciadora de $A_{\tilde{I}}$ formado por los elementos

$$A_{\tilde{I}} = \{x \in A / \tilde{I}(x) = \tilde{I}(0)\}$$

Entre los dos anillos cocientes $A/\tilde{I}$ y $A/A_{\tilde{I}}$ existe una clara correspondencia como muestra el siguiente teorema:

Teorema 2.8 Dado un L -ideal difuso $\tilde{I}$ del anillo A , y dados los anillos $A/\tilde{I}$ y $A/A_{\tilde{I}}$, existe entre ellos un isomorfismo definido por:

$$\begin{aligned} A/\tilde{I} &\longrightarrow A/A_{\tilde{I}} \\ (x + \tilde{I}) &\longrightarrow [x] \end{aligned}$$

para cualquier $x \in A$.

Teorema 2.9 Sean A y A' dos anillos, si $f : A \rightarrow A'$ es un epimorfismo, entonces existe una correspondencia inyectiva preservando el orden entre los L -ideales difusos de A' y los de A que son constantes sobre el núcleo de f . Además si $\tilde{I} : A \rightarrow L$ y $\tilde{I}' : A' \rightarrow L$ son L -ideales difusos de A y A' respectivamente (con $\tilde{I}$ constante sobre el núcleo de f), entonces $f(A_{\tilde{I}}) = A'_{f(\tilde{I})}$ y $f^{-1}(A'_{\tilde{I}'}) = A_{f^{-1}(\tilde{I}')}.$

Definición 2.10 Dado un L -ideal difuso $\tilde{I}$ del anillo A , se dice que es primo (ideal difuso primo sobre un retículo o L -ideal difuso primo) si se verifica

$$\forall x, y \in A \text{ tales que } \tilde{I}(x \cdot y) = \tilde{I}(0), \text{ entonces } \tilde{I}(x) = \tilde{I}(0) \text{ ó } \tilde{I}(y) = \tilde{I}(0) \quad (9)$$

Es evidente que si I es un ideal nítido primo, entonces verifica la condición (9), y por tanto la definición anterior no es más que una generalización del mismo concepto en nítidos.

En el siguiente teorema se dan una serie de equivalencias que permitirán demostrar que un L -ideal difuso es primo, sin necesidad de utilizar la definición 2.10.

Teorema 2.11 Sea $\tilde{I} : A \rightarrow L$ un L -ideal difuso, entonces las siguientes proposiciones son equivalentes:

- i) $\tilde{I}$ es un L -ideal difuso primo,
- ii) $A_{\tilde{I}}$ es un ideal primo,
- iii) $A/A_{\tilde{I}}$ es íntegro,
- iv) $A/\tilde{I}$ es íntegro.

Corolario 2.12 La correspondencia biunívoca del teorema 2.9 conserva L -ideales primos.

Definición 2.13 Dado un L -ideal difuso $\tilde{I} : A \rightarrow L$, se dice que es trivial si $\tilde{I}(x) = \tilde{I}(1), \forall x \in A$, o si $\tilde{I}(1) = \tilde{I}(0)$.

Proposición 2.14 Un L -ideal difuso $\tilde{I}$ es trivial si y sólo si el ideal nítido $A_{\tilde{I}}$ es trivial.

3 IDEALES DIFUSOS MAXIMALES SOBRE UN RETÍCULO

Definición 3.1 Dado $\tilde{I}$ un L -ideal difuso en el anillo A , se dice que es maximal si el conjunto nítido $A_{\tilde{I}}$ es un ideal nítido maximal.

Lema 3.2 Dado $\tilde{I}$ un L -ideal difuso en el anillo A , si $\tilde{I}$ es maximal, entonces sólo puede tomar un valor distinto de $\tilde{I}(0)$, en cualquier elemento del anillo A , o sea, es la función característica de $A_{\tilde{I}}$.

Formas equivalentes de comprobar que un L -ideal difuso es maximal, las proporciona el siguiente teorema.

Teorema 3.3 Dado $\tilde{I}$ un L -ideal difuso en el anillo A , las siguientes afirmaciones son equivalentes:

- i) $\tilde{I}$ es un L -ideal difuso maximal,
- ii) $A/A_{\tilde{I}}$ es un cuerpo,
- iii) $A/\tilde{I}$ es un cuerpo.

Lema 3.4 Dado $\tilde{I}$ un L -ideal difuso maximal en el anillo A , el α -corte de $\tilde{I}$, $\tilde{I}^\alpha$, para cualquier $\alpha \in L$, es ó bien el vacío, ó bien el ideal nítido maximal, ó bien A .

El siguiente lema, además de ser útil en la demostración de la proposición 3.6, establece una clara relación entre los L -ideales difusos primos definidos en 2.10 y los L -ideales difusos maximales definidos en 3.1.

Lema 3.5 Dado $\tilde{I}$ un L -ideal difuso en el anillo A , si $\tilde{I}$ es maximal, entonces $\tilde{I}$ es primo.

Demostración

Es consecuencia directa de 2.11.

Corolario 3.6 La aplicación considerada en el teorema 2.9 preserva los L -ideales difusos maximales.

Los ideales maximales son fundamentales en toda el álgebra conmutativa. El teorema que sigue y sus corolarios aseguran que siempre se tiene un número suficiente de ellos.

Teorema 3.7 Cada anillo $A \neq 0$ tiene, por lo menos, un L -ideal difuso maximal.

La demostración de este teorema es trivial sin más que aplicar un resultado análogo para ideales nítidos. Ver [2].

Proposición 3.8 Si $\text{Im}(\tilde{I})$ es un retículo totalmente ordenado e $\tilde{I} \neq (1)$ es un L -ideal difuso de A , existe un L -ideal difuso maximal $\tilde{M}$ de A que contiene a $\tilde{I}$.

Demostración

Un L -ideal difuso maximal que contiene a $\tilde{I}$, no es más que el conjunto difuso $\tilde{M}$ definido por

$$\tilde{M}(x) = \begin{cases} \tilde{I}(0) & \text{si } x \in M \\ \tilde{I}(1) & \text{si } x \notin M \end{cases}$$

donde M es el ideal maximal nítido que contiene al ideal $\bigcup_{\alpha > \tilde{I}(1)} \tilde{I}^\alpha$.

Observación. Si el anillo A es local, entonces también se verifica la proposición anterior, bastaría tomar $M = \{\text{ideal maximal de } A\}$.

Destacar que es importante en la demostración que $\text{Im}(\tilde{I})$ sea totalmente ordenado como se ve en el siguiente ejemplo:

Ejemplo 3.9 Sea $L = \{1, 2, 3, 6\}$ el conjunto de los divisores del 6 con la relación de orden "divide a". El siguiente L -ideal difuso de $\mathbb{Z}$ no está contenido en ningún maximal:

$$\tilde{I}(x) = \begin{cases} 6 & \text{si } x \in 6 \cdot \mathbb{Z} \\ 3 & \text{si } x \in 3 \cdot \mathbb{Z} - 6 \cdot \mathbb{Z} \\ 2 & \text{si } x \in 2 \cdot \mathbb{Z} - 6 \cdot \mathbb{Z} \\ 1 & \text{en otro caso} \end{cases} = m.c.d.(x, 6)$$

Este ejemplo da lugar a la definición de L -ideal difuso debilmente máximo que abordaremos en un trabajo posterior.

Definición 3.10 Sea $\tilde{I}$ un L -ideal difuso, y sea $L_1 = \{\text{los elementos minimales de } (\text{Im}(\tilde{I}) - \{\tilde{I}(1)\})\}$ se dice que $\tilde{I}$ es debilmente maximal si se verifica que $\bigvee_{\alpha \in L_1} \alpha = \tilde{I}(0)$ e $\tilde{I}^\alpha$ es un ideal maximal para todo $\alpha \in L_1$.

Esta es una generalización de L -ideal difuso maximal y que verifica que todo L -ideal difuso esta contenido en un L -ideal difuso debilmente maximal.

Corolario 3.11 Cada elemento de A que no es una unidad está contenido en un L -ideal difuso maximal.

En la siguiente proposición se van a relacionar los L -ideales difusos maximales con los anillos locales.

Proposición 3.12 Las siguientes afirmaciones son equivalentes:

- (a) Un anillo A es local.
- (b) El $\tilde{I}(0)$ -corte de $\tilde{I}$ coincide, para todo $\tilde{I}$, L -ideal difuso maximal.
- (c) Para todo retículo L , todo L -ideal difuso está contenido en algún L -ideal maximal.

Demostración

La demostración de la primera equivalencia es inmediata, mientras que la segunda equivalencia es consecuencia de la observación de 3.8 y de una construcción similar al ejemplo 3.9.

Proposición 3.13 i) Sea A un anillo, y sea $\tilde{M} : A \rightarrow L$ un L -ideal difuso de A , tal que $\forall x \notin A_{\tilde{M}}$, x es invertible. Entonces A es local y $\tilde{M}$ es un L -ideal difuso maximal de A .
 ii) Sea A un anillo y $\tilde{M}$ un L -ideal difuso maximal de A tal que cada elemento de $1 + A_{\tilde{M}}$ es invertible en A . Entonces A es un anillo local.

4 L -IDEALES DIFUSOS FUERTEMENTE PRIMOS

Definición 4.1 Un L -ideal difuso $\tilde{I} : A \rightarrow L$ en A se dice fuertemente primo si $\tilde{I}(x \cdot y) = \tilde{I}(x)$ o $\tilde{I}(x \cdot y) = \tilde{I}(y)$, $\forall x, y \in A$.

La siguiente proposición proporciona una definición equivalente:

Proposición 4.2 Sea $\tilde{I}$ un L -ideal difuso en A , las siguientes afirmaciones son equivalentes:

- i) $\tilde{I}$ es fuertemente primo,
- ii) $\tilde{I}(x \cdot y) = \tilde{I}(x) \wedge \tilde{I}(y)$, $\forall x, y \in A$ e $\text{Im}(\tilde{I})$ es un retículo totalmente ordenado,
- iii) $\tilde{I}^\alpha$ es un ideal nítido primo, $\forall \alpha \leq \tilde{I}(0)$.

Conviene aclarar que carece de sentido el hablar de L -ideales difusos fuertemente maximales, pues estos deberían ser, por analogía con los primos, aquellos que tienen todos los α -cortes maximales, pero en este caso, lo verifican todos los maximales difusos. La siguiente proposición nos relaciona los L -ideales difusos fuertemente primos y los maximales.

Proposición 4.3 Sea $\tilde{I}$ un L -ideal difuso en el anillo A . Si $\tilde{I}$ es maximal, entonces $\tilde{I}$ es fuertemente primo.

El recíproco de esta proposición no es en general cierto, a no ser que se impongan algunas restricciones, como se ve en el siguiente corolario.

Corolario 4.4 Sea A un anillo de dimensión uno, y sea $\tilde{I}$ un L -ideal difuso fuertemente primo de A . Si $\tilde{I}$ verifica que el ideal nítido $A_{\tilde{I}} \neq (0)$, entonces $\tilde{I}$ es maximal.

5 IDEALES DIFUSOS FINITO VALORADOS SOBRE UN RETÍCULO

Definición 5.1 Se dice que $\tilde{I} : A \rightarrow L$ es un L -ideal difuso n -valorado si $\tilde{I}(A)$ es un conjunto finito de n elementos. En general llamaremos L -ideal difuso finito valorado a cualquier L -ideal difuso que cumpla la condición anterior para algún n .

Lema 5.2 $\tilde{I}$ es un L -ideal difuso 2-valorado sobre A , si y solamente si, es la función característica de un ideal de A .

En particular los L -ideales difusos maximales son 2-valorados.

Proposición 5.3 Si A es un anillo simple y $\tilde{I} : A \rightarrow L$ es un L -ideal difuso, entonces $\tilde{I}(x) = \tilde{I}(1)$, $\forall 0 \neq x \in A$, es decir, $\tilde{I}$ es la función característica de A o de $(0) = \{0\}$.

Proposición 5.4 Si $\tilde{I} : A \rightarrow L$ es un L -ideal difuso primo, $A_{\tilde{I}} \neq \{0\}$ y A es alguno de estos anillos:
 o bien un dominio de ideales principales.
 o bien booleano.
 o bien artinianiano
 entonces $\tilde{I}$ es la función característica de $A_{\tilde{I}}$.

Demostración

consecuencia inmediata de 3.2

Proposición 5.5 Sean $\tilde{I} : A \rightarrow L$ y $\tilde{I}' : A \rightarrow L$ dos L -ideales difusos de A . El subconjunto difuso de A formado por la intersección de los dos ideales $\tilde{I} \cap \tilde{I}'$ es un nuevo L -ideal difuso de A .

Demostración

Basta considerar la desigualdad:

$$(a \vee b) \wedge c \geq (a \wedge c) \vee (b \wedge c)$$

que se verifica para todo $a, b, c \in L$.

En la proposición anterior se está considerando la definición clásica de intersección dada por [5]

$$\forall x \in A, \quad \tilde{I} \cap \tilde{I}'(x) = \tilde{I}(x) \wedge \tilde{I}'(x)$$

En particular, utilizando la proposición 5.5, se obtiene el siguiente resultado:

Proposición 5.6 Si $\tilde{D} : A \rightarrow L$ es un L -subconjunto difuso de A y L es un retículo completo inferiormente, entonces la intersección de todos los L -ideales difusos que contienen a $\tilde{D}$ existe, y por lo tanto existe el L -ideal difuso más pequeño que contiene a $\tilde{D}$.

Demostración

basta considerar la misma desigualdad anterior y que retículo completo inferiormente, quiere decir que todo subconjunto de L tiene ínfimo.

Definición 5.7 Sea $\tilde{D} : A \rightarrow L$ un L -subconjunto difuso de A . El L -ideal difuso más pequeño que contiene a $\tilde{D}$ se llama L -ideal difuso generado por $\tilde{D}$ y se denota por $\langle \tilde{D} \rangle$.

Se establece a continuación un lema necesario en la demostración del siguiente teorema.

Lema 5.8 Si $\tilde{I} : A \rightarrow L$ es un L -ideal difuso de A , entonces se verifica que:

Si $\tilde{I}(x) = a \Rightarrow \tilde{I}^a$ es el menor α -corte de $\tilde{I}$ que contiene a x .

Teorema 5.9 Sea L un retículo distributivo. Un L -ideal difuso $\tilde{I} : A \rightarrow L$ es finito valorado si y sólo si está generado por un L -conjunto difuso finitamente valorado $\tilde{D} : A \rightarrow L$.

Demostración

($\Rightarrow$) La implicación hacia la derecha es trivial.

($\Leftarrow$) Construimos primeramente el L -ideal generado por $\tilde{D} : A \rightarrow L$.

Sea $L' = \text{Im}(\tilde{D})$, el completado de $\text{Im}(\tilde{D})$. Se tiene que L' es un subretículo finito de L .

Sea a un elemento de L' , se denotará por L_a a

$$L_a = \{b \in L' \text{ tales que } b \geq a\}$$

Sea $M_a = \tilde{D}^{-1}(L_a)$ y sea $I_a = \langle M_a \rangle$.

Se tiene que si $a \leq b$ entonces $I_b \subseteq I_a$.

Se define el siguiente L -conjunto difuso que veremos es un L -ideal difuso.

$$\tilde{I}(x) = \bigvee_{x \in I_a} a$$

Este L -conjunto difuso es un L -ideal difuso, pues verifica las condiciones de la proposición 2.2 como se sigue de:

$$\tilde{I}(x - y) = \bigvee_{x-y \in I_a} a \quad (10)$$

$$\tilde{I}(x) \wedge \tilde{I}(y) = \left(\bigvee_{x \in I_b} b \right) \wedge \left(\bigvee_{y \in I_c} c \right) =$$

$$\bigvee_{\substack{x \in I_b \\ y \in I_c}} (b \wedge c) \quad (11)$$

Donde la igualdad (11) se obtiene sin más que aplicar que L es distributivo.

Se tiene que $\begin{cases} x \in I_b \subseteq I_{b \wedge c} \\ y \in I_c \subseteq I_{b \wedge c} \end{cases} \Rightarrow x - y \in I_{b \wedge c}$, luego se está tomando supremo en (11) sobre un conjunto con menos elementos que en (10), luego el supremo de (11) será menor o igual que el de (10)

$$\bigvee_{\substack{x \in I_b \\ y \in I_c}} (b \wedge c) \leq \bigvee_{x-y \in I_a} a$$

luego $\tilde{I}(x - y) \geq \tilde{I}(x) \wedge \tilde{I}(y)$.

Veamos que se verifica que $\tilde{I}(x \cdot y) \geq \tilde{I}(x) \vee \tilde{I}(y)$:

$$\tilde{I}(x \cdot y) = \bigvee_{x \cdot y \in I_a} a \quad (12)$$

$$\begin{aligned} \tilde{I}(x) \vee \tilde{I}(y) &= \left(\bigvee_{x \in I_b} b \right) \vee \left(\bigvee_{y \in I_c} c \right) = \\ &= \bigvee_{\substack{x \in I_a \\ \text{ó } y \in I_a}} a \end{aligned} \quad (13)$$

Donde la igualdad (13), se obtiene sin más que aplicar la propiedad asociativa.

Como antes, si $x \in I_a$ entonces $x \cdot y \in I_a$ y si $y \in I_a$, entonces $x \cdot y \in I_a$, luego tomamos supremo sobre un conjunto más pequeño.

Luego $\tilde{I}(x \cdot y) \geq \tilde{I}(x) \vee \tilde{I}(y)$

Veamos que contiene a $\tilde{D}$.

Sea $x \in A$ entonces $\tilde{D}(x) = a$, por lo tanto $x \in M_a \Rightarrow x \in I_a \Rightarrow \tilde{I}(x) \geq a = \tilde{D}(x) \Rightarrow \tilde{I} \supseteq \tilde{D}$.

Por lo tanto $\tilde{I}$ es un L -ideal difuso que contiene a $\tilde{D}$.

Y por último, veamos que es el menor:

Sea $\tilde{J} : A \rightarrow L$ un L -ideal difuso en A que contiene a $\tilde{D}$, por lo tanto

$$\forall x \in A, \quad \tilde{D}(x) \leq \tilde{J}(x)$$

luego $M_a \subseteq \tilde{J}^a$, con lo que $I_a \subseteq \tilde{J}^a$. Entonces se tiene que si $x \in I_a$, $x \in \tilde{J}^a$, y aplicando el lema 5.8 se tiene que $\tilde{J}(x) \geq a$ para todo $a \in L'$ verificando que $x \in I_a$, lo que implica que

$$\bigvee_{x \in I_a} a \leq \tilde{J}(x) \Rightarrow$$

$$\tilde{I}(x) \leq \tilde{J}(x)$$

con lo que el teorema queda demostrado. $\square$

6 ANILLOS ARTINIANOS Y NOETHERIANOS

Vamos a analizar ahora el caso en el que el referencial es un anillo noetheriano y el caso de anillo artiniano, pero primero tendremos que dar una definición:

Definición 6.1 Se dice que un conjunto parcialmente ordenado $\mathcal{P}$ es de longitud finita si la longitud de todos los caminos entre dos elementos cualesquiera de $\mathcal{P}$ es finita.

Teorema 6.2 Dado un anillo A , las siguientes afirmaciones son equivalentes:

- i) A es artiniano,
- ii) todos y cada uno de sus L -ideales difusos $\tilde{I}$ verifican que $\text{Im}(\tilde{I})$ es de longitud finita,

Puesto que todo anillo artiniano es noetheriano, entonces se sigue de este teorema que todo anillo que tenga todos sus L -ideales difusos $\tilde{I}$, con $\text{Im}(\tilde{I})$ de longitud finita, es noetheriano. Sin embargo, el recíproco no es cierto, como muestra el siguiente ejemplo.

Ejemplo 6.3 Si se considera el anillo noetheriano $\mathbb{Z}$ formado por los números enteros, y la cadena de ideales nítidos $I_1 \supseteq I_2 \supseteq \dots \supseteq I_k \supseteq \dots$ con $I_n = 2^n \mathbb{Z}$. Entonces se puede definir el siguiente L -ideal difuso en el anillo $\mathbb{Z}$:

$$\tilde{I}(x) = \begin{cases} 1 & \text{si } x = 0 \\ \frac{n}{n+1} & \text{si } x \in I_n - I_{n+1} \\ 0 & \text{si } x \in \mathbb{Z} - I_1 \end{cases}$$

que evidentemente no es finito valorado, y tampoco $\text{Im}(\tilde{I})$ es de longitud finita.

7 CONCLUSIONES

En este trabajo se ha conseguido extender al marco de los L -conjuntos difusos, algunos importantes resultados de álgebra difusa, utilizando en todo momento subestructuras difusas de una estructura nítida como es un anillo. Esto no es más que un primer estudio de estas estructuras, que se espera seguir completando y profundizando, intentando poder extenderlo a otras estructuras algebraicas como los A -módulos, A -álgebras, etc, e incluso a estructuras difusas, es decir, desarrollar un estudio paralelo para anillos difusos, A -módulos difusos, A -álgebras difusas, etc...; así como estudiar los L -ideales difusos debilmente maximales cuya definición hemos adelantado en este trabajo.

Agradecimientos

Agradecer la colaboración prestada al Prof. José Manuel Aroca en la realización de este trabajo.

Referencias

- [1] J.M. Anthony and H. Sherwood (1979). Fuzzy groups redefined. *J. Math. Anal. Appl.* **69**:124-130.
- [2] M.F. Atiyah and L.G. MacDonald (1980). Introducción al álgebra conmutativa. Editorial Reverté, S.A.
- [3] P. Bhattacharya (1986). Fuzzy subgroups II. *Inform. Sci.* **38**:293-297.
- [4] V.N. Dixit, R. Kumar and N. Ajmal (1991). Fuzzy ideals and fuzzy prime ideals of a ring. *Fuzzy Sets and Systems* **44**:127-138.
- [5] D. Dubois and H. Prade (1980). Fuzzy Sets and Systems. Theory and Applications. Academic Press, Inc.
- [6] J.A. Goguen (1967). L -fuzzy sets. *J. Math. Anal. Appl.* **18**:145-174.
- [7] G.J. Klir and T.A. Folger (1988). Fuzzy Sets, Uncertainty, and Information. Prentice-Hall International (UK) Limited
- [8] R. Kumar (1991). Fuzzy nil radicals and fuzzy primary ideals. *Fuzzy Sets and Systems* **43**:81-93.
- [9] H.V. Kumbhojkar and M.S. Bapat (1990). Not-so-fuzzy fuzzy ideals. *Fuzzy Sets and Systems* **37**:237-243.
- [10] D.S. Malik (1991). Fuzzy relations on rings and groups. *Fuzzy Sets and Systems* **43**:117-123.
- [11] U. M. Swamy and D.V. Raju (1991). Algebraic fuzzy systems. *Fuzzy Sets and Systems* **41**:187-194.
- [12] L.A. Zadeh (1965). Fuzzy sets. *Information and Control* **8**:338-353.

XFL: UN LENGUAJE PARA LA DEFINICIÓN DE SISTEMAS DIFUSOS

Diego R. López
drlopez@cica.es

Santiago Sánchez-Solano
santiago@cnm.us.es

Ángel Barriga
barriga@cnm.us.es

Resumen

En esta comunicación describimos las principales características de XFL, un lenguaje para la especificación de sistemas basados en lógica difusa que permite realizar definiciones independientemente de las técnicas particulares empleadas para las operaciones difusas. XFL permite también el empleo de bases de reglas jerárquicas para la definición de sistemas arbitrariamente complejos. Este lenguaje ha permitido el desarrollo de varias herramientas de desarrollo (hardware y software), de las que se presenta una breve descripción.

Palabras Clave: lógica fuzzy, especificación, lenguajes, herramientas de desarrollo.

1 INTRODUCCIÓN

La mayor parte de los sistemas actuales para el desarrollo de aplicaciones basadas en lógica difusa son capaces de manejar un conjunto limitado de operaciones. Por operaciones difusas entendemos aquí las funciones que definen el tratamiento de la información en el sistema (funciones de implicación, conectivas, métodos de defuzzificación, etc). Este hecho implica que la definición de un sistema difuso con alguna de estas herramientas está, de alguna manera, ligado a las operaciones implementadas por la herramienta en cuestión y ello introduce, básicamente, tres clases de limitaciones. En primer lugar, surgen problemas en cuanto a la comparabilidad de sistemas similares realizados con herramientas diferentes. En segundo, los detalles orientados a la implementación concreta pueden ocultar la estructura original del sistema en desarrollo. Por último, esta situación hace difícil la comparación entre las diferentes alternativas para cada operación (debe recurrirse a métodos ad-hoc, [3]) y la evaluación de nuevos formalismos.

Por otra parte, las herramientas de desarrollo actuales no ofrecen facilidades a la hora de trabajar con sistemas de gran complejidad. Así, por ejemplo, no suelen permitir la

agregación jerárquica de bases de reglas [2] o que los antecedentes de una regla puedan combinarse mediante conectivas en estructuras de una complejidad arbitraria.

Al plantearnos la realización de una nueva versión del entorno de desarrollo de sistemas difusos Xfuzzy [8], uno de nuestros primeros objetivos fue basarla en un método de especificación que obviara los problemas descritos anteriormente. El método de especificación debía ser independiente de las operaciones concretas empleadas en cada caso. Aún más, el empleo de cualquiera de estas operaciones (o combinación de ellas) debería ser configurable para las herramientas que emplearan este método. Por otro lado, no debía imponer limitación alguna en cuanto a la complejidad del sistema especificado, tanto en la estructura de las bases de reglas que lo compusieran, como en la definición de las mismas reglas. Finalmente, el método de especificación debía demostrar su validez proporcionando herramientas flexibles y con capacidad de producir implementaciones rápidas y fáciles de integrar con otros módulos (no necesariamente basados en lógica difusa) para conformar un sistema completo. Este artículo presenta los resultados de este esfuerzo: el lenguaje XFL y las herramientas basadas en él.

2 EL LENGUAJE XFL

XFL es un lenguaje para definición de sistemas basados en lógica difusa que facilita la descripción, verificación y síntesis de módulos difusos, de acuerdo con los objetivos descritos en el apartado anterior. Constituye la base de una serie de herramientas de desarrollo de sistemas difusos en las que estamos trabajando en la actualidad.

La definición de un sistema difuso usando XFL consta de dos partes fundamentales: las declaraciones de los tipos y la especificación del (o de los) módulo(s) que compone(n) el sistema. La primera incluye la definición de las clases de variables sobre las que va a operar el sistema, incluyendo sus universos de discurso y las funciones de pertenencia asociadas a cada uno de los conjuntos difusos sobre los que se apoyan. En la segunda se define la estructura y contenido de la base de reglas que definen el comportamiento del sistema.

XFL no impone ninguna restricción sobre la complejidad de las bases de reglas que describen un sistema difuso. Esto es cierto tanto en la estructura de una regla individual como en lo relativo al establecimiento de relaciones entre bases de reglas dentro de un mismo sistema. Una regla individual es una combinación arbitraria de predicados relativos a las funciones de pertenencia de los antecedentes, utilizando las conectivas *and*, *or* y *not*. Estos predicados pueden agruparse mediante el uso de paréntesis y las reglas tradicionales de precedencia, asociatividad y distributividad son aplicadas, garantizando así una interpretación "natural" de las reglas. Si varias reglas son aplicables a un consecuente determinado en una base de reglas, los resultados de cada una de ellas son agregados mediante el uso de una conectiva implícita, *c*, y el valor de la función de pertenencia de cada una de las salidas se obtiene componiendo los valores de las funciones de pertenencia individuales mediante la conectiva *also* [7].

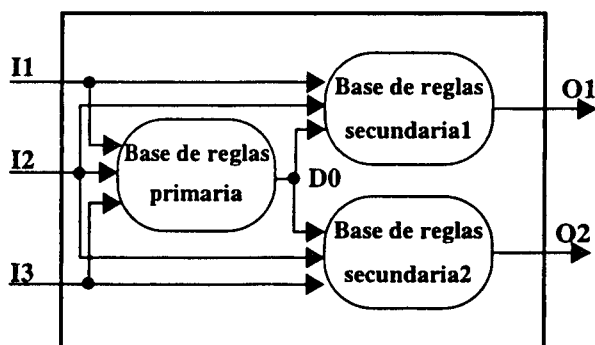


Figura 1: Modelo para una base de reglas jerárquica mediante composiciones XFL

A la hora de definir una estructura compleja de la base de reglas, XFL permite dos modos de composición de las mismas. El modo paralelo se limita a partir una base de reglas determinada en varios conjuntos (que pueden compartir entradas comunes), de manera que las salidas son independientes. No implica ninguna relación entre las bases de reglas compuestas. Este modo de composición puede resultar útil a la hora de integrar varias bases de reglas ya existentes en otra con mayor número de entradas y/o salidas o cuando se pretenda dotar de mayor simplicidad al formato de las bases de reglas de un sistema complejo. El modo serie supone que al menos una de las salidas de la primera base de reglas actúa como entrada de la segunda. Las salidas de la primera base de reglas determinan el grado de soporte de cada función de pertenencia en las entradas correspondientes de la segunda base de reglas, permitiendo así el establecimiento de bases de reglas jerarquizadas.

Los dos modos de composición pueden ser, de la misma manera que en el caso de una regla individual, combinados de manera arbitraria para definir la estructura de la base de reglas del sistema. Así, por ejemplo, una base de reglas compuesta en serie con otras dos compuestas en paralelo entre sí se corresponde con un modelo en el que una base de reglas primaria controla la activación de las reglas en las dos bases secundarias, que son las que proporcionan la salida del sistema (Figura 1).

Aunque una especificación XFL es independiente del método de defuzzificación empleado, ciertas características particulares del lenguaje están orientadas a facilitar el empleo de las técnicas más habituales. El lenguaje proporciona construcciones para facilitar el uso de dos paradigmas en los que basar los métodos de defuzzificación y, como se verá más adelante, la definición externa de métodos de defuzzificación para las herramientas que usan XFL se basa en ellos. Estos paradigmas están basados en la estructura básica de los métodos descritos en la literatura [4],[5],[6]. El primero consiste en realizar una exploración a través del universo de discurso de la variable de salida, utilizando los valores de cada punto y el valor de la función de pertenencia agregada en él. Dentro de esta clase incluimos el método del Centro de Gravedad, el de la Media de los Máximos, etc. En el segundo la exploración se realiza sobre el "universo lingüístico" de la variable, es decir, sobre los conjuntos difusos definidos para ella. Pueden emplearse el grado de activación de cada función de pertenencia y los parámetros que las definen. Dentro de esta clase incluimos la mayoría de los llamados métodos simplificados, como el de la Altura o los métodos de calidad.

3 DEFINICIÓN DE TIPOS

XFL soporta un conjunto de tipos predefinidos de funciones de pertenencia que incluye a la gran mayoría de las funciones descritas en la literatura y también proporciona un método de definición de funciones lineales a tramos. La característica más importante de la manera en que se definen los universos de discurso y funciones de pertenencia (lo que llamamos definición de tipos) en XFL es la incorporación de mecanismos de herencia. La intención es simplificar la especificación de sistemas complejos y facilitar la reutilización de módulos previamente escritos. La definición de un tipo en XFL tiene el siguiente formato:

```

type identificador tipoBase {
    funcionDePertenencia
    funcionDePertenencia ... }
  
```

donde tipoBase contiene una referencia al tipo (o tipos) del que se deriva el tipo definido. Esta referencia puede ser a un tipo predefinido o a tipos cuya definición se ha realizado con anterioridad en la especificación.

XFL proporciona dos tipos predefinidos: *integer* y *real*. Si el tipo que se define es un tipo derivado, tipoBase puede contener una lista de tipos previamente definidos, cuya única restricción es que deben derivar en última instancia del mismo tipo predefinido. Un tipo derivado hereda automáticamente todas las funciones de pertenencia definidas para su(s) antecesor(es), con una regla de precedencia para el caso de que una misma función aparezca en más de uno de ellos: la definición correspondiente al último elemento de la lista es la que toma efecto.

La definición de una nueva función de pertenencia (o la sobrecarga de una ya definida para un tipo derivado) corresponde a una estructura de la forma:

```
identificador claseDeFuncion ( listaDePuntos )
```

donde listaDePuntos contiene los valores de los puntos dentro del universo de discurso que son significativos para definir la función, a partir de la clase definida en claseDeFuncion. Estos valores no tienen otra restricción que el asegurar que la definición es coherente con la clase (p. ej., en el caso de una función triangular, que el vértice y los puntos que definen la base no sean todos el mismo) y que la función tome al menos un valor no nulo dentro del universo de discurso. La Tabla 1 muestra las clases de función de pertenencia que soporta XFL junto con los parámetros que las definen.

Tabla 1: Clases de funciones de pertenencia

Función	listaDePuntos
triangle	base, vértice, base
rectangle	base, base
trapezoid	b. mayor, b. menor, b. menor, b. mayor
bell, $e^{\left(\frac{(x-a)}{b}\right)^2}$	a, b
sigma, $\frac{1}{1 + e^{(x-a)/b}}$	a, b
delta	valor
points	punto:valor, punto:valor, ...

4 DEFINICIÓN DE LA BASE DE REGLAS

El elemento básico en la definición de la base de reglas que determina el comportamiento de un sistema difuso en XFL se denomina módulo. Cada módulo consta de una declara-

ción de las variables (de entrada y salida) que intervienen en él y de la especificación de su estructura. En términos generales, la definición de un módulo tiene la siguiente estructura:

```
identificador ( listaDeVariables ) {  
    estructuraDelModulo }
```

donde el identificador permite referenciar al módulo en la estructura de otro módulo más complejo. XFL requiere que, en cualquier especificación, exista un módulo llamado *system*, que es el que define el comportamiento global del sistema y cuyas variables de entrada/salida son las del sistema.

XFL permite la definición de bases de reglas de estructura arbitrariamente compleja. Esta característica se manifiesta en que la definición de la estructura de un módulo puede hacerse por medio de dos construcciones diferentes. Por un lado, pueden especificarse las reglas que definen el módulo y, por otro, el módulo puede definirse como formado a su vez por otros módulos compuestos entre sí. Para un módulo del que se define su estructura en términos de reglas estructuraDelModulo está constituido por:

```
rulebase { regla  
    ... }
```

Mientras que para un módulo cuya estructura se defina en varios módulos interconectados, estructuraDelModulo toma la siguiente forma:

```
components { referenciasAModulos }
```

Cada regla está constituida por la siguiente estructura:

```
if antecedente -> consecuente
```

El antecedente de una regla es una combinación arbitraria de expresiones que ligan alguna de las variables de entrada con alguna de las funciones de pertenencia definidas para su tipo. Cada una de estas expresiones se forma del siguiente modo:

```
identDeVariable is identDeFuncionDePertenencia
```

Los operadores que combinan las expresiones del antecedente de la regla se identifican mediante el carácter ~ para el operador *not*, el carácter & para el operador *and* y el carácter | para el operador *or*. Las reglas habituales de asociatividad, distributividad y precedencia son aplicables. Por supuesto, las expresiones pueden agruparse mediante el empleo de paréntesis.

El consecuente de una regla está constituido por una lista (separada por comas) de expresiones que ligan alguna de las variables de salida con alguna de las funciones de pertenencia definidas para su tipo.

Si un módulo se especifica en términos de composición de una serie de módulos componentes, la definición de su estructura está constituida por una combinación arbitraria de referencias a otros módulos combinados por medio de los operadores de composición paralela (símbolo `[]`) y de composición serie (símbolo `;`). Una referencia a un módulo se realiza incluyendo su identificador e instanciando sus variables de entrada/salida a través de referencias a las variables de entrada/salida del módulo padre o el empleo de referencias "dummy" (para el caso de composiciones serie).

La conexión entre variables del módulo padre y entre las variables de enlace empleadas para los módulos conectados en serie se realiza de manera implícita a través del nombre de la variable. Por supuesto, se aplican las reglas obvias de compatibilidad de tipos y modos (entrada/salida) entre las variables. Las variables de entrada del módulo padre pueden usarse libremente como variables de entrada de cualesquiera de los módulos que lo componen, independientemente de la estructura de que se trate. Por el contrario, por motivos obvios, cada variable de salida del módulo padre sólo puede tener una instancia en los módulos hijos. Todas las variables de un módulo hijo deben estar ligadas, bien a variables del módulo padre, bien por corresponder a variables de enlace entre módulos compuestos en modo serie.

Como ejemplo, la Figura 2 muestra la estructura de la base de reglas para el sistema considerado en la Figura 1.

```

primaria (tI1 ? pi1, tI2 ? pi2,
          tI3 ? pi3, tD0 ! ps) {
  rulebase {
    <regla1>
    . . .
    <reglaN>
  }
}
secundaria1 (tD0 ? s1s, tI1 ? si1,
             tI2 ? si2, tO1 ? o1) {
  rulebase {
    <regla1>
    . . .
  }
}
secundaria2 (tD0 ? s2s, tI2 ? si2,
             tI3 ? si3, tO2 ! o2) {
  rulebase {
    <regla1>
    . . .
  }
}

system (tI1 ? I1, tI2 ? I2, tI3 ? I3,
        tO1 ! O1, tO2 ! O2) {
  components {
    primaria (I1, I2, I3, D0) ;
    (secundaria1 (D0, I1, I2, O1) [])
    secundaria2 (D0, I2, I3, O2))
  }
}

```

Figura 2: Especificación XFL para la base de reglas de la Figura 1

5 DEFINICIÓN DE LAS OPERACIONES DIFUSAS

Aunque constituye un elemento que, intencionadamente, está fuera de la estructura de XFL, el mecanismo de definición de las operaciones difusas juega un papel fundamental a la hora de poder utilizar el lenguaje como base para herramientas de desarrollo de sistemas difusos. En esencia, cada herramienta que permite la traducción de una especificación XFL a un formato (hardware o software) apropiado para su implementación debe disponer de un método de configuración por el que se definan (en términos del lenguaje destino) las operaciones difusas susceptibles de ser empleadas. Una herramienta basada en XFL debe soportar mecanismos para identificar estas operaciones, de manera que utilizándolos pueda generarse una implementación completa del sistema en desarrollo. En este apartado presentamos el estado actual de las herramientas disponibles, para las que se ha diseñado un mecanismo de definición que consideramos bastante general y extensible a prácticamente la totalidad de los lenguajes destino posibles.

En las herramientas actualmente disponibles para XFL la selección de las operaciones difusas que se emplearán en la implementación de un sistema se realiza mediante una serie de directivas que, incluidas en el fichero fuente, identifican las operaciones definidas en un fichero de configuración. La directivas se muestran en la Tabla 2, indicándose su función y el símbolo de inicio de la sección correspondiente del fichero de configuración

Tabla 2: Directivas de configuración

Directiva	Función	Símbolo de inicio
#and	operador and	%Tnorm
#or	operador or	%Tconorm
#not	operador not	%Negation
#implication	función de implicación	%Implication
#composition	método de composición	(*) Se toma de las T-conormas
#also	mecanismo de agregación	(*) Se toma de las T-conormas
#defuzzification	método de defuzzificación	%Defuzzification

Dentro de cada una de las secciones, la definición de cada operación consta de dos partes: el identificador de la misma, que va en una línea aparte, y el cuerpo de su definición, que se incluye a continuación de la línea que contiene el identificador en líneas sucesivas que deben comenzar con un tabulador. Los cuerpos de las definiciones de las operaciones (salvo en el caso de los métodos de defuzzificación que describiremos más adelante) deben constar del cuerpo de una construcción nativa del lenguaje destino (función, procedimiento, macro, etc.) que retorne el resultado de la operación, usando unos parámetros predefinidos (*_a* y *_b*) para los operandos.

Para los métodos de defuzzificación la definición es más compleja: las líneas de definición van precedidas de símbolos que indican los requisitos del método en cuanto a las funciones de pertenencia a los que son aplicables y el punto dentro del proceso de defuzzificación en el que se aplican las operaciones. Se dispone de símbolos para especificar la inicialización de variables, la asignación del valor final de salida y bucles sobre el universo de discurso o lingüístico de las variables.

En cualquiera de las definiciones (ya sea de operaciones simples o de métodos de defuzzificación) se pueden incluir líneas con sentencias del lenguaje destino, utilizando la construcción *%proc* para definir procedimientos o funciones, y la construcción *%data* para definir estructuras de datos. La Figura 3 muestra un ejemplo de fichero de configuración para el traductor de XFL a C (XFC) en el que aparecen algunas de las características aquí descritas.

6 HERRAMIENTAS BASADAS EN XFL

El empleo de XFL como lenguaje de especificación ha permitido el desarrollo de un conjunto de herramientas que permiten explotar sus características para la puesta a punto e implementación de sistemas basados en lógica difusa. Las características de XFL han hecho posible que estas herramientas estén dirigidas tanto a implementaciones hardware como software y nos permiten prever la potencial aparición, en un futuro próximo, de nuevas herramientas dedicadas a otros aspectos del desarrollo de sistemas difusos.

El núcleo de todas las herramientas basadas en XFL está formado por un conjunto de funciones (que denominamos la librería XFL) capaces de analizar sintáctica y semánticamente una especificación y almacenarla en forma de un árbol de sintaxis abstracta para su posterior uso por los diferentes programas que las empleen. Estas funciones incluyen también la explotación de los mecanismos de configuración para las operaciones difusas descritos en el apartado anterior.

Para el desarrollo de sistemas difusos en software se dispone de XFC, un programa capaz de producir el motor de inferencia difuso correspondiente a una especificación XFL en C. El motor de inferencia se implementa a través de una función C que puede ser invocada desde cualquier otra parte de una aplicación general que necesite un módulo con estas características.

A la hora de realizar implementaciones hardware se cuenta con dos herramientas: XFPG, cuya salida permite la realización de un controlador difuso por medio de lógica combinacional, y XFVHDL [1], que produce la definición VHDL de un sistema difuso especificado con XFL. La primera herramienta utiliza la salida proporcionada por XFC para construir un programa capaz de generar tablas de verdad en un formato aceptable por herramientas de minimización lógica y obtener así la expresión, en términos directamente implementables en dispositivos como PLAs o FPGAs, del sistema difuso descrito usando XFL. La segunda herramienta produce una *entity* VHDL susceptible de ser integrada con otros elementos similares en la definición de un sistema más complejo o que puede constituir un circuito integrado por sí misma.

También se dispone de una herramienta de aprendizaje basada en retropropagación, XFBPA. Este programa permite el ajuste de los parámetros que definen las funciones de pertenencia y la eliminación de reglas cuyo máximo grado de activación no superen un determinado umbral. El aprendizaje no limita la estructura de las reglas y permite ajustar cualquier sistema descrito en XFL respetando la flexibilidad de la descripción.

Todas estas herramientas pueden utilizarse de modo independiente o a través de un entorno gráfico común para el desarrollo de sistemas difusos, denominado Xfuzzy 2.0. Además de integrar las funcionalidades de las herramientas descritas anteriormente, Xfuzzy ofrece un entorno gráfico de desarrollo para especificaciones XFL y mecanismos de simulación para la evaluación y puesta a punto de los sistemas difusos. La simulación por medio de Xfuzzy permite la composición del sistema con uno o más módulos que definen el comportamiento de su entorno. Estos módulos pueden ser definidos en C, XFL e incluso estar constituidos por una serie de valores numéricos. Los resultados de las simulaciones pueden obtenerse en varios formatos, gráficos o numéricos.

7 CONCLUSIONES

Hemos presentado aquí los principales aspectos de XFL, un lenguaje flexible y potente para la definición de sistemas basados en lógica difusa. Sus características más significativas radican en su capacidad para definir sistemas potencialmente complejos y en su independencia frente a aspectos tecnológicos concretos. Creemos que estas características permiten su empleo en un amplio espectro de aplicaciones de la lógica difusa y en el estudio mismo de los mecanismos que deben emplear los sistemas basada en ella (conectivas, métodos de defuzzificación, etc.). La aplicabilidad de XFL a diferentes dominios ha quedado demostrada por los campos de aplicación de las herramientas basadas en él y confiamos en que el desarrollo de nuevas herramientas y el empleo de XFL en distintas aplicaciones contribuya a enriquecer las capacidades del lenguaje.

Referencias

- [1] A. Barriga, S. Sánchez-Solano, C. J. Jiménez, D. Galán, D. R. López, "Automatic Synthesis of Fuzzy Logic Controllers", *Mathware & Soft Computing* (pendiente de publicación).
- [2] H. R. Berenji et al., "A Hierarchical Approach to Designing Approximate Reasoning-Based Controllers for Dynamic Physical Systems", *Uncertainty in Artificial Intelligence* (1991) pp. 331-343.
- [3] E. Cárdenas, J. C. Castillo, O. Cordón, A. Peregrín, "Estudio comparativo de sistemas de inferencia y métodos de defuzzificación aplicados al control difuso del péndulo invertido", *Proc. Tercer Congreso Español sobre Tecnologías y Lógica Fuzzy* (1993) pp. 259-266.
- [4] H. Hellendoorn and C. Thomas, "Defuzzification in fuzzy controllers", *Jour. of Intelligent and Fuzzy Systems*. Vol. 1 (1993), pp. 109-123.
- [5] R. Jager, H. B. Verbruggen, P. M. Bruhx, "The Role of Defuzzification Methods in the Application of Fuzzy Logic", *SICICA'92*.
- [6] M. Mizumoto and H. Zimmermann, "Comparison of fuzzy reasoning methods", *Fuzzy Sets and Systems*, vol. 8 pp. 253-283, 1982.
- [7] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller", Parts I & II, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20 N. 2 (1990) pp. 404-432.
- [8] J. Masa, A. Barriga, S. Sánchez-Solano, "Un entorno de simulación para sistemas de control basados en lógica difusa", *Proc. VII Congreso de Diseño de Circuitos Integrados DCIS'92* (1992) pp. 547-548.

RED NEURONAL PARA FUZZYFICACIÓN Y RECONOCIMIENTO DE PATRONES

Francisco J. López Aligué
Departamento de Electrónica
Universidad de Extremadura
aligue@ba.unex.es

M^a Isabel Acevedo Sotoca
Departamento de Electrónica
Universidad de Extremadura
acevedo@ba.unex.es

Resumen.

Presentamos una nueva definición de la función de transferencia neuronal, que junto con un eficaz esquema de adaptación de los pesos consigue una óptima separación entre las respuestas dadas ante los distintos prototipos empleados en la definición de las clases. El algoritmo de aprendizaje es sumamente rápido y permite efectuar una clasificación de las entradas muy robusta.

Palabras Clave : Reconocimiento de patrones, fuzzyficación, redes neuronales.

1. INTRODUCCIÓN

Las redes neuronales multicapa han recibido especial atención como elementos claves para el Reconocimiento de Patrones desde hace ya tiempo. En esta línea, la teoría de los conjuntos Difusos ofrece interesantes posibilidades entre las que destaca el empleo de neuronas definidas por funciones especiales intentando así evitar los conflictos que producen las diferencias existentes entre las entradas futuras y las usadas como prototipos, origen de la dificultad que implica llevar a cabo una correcta clasificación.

Este trabajo se ha dirigido a la búsqueda de estructuras simples altamente repetitivas de elementos más o menos semejantes a las neuronas biológicas en cuanto a su función de transferencia se refiere, y a los mecanismos de adaptación de los pesos. Aquí, partimos de la necesidad de establecer clasificaciones seguras entre las entradas a la red, para lo que solamente se

dispone de la información de la propia entrada que va a servir de prototipo, es decir, de su distribución de pixels iluminados y oscuros, que se convierten en datos con valores comprendidos entre dos límites 0 y 1. Así, lo que hacemos es definir la red como una substanciación de los prototipos que deseamos utilizar como definitorios de las distintas clases, de forma que los valores de los pesos se obtienen del valor que las propias entradas marcan.

2. ASPECTOS BÁSICOS DEL MODELO

En Reconocimiento de Patrones, el método de asignación de la clase a que cada entrada pertenece se basa en la información guardada en la red en forma de esquema de conexión de las entradas externas con las neuronas, a través de los correspondientes contactos, llamados sinapsis, calculadas según los prototipos que se hayan utilizado en una primera fase de "aprendizaje". Admitiendo que el diagrama de conectividad de las entradas de cada neurona queda fijado para todo el resto del proceso una vez decidido inicialmente en la fase de aprendizaje, la memoria de la red reside en los valores de los pesos asociados a las entradas a cada neurona en esa fase inicial. Así, la configuración de entrada correspondiente al conjunto total de los pixels del prototipo podemos denominarla como

$$X_i = (x_{1i}, x_{2i}, \dots, x_{pi}) \in I^P = [0,1]^P$$

donde $x_{ji} \in [0,1]$, representa el valor del pixel "j" de la imagen X_i de entrada. Del conjunto completo de los pixels de la imagen de entrada, cada neurona recibe unos cuantos, que denotaremos por z_{hk} , y tales que forman el verdadero vector de

entrada a la neurona k-ésima

$$Z_k = (z_{1k}, z_{2k}, \dots, z_{rk}) \in I^r = [0,1]^p \quad r \leq p$$

y siempre con el mismo orden. Es decir, se trata de un conjunto ordenado obtenido a partir del conjunto total de la configuración de entrada.

Por regla general, los vectores de entrada serán de naturaleza típicamente analógica, tal y como también lo son los pesos asociados a estos. De esta forma, sea

$$W_k = (w_{1k}, w_{2k}, \dots, w_{rk}); w_{hk} \in [0,1]$$

el vector de pesos asociado a la entrada Z_k a la neurona k-ésima. En el caso más general, se ha adoptado la función

$$y_k = \sigma \left(\sum_{i=1}^R w_k^i x_k^i - \theta \right) \quad (1)$$

como más representativa de los modelos convencionales, que supone, en realidad, una medida de la *similitud* existente entre el patrón almacenado W_k y el de entrada X_k . Cuando esta similitud supera el umbral θ , la salida pasa a valer 1, mientras que para funciones distintas, la actividad de la neurona guarda una relación directa más o menos lineal con la similitud existente.

Sin embargo, la medida de la similitud puede hacerse de forma mucho más simple, pero para ello es preciso hacer antes un detenido análisis de la estructura y funcionamiento de una única capa junto con su método de aprendizaje, fundamental para la propia definición del modelo que aquí vamos a analizar.

3. EL MODELO NEURONAL Y SU ESQUEMA DE APRENDIZAJE

El modelo neuronal que presentamos es genérico y permite plasmarse en las estructuras prácticas posteriores con gran facilidad. El aprendizaje se realiza de una forma bastante inmediata: Tomamos las neuronas de la primera capa, que admitiremos distribuidas sobre un cuadrado de dimensión $N \times N$. Con ellas, formamos tantos grupos como prototipos queramos utilizar para definir las distintas clases, siempre que cada prototipo sirva para identificar una única clase; en caso contrario, se definen subclases, de forma que se siga manteniendo una correspondencia biunívoca entre el conjunto de los prototipos y el de las clases. Para poder detectar el mayor número posible de

los pixels que conforman las imágenes, parece lógico distribuir las neuronas (y sus entradas) sobre la mayor extensión posible del plano de imagen, para lo que se puede utilizar una estrategia similar a la empleada en el preprocesamiento de imágenes por teselación.

Entonces, llega el momento de asignar cada grupo de neuronas a un prototipo haciendo coincidir los pesos w_{hk} de las conexiones a los valores de las componentes del vector de entrada z_{hk} que le correspondan. Este proceso es prácticamente inmediato y se realiza una sola vez para cada neurona. Así, para aquellas neuronas elegidas para ese determinado prototipo, se define un umbral θ (*umbral de actividad*), de forma que

$$\begin{aligned} \forall z_{hk} > \theta, w_{hk} &= z_{hk} \\ \forall z_{hk} \leq \theta, w_{ij} &= -1 \end{aligned} \quad (2)$$

Obsérvese que cuando la señal de entrada no supera el umbral θ , que viene impuesto por la necesidad de eliminar del proceso de aprendizaje la componente de ruido presente siempre en la red, se asigna un valor negativo al peso asociado. Más adelante nos ocuparemos de la programación del resto de las capas de la Red.

Ahora, ya podemos abordar el diseño de la función que deberá llevar a cabo nuestra neurona. Se trata de detectar la existencia de similitudes entre la imagen de entrada y el patrón almacenado, los cuales son parte de la imagen total y del prototipo total respectivamente. Así pues, la función que defina nuestra neurona, deberá poner en evidencia lo que se parece la entrada actual al patrón almacenado. Cuando se presenta una determinada entrada z_{hk} en un punto en que su peso asociado es nulo para el prototipo, nos encontramos en una situación que debe ser tomada como una diferencia típica y, por tanto, calculada su medida como tal. Así, llamando I_{hk} la excitación que llega a la neurona k a través de la sinapsis h, recibida de la señal de entrada z_{hk} , podemos definir su cálculo según el valor de la entrada:

$$\begin{aligned} \forall z_{hk} > \theta, \text{ Si } w_{hk} > 0 \text{ entonces} \\ I_{hk} &= 1 - |z_{hk} - w_{hk}| \end{aligned} \quad (3.a.)$$

Que, dado (2) implica que si $w_{hk} = 0$, $Y_{hk} = -z_{hk}$, es decir, se comporta como una entrada inhibidora, mientras que para $w_{hk} > 0$, se trata de una entrada excitadora. De la misma forma,

$$I_{hk} = 0 \quad \forall z_{hk} \leq \theta \quad (3.b.)$$

por lo que, en consecuencia, la actividad resultante de la neurona puede definirse ya como

$$y_k = \frac{1}{n^*} \sum_{j=1}^n I_{hj} \quad \text{si y sólo si} \quad \sum_{j=1}^n I_{hj} \geq 0$$

$$= 0 \quad \text{cualquier otro caso} \quad (4)$$

donde n^* = Número de pesos positivos, $w_{hk} > 0$ ($n^* \leq n$), es el coeficiente de normalización para que el valor final de la función esté comprendido entre 0 y 1. Un aspecto de máxima trascendencia en esta formulación es el hecho de que una entrada nula no se emplea para la evaluación de similitud, puesto que lo contrario equivale a admitir que la ausencia total de imagen, es decir, entrada de total oscuridad, tendría una pertenencia bastante elevada a casi todas las clases, dado que un buen número de pixels de los prototipos son nulos.

De esta forma, nuestra función determina el grado de similitud existente entre la entrada y el prototipo. En efecto, para el caso de coincidencia total,

$$|z_{hk} - w_{hk}| = 1 \quad \forall h \Rightarrow I_{hk} = 1,$$

mientras que en el caso contrario de total disparidad, tendremos

$$|z_{hk} - w_{hk}| = 0 \quad \forall h \Rightarrow I_{hk} = 0.$$

En las situaciones normales, I_{hk} tomará valores intermedios. En realidad, la salida de la neurona es una auténtica medida de la función de pertenencia a cada clase de cada entrada definida por el prototipo W_{hk} . En este sentido, nuestra neurona debe ser vista como una neurona fuzzy debido al hecho de que cada neurona calcula la similitud existente entre la entrada y su prototipo memorizado, independientemente del resto de las neuronas.

4. LA RED

En la descripción del apartado anterior, el cálculo del valor de la salida de las distintas neuronas se ha hecho sin tomar en consideración el umbral de las mismas. Como consecuencia de los resultados experimentales, es de vital importancia que la salida de cada capa sea procesada por un filtro de nivel, de forma que el valor final de las salidas individuales que pasan de esta capa a la siguiente

dependa de la superación o no de un determinado valor. En caso afirmativo, el valor de salida se aumenta hasta el máximo posible, mientras que en caso contrario, se reduce a cero. Lo dicho para la conexión entre la primera capa y la segunda, es perfectamente extensible a la conexión entre ésta y la tercera y así sucesivamente. De esta forma, la red queda configurada como una solución de capas de procesamiento y de filtrado alternativamente, tal y como muestra la Figura 1. Una vez programada la primera capa, se utilizan sus salidas, que ya son fijas para cada entrada externa, para la programación de los pesos de la segunda capa, lo que se lleva a cabo de acuerdo con la misma estrategia que en el caso anterior.

Sin embargo, el funcionamiento de la segunda capa y posteriores se podrá realizar empleando umbrales distintos de 1 para las capas de filtrado intermedias, contrariamente a lo que no sucede en el aprendizaje, en que se mantiene siempre al máximo valor. Un umbral 0 significa la eliminación del filtrado entre capas, es decir, efectuar una clasificación muy cercana al concepto fuzzy, mientras que el umbral 1 supone convertir la capa en un clasificador crisp de las imágenes de entrada. Umbrales intermedios servirán para permitir eliminaciones de ruidos y distorsiones.

El proceso puede repetirse indefinidamente hasta que todas las capas tengan sus valores sinápticos calculados lo que, en casos de simulaciones prácticas, aun con un número muy elevado de neuronas supone un tiempo prácticamente despreciable que, además, sólo debe ejecutarse una vez al principio de su instalación.

De acuerdo con las definiciones anteriores, la salida de cada neurona nos dará un valor comprendido entre los límites 0 y 1. Así, ante un determinado prototipo, algunas neuronas, las que lo reconozcan como propio, darán una respuesta de máxima intensidad, mientras que para las demás, la mayoría, la salida dependerá de lo que su entrada se parezca al vector que tuvieran almacenado. La clasificación de la imagen de entrada como perteneciente a una determinada clase, lo que nos exige que las salidas de la capa constituyan una configuración que pueda asociarse con la que se obtiene para algún prototipo de los que se hayan utilizado en su identificación. Como es obvio, según la definición (2), lo que en realidad vamos a obtener será una distribución de valores más o menos parecida a la obtenida ya

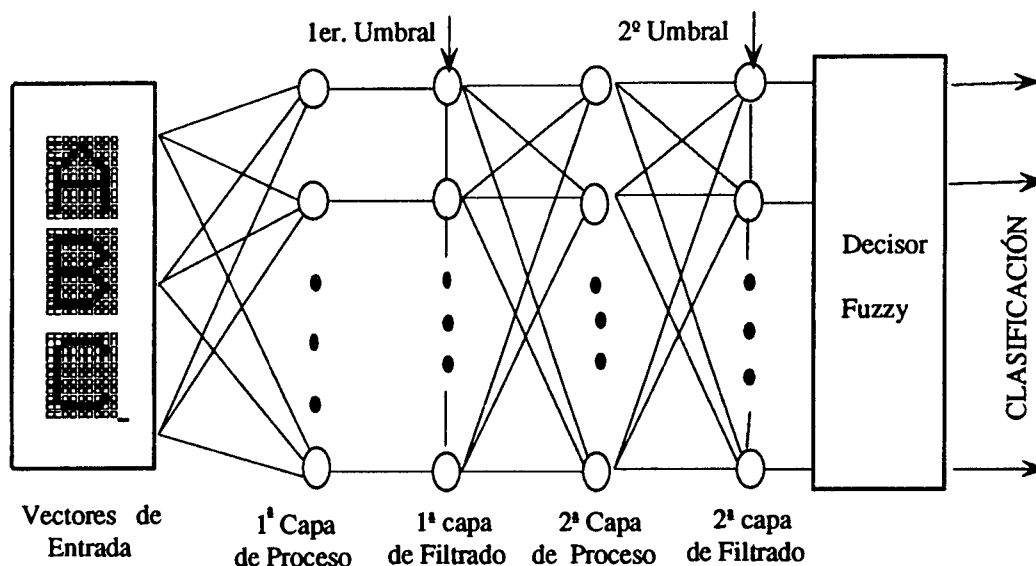


Figura 1. Esquema básico de la Red. Tanto las imágenes de entrada como las conexiones mostradas o el número de capas lo son exclusivamente a modo de ejemplo.

para los prototipos. Por regla general, incluso la respuesta a cada uno de tales prototipos contendrá varias salidas excitadas de neuronas comunes a la obtenida con otros prototipos.

En el caso general, la salida de la red participará mayoritariamente de la configuración obtenida para el prototipo de la clase al que la entrada pertenezca, pero dadas las distorsiones y ruidos presentes, en la salida se mezclarán los valores correspondientes al prototipo con los de los demás con los que exista alguna semejanza. La solución consiste en enviar las salidas de la red a una nueva capa como la anterior, donde se efectúe un nuevo proceso de clasificación.

Sin embargo, esta conexión entre capas no la hacemos de forma directa, sino que lo que realmente se envía es una información de similitud. Tal y como hemos dicho antes, introducimos una capa de *filtrado* entre las que denominamos de *procesamiento*. Dicha capa de interacción se encarga de efectuar un tratamiento de las salidas procedentes de la primera capa de procesamiento, la llamada de *entrada*, para lo que únicamente se emplea el umbral con el que se decide si el valor procedente de la capa anterior se asigna a 1 o a 0.

De esta forma, la primera capa recibe las imágenes directas del exterior, efectúa una primera clasificación y envía su salida a la primera capa de filtrado. Sin embargo, al ir generando todas las distintas configuraciones de salida correspondientes a cada uno de los distintos prototipos, las muestras de entrada que llegan a las neuronas de la capa pueden ser semejantes, en algunos casos, para distintos prototipos. Así, por ejemplo, una cierta neurona recibe una muestra de una entrada similar al que recibe para otra.

Ahora, la salida de la capa que se obtiene al presentarle, cuando ya estamos en la etapa de reconocimiento, el primer prototipo, contendrá, con actividades comprendidas entre 0 y 1, las salidas que se le asignaron de entrada como propias, más todas aquellas para las que diera la casualidad de que sus muestras con la entrada de otro u otros prototipos. La estructura multicapa se encarga de ir reduciendo progresivamente este solape hasta valores mínimos. Al final, nuestro sistema incorpora un decisor Fuzzy desarrollado por los autores y que permite llegar a decisiones de gran seguridad y robustez.

5. ASPECTOS PRÁCTICOS

Se ha desarrollado un simulador genérico propio, basado en una arquitectura multiprocesador con un operativo de tiempo real (basado en VxWorks), con el que, hemos implementado una Red de cuatro capas con 10.000 neuronas por capa y 50 entradas por neurona. El aprendizaje se realiza poniendo los umbrales a un valor intermedio de 0,5 para todas las capas, mientras que el reconocimiento se hace con un umbral 1 para la salida de la 1ª capa, y 0,9 para las demás (eliminación de ruido únicamente). La capa de entrada lleva a cabo ciertos procesos de pretratamiento de imágenes (eliminación de giros, escalado, centrado, etc.) que consiguen evitar deformaciones y defectos de la entrada.

Se ha desarrollado una estrategia de aprendizaje diferenciada para la primera capa, y que consiste en establecer conexiones de las neuronas de la 1ª capa con, únicamente, los entornos de los pixels de los prototipos que estén activados. De esta forma, se consiguen tres grandes ventajas :

a) Se pueden tratar imágenes de una gran finura de grano con un número relativamente bajo de neuronas.

b) Se eliminan todos los ruidos y señales sin significado de forma automática.

c) Al conectarse las entradas de las neuronas a pixels verdaderamente significativos, éstos pueden estar geográficamente muy distantes, con lo que las neuronas pasan a detectar similitudes espaciales casi globales, perdiendo la restricción de localidad que se suele achacar a las redes convencionales.

Este tipo de red está dando excelentes resultados para aplicaciones prácticas en las que se precisa de una gran capacidad de distinción entre prototipos bastante similares y con una elevada definición de imagen de entrada, como es el caso de la selección de razas animales, en que los rasgos fenotípicos son determinantes. Sin llegar hasta este extremo, van apareciendo numerosas aplicaciones de sumo interés y que ponen de manifiesto la versatilidad de nuestro modelo. Una primera aplicación de este tipo de señales surge de la misma raíz de la clasificación. Nos planteamos la siguiente situación: se recibe una señal analógica, que podría corresponder perfectamente a algún significado lingüístico común, tal y como ocurre en los sistemas de control fuzzy, que debe ser asignada a una determinada clase. La fuzzyficación es un proceso absolutamente similar al que ejecuta nuestro modelo neuronal. En efecto, la idea básica consiste en calcular cuanto se parece la señal de entrada a las que hemos usado como prototipos, de forma que el problema vuelve a plantearse en términos de similitud entre señales. Para nosotros, la solución del problema consiste en emplear la misma definición neuronal, pero ahora con una gran simplificación consistente en que la dimensión de la entrada se ha reducido a 1, y que bastará con disponer de tantas neuronas como clases queramos establecer. De esta manera, todas las neuronas trabajaran en paralelo y el resultado se obtiene en un único paso de simulación, reduciendo el tiempo de cómputo al mínimo.

En nuestro ejemplo se trata de definir 6 categorías a las que hay que asociar la señal de entrada. Para simplificar, supondremos que dicha señal está comprendida en el intervalo (0,1), y admitiremos que las seis categorías están uniformemente distribuidas sobre todo el rango de posible variación. En consecuencia, la red deberá disponer de seis neuronas, cada una de ellas con una sola entrada, la cual es, además, común a

todas las neuronas y constituye la *Señal de Entrada*, procedente del exterior. Se trata, por tanto, de una situación opuesta a la que hemos analizado en el caso anterior, puesto que el número de elementos en juego es ahora mínimo, frente al elevado número de entradas externas y de neuronas de entonces. Además, aquí solamente necesitamos una capa, frente al número de las que allí empleábamos. Sin embargo, la neurona empleada y su algoritmo de aprendizaje son exactamente los mismos, por lo que no es precisa ninguna redefinición del modelo. Tan sólo, debe tenerse en cuenta que ahora $n^* = 1$, y que no existen entradas inhibidoras. El resultado se muestra en la figura 2, donde aparecen las seis salidas como las del fuzzyficador equivalente, cada una de ellas procedente de una de las neuronas de la red.

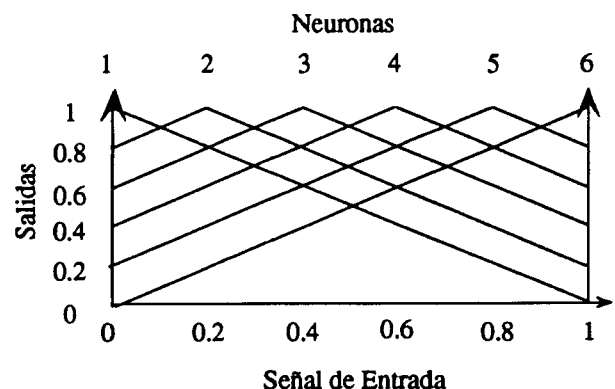


Figura 2. Respuesta de las seis neuronas del fuzzyficador ante los prototipos del aprendizaje

La entrada común llega a todas ellas y el proceso de aprendizaje se lleva a cabo de la forma convencional, es decir, presentando una señal de entrada 0 y obligando a que ese sea el valor del peso asociado de la primera neurona. A continuación, se presenta una entrada de 0.2 y se asigna este valor a la segunda neurona, y así sucesivamente, hasta llegar a la sexta neurona, cuyo peso se equipara al valor 1.

Este no es el tipo de clasificación más adecuado para un fuzzyficador convencional, puesto que, en la mayoría de los casos, se exige una mayor separación entre clases. Lo habitual (Figura 3) en los sistemas de control es exigir que cada valor de entrada se clasifique entre sólo dos clases, salvo que coincida con uno de los prototipos, en cuyo caso deberá asignarse exclusivamente a la clase que tal prototipo define.

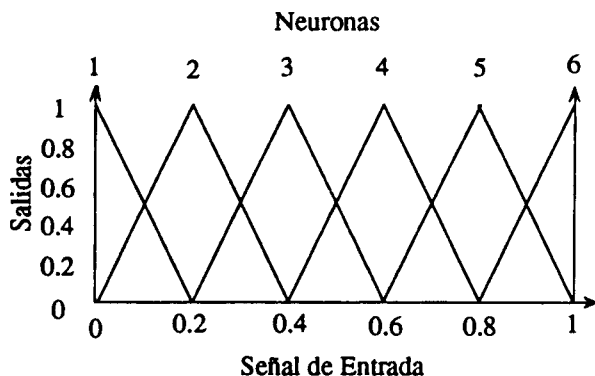


Figura 3. Resultado de introducir una capa de filtrado a la salida de la red neuronal.

La forma de conseguir esta respuesta a partir de nuestra red es bastante simple, y sus resultados pueden comprobarse en la Figura 3, donde lo significativo de la misma estriba en el hecho de que nuevamente nos encontramos con la conveniencia de introducir una función no lineal que modifique la función que internamente ejecuta la neurona que, en este caso particular, hemos elegido con el perfil de la rampa cuyo aspecto resulta totalmente familiar para los casos de fuzzyficación convencionales.

6. CONCLUSIONES

Nuestro objetivo era encontrar un modelo de neurona, y de red neuronal, que nos permitiera ejecutar ciertos procesos de reconocimiento de patrones de una forma analítica no convencional, partiendo de la definición arbitraria de similitud. Para ello, optamos por la construcción de una red formada por elementos de procesamiento no necesariamente significativos desde un punto de vista biológico. Sin embargo, hemos llegado a una estructura mucho más relacionada con los elementos neuronales biológicos de lo que podríamos haber supuesto: La función no lineal de salida de nuestros elementos de procesamiento incluye la función sigmoide, que la naturaleza ha adoptado como única. Está claro que nuestra simulación software puede hacerse con funciones sigmoides, umbral o cualquier otro. En el fondo son distintas versiones de un mismo hecho, y es que se trata de un único modelo neuronal, capaz de adquirir el funcionamiento que en cada caso convenga, sin más que adaptar adecuadamente la función de salida (sigmoide, umbral, etc.).

Por otra parte, es destacable el hecho de que el aprendizaje, de tipo supervisado, es inmediato y

no precisa de algoritmos de adaptación de los pesos, lo que supone una buena ventaja sobre los más tradicionales.

Agradecimientos

Los autores desean expresar su agradecimiento a la C.I.C.Y.T. por su ayuda a través del programa TIC 92 - 0254, y a la Junta de Extremadura (Consejería de Educación).

Referencias

- [1] Bezdeck, J. (1994), *Fuzzy Models for Pattern Recognition*. IEEE Press.
- [2] Carpenter, G.A. (1991), Neural models for pattern recognition and associative memory, en *Pattern Recognition by Self-Organizing Neural Networks*, MIT Press.
- [3] Chua, L. y Yang, L. (1988), Cellular neural networks: Theory. *IEEE Trans. on CAS*, **35**, 10: 1257-1272
- [4] Grossberg, S., (1991). Neural pattern discrimination, en *Pattern Recognition by Self-Organizing Neural Nets*, MIT Press.
- [5] Kosko, B., (1992). *Neural Networks and Fuzzy Systems*. Prentice Hall, N.J.
- [6] Kung, S.Y. y Taur, J.S., (1995). Decision-based neural networks with signal/image classification applications, *IEEE Trans on Neural Networks*, **6**, 1: 170-181.
- [7] López Aligué, F.J., Acevedo Sotoca, I., y Jaramillo Morán, M.A., (1991). The fuzziness of fuzzy partitions, *Pattern Recognition Letters*, **12**: 265-271.
- [8] Micheli-Tzanakou, E., (1988). Neural aspects of vision and related technological advances, *Proc. of the IEEE*, **76**, 9: 1130- 1142.
- [9] Mitra, S., y Pal, S.K., (1995). Fuzzy multi-layer perceptron, inferencing and rule generation, *IEEE Trans. on Neural Networks*, **6**, 1: 51-63,
- [10] Pedrycz, W., (1992). Fuzzy Neural Networks with reference neurons as pattern classifiers, *IEEE Trans. on Neural Networks*, **3**, 5: 770-775.
- [11] Pedrycz, W., (1995). A reconfigurable Fuzzy Neural network with in-situ learning, *IEEE Micro*, **15**, 4: 19-29.
- [12] Seiler, G., Schuler, A.J. y Nossek, J.A. (1993). Design of robust cellular neural networks. *IEEE Trans. on CAS*, **40**, 5: 358-364.

APLICACIÓN DE TÉCNICAS TEMPORALES DIFUSAS A LA DISCRIMINACIÓN DE VOCABLOS SIMPLES

Daniel Martínez

Orestes Mas

Martín Rolando

Universitat Politècnica de Catalunya - Dept. T.S.C.

c/Gran Capità s/n.

08024 BARCELONA

e-mail: conomc@tsc.upc.es

Resumen

En este trabajo se presenta una aplicación de la Teoría de Conjuntos Difusos al reconocimiento de vocablos simples. El método propuesto soslaya el problema de tratar la duración y posición variable de los fonemas mediante el uso de conceptos temporales difusos en las reglas de inferencia. Después de una descripción de la estructura general del sistema, se pasa a la explicación pormenorizada de cada una de las secciones: Etapa de adquisición de la señal vocal, Etapa de clasificación difusa de los distintos fonemas y Etapa de discriminación temporal de los mismos. Posteriormente se discute la realización física de los conceptos temporales difusos mediante filtros lineales. Para finalizar, se exponen algunos resultados obtenidos y las conclusiones.

Palabras Clave: Conceptos Temporales Difusos, Inferencia Difusa, Reconocimiento de voz

1. INTRODUCCIÓN

Desde que en la década de los 50 se empezó a abordar de forma sistemática el problema del reconocimiento automático del habla, uno de los obstáculos que primero se presentaron fue el de compensar la *duración e instante de pronunciación* variables de un mismo vocablo pronunciado en distintos momentos o por distintas personas. Para minimizar el problema se introdujo la llamada *técnica de normalización temporal no lineal*, que ajusta la duración de los vocablos a un valor constante para poder compararlos. El inconveniente de esta técnica es que requiere la realización de cálculos más o menos complejos que impiden su implementación mediante técnicas analógicas, más económicas y rápidas que las digitales.

En el marco anteriormente expuesto, puede sacarse partido de modelar la incertidumbre en cuanto a la posición temporal y duración de los fonemas mediante conceptos difusos temporales. El hacerlo, sin embargo, presenta cierta problemática específica en cuanto a modelación de los citados conceptos y inferencia difusa en su presencia, problemática que debe ser previamente resuelta. En este sentido son de utilidad los trabajos de Mas, Palà y Miró, que permiten expresar el razonamiento aproximado en presencia de conceptos temporales difusos mediante una *convolución* [1], [2].

A continuación presentamos un ejemplo de aplicación de los conceptos temporales difusos a la resolución de un problema sencillo: el objetivo del sistema diseñado es el de discriminar entre los vocablos SÍ y NO, pronunciados por distintos locutores, de forma totalmente analógica.

2. ESTRUCTURA GENERAL

El diagrama de bloques del sistema discriminador se muestra en la Figura 1, y lo constituyen los bloques siguientes:

- Bloque de captación y parametrización de la señal de voz
- Bloque de clasificación difusa de los sonidos en cuatro categorías básicas: Silencios, Nasaes, Vocales y Silbantes.
- Discriminación entre los vocablos "SI" y "NO" mediante las dos reglas temporales difusas siguientes:

If (voz ANTES is Silbante)

and (voz AHORA is Vocal)

then (output is SÍ)

If (voz ANTES is Nasal)

and (voz AHORA is Vocal)

then (output is NO)

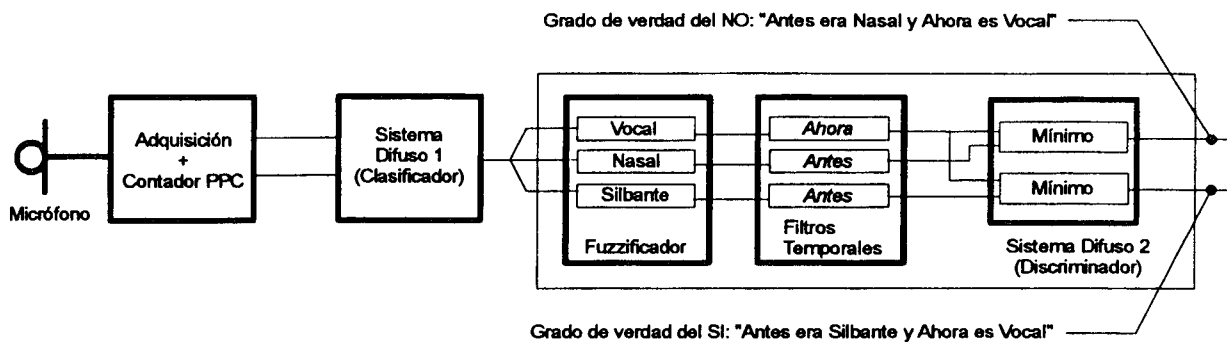


Figura 1: Diagrama de bloques del discriminador de vocablos

Es importante señalar que no se emplea ningún tipo de memoria, por lo que el reconocimiento se realiza en tiempo real.

En las secciones siguientes se estudian más detalladamente cada uno de los procesos mencionados.

3. ADQUISICIÓN Y PARAMETRIZACIÓN DE LA SEÑAL VOCAL

En todo sistema de reconocimiento automático del habla, la primera acción a realizar suele ser un proceso interpretativo aplicado a la señal de voz para extraer de ella aquellos parámetros que mejor la caractericen. Durante este proceso, el objetivo primordial es reducir notablemente la cantidad de información a manejar mediante la eliminación de las redundancias presentes en la señal de voz [3]. Debemos, pues, plantearnos elegir un método de parametrización adecuado a nuestras necesidades.

Sea cual sea la parametrización escogida, deberá ser a la vez *sencilla de obtener y manejar* (por motivos económicos) y *significativa* (suficiente para permitir alcanzar los objetivos de reconocimiento propuestos). Para ello se ha escogido como parámetro la *densidad media de cruces por cero* de la señal de voz en un cierto periodo. A pesar de ser una representación bastante "tosca" de la voz, dicho parámetro nos proporciona una estimación aproximada del contenido frecuencial de la señal [4], [5], suficiente para nuestros propósitos. Para aumentar la información disponible se ha decidido calcular este parámetro dos veces, una para las componentes paso-bajo de la señal de voz y otra para las componentes paso-alto, como se describe en el siguiente párrafo.

A grandes trazos la etapa de adquisición y parametrización funciona de la siguiente manera: captamos la señal de voz mediante un micrófono *electret* omnidireccional, seguido de un preamplificador que filtra la señal paso banda (principalmente elimina la componente continua que nos da el *electret*). A la salida

del preamplificador dividimos la señal en dos ramas, una de las cuales se filtra paso alto ($f_c=3\text{KHz}$) y la otra paso bajo ($f_c=1\text{KHz}$). Es importante evitar la saturación del preamplificador y de cualquier etapa de los filtros, ya que ésta provocaría la aparición de componentes en alta frecuencia que falsearían los valores posteriores de los resultados. Por ello, en aplicaciones donde el nivel de señal presente a la entrada no esté predeterminado resulta de interés la incorporación de un circuito de control automático de ganancia (CAG) en algún punto del camino de la señal. La salida de esta etapa la constituyen, pues, dos señales que son las componentes paso-bajo y paso-alto de la señal de voz.

A continuación cada una de estas dos componentes se lleva a un circuito detector de cruces por cero, y la salida de éstos se lleva a sendos contadores digitales. La lectura periódica (y posterior puesta a cero) del valor contenido en los citados contadores nos proporciona la parametrización requerida. El periodo durante el cual se acumulan cruces por cero se ha tomado de *10 milisegundos*, por ser un intervalo de tiempo durante el cual las características de la señal vocal pueden considerarse aproximadamente estacionarias [6].

La Figura 2 nos muestra los pasos por cero de un *SÍ* y un *NO* (seguidos y por este orden). La primera gráfica es la de los pasos por cero de la señal paso bajo y la segunda la de los pasos por cero de la señal paso alto. A simple vista parece posible distinguir entre los distintos fonemas que componen los vocablos "*SÍ*" y "*NO*" utilizando únicamente los pasos por cero de la señal filtrada, con lo que es plausible construir un sistema difuso que efectúe esta clasificación basándose en las mismas reglas que sigue un experto para realizarla. La realización de este sistema difuso se comenta en la siguiente sección.

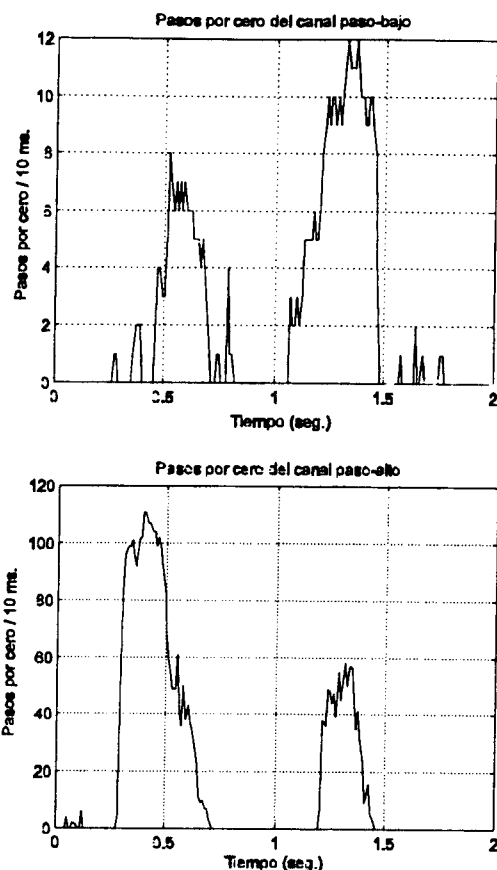


Figura 2: Pasos por cero de la señal paso-bajo y paso-alto

4. PRIMER SISTEMA DIFUSO: CLASIFICACIÓN DE FONEMAS

A la vista de las gráficas de la Figura 2 pueden apreciarse claramente las diferencias entre los cuatro fonemas que se han pronunciado (/s/, /i/, /n/ y /o/). Observamos que la /s/ (de 0 a 0.3 s) tiene muchos pasos por cero (PPC) en la banda paso-alto y pocos PPC en la paso-bajo. En general, este comportamiento será parecido para los fonemas silbantes y fricativos, como /sh/ o /f/. Las vocales (/i/ de 0.3 a 0.5s y /o/ de 1 a 1.2s) tienen bastantes PPC en la banda paso-bajo y la mitad de PPC en la paso-alto que la /s/. Por último, la /n/ no tiene PPC en la banda paso-alto y tiene algunos PPC en la paso-bajo. Es de destacar que las diferencias observadas en los cruces por cero se han puesto de manifiesto utilizando conscientemente expresiones difusas, como “muchos” o “bastantes”, con lo cual un sistema difuso que efectuase la clasificación de fonemas basándose en los mencionados cruces podría expresarse mediante el conjunto de 5 reglas siguientes, en las que “PPCPB” significa “Número de cruces por cero en el canal paso-bajo en 10 milisegundos”, y “PPCPA” lo mismo para el canal paso-alto:

Reglas:

- If (PPCPB is Bajo) and (PPCPA is Bajo) then (output is Silencio)
- If (PPCPB is Medio) and (PPCPA is Bajo) then (output is Nasal)
- If (PPCPB is Alto) and (PPCPA is Medio) then (output is Vocal)
- If (PPCPB is Medio) and (PPCPA is Medio) then (output is Vocal)
- If (PPCPB is Bajo) and (PPCPA is Alto) then (output is Silbante)

El clasificador difuso de fonemas será, pues, un sistema difuso con dos entradas (cruces por cero paso alto y paso bajo), y una salida numérica cuyo valor nos indicará qué tipo de fonema está presente en la entrada en cada momento. El siguiente cuadro muestra la correspondencia entre el valor de la salida y la categoría de fonema reconocido:

Fonema	Salida
“Silencio”	0
“Nasal” (/u/, /n/, /m/,...)	1
“Vocal” (/a/, /e/, /i/, /o/)	2
“Silbante” (/s/, /sh/, /f/)	3

Las funciones de pertenencia que caracterizan los conceptos difusos “Bajo”, “Medio” y “Alto” para cada una de las entradas pueden obtenerse haciendo uso de distintas técnicas, de las cuales se han utilizado básicamente dos. La primera de ellas, manual, hace uso de criterios estadísticos aplicados a la observación de múltiples gráficas como las de la Figura 2, resultado de pronunciar múltiples “sí” y “no” por varios locutores. La segunda, automática, se basa en el entrenamiento de un sistema neuro-difuso ANFIS [7], con un número suficiente de muestras. Es de destacar que ambas técnicas han arrojado resultados muy similares, debido probablemente a que la clasificación que se quiere hacer no plantea excesivas dificultades.

La Figura 3 presenta la salida del sistema clasificador cuando ponemos a sus entradas las señales de la Figura 2. En ella puede observarse que la señal está básicamente cuantizada a los cuatro valores deseados según la combinación presente a las entradas.

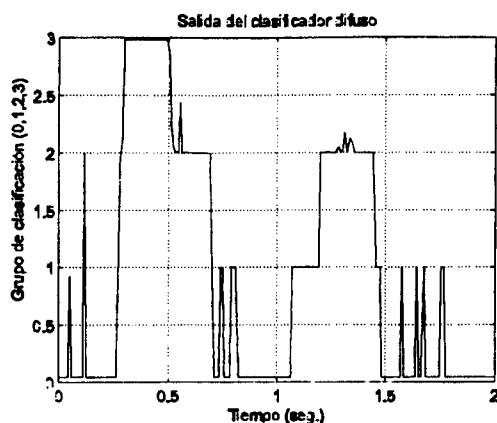


Figura 3: Salida del sistema clasificador difuso

5. SEGUNDO SISTEMA DIFUSO: DISCRIMINADOR BASADO EN CONCEPTOS TEMPORALES DIFUSOS

Hasta ahora el sistema se limita a clasificar cada fonema en una de las cuatro categorías definidas. Para proseguir en el proceso de discriminación entre el "SÍ" y el "NO" es preciso además tener en cuenta el *orden* en que se producen los distintos fonemas, puesto que de no ser así el sistema respondería igual a SÍ que a ÍS.

Al considerar el *orden* de aparición de los fonemas debemos introducir conceptos temporales en nuestros razonamientos. Por ende, dichos conceptos temporales no pueden ser demasiado precisos, pues el instante de emisión y la duración de los distintos fonemas variará entre dos producciones del mismo vocablo pronunciadas en instantes distintos y/o por distintos locutores. Podemos pensar, pues, en diseñar un sistema difuso que nos discrimine entre un "SÍ" y un "NO" (representados mediante dos valores distintos de la salida) a partir de la señal de salida del clasificador anterior (que denotaremos por x) y el conjunto de reglas siguiente:

- If (x AHORA is Vocal fuerte) and (x ANTES is Silbante) then (output is SÍ)
- If (x AHORA is Vocal fuerte) and (x ANTES is Nasal) then (output is NO)

Obsérvese que, en los antecedentes de las reglas anteriores nos aparecen conceptos temporales, como ANTES y AHORA, que son difusos y que por lo tanto no hacen referencia a un único instante de tiempo. Ello impide aplicar la regla composicional de inferencia en la forma habitual, pensada para operar las entradas instante a instante.

En esta situación resulta de evidente utilidad la formulación propuesta por Mas *et al.* citada anteriormente [2], la cual permite encontrar el grado de

verdad de antecedentes de reglas difusas cuando en ellas se mezclan conceptos difusos espaciales y temporales. Someramente, dicha formulación expresa el grado de verdad de una regla difusa como la *convolución* entre una función que representa el concepto temporal difuso (p. ej. "Antes") y otra función que representa el concepto espacial (p. ej. "Grande"). Así expresado, el grado de verdad del antecedente de la regla puede calcularse instante a instante como la salida de un filtro lineal cuya respuesta al impulso sea el concepto temporal difuso deseado ($\mu_{Antes}(t)$) y cuya entrada sea $\mu_{Grande}(x(t))$ [1], [2].

Las funciones de pertenencia AHORA y ANTES se realizan, pues, con dos filtros lineales. A la salida de cada uno tendremos el grado de verdad, de 0 a 1, de una premisa determinada, por ejemplo "ANTES is Nasal". El grado de verdad de un antecedente completo se obtendrá haciendo el mínimo (and) de las dos premisas que lo componen. Una vez obtenido el grado de verdad del antecedente, denotado por v , se encuentra genéricamente el conjunto difuso de salida B' como:

$$\mu_{B'}(y) = v \wedge \mu_B(y)$$

en donde B es el conjunto difuso original, B' es el modificado y " $\wedge$ " es el operador *mínimo* o *producto*, según efectuemos una inferencia tipo Mamdani o tipo Larsen [8].

Así, la salida final del sistema consistirá en dos señales, una por cada regla, que oscilarán entre los valores 0 y 1. Si una de las dos supera un umbral predefinido (por ejemplo, 0.8) se habrá producido el reconocimiento de una de las dos palabras.

6. REALIZACIÓN DE LOS FILTROS AHORA Y ANTES

A la hora de realizar los filtros temporales (o funciones de pertenencia temporales), tenemos dos opciones: la *digital* o la *analógica*.

La *opción digital* nos permite un diseño muy exacto, ya que podemos implementar cualquier respuesta impulsional utilizando, por ejemplo, filtros FIR. Sin embargo esta opción acarrea una mayor complejidad a la hora de realizar el filtro (memorias, reloj,...).

La *opción analógica* implica el diseño de filtros realizables mediante alguna estructura de componentes (pasivos, activos, etc.), y dado que la mayor parte de la teoría de diseño de filtros está encaminada al diseño en el dominio frecuencial, esta tarea resulta algo difícil. En cambio una vez obtenido el diseño su implementación es inmediata: se construye el filtro y se excita con la señal deseada, a la salida tenemos la respuesta que buscamos. En el caso analógico, sin embargo, puede suceder que no

todos los conceptos temporales que deseemos sean realizables mediante un filtro lineal. Por ejemplo, una exigencia excesiva en las especificaciones puede conducir a filtros inestables o de orden prohibitivamente elevado. Aparece entonces la necesidad de obtener conceptos temporales aproximados a los deseados, pero que sean fáciles de realizar. Esto puede llevar, por ejemplo, a aceptar filtros con polos complejos conjugados, que cuya respuesta al impulso presenta oscilaciones, dando lugar a funciones de pertenencia que pueden tomar valores negativos pequeños en alguna ocasión.

Siguiendo el criterio de simplicidad y bajo coste escogemos la opción analógica. La Figura 4 muestra la respuesta impulsional del sistema AHORA y ANTES analógico que utilizamos finalmente, las funciones de transferencia son:

$$H_{\text{ahora}}(s) = \frac{100}{s + 100}$$

$$H_{\text{antes}}(s) = \frac{1.3 \cdot 10^8}{s^6 + 87s^5 + 3778s^4 + 10^5s^3 + 2 \cdot 10^6s^2 + 2.2 \cdot 10^7s + 1.3 \cdot 10^8}$$

donde $H_{\text{ahora}}(s)$ se puede realizar con un simple circuito RC, y $H_{\text{antes}}(s)$ con un filtro activo utilizando tres amplificadores operacionales.

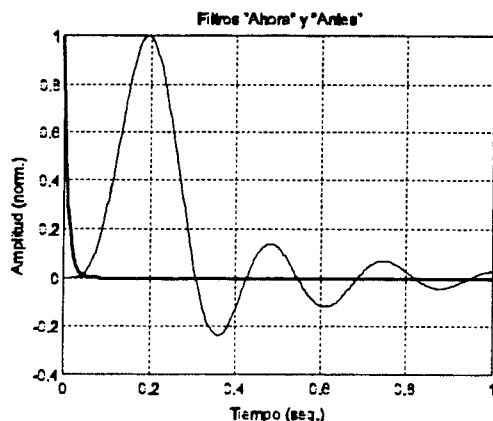


Figura 4: Respuestas al impulso de los filtros "Ahora" y "Antes"

7. RESULTADOS

Los resultados que presentaremos a continuación han sido obtenidos utilizando un programa de simulación en el entorno Matlab. Los datos de partida son los pasos por cero extraídos de una placa que contiene el micrófono y todo el sistema de adquisición de parámetros.

La Figura 5 muestra dos curvas, correspondientes al grado de verdad (o grado de reconocimiento) de los

vocablos "SI" y "NO", respectivamente. Concretamente, se muestran dichos grados de verdad cuando se pronuncia la secuencia de palabras "Sí...No" delante del micrófono. Como puede observarse, el grado de verdad del "SI" toma un valor alto justamente después de haberse producido una combinación de sonido silbante seguido por un sonido vocálico. De forma parecida, el grado de verdad del "NO" presenta un máximo después de haberse dado un sonido nasal seguido de uno vocálico. Ambos máximos tienen la suficiente amplitud como para permitir su fácil detección mediante, por ejemplo, un circuito comparador-temporizador que active un diodo LED durante un cierto tiempo.

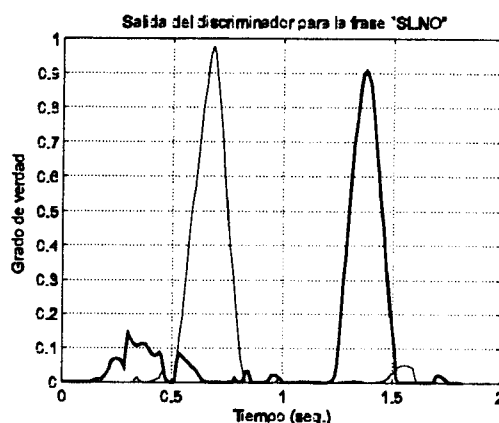


Figura 5: Salida del discriminador habiendo pronunciado la frase "SI..NO"

8. CONCLUSIONES Y LÍNEAS FUTURAS

Como se ha puesto de manifiesto, el uso de técnicas temporales difusas permite discriminar fenómenos que se producen en instantes distintos, pero cuya duración y localización (además de su amplitud) son imprecisas. Dichas técnicas se han aplicado con éxito a la construcción de un prototipo capaz de discriminar entre los vocablos "SI" y "NO" usando íntegramente tecnología analógica, lo que lo convierte en un dispositivo muy económico y de reducidas dimensiones, que admite una fácil integración si se toman como base sistemas de inferencia difusa analógicos existentes actualmente. En este sentido, el presente prototipo, desarrollado inicialmente con componentes discretos, se prevé integrarlo usando el controlador difuso analógico propuesto por Vidal y Rodríguez-Vázquez [9], [10].

Por otra parte, se ha comprobado que la parametrización de la señal vocal mediante los cruces por cero en las bandas paso-bajo y paso-alto de la señal es más que suficiente para los propósitos marcados. Esto sugiere la posibilidad de ampliar el número de palabras a reconocer manteniendo básicamente la misma parametrización. En

este sentido, se tiene proyectada la construcción de un prototipo capaz de reconocer los números del cero al nueve, con independencia del locutor. También se puede pensar en usar una parametrización menos tosca de la señal vocal para aumentar la potencia de reconocimiento del dispositivo.

Como nota final, comentar que el potencial campo de aplicación de esta técnica es amplio, aunque a juzgar por su económica implementación sus aplicaciones más inmediatas podrían centrarse en juguetería (control por voz económico), telefonía (marcaje por voz) y, en general, cualquier aplicación en donde se requiera discriminar entre un número reducido de vocablos sin tener que recurrir a las técnicas clásicas de procesamiento de señal, más precisas pero también más costosas.

Agradecimientos

Quisiéramos agradecer especialmente a Fernando Vidal-Verdú, de la Universidad de Málaga, la ayuda prestada en la realización de este proyecto.

Referencias

- [1] Orestes Mas y Pere Palà, "Étude des systèmes de Temps Flou et sa Réalisation avec Circuits Analogiques", *Actes des Rencontres francophones sur la Logique Floue et ses applications* pp. 208-214, Paris, Noviembre 1995.
- [2] Orestes Mas *et al.*, "Razonamiento aproximado en presencia de conceptos temporales difusos", *Actas del VI Congreso de la Asociación Española de Tecnologías y Lógica Fuzzy (ESTYLF'96)*, Oviedo, Septiembre 1996.
- [3] Francisco Casacuberta y Enrique Vidal, *Reconocimiento automático del habla*; Marcombo, Barcelona, 1987.
- [4] Mabo Robert Ito and Robert W. Donaldson, "Zero-Crossing Measurements for Analysis and Recognition of Speech Sounds", *IEEE Trans. On Audio and Electroacoustics*, vol. AU-19, No. 3, pp. 235-242, Septiembre 1971.
- [5] J.A. Rausell *et al.*, "Parametrización de la voz mediante μP (parte II)", *Mundo Electrónico*, No. 142, pp. 51-57, 1984.
- [6] Robert W. A. Scarr, "Zero Crossings as a Means of Obtaining Spectral Information in Speech Analysis", *IEEE Trans. On Audio and Electroacoustics*, vol. AU-16, No. 2, pp. 247-255, Junio 1968.
- [7] Jyh-Shing R. Jang, "ANFIS: Adaptive Network-Based Fuzzy Inference System", *IEEE Trans. On Systems, Man and Cybernetics*, vol. 23, No. 3, pp. 665-685, May/June 1993.
- [8] Jerry M. Mendel, "Fuzzy logic systems for engineering: a tutorial", *Proceedings IEEE*, vol. 83, No. 3, pp. 345-377, March 1995.
- [9] Fernando Vidal y Ángel RodríguezVázquez, "A design approach for Neuro/Fuzzy chips", *Actas del IV Congreso de la Asociación Española de Tecnologías y Lógica Fuzzy (FUZZY'94)*, pp. 183-188, Blanes, Septiembre 1994.
- [10] Fernando Vidal y Ángel RodríguezVázquez, "Aprendizaje supervisado de reglas en un controlador difuso analógico", *Actas del V Congreso de la Asociación Española de Tecnologías y Lógica Fuzzy (ESTYLF'95)*, pp. 239-244, Murcia, Septiembre 1995.

RAZONAMIENTO APROXIMADO EN PRESENCIA DE CONCEPTOS TEMPORALES DIFUSOS

Orestes Mas

conomc@tsc.upc.es

Pere Palà

pere@tsc.upc.es

Joan M. Miró

conjms@tsc.upc.es

Universitat Politècnica de Catalunya - Dept. T.S.C.

c/Gran Capità s/n.

08024 BARCELONA

Resumen

El presente trabajo estudia cómo efectuar el proceso de inferencia en presencia de conceptos temporales difusos. Primeramente se hace un estudio del tipo de conceptos difusos que aparecerán en problemas del entorno de ingeniería. En segundo lugar se expone el formalismo y la notación usadas para describir los citados conceptos. Posteriormente se muestra un algoritmo que permite efectuar el proceso de inferencia mediante el uso de una suma de convolución (integral para el caso continuo). Se muestra finalmente cómo utilizar el algoritmo propuesto para la implementación *hardware* de la inferencia difusa en presencia de conceptos temporales.

Palabras Clave: Inferencia Difusa, Conceptos Temporales Difusos, Convolución.

1. INTRODUCCIÓN

Desde que el Prof. Zadeh introdujo, hace ahora 31 años, la Teoría de Conjuntos Difusos [1], esta rama del conocimiento ha experimentado un desarrollo constante, multiplicándose sus aplicaciones en aquellas situaciones en las que interese implementar de forma automática procesos de razonamiento similares a los humanos. En todas ellas, el núcleo central lo constituye un Sistema Lógico Difuso (SLD), el cual, en esencia, realiza una transformación no lineal entre un vector de datos de entrada (x) y otro vector de salida (y) en base a una serie de reglas difusas. Estos vectores describen en su gran mayoría magnitudes físicas o psicológicas difusas como temperatura, color, altura, satisfacción, etc., cuyo denominador común, borrosidad aparte, es que son *atemporales*.

Sin embargo, existen ciertas situaciones en donde aparece el *tiempo* de forma difusa, como por ejemplo problemas de diagnóstico médica [2], consulta de bases de

datos [3], de reconocimiento de voz, control, etc. A consecuencia de ello, distintos autores han propuesto en los últimos años varios formalismos para representar y tratar magnitudes temporales de forma imprecisa en general, y difusa en particular. En esta última situación resultan especialmente interesantes las formulaciones de Dubois y Prade [4], y Barro *et al.* [5], las cuales, basándose en la teoría de la posibilidad, introducen conceptos como *fecha e intervalo* de cara a posibilitar el tratamiento de problemas en donde aparezcan conceptos temporales difusos de forma explícita. Un ejemplo de este tipo lo encontramos en la sentencia (1):

“La duración de la fiebre fue de unos cuatro días” (1)

Otro tipo de situación muy distinta se da cuando en la descripción del problema aparecen conceptos temporales de forma implícita. En efecto, considérese la proposición siguiente, en la que la variable “temperatura” es una función dependiente del tiempo que proviene de un sensor:

“La temperatura antes era grande” (2)

En esta expresión, es evidente que el concepto “grande” es difuso, pero el concepto “antes” lo es también. Por ende, el concepto “antes” aparece aquí en forma implícita, pues no nos interesa conocer su valor sino el valor que tomó la variable temperatura en ese lapso difuso de tiempo.

El presente trabajo estudia cómo efectuar el proceso de inferencia en presencia de conceptos temporales difusos. Por otra parte se ha hecho especial hincapié en orientar los resultados obtenidos hacia una realización *hardware* sencilla de los algoritmos, dado que buena parte de los esfuerzos de investigación en el campo de los sistemas difusos se destinan actualmente al desarrollo de circuitos electrónicos que, de forma digital o analógica, integran todas las funciones de un SLD y permiten la ejecución en tiempo real de los mencionados algoritmos.

2. REPRESENTACIÓN DE CONCEPTOS TEMPORALES DIFUSOS

Siguiendo la formulación de Dubois y Prade citada anteriormente [4], podemos representar conceptos temporales difusos mediante distribuciones de posibilidad temporales, es decir, mediante funciones que mapean el eje de tiempos en el intervalo $[0, 1]$. Denotaremos a un elemento genérico del eje de tiempos (soporte) con la letra τ . A efectos de acotar el conjunto de situaciones que pueden presentarse en el estudio de conceptos temporales difusos, conviene clasificar éstos en distintas categorías: En primer lugar distinguiremos entre conceptos temporales *abruptos* y *difusos*, representando los primeros mediante *singletons* y los segundos con funciones de pertenencia cuyo soporte incluya más de un elemento. En segundo lugar, y sin que sea excluyente con el caso anterior, diferenciaremos entre conceptos temporales *absolutos* y *relativos al instante actual*. La diferencia fundamental entre estos dos tipos es que las funciones de pertenencia de los segundos dependen de τ y de t (siendo t el instante actual), mientras que las de los primeros dependen sólo de τ .

Considerando, pues, lo expuesto, definiremos nuestros conceptos temporales difusos mediante funciones de pertenencia que dependerán, en general, de t y τ . Estudiando los distintos tipos de conceptos temporales difusos que aparecen en las aplicaciones de ingeniería, se observa que dichos conceptos verifican mayoritariamente las propiedades siguientes:

- a) **No se trabaja con valores futuros de las variables:** No es posible razonar en base a valores futuros de las entradas en el antecedente de las reglas, puesto que no son conocidos en el momento actual. Tampoco puede haber valores futuros de las salidas en el consecuente de las reglas puesto que no puede modificarse la salida futura en el instante actual. Esto conlleva que las funciones de pertenencia de los conceptos temporales sean cero para $\tau > t$.

$$\mu(t, \tau) = 0 \quad \forall \tau > t$$

- b) **Los conceptos temporales son relativos al instante actual:** Los conceptos temporales difusos que manejamos en aplicaciones de ingeniería raramente -o nunca- se refieren a fechas concretas del pasado, sino que casi todos describen conceptos relativos al instante actual. En esta situación, las funciones de pertenencia no son bidimensionales sino que dependen únicamente de la diferencia entre t y τ .

$$\mu(t, \tau) = \mu(t - \tau)$$

- c) **No se pueden modificar valores pasados de la salida.** Debido a ello, en el consecuente de las reglas sólo podrán aparecer los valores actuales de

las salidas (no podrán contener conceptos temporales difusos).

A modo de ejemplo se muestra en la Figura 1 la representación de los conceptos temporales "hace 2 segundos" (abrupto) y "antes" (difuso), correspondiendo τ al eje de tiempos (todos los instantes posibles) y t al instante actual. Nótese que el concepto "hace 2 segundos", al ser abrupto, se representa mediante un *singleton* situado en $\tau = t-2$, mientras que el concepto "antes" se representa mediante un conjunto difuso con soporte τ . En este caso, los dos conceptos temporales son relativos al instante actual. Obsérvese también que, al ser conceptos relativos, su posición en el eje de tiempos cambia continuamente a medida que t crece.

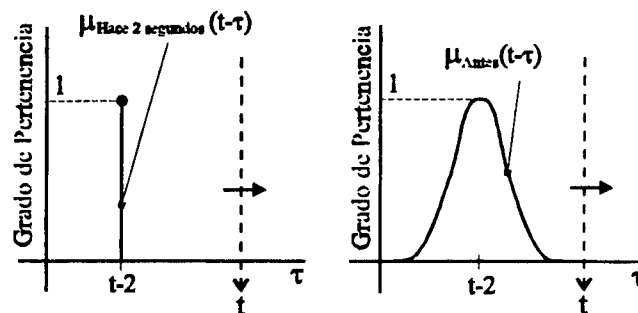


Figura 1: Representación de conceptos temporales abruptos y difusos.

Finalmente, destacaremos que puede emplearse una formulación equivalente para el caso discreto, denotando por k un elemento genérico del soporte y por n al valor actual, y describiendo nuestros conceptos temporales mediante funciones de pertenencia discretas de la forma $\mu(n-k)$.

3. EXPRESIONES DIFUSAS DEPENDIENTES DEL TIEMPO

Llamaremos *reglas de tiempo difuso* a aquellas reglas difusas SI-ENTONCES en las que aparezcan conceptos temporales de forma implícita, en el sentido definido en (2). Nuestro objetivo es establecer cómo debe realizarse la inferencia difusa en presencia de los mencionados conceptos. Para ello partimos de la *regla composicional de inferencia* [6], que nos permite razonar en base al *modus ponens* generalizado, es decir, nos permite encontrar un conjunto difuso de salida B' partiendo de una entrada A' y de una regla Si-Entonces que relaciona A con B (Tabla 1)

Tabla 1: Modus Ponens Generalizado

Premisa 1 (hecho):	x es A'
Premisa 2 (regla):	Si x es A entonces y es B
Consecuente (conclusión):	y es B'

Asumiendo que la implicación difusa $A \rightarrow B$ se expresa mediante una relación difusa R en $X \times Y$, la regla composicional de inferencia puede expresarse mediante la expresión

$$\mu_{B'}(y) = \sup_x \{ \mu_{A'}(x) * \mu_R(x, y) \} \quad (3)$$

en donde el operador $*$ representa cualquier t-norma difusa. Dado que cualquier regla difusa que posea más de un antecedente y/o más de un consecuente admite su descomposición en un conjunto de reglas difusas con un sólo antecedente y consecuente [7], [8], efectuaremos la discusión para este último caso. Según la t-norma y el operador implicación que se escojan se obtendrán diversas particularizaciones de la regla composicional de inferencia. Tomando el mínimo como t-norma a la vez que como implicación (implicación de Mamdani) obtenemos

$$\begin{aligned} \mu_{B'}(y) &= \sup_x \{ \mu_{A'}(x) \wedge (\mu_A(x) \wedge \mu_B(y)) \} \\ &= \sup_x \{ \mu_{A'}(x) \wedge \mu_A(x) \} \wedge \mu_B(y) \\ &= w \wedge \mu_B(y) \end{aligned} \quad (4)$$

En otras palabras, debemos primero encontrar el grado de parecido w entre A y A' como el supremo de $\mu_{A'}(x) \wedge \mu_A(x)$; entonces la función de pertenencia de B' es igual a la de B recortada al valor w . La parte más importante del proceso de inferencia se reduce, pues, a encontrar el citado valor w , que no es más que el *grado de verdad del antecedente* de la regla difusa para una situación de entrada concreta.

4. OBTENCIÓN DEL GRADO DE VERDAD DEL ANTECEDENTE

En este apartado se expone el método para encontrar el grado de verdad del antecedente de reglas difusas en las que aparece el tiempo en forma implícita. Sin pérdida de generalidad, haremos la discusión para el caso discreto, indicando al final las expresiones que resultan para el caso continuo. Sea pues un sistema difuso con una secuencia de entrada $x(n)$ y cuya salida es la secuencia $y(n)$. En un entorno de ingeniería, estas señales se conectan a los distintos sensores y actuadores de los que se compone el sistema, por lo que en general las consideraremos señales abruptas (*crisp*). Supongamos que el antecedente de una de las reglas difusas contenidas en la base de conocimiento del sistema es

$$"x \text{ antes era grande.}" \quad (5)$$

Si el concepto "antes" fuese abrupto y, por ejemplo, equivalente a "hace 4 muestras", el grado de verdad de la expresión (5) se obtendría fácilmente encontrando cómo era de grande el valor de x hace cuatro muestras

$$w = \mu_{\text{Grande}}(x(n-4)). \quad (6)$$

Tomemos ahora el concepto "antes" como difuso, describiéndolo mediante una función de pertenencia que asigna a cada instante de tiempo k su grado de pertenencia al concepto "antes", es decir $\mu_{\text{Antes}}(n-k)$. En notación de Zadeh [1],

$$\begin{aligned} "Antes" &= \left\{ \frac{\mu_{\text{Antes}}(n-k)}{k} \right\} \\ &= \left\{ \frac{\mu_{\text{Antes}}(m)}{n-m} + \frac{\mu_{\text{Antes}}(m-1)}{n-m+1} + \dots + \frac{\mu_{\text{Antes}}(0)}{n} \right\} \end{aligned} \quad (7)$$

con ello, el grado de verdad (6) se convierte en

$$w = \mu_{\text{Grande}}(x(\text{Antes})) \quad (8)$$

en donde $x(\text{Antes})$ es la imagen del conjunto difuso "Antes" por la función (*crisp*) x . Dicha imagen resulta ser a su vez un conjunto difuso en el universo X , al que llamaremos "XAnterior" ("XA"). Su función de pertenencia, $\mu_{XA}(x)$, puede calcularse usando el *principio de extensión* de Zadeh [9], mediante la expresión

$$\begin{aligned} \mu_{XA}(x) &= \sup_k \{ \mu_{\text{Antes}}(n-k) \} \\ "XAnterior" &= \left\{ \frac{\mu_{XA}(x)}{x} \right\} = \left\{ \frac{\sup_k \{ \mu_{\text{Antes}}(n-k) \}}{x(k)} \right\} \end{aligned} \quad (9) \quad (10)$$

es decir, el conjunto difuso imagen, "XA", contiene todos aquellos elementos de X que son imagen de algún $k \in \text{"Antes"}$ a través de la función $x(k)$. El grado de pertenencia de un elemento cualquiera $x_0 = x(k_0)$ al conjunto "XA" es el mismo que el grado de pertenencia del elemento k_0 al conjunto "Antes", con la salvedad de que si dos o más k 's distintas tienen la misma imagen, se toma el grado de pertenencia mayor. Todo el proceso se ilustra en la Figura 2.

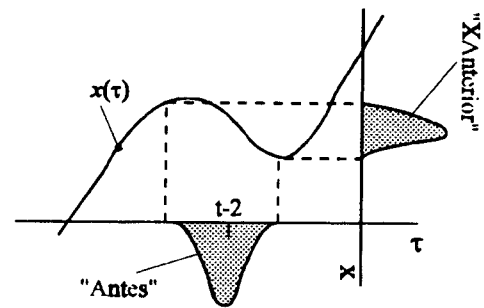


Figura 2: Aplicación del principio de extensión al conjunto "Antes".

Llegados a este punto, encontrar el grado de verdad w de la expresión (5) se reduce a encontrar cómo es de grande el conjunto XAnterior, situación que se muestra de forma gráfica en la Figura 3. Adviértase que no se trata de comparar los dos conjuntos difusos, sino que lo que hay que encontrar es el *grado de inclusión* del conjunto "XA"

en el conjunto "Grande". En efecto, en el supuesto de que la función $x(k)$ fuese constante e igual a un valor que considerásemos "Grande", la imagen por x de cualquier conjunto difuso temporal sería un *singleton* en X , ubicado dentro del conjunto "Grande"; en esta situación, deberíamos concluir que el valor de $x(k)$ ha sido grande en todo momento, con lo que el grado de verdad de (5) debería ser 1 aunque, de hecho, los conjuntos "XA" y "Grande" sean muy distintos.

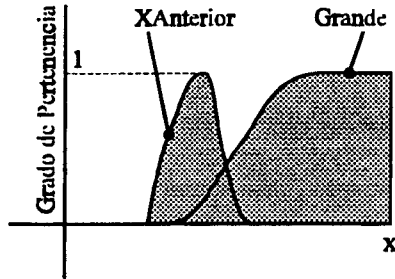


Figura 3: Conjuntos difusos "XAnterior" y "Grande".

Aunque es posible proponer varias fórmulas que representen el grado de inclusión de un conjunto difuso A en otro conjunto difuso B, que denotaremos por $S(A,B)$, la más extendida es [10]

$$S(A,B) = \frac{1}{|A|} \left(|A| - \sum_{x \in X} \max(0, \mu_A(x) - \mu_B(x)) \right) \quad (11)$$

donde $|A|$ representa la *cardinalidad escalar* del conjunto difuso A, definida como

$$|A| = \sum_{x \in X} \mu_A(x). \quad (12)$$

Dado que

$$\max(0, \mu_A(x) - \mu_B(x)) = \begin{cases} 0 & \text{si } \mu_A(x) < \mu_B(x) \\ \mu_A(x) - \mu_B(x) & \text{si } \mu_A(x) \geq \mu_B(x) \end{cases} \quad (13)$$

la ecuación (11) puede ponerse fácilmente como

$$\begin{aligned} S(A,B) &= \frac{1}{|A|} \left(\sum_{x \in X} \min(\mu_A(x), \mu_B(x)) \right) \\ &= \frac{|A \cap B|}{|A|} \\ &= \frac{\sum_{x \in X} (\mu_A(x) \wedge \mu_B(x))}{\sum_{x \in X} \mu_A(x)} \end{aligned} \quad (14)$$

Aplicando el resultado general dado por (14) al cálculo del grado de verdad de la expresión (5), se obtiene

$$w = \frac{\sum_{x \in X} \mu_{XA}(x) \wedge \mu_{Grande}(x)}{\sum_{x \in X} \mu_{XA}(x)} \quad (15)$$

Puesto que x es, de hecho, una función del tiempo k , los sumatorios de la expresión (15) pueden realizarse en la variable k :

$$w = \frac{\sum_{k \in K} \left(\sup_x \{ \mu_{Antes}(n-k) \} \wedge \mu_{Grande}(x(k)) \right)}{\sum_{k \in K} \sup_x \{ \mu_{Antes}(n-k) \}} \quad (16)$$

Como se ha comentado anteriormente, en la ecuación (16) el supremo debe de tomarse entre todos los grados de pertenencia $\mu_{Antes}(n-k_i)$ correspondientes a aquellas k_i que tengan la misma imagen $x(k_i)$. Sin embargo cabe señalar que esta situación, aunque posible en teoría, no se dará nunca en la práctica debido a que, aún en el caso de una entrada $x(k)$ constante, bastarán variaciones infinitesimales (debidas al ruido u otras causas) en el valor de $x(k)$ para poder considerar que cada k tiene una imagen distinta. En este caso, pues, podemos obviar la función *supremo* en la ecuación (16), con lo que ésta se convierte en la ecuación (17), en la que los índices del sumatorio se han puesto teniendo en cuenta que cualquier concepto temporal difuso es cero para $k > n$:

$$w = \frac{\sum_{k=-\infty}^n (\mu_{Antes}(n-k) \wedge \mu_{Grande}(x(k)))}{\sum_{k=-\infty}^n \mu_{Antes}(n-k)} \quad (17)$$

El resultado (17) tiene validez general, y admite su implementación directa. Hay que notar, sin embargo, que si bien en el numerador de (17) debe usarse el *mínimo* como t-norma difusa, el uso de otras t-normas válidas arrojará resultados parecidos, algunos de los cuales pueden ser de interés desde el punto de vista de la realización. Por ejemplo, tomando el *producto* como t-norma difusa, la ecuación (17) se convierte en

$$w \cong \sum_{k=-\infty}^n \bar{\mu}_{Antes}(n-k) \cdot \mu_{Grande}(x(k)) \quad (18)$$

en donde $\bar{\mu}_{Antes}(n-k)$ es el concepto temporal "Antes" normalizado a área unidad. La expresión equivalente a (18) para el caso continuo es

$$w \cong \int_{-\infty}^t \bar{\mu}_{Antes}(t-\tau) \cdot \mu_{Grande}(x(\tau)) \cdot d\tau \quad (19)$$

Los resultados (18) y (19) son de extraordinario interés, pues expresan el grado de verdad de (5) en forma de una *suma de convolución* (*integral de convolución* para el caso continuo), cuya realización *hardware* es inmediata usando técnicas digitales o analógicas clásicas. En efecto, a la vista de las expresiones (18) y (19) podemos encontrar el grado de verdad w como la salida de un sistema lineal con respuesta impulsional dada por $\bar{\mu}_{Antes}(n)$ y cuya entrada sea $\mu_{Grande}(x(n))$, como se muestra en la Figura 4. El diagrama de bloques es idéntico para el caso continuo, simplemente sustituyendo

n por t . El proceso es fácilmente extensible a más de una variable de entrada y más de una regla, con la simple adición de un filtro lineal por cada concepto temporal difuso que se use en los razonamientos.

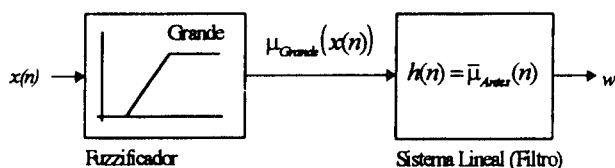


Figura 4: Realización de la ecuación (18)

5. CONCLUSIONES

El desarrollo expuesto en los apartados anteriores muestra que el proceso de inferencia difusa tradicional se ve modificado cuando en las reglas aparecen conceptos temporales difusos. Sin embargo, las modificaciones se reducen básicamente a la incorporación de un filtro lineal a la salida del fuzzificador, y antes de afectar al consecuente de la regla. Esto tiene la ventaja de poder construirse analógicamente de forma muy económica. Indicaciones sobre la implementación concreta de las respuestas impulsionales dadas por $\bar{\mu}_{Antes}(n)$ pueden encontrarse en [11].

Cabe señalar que la inclusión de un filtro dentro de un sistema difuso tradicional convierte a éste último en un sistema dinámico no lineal, con lo que puede presentar fenómenos como oscilaciones, caos, etc. En este sentido, se está en fase de estudio de una familia de osciladores basados en técnicas temporales difusas. Asimismo, se han utilizado estas técnicas para construir un módulo capaz de discriminar entre un conjunto reducido de vocablos pronunciados por distintos locutores, en un entorno de reconocimiento sencillo de la palabra.

Referencias

- [1] L.A.Zadeh, "Fuzzy Sets", *Information and Control*, vol. 8, pp. 338-353, 1965.
- [2] Roque Marín *et al.*, "Una aproximación al razonamiento temporal borroso en medicina", *Actas del IV Congreso Español sobre Tecnologías y Lógica Fuzzy*, pp. 53-58, Murcia, 1994.
- [3] S. De *et al.*, "Natural Language Query Processing in a Temporal Database", *Data and Knowledge Eng.*, vol. 1, pp. 3-15, 1985.
- [4] D.Dubois and H.Prade, "Processing Fuzzy Temporal Knowledge", *IEEE Trans. On System, Man and Cybernetics*, vol. 19, 1989.

- [5] S.Barro, R.Marín, J.Mira and A.R.Paton, "A model and a language for the fuzzy representation and handling of time", *Fuzzy Sets and Systems*, vol. 61, pp. 153-175, 1994.
- [6] L.A.Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", *IEEE Trans. On System, Man and Cybernetics*, vol. SMC-3, No. 1, 1973.
- [7] Jerry M. Mendel, "Fuzzy Logic Systems for Engineering: A Tutorial", *Proceedings of the IEEE*, vol. 83, No. 3, March 1995.
- [8] Jyh-Shing Roger Jang and Chuen-Tsai Sun, "Neuro-Fuzzy Modeling and Control", *Proceedings of the IEEE*, vol. 83, No. 3, March 1995.
- [9] L.A.Zadeh, "The concept of a linguistic variable and its application to approximate reasoning. Part Y", *Information Sciences*, Vol. 8, pp. 199-251, 1975.
- [10] George J. Klir and Bo Yuan, *Fuzzy Sets and Fuzzy Logic. Theory and Applications*, Prentice-Hall, Upper Saddle River, New Jersey, 1995.
- [11] Orestes Mas y Pere Palà, "Étude de Systèmes de Temps Flou et leur réalisation avec circuits analogiques", *Rencontres Francophones sur la Logique Floue et ses Applications*, pp. 208-214, Paris, Noviembre 1995.

EVOLUCIÓN GENÉTICA DE CONTROLADORES BORROSOS

Vicente Matellán
Laboratorio de Agentes Inteligentes
Departamento de Informática
Universidad Carlos III de Madrid
E-mail: vmate@ia.uc3m.es

José M. Molina
Laboratorio de Agentes Inteligentes
Departamento de Informática
Universidad Carlos III de Madrid
E-mail: molina@ia.uc3m.es

Camino Fernández
Grupo de Vida Artificial
Departamento de Informática
Universidad Carlos III de Madrid
E-mail: camino@inf.uc3m.es

Resumen

Este trabajo aborda el aprendizaje de controladores borrosos básicos para robots autónomos. Se presenta un método para la adaptación de comportamientos reactivos básicos implementado en forma de controladores borrosos y aplicando un algoritmo genético a la evolución del sistema de reglas borrosas. Se muestran los experimentos realizados en cuanto a la evolución de las reglas de control basadas en conceptos simbólicos representados como etiquetas lingüísticas. Para las reglas se ha utilizado lógica borrosa y para probar los conjuntos de reglas obtenidos en cada generación del algoritmo genético se ha utilizado un robot real. El individuo que presenta los mejores resultados es el elegido para la generación de una nueva población. Este proceso se conoce con el nombre de estrategia *elitista*. Todos los individuos se prueban en un entorno real. Al final del proceso, los individuos de la última generación ofrecen un conjunto de reglas que proporciona mejores resultados que los obtenidos por un diseñador inexperto.

Palabras clave: Robots Autónomos, Algoritmos Genéticos, Aprendizaje.

1 DESCRIPCIÓN DEL PROBLEMA

Los controladores borrosos se han utilizado profusamente para el control de distintos tipos de robots autónomos. Por ejemplo, en [1] se presenta la arquitectura LIFIA. Esta arquitectura es una jerarquía de niveles en la que cada uno de ellos trabaja de forma asíncrona, con su propio nivel de abstracción de datos, en la que el módulo de navegación consiste en un

módulo reactivo basado en lógica borrosa. Otro ejemplo clásico de sistema autónomo controlado mediante comportamientos borrosos es [2]. Este sistema consiste en un coche que se mueve de forma autónoma gracias a un grupo de conjuntos de reglas borrosos.

Para presentar el mecanismo de aprendizaje utilizado en la adaptación de las reglas de control borrosas, veremos en un primer subapartado algunos conceptos básicos relativos a los controladores que utilizan lógica borrosa. Utilizaremos el segundo subapartado para presentar la estrategia de diseño y el tercero para describir el proceso de adaptación.

1.1 CONTROLADOR BORROSO

El primer paso en el diseño de un controlador borroso es la selección de una descripción adecuada de las entradas relevantes para el control, como puede ser la *distancia* a los obstáculos, de forma análoga a la que utilizaría una persona para su descripción. Así es que, dada una distancia numérica d_i a un obstáculo detectado por un sensor, D_i se define como el rango de todos los posibles valores de la distancia d_i . Con el fin de manejar la incertidumbre que subyace bajo la apariencia de entradas de los sensores (distorsionada después del proceso de adquisición), los valores numéricos de la distancia d_i se pueden traducir a etiquetas simbólicas cualitativas a través de un proceso de "fuzzificación" [3], transformando la entrada de los sensores en variables lingüísticas.

Una variable lingüística, según [3], es una variable cuyos valores son frases de un lenguaje natural o artificial, es decir, la concatenación de términos atómicos: etiquetas (adjetivos), modificadores (como "muy", "poco", "algo", etc.), la negación y las marcas (paréntesis). El significado de una variable lingüística se define como un subconjunto borroso para el que el valor de la variable lingüística se utiliza como una etiqueta. Un subconjunto borroso A de un universo del discurso U viene dado por una función de pertenencia $\mu : U \rightarrow [0, 1]$ que relaciona con cada elemento y de

U un número $\mu_A(y)$ que representa el grado de pertenencia de y a A . La operación de "fuzzificación" (que depende de la aplicación) produce como efecto la transformación de un conjunto no borroso o cantidad en un conjunto borroso. Conviene destacar en este punto, que el valor de, por ejemplo, la variable lingüística *distancia* (una etiqueta natural como *cerca*) representa un significado mucho menos preciso que su correspondiente valor numérico de cuántos centímetros hay hasta el obstáculo.

Utilizando estos conceptos, para cada d_i , una variable lingüística Ld_i se introduce junto con su conjunto de valores $ld_{i1}, ld_{i2}, \dots, ld_{im}$, cuya cardinalidad es m_i . Cada ld_{ij} del conjunto etiqueta un subconjunto borroso en el universo del discurso D_i , con la función de pertenencia $\mu_{ld_{ij}}(d_i)$. Los valores de la función de pertenencia de una etiqueta tienen que ver con la dificultad de asignar dicha etiqueta a un calor d_i obtenido a partir de los sensores del robot. La operación de "fuzzificación" adoptada que afecta a los valores numéricos d_i , provocará su transformación en un "singleton" borroso [3] o subconjunto borroso cuya base es un único punto en D_i con la función de pertenencia igual a uno.

Un Algoritmo Relacional Difuso (FRA) [3] almacenará el conocimiento que se requiere para controlar un robot autónomo mediante un proceso de razonamiento borroso. Los FRA se componen de un conjunto finito de sentencias condicionales del tipo **IF** Ld_i **IS** ld_{ij} **THEN** $LMOT_i$ **IS** lm_k , donde el antecedente puede estar compuesto de conjunciones y/o disyunciones borrosas sobre las variables lingüísticas Ld_i y el consecuente de afirmaciones borrosas sobre $LMOT_i$, la variable lingüística que identifica la velocidad del motor i cuyo valor es $lm_1, lm_2, lm_3 \dots lm_n$. La implicación de Mandani [4] se ha elegido para la asignación de significado en estas sentencias condicionales. Así, el subconjunto borroso de pares ordenados (d_i, s) , donde $d_i \in D_i$ y $m \in MOT$, del producto cartesiano de $(ld_{ij} \times lm_k)$ con el grado de pertenencia dado por $\min(\mu_{ld_{ij}}(d_i), \mu_{lm_k}(s))$, donde s es la "defuzzificación" de $LMOT$ y MOT representa su dominio numérico (universo del discurso de $LMOT$).

El último aspecto a considerar es la estrategia de inferencia para manejar el conocimiento contenido en el FRA. La regla composicional de inferencia (CRI) propuesta por [3], extensión de la conocida regla de *modus ponens*, sirve como mecanismo de inferencia para obtener el subconjunto borroso de MOT , cuya función de pertenencia se obtiene después del producto max-min de versiones discretizadas de $\mu_{ld_{ij}}(d_i)$ y $\mu_{ld_{ij}}(d_i) \times ls_k(d_i, s)$, representado como matrices relacionales [3]. Dado que puede haber varias sentencias condicionales formando parte del FRA, el significado de $LMOT$ será la intersección de los signifi-

cados intermedios resultantes de cada aplicación del CRI ("min" de todos los consecuentes inducidos de las funciones de pertenencia). Finalmente, el proceso de "defuzzificación" adoptado para $LMOT$ será una versión simplificada del procedimiento del Centro de Gravedad. Este método trata las reglas separadamente. Cada regla produce un nivel de activación para las etiquetas λ_i . Consideremos $C_{i,k}$ como la representación numérica de cada etiqueta l, k (e.g. los centros de gravedad). Entonces, la salida se considera como una nueva especie de suma ponderada: $sum = (\sum \lambda_i C_{i,k}) / (\sum \lambda_i)$.

1.2 DISEÑO DE UN ALGORITMO RELACIONAL BORROSO

El diseño de las reglas para controlar un sistema autónomo utilizando un FRA no resulta un asunto trivial. Consideremos un sistema, similar al robot que hemos utilizado, con ocho señales de entrada (sensores) y dos salidas (motores). Codificaremos cada uno de los sensores de entrada con solo dos etiquetas lingüísticas (*near, far*) y cada una de las salidas con cinco etiquetas lingüísticas (*fast-advance, advance, stop, back, fast-back*). El número de posibles reglas borrosas es 6.400. Pero si hemos utilizado un alto grado de granularidad en nuestro sistema, por ejemplo, su hubiéramos codificado utilizando cinco etiquetas lingüísticas, el número de reglas habría sido 9.765.625. Utilizando una codificación de ocho etiquetas lingüísticas por variable, el número de reglas sería del orden de billones (1.073.741.824).

Lógicamente, solo se necesitan unas pocas reglas para obtener un comportamiento inteligente. El problema es cómo elegir las reglas. Hasta ahora, el método más comunmente utilizado ha sido preguntarle al experto las reglas. Sin embargo, los humanos suelen pensar de forma antropomórfica, lo que puede causar problemas.

Consideremos, por ejemplo, un robot autónomo simple como el de la figura 1. Este robot tiene dos sensores de proximidad (*sensor1* y *sensor2*) y dos motores (*motor1* y *motor2*). Cuando se le pide al experto que escriba las reglas que le permitan deambular por el mundo sin chocar con nada, las reglas obtenidas son algo parecido a **IF** *sensor1* **IS** *near* **THEN** *Motor1* **IS** *fast-advance*, lo que significa que cuando el sensor izquierdo percibe un obstáculo el motor izquierdo incrementa su velocidad para alejarse de él girando hacia la derecha. Análogamente, se definirá la regla simétrica.

Si se prueba este grupo de reglas en un robot, se comprueba que el robot comienza a incrementar su velocidad paulatinamente. El problema se puede definir

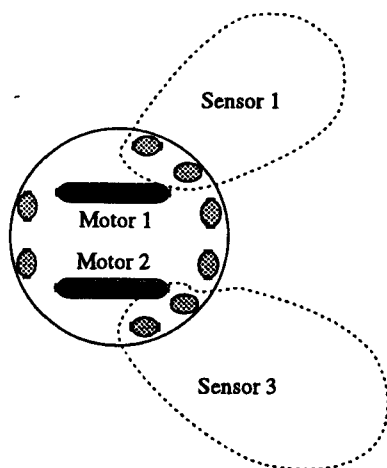


Figura 1: Robot para el Ejemplo 1

como un aumento continuo de la entropía. Se puede corregir fácilmente escribiendo las reglas de forma inversa - *IF sensor1 IS near THEN M2 IS slow* - y añadiendo una nueva regla para incrementar la velocidad cuando no hay obstáculos. Como una primera meta, pretendemos comprobar si este problema aparece cuando las reglas se obtienen a través de evolución genética.

Otro problema con las reglas diseñadas por humanos es que las diseñan para un mundo teórico: la mente humana. Esto significa que las reglas y las etiquetas lingüísticas se tienen que retocar muchas veces hasta alcanzar un comportamiento aceptable.

2 ADAPTACIÓN DE COMPORTAMIENTOS BORROSOS

Se pueden encontrar en la literatura algunos trabajos acerca del uso de métodos genéticos de aprendizaje de planes para autónomos, por ejemplo en [5], se presenta un método genético para formular nuevos conjuntos de reglas de decisión de bajo nivel para movimientos de robots. Cada regla comprueba que ciertas condiciones se cumplen (obstáculos detectados, posición de la meta, etc.), y ejecuta los operadores correspondientes (mover hacia delante, mover hacia atrás, girar a la derecha, girar a la izquierda). La competición genética tiene lugar entre conjuntos de reglas de comportamiento (plan) después de probar los planes.

Se han llevado a cabo algunos trabajos con relación a la evolución genética de conjuntos de reglas. Por ejemplo, en [6] se presenta un algoritmo genético para el diseño de las reglas borrosas capaces de centrar un carrito aplicando una fuerza simple. Otro ejemplo, [7],

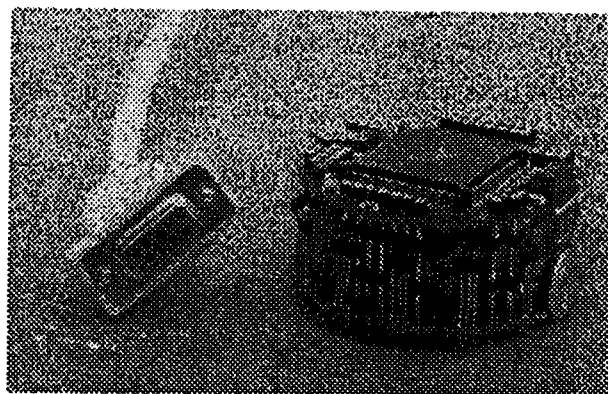


Figura 2: El Mini-Robot Khepera.

donde se presenta la aplicación de este método a tres sistemas físicos diferentes: un sistema de nivel de líquidos, un sistema de pH y un sistema aplicado al control de satélites. En cada una de estas aplicaciones, los controladores borrosos diseñados genéticamente mejoran los diseñados por los humanos. Sin embargo, se han hecho pocos experimentos sobre la evolución de las etiquetas lingüísticas borrosas. En [8] se aplica un método genético al problema clásico de control del péndulo invertido, determinando cada función de pertenencia para un modelo borroso TSK, el número de reglas borrosas y los parámetros de los consecuentes de las reglas.

Por último, se han propuesto algunos trabajos sobre la evolución de controladores borrosos aplicados al control de vehículos autónomos. Por ejemplo, en [9] se presenta un método general para la evolución de controladores borrosos basados en reglas. Otra metodología basada en una estructura jerárquica priorizada utilizando un algoritmo genético se aplica al control de un vehículo autónomo, [10].

En el resto de esta sección describiremos brevemente el robot utilizado en nuestros experimentos y presentaremos nuestro trabajo: evolución genética de un controlador borroso para el control de un robot.

2.1 EL MINI-ROBOT KHEPERA

El robot que se ha utilizado es el mini-robot KHEPERA [11], ver figura 2, que es un mini-robot comercial que ha sido desarrollado en el LAMI (EPFL, Lausane, Suiza). Este robot tiene forma circular con un diámetro de 5,5 cm., una altura de 3 centímetros y un peso de 70 gramos. Se mueve gracias a dos ruedas controladas por dos motores independientes que le permiten movimiento en cualquier dirección. El robot dispone también de ocho sensores de proximidad por infrarrojos.

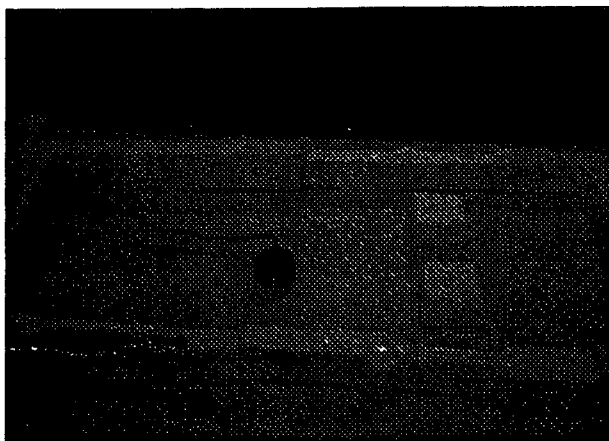


Figura 3: Entorno de Experimentación

El corazón del robot es el controlador 68331 de Motorola con 256 Kbytes de RAM y 512 Kbytes de ROM que maneja las rutinas de entrada/salida y puede comunicarse via puerto serie con un servidor. Dispone de sus propias baterías, lo que le permite trabajar de forma autónoma, aunque también puede hacerlo conectado a una estación de trabajo via puerto serie, lo que le permite hacer uso de los recursos de la misma.

Hemos preferido utilizar un robot real en lugar de una simulación por dos razones. En primer lugar, porque una perfecta simulación de un robot tan simple como el Khepera requiere un importante esfuerzo computacional. Por ejemplo, la simulación de los sensores, teniendo en cuenta la iluminación del laboratorio, orientación del robot, etc.; requeriría enormes cálculos. Y en segundo lugar, incluso una muy buena simulación sería incapaz de tener en cuenta todas las posibles contingencias como pueden ser la inercia, la fricción, los fallos del hardware, etc.

2.2 ADAPTACIÓN DE REGLAS BORROSAS

Nuestra primera meta consistió en comprobar si las reglas obtenidas a través de evolución genética eran capaces de controlar correctamente un robot autónomo. Así es que, se le proporcionaron al robot algunos conceptos previos como cerca, lejos, etc. para los sensores y *despacio* y *deprisa* para los motores. El programa genético deberá ser capaz de combinar estos conceptos con éxito. El robot se colocó dentro de un entorno sencillo consistente en un área rectangular con dos obstáculos situados como indica la figura 3. Las paredes eran de goma-espuma y la superficie la de una mesa de madera. La mesa estaba situada en un laboratorio iluminado siempre con luz artificial.

Cada uno de los sensores del robot devuelve un entero

cuyo rango va de cero (no se detecta ningún obstáculo) a 1023 (se detecta un obstáculo justo delante del sensor). La velocidad de los dos motores viene fijada por un entero. En este caso, el rango que hemos utilizado va de -10 (máxima velocidad hacia atrás) a 10 (máxima velocidad hacia delante). Aunque el robot tiene ocho sensores, los hemos agrupado de forma que el robot resultante se puede considerar equivalente al que aparece en la figura 1.

Para la realización del experimento hemos realizado cinco particiones borrosas de cada una de las cuatro variables (*sensor1*, *sensor2*, *motor1* y *motor2*). En este sentido, las estrategias de control individuales son dos tablas de 5x5, una para cada motor. Cada tabla relaciona las funciones de pertenencia de cada uno de los sensores con las de un motor. Las tablas están codificadas como cromosomas con alelos 0, 1, 2, 3, 4, 5 que corresponden a las funciones de pertenencia *fast-back*, *back*, *stop*, *fast-advance*, *advance* y *Nulo*. Donde el último indica que no hay ninguna relación entre las dos funciones de pertenencia.

Consideramos ambas tablas como un *genotipo*. De modo que cada individuo tiene un cromosoma de longitud cincuenta. Los alelos son las cinco funciones de pertenencia sobre el dominio de los motores más el código nulo. El *fenotipo* es el comportamiento que produce el controlador borroso. El comportamiento se obtiene aplicando las operaciones habituales: fuzzificación, max-composición y defuzzificación por el método del centroide. Nuestro criterio de adecuación, Θ , es función de tres variables medidas directamente sobre el robot Khepera y otra más sobre el genotipo.

$$\Theta = \frac{V(1 - \sqrt{D})(1 - I)}{\text{reglas}}$$

La variable I representa el valor normalizado del sensor que presenta el nivel más alto de activación:

$$I = \frac{\text{sensor}}{1023}$$

V es la velocidad media de rotación de las dos ruedas:

$$V = \frac{\text{media}}{10}$$

y D es el valor absoluto normalizado de la diferencia entre la velocidad de las dos ruedas:

$$D = \frac{|v_1 - v_2|}{20}$$

Lo que hace que la función Θ se maximice cuando se evitan obstáculos, se mantiene una dirección recta, se

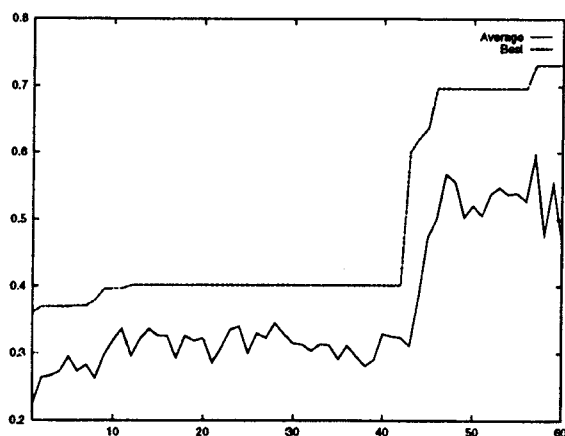


Figura 4: *Fitness* Medio vs. Mejor.

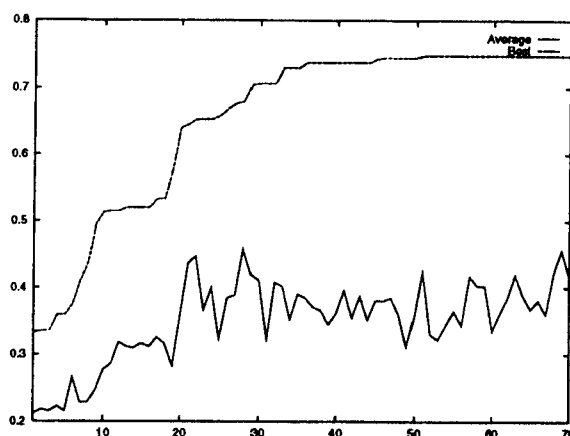


Figura 5: *Fitness* Medio vs. Mejor.

mantiene una velocidad (el robot no está parado) y se controla utilizando pocas reglas.

La evolución del entrenamiento se realizó a través de un algoritmo genético estándar que consiste en 100 generaciones de una población de 100 individuos. El operador de mutación cambia un código borroso un nivel arriba o abajo o a un valor nulo. El operador de crossover es el crossover estándar de dos puntos. El tiempo de vida de cada individuo se ha fijado en 30 segundos. Hemos utilizado una estrategia *elitista*, lo que significa que el mejor individuo promociona automáticamente a la siguiente generación.

En la figura 4 se puede apreciar la evolución de la adecuación media de cada generación y la adecuación del mejor individuo de cada generación en un experimento utilizando la siguiente configuración:

Obstáculos 1
Mutación 0.2
Crossover 0.2

La figura 4 nos permite evaluar el proceso de aprendizaje. Se comprueba que, aunque permitimos 100 generaciones, alrededor de la generación número 60, en la mayor parte de los experimentos, la población ha aprendido a evitar obstáculos y muchos de los individuos de las últimas generaciones utilizan menos de 40 reglas.

La figura 5 muestra el resultado del mismo experimento utilizando una configuración diferente:

Obstáculos 1
Mutación 0.4
Crossover 0.4

Esta figura muestra que si las probabilidades de mutación y crossover se incrementan, entonces la velocidad del proceso de aprendizaje crece. También se han llevado a cabo experimentos incrementando solo uno de

estos factores y variando el número de individuos de cada generación y el número de generaciones.

En la figura 6 se muestra la contribución de las tres partes que componen la función Θ frente al total de adecuación. Los dos valores (partes y total) que se muestran en la figura corresponden a la media de individuos de cada población. También se han probado las consecuencias de utilizar un factor para mejorar una de las habilidades, como puede ser "correr más deprisa" contra "mantenerse a salvo".

3 CONCLUSIONES

En este artículo se propone un método para evolucionar reglas de alto nivel utilizando algoritmos genéticos clásicos. A través de estos procesos de evolución, el mini-robot Khepera ha sido capaz de desarrollar un comportamiento autónomo que le permite sobrevivir en su entorno. El papel de los investigadores se ha limitado a proporcionarle el criterio de supervivencia y la estructura del controlador borroso.

El conjunto de reglas medio de los individuos de las últimas generaciones muestra que este método ofrece soluciones basadas en el número de reglas, que ha resultado ser menor de lo esperado, lo que constituye otro resultado fundamental del experimento.

Nuestro trabajo actual va encaminado hacia el uso de la misma aproximación en la emergencia de funciones de pertenencia borrosas y en la evolución de las dos partes del controlador borroso a la vez. Hasta la fecha se han obtenido algunos resultados interesantes en lo que se refiere a la evolución por separado de las funciones de pertenencia. Confiamos a su vez en la posibilidad de hacer evolucionar ambas al mismo tiempo.

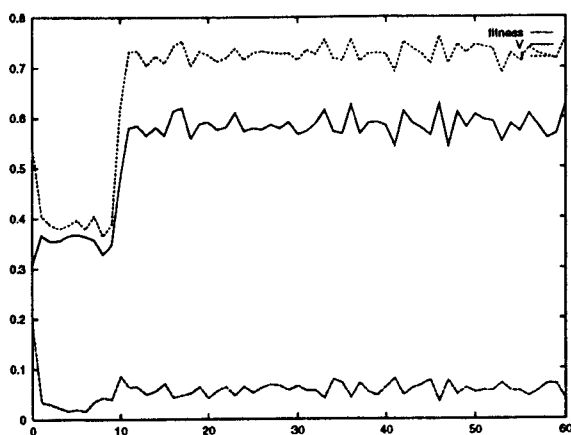


Figura 6: *Fitness* vs. los factores que lo componen.

Referencias

- [1] Patrick Reignier. Fuzzy logic techniques for mobile robot obstacle avoidance. *Robotics and Autonomous Systems*, (12):143-153, 1994.
- [2] Francois G. Pin and Yutaka Watanabe. Driving a car using reflexive fuzzy behaviors. In *Proceedings of the Second International Conference on Fuzzy Systems*, San Francisco, USA, 1993.
- [3] L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *Trans. on Systems, Man and Cybernetics*, 1(1), 1973.
- [4] Z. Zhong I. B. Turksen. An approximate analogical reasoning approach based on similarity measures. *Trans. on Systems, Man and Cybernetics*, 18(6), 1988.
- [5] Diane J. Cook. Using analytic and genetic methods to learn plans for mobile robots. In *Proceedings of the SPIE Conference on the Applications of Artificial Intelligence*, pages 327-336, 1993.
- [6] Philip Thrift. Fuzzy logic synthesis with genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 509-513, 1993.
- [7] Chuck Karr. Applying genetics to fuzzy logic. *AI Expert*, pages 38-43, March 1991.
- [8] Michael A. Lee and Hideyuki Takagi. Integrating design stages of fuzzy systems using genetic algorithms. In *Proceedings of the Second International Conference on Fuzzy Systems*, pages 612-617, 1993.
- [9] Mark G. Cooper. Evolving a rule-based fuzzy controller. *Simulation*, 65(1), 1995.
- [10] F. Hoffmann and G. Pfister. Automatic design of hierarchical fuzzy controller using genetic algorithms. In *Proceedings of the EUFIT 94*, Aachen, Germany, 1994.
- [11] Francesco Mondada and Franzi P. Ienne. Mobile robot miniaturisation: A tool for investigation in control algorithms. In *Proceedings of the third International Symposium on Experimental Robotics*, Kyoto, Japan, 1993.

JUEGOS SIMPLES DIFUSOS

Elisenda Molina

Dpto. Estadística e I. Operativa
Facultad de CC. Matemáticas
Universidad Complutense de Madrid
e-mail: emolina@sunes1.mat.ucm.es

Juan Tejada

Dpto. Estadística e I. Operativa
Facultad de CC. Matemáticas
Universidad Complutense de Madrid
e-mail: tejada@sunes1.mat.ucm.es

Resumen

Una clase de juegos n -personales de especial interés es la compuesta por los juegos simples, dicho interés se centra en que todo cuerpo legislativo de voto puede ser representado mediante un juego simple. En estos juegos el elemento cuantitativo puede ser tratado como algo secundario, el principal objetivo es formar ciertas coaliciones "decisivas". En este trabajo estudiamos la extensión de los juegos simples al caso en que se considere la formación de coaliciones difusas con el fin de recoger la imprecisión no aleatoria inherente a algunos problemas. Se analiza su estructura y se introducen distintos tipos de juegos.

Palabras clave: juegos cooperativos, juegos simples, coalición minimal ganadora, comité, coalición difusa.

El uso de conceptos ligados a la teoría de conjuntos difusos en los juegos n -personales permite mejorar la modelización de situaciones conflictivas en las que intervienen un conjunto de personas, permitiendo incorporar al modelo información de carácter ambiguo o impreciso no asimilable a ningún fenómeno aleatorio o probabilístico. En este trabajo consideraremos como elemento difuso la existencia de coaliciones difusas (Aubin [1]) en juegos n -personales cooperativos con utilidad transferible. Formalmente, una coalición difusa es un subconjunto difuso del conjunto N de jugadores, que se puede identificar con un vector $\tau \in [0, 1]^n$, donde τ_i indica el grado de pertenencia o tasa de participación del jugador i en la coalición difusa τ . Notaremos por $\mathcal{F}(N)$ al conjunto de coaliciones difusas de N . En particular, nos centraremos en el estudio de aquellos juegos que surgen al extender el concepto de

juego simple clásico al contexto de juegos cooperativos difusos.

Definición 1 (Aubin, 1974) *Un juego n -personal cooperativo con utilidad transferible y coaliciones difusas en forma de función característica viene dado por el par (N, v^*) , donde $N = \{1, \dots, n\}$ es el conjunto de jugadores y $v^* : [0, 1]^n \rightarrow \mathbb{R}$ es la función característica del juego que asigna a cada coalición difusa $\tau \in [0, 1]^n$ su valor $v^*(\tau)$, verificándose que $v^*(\emptyset) = 0$.*

Definición 2 *Definimos un juego simple difuso como un juego n -personal cooperativo con utilidad transferible y coaliciones difusas en forma de función característica, verificando:*

- (i) $v^*(\tau) \in [0, 1] \quad \forall \tau \in \mathcal{F}(N)$
- (ii) v^* es un juego monótono :
$$v^*(\tau) \leq v^*(\tau') \quad \forall \tau \subseteq \tau'$$
- (iii) $v^*(\bar{1}) = 1$

Estos juegos permiten aproximar situaciones reales cuyas características no pueden ser recogidas por los modelos clásicos. Por ejemplo, una votación en la que los jugadores emiten un voto de aceptación o rechazo hacia una cierta propuesta se puede modelizar mediante un juego simple con coaliciones nítidas ; sin embargo, si se quiere modelizar una negociación en la que los jugadores expresen opiniones vagas el modelo nítido resulta inadecuado, haciéndose necesaria la incorporación de coaliciones difusas en el modelo. Cada propuesta genera una coalición difusa τ en la que la tasa de participación de cada jugador viene dada por el grado de aceptación de la propuesta por parte de dicho jugador, el valor $v^*(\tau)$ de la coalición difusa τ representa el grado de consenso al que se llega si se forma esa coalición.

En el trabajo se estudian las propiedades estructurales de este tipo de juegos, para ello se extienden conceptos propios de los juegos simples nítidos al caso que nos

ocupa, y se introducen algunas clases de juegos simples difusos, analizándose su estructura así como las situaciones que modelizan.

1 COALICIONES GANADORAS Y COALICIONES DE BLOQUEO

En el modelo clásico el concepto de "coalición ganadora" es un concepto nítido: los miembros de una coalición $S \subseteq N$ disponen del poder suficiente como para conseguir que una propuesta sea aprobada en caso de ser aceptada por todos ellos (S ganadora), o bien, no disponen de dicho poder (S perdedora); mientras que en el modelo difuso se le debe dar un tratamiento difuso. La propiedad "ser una coalición ganadora" es una propiedad difusa, siendo $v^*(\tau)$ el grado en que la coalición τ verifica dicha propiedad. Así, el conjunto de coaliciones ganadoras del juego resulta ser un subconjunto difuso $\widetilde{W}$ de $\mathcal{F}(N)$ cuya función de pertenencia viene dada por la función característica del juego, $\mu_{\widetilde{W}} \equiv v^*$. Análogamente, el concepto de "coalición minimal ganadora" debe ser considerado como un concepto difuso, a continuación se introduce el concepto de *coalición minimal ganadora a nivel s* a partir del cual se define el *conjunto difuso de coaliciones minimales ganadoras*, conjunto que, al igual que ocurre en el caso nítido, determina la función característica del juego (proposición 1).

Definición 3 Dado el juego simple difuso (N, v^*) tal que v^* es continua por la derecha, decimos que $\tau^s \in \mathcal{F}(N)$ es una *coalición minimal ganadora a nivel $s \in [0, 1]$* si

- (i) $\tau^s \in \widetilde{W}_s$,
- (ii) $\tau \notin \widetilde{W}_s, \forall \tau \subset \tau^s$

donde $\widetilde{W}_s$ es el s -corte débil del conjunto difuso $\widetilde{W}$ de coaliciones ganadoras.

Definición 4 Dado el juego difuso (N, v^*) , v^* continua, definimos el conjunto difuso $\widetilde{MW}$ de *coaliciones minimales ganadoras* como el subconjunto difuso de $\mathcal{F}(N)$ con función de pertenencia

$$\mu_{\widetilde{MW}}(\tau) = \begin{cases} v^*(\tau) & \text{si } \tau \text{ es una coalición} \\ & \text{minimal ganadora a} \\ & \text{nivel } s = v^*(\tau) \\ 0 & \text{en otro caso} \end{cases}$$

El conjunto de coaliciones minimales ganadoras está bien definido, esto es, si v^* es una función continua, entonces se verifica

$$\{s \in [0, 1] / \tau \text{ es una coalición minimal ganadora a nivel } s\} \subseteq \{v^*(\tau)\}$$

Proposición 1 Sea (N, v^*) un juego simple difuso tal que v^* es una función continua, entonces

$$v^*(\tau) = \max\{\mu_{\widetilde{MW}}(\tau') / \tau' \subseteq \tau, \tau' \in \text{Sop}(\widetilde{MW})\}$$

El análisis del juego a través del conjunto $\mathcal{B}$ de coaliciones de bloqueo (análisis del juego dual) se centra en el establecimiento del valor de una coalición en función del poder que tienen sus miembros para conseguir que una cierta propuesta sea desestimada si es rechazada por todos ellos. Como en el caso anterior, el *conjunto de coaliciones de bloqueo* $\widetilde{B}$ de un juego simple difuso es también un conjunto difuso de $\mathcal{F}(N)$. Sea (T, S, n) la tripleta de De Morgan fuerte que define la estructura de conjuntos de $\mathcal{F}(N)$, siendo n una negación estricta y fuerte, entonces la función de pertenencia de $\widetilde{B}$ viene dada por $\mu_{\widetilde{B}}(\tau) = n(v^*(co(\tau)))$, donde $co(\tau)$ es la coalición complementaria de τ . Intuitivamente, el grado en que una coalición bloquea el juego viene dado por el grado en que la coalición complementaria no verifica la propiedad ser ganadora (grado de desacuerdo al que se llega si el grado en que los jugadores no aceptan la propuesta viene dado por τ). En particular, dado el juego simple (N, v^*) , se define el *juego dual* como $v_D^*(\tau) = n(v^*(co(\tau)))$. A continuación se define el concepto de *coalición minimal de bloqueo a nivel s* , a partir del cual se define el conjunto difuso $\widetilde{MB}$ de *coaliciones minimales de bloqueo* como en el caso anterior (definición de $\widetilde{MW}$), obteniéndose un resultado de caracterización análogo al establecido en la proposición 1.

Definición 5 Dado el juego simple difuso (N, v^*) , v^* continua por la izquierda, decimos que $\tau^b \in \mathcal{F}(N)$ es una *coalición maximal bloqueada a nivel $s \in [0, 1]$* si

- (i) $\tau^b \notin \widetilde{W}_{s'}$,
- (ii) $\tau \in \widetilde{W}_{s'}, \forall \tau \supset \tau^b$

donde $s' = n(s)$ y $\widetilde{W}_{s'}$ es el s' -corte fuerte del conjunto difuso $\widetilde{W}$ de coaliciones ganadoras.

Definición 6 Dado el juego simple difuso (N, v^*) , v^* continua por la izquierda, decimos que $\tau \in \mathcal{F}(N)$ es una *coalición minimal de bloqueo a nivel $s \in [0, 1]$* si la coalición difusa complementaria $co(\tau)$ es una *coalición maximal bloqueada a nivel s*

Proposición 2 Dado el juego simple difuso (N, v^*) , v^* continua, sea (N, v_D^*) el juego dual, entonces $\widetilde{MW}(v_D^*) = \widetilde{MB}(v^*)$

Atendiendo a la relación entre el conjunto de coaliciones minimales ganadoras y el conjunto de coaliciones minimales de bloqueo los juegos simples difusos se pueden clasificar en juegos *propios* ($\mathcal{W} \subseteq \tilde{B}$), *fuertes* ($\tilde{W} \supseteq \tilde{B}$) y *decisivos* ($\tilde{W} = \tilde{B}$).

2 SIGNIFICACIÓN

En un juego simple nítido un jugador J_i es no significativo ($v(S) = v(S \cup \{i\}) \forall S \subseteq N \setminus \{i\}$) o, en algún momento su postura es decisiva (J_i significativo); sin embargo, en el caso difuso, la influencia de un jugador en el juego varía en *tipo* e *intensidad*. Con el fin de recoger las distintas formas en que un jugador puede influir en el juego se introducen los siguientes conceptos de *significación*.

Definición 7 (Aubin) J_i es un jugador títere (*dummy*) en el juego difuso (N, v^*) si la función característica v^* es independiente de la tasa de participación del jugador J_i .

Notación: $(t_i, \tau) = (\tau_1, \dots, \tau_{i-1}, t_i, \tau_{i+1}, \dots, \tau_n)$

Definición 8 Sea (N, v^*) un juego simple difuso. Decimos que el jugador J_i es

- (i) α -significante si $\forall t' \neq t'' \exists (\cdot, \tau)$ tal que $v^*(t'_i, \tau) \neq v^*(t''_i, \tau)$
- (ii) β -significante si $\forall (\cdot, \tau) \exists t' \neq t''$ tal que $v^*(t'_i, \tau) \neq v^*(t''_i, \tau)$
- (iii) β -significante débil si $\forall (\cdot, \tau) \in (0, 1)^{n-1} \exists t' \neq t''$ tal que $v^*(t'_i, \tau) \neq v^*(t''_i, \tau)$
- (iv) fuertemente-significante si $\forall t' \neq t'' \forall (\cdot, \tau) v^*(t'_i, \tau) \neq v^*(t''_i, \tau)$
- (v) veto-significante si $\forall (\cdot, \tau) v^*(0_i, \tau) = 0$
- (vi) decisivo si $\exists (\cdot, \tau)$ tal que $v^*(0_i, \tau) = 0$ y $v^*(1_i, \tau) = 1$
- (vii) fuertemente-decisivo si $\forall (\cdot, \tau) v^*(0_i, \tau) = 0$ y $v^*(1_i, \tau) = 1$

Proposición 3 Sea (N, v^*) un juego simple difuso tal que v^* es una función continua. Se verifica

- (i) i es un jugador títere $\iff$
 $\tau_i = 0 \quad \forall \tau \in \text{sop}(\tilde{M}\tilde{W}) \iff$
 $\tau_i = 0 \quad \forall \tau \in \text{sop}(\tilde{M}\tilde{B})$

(ii) Si

$$\{t \in [0, 1] / \exists \tau \in \text{sop}(\tilde{M}\tilde{W}) \text{ tal que } \tau_i = t\} = (0, 1)$$

entonces i es α -significante.

(iii) Si

$$\{\tau \subseteq N \setminus \{i\} / \exists t \in (0, 1) \text{ tal que } \sigma = (t_i, \tau) \in \text{sop}(\tilde{M}\tilde{W})\} = \mathcal{F}(N \setminus \{i\})$$

entonces, i es β -significante.

(iv) Si i es fuertemente significativa $\implies$

$(t_i, \bar{0}) \subseteq \{i\}$ es una coalición minimal ganadora a nivel $s = v^*(t_i, \bar{0}) \forall t \in (0, 1)$.

En particular, si todos los jugadores son fuertemente significativos $\implies \mu_{\tilde{M}\tilde{W}} \equiv v^*$

(v) Si i es un jugador veto significativo $\implies$

$$\tau_i > 0 \quad \forall \tau \in \text{sop}(\tilde{M}\tilde{W})$$

En la proposición anterior se establece la relación entre los distintos tipos de significación y el conjunto difuso de coaliciones minimales ganadoras. El siguiente diagrama muestra las relaciones entre los distintos tipos de jugadores.

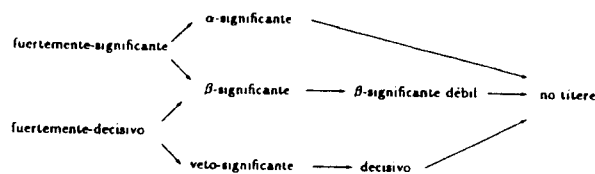


Figura 1

En cuanto a la extensión de los conceptos de *dictador* y *jugador veto* la definición que empleamos es una adaptación de la introducida por Montero [3].

3 COMITÉS

Los conceptos de *juego compuesto* y *comité* se extienden trivialmente sin más que tener en cuenta que todos los juegos que aparecen involucrados en su definición son, en este caso, juegos simples difusos. En un *juego compuesto* los votos de los jugadores se agregan jerárquicamente: en una primera etapa los miembros de cada *comité* (subgrupo de jugadores que emite conjuntamente su voto) M_j agregan sus votos mediante un cierto "subjuego", para posteriormente, agregar mediante el juego compuesto el voto resultante de cada una de dichas agregaciones previas.

4 CLASES RELEVANTES DE JUEGOS SIMPLES DIFUSOS

Por último, se introducen algunas clases de juegos simples difusos: se analizan las situaciones que permiten

representar y se estudia su estructura. Los tipos de juegos tratados son:

- (a) *juegos simétricos* : extienden los juegos simétricos clásicos basándose en operadores triangulares (*t*-normas o *t*-conormas) u operadores OWA (Yager [5]).
- (b) *juegos jerárquicos de agregación de opiniones*: basados en la representación de un juego simple nítido en función del conjunto de coaliciones minimales ganadoras. Son un caso particular de reglas de agregación jerárquicas basadas en cubrimientos (Cutello y Montero [2]).
- (c) *juegos de cuotas* : generalizan los juegos simples clásicos de mayoría ponderada.
- (d) *juegos multilineales* : basados en la extensión multilineal definida por Owen [4].

Los juegos multilineales surgen cuando es posible la determinación del valor de las coaliciones nítidas y es válida la hipótesis de linealidad para su extensión a las coaliciones difusas. En cuanto a los juegos de cuotas, dado el caso en que los jugadores dispongan de un cierto capital para invertir en un proyecto, permiten incorporar al modelo la posibilidad de que los jugadores administren su capital sin restricción alguna (su participación en una coalición difusa no conlleva la cesión de la totalidad de su capital), así como el que el proyecto pueda ser llevado a cabo a distintos niveles. Anteriormente se ha hecho alusión al empleo de un juego simple con el fin de describir las reglas que rigen un proceso de negociación en el que los jugadores expresan opiniones difusas. En este sentido, se tiene que mediante un juego jerárquico de agregación de opiniones se pueden modelizar las negociaciones previas a una votación, votación que en última instancia es nítida (los jugadores emiten un voto favorable o desfavorable); en dicho caso, la distribución del poder entre los jugadores está nítidamente definida mediante un juego simple con coaliciones nítidas y, por tanto, las relaciones de poder entre los jugadores deben ser mantenidas durante el proceso de negociación (durante el cual la propuesta inicial puede verse modificada).

Referencias

- [1] J.P.AUBIN (1974). Coeur et valeur des jeux flous à paiements latéraux. *C.R. Acad. Sci. Paris* **279** A: 891-894.
- [2] V.CUTELLO AND J.MONTERO (1994). Hierarchies of aggregation operators. *International Journal of Intelligent Systems* **9**: 1025-1045.

- [3] J.MONTERO (1985). A note on Fung-Fu's theorem. *Fuzzy Sets and Systems* **17**: 259-269.
- [4] G.OWEN (1972). Multilinear Extensions of Games. *Management Science* **18**: 64-79.
- [5] R.R.YAGER (1988). On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE transactions on systems, man and cybernetics* **18**: 183-190.

ε -Particiones y α -Equivalencias

Susana Montes
Dept. Matemáticas
Universidad de Oviedo
e-mail: smr@pinon.ccu.uniovi.es

Jorge Jiménez
Dept. Matemáticas
Universidad de Oviedo
e-mail: meana@etsiig.uniovi.es

Pedro Gil
Dept. Matemáticas
Universidad de Oviedo
e-mail: pedro@pinon.ccu.uniovi.es

Resumen

El propósito de este trabajo es estudiar un tipo especial de relaciones difusas, como serán las α -equivalencias, así como considerar la relación que une a éstas con la familia de ε -particiones del referencial. También se estudian algunas equivalencias clásicas entre conjuntos, particiones y relaciones difusas.

Palabras Clave: Conjunto difuso, ε -particiones, α -equivalencias y ε -igualdades.

1 INTRODUCCIÓN

Desde que Zadeh [22] introdujo la definición de conjunto difuso, numerosos autores [6], [16], [19], ..., han estudiado las relaciones difusas, y dentro de éstas, ha despertado especial interés el estudio de las relaciones binarias difusas.

Un tipo muy importante de relaciones binarias nítidas son las relaciones de equivalencia R , es decir, los subconjuntos del producto cartesiano del conjunto nítido X consigo mismo, formados por los elementos

$$\{(x, y) \in X \times X / xRy\}$$

que verifican las siguientes propiedades:

- Reflexiva: $xRx, \forall x \in X$.
- Simétrica: $xRy \iff yRx, \forall x, y \in X$.
- Transitiva: Si xRy y yRz entonces $xRz, \forall x, y, z \in X$.

En la teoría de conjuntos difusos se han dado numerosas extensiones de estas propiedades [9], lo que ha dado lugar a diferentes extensiones de las relaciones de equivalencia [7], [19], [23], etc.

Puesto que en el caso nítido toda relación de equivalencia queda completamente caracterizada por la partición que genera y viceversa, también en el caso difuso se han intentado emparejar las relaciones binarias de

"equivalencia" y las particiones difusas [1], [3], [4], [5], [11], [13], [16], [21], etc. En la mayoría de los casos la definición de partición resultante es la dada por Ruspini [18], o alguna modificación de ésta [2].

El presente trabajo define y estudia las α -equivalencias, tipo de relaciones binarias difusas que generarán y caracterizarán en cierta medida, las ε -particiones [14], particiones difusas que extienden a las particiones nítidas, recogiendo las ideas de "recubrimiento" del total por subconjuntos "disjuntos dos a dos", y la borrosidad que existe en la familia de conjuntos que forman la partición.

2 PRELIMINARES

2.1 ε -PARTICIONES DIFUSAS

Definición 2.1 Sea $\tilde{A}$ un conjunto difuso sobre un referencial X . Se dice que la familia $\{\tilde{A}_1, \tilde{A}_2, \dots, \tilde{A}_n\}$ de subconjuntos difusos de $\tilde{A}$ es una ε -partición difusa de $\tilde{A}$, con $\varepsilon \in [0, 1]$, si verifican que :

$$(\tilde{A}_i \cap \tilde{A}_j)_{\bar{\alpha}} = \emptyset, \forall i, j = 1, 2, \dots, n. \text{ con } i \neq j, \forall \alpha \geq \varepsilon$$

$$(\bigcup_{i=1}^n \tilde{A}_i)_{\alpha} = (\tilde{A})_{\alpha}, \forall \alpha \in (\varepsilon, 1 - \varepsilon]$$

siendo $(\tilde{D})_{\alpha}$ el α -corte débil y $(\tilde{D})_{\bar{\alpha}}$ el α -corte fuerte del conjunto $\tilde{D}$ según las definiciones tradicionales de estos conceptos [8].

Por lo tanto, las anteriores condiciones se pueden reescribir como:

$$\{x \in X / \tilde{A}_i \cap \tilde{A}_j(x) > \alpha\} = \emptyset$$

$$\forall i, j = 1, 2, \dots, n. \text{ con } i \neq j, \forall \alpha \geq \varepsilon$$

$$\{x \in X / \bigcup_{i=1}^n \tilde{A}_i(x) \geq \alpha\} = \{x \in X / \tilde{A}(x) \geq \alpha\}$$

$$\forall \alpha \in (\varepsilon, 1 - \varepsilon]$$

Cuando se consideran las operaciones de unión e intersección clásicas [12] del máximo y el mínimo de las funciones de pertenencia, es inmediato que tales condiciones equivalen a las siguientes, más ilustrativas de

la importancia del valor de ϵ en esta definición de partición difusa:

$$\begin{aligned} (\tilde{A}_i \cap \tilde{A}_j)(x) &\leq \epsilon, \forall i, j = 1, 2, \dots, n, i \neq j \quad \forall x \in X \\ \bigcup_{i=1}^n \tilde{A}_i(x) &\geq 1 - \epsilon, \forall x \in X / \tilde{A}(x) \geq 1 - \epsilon \\ \bigcup_{i=1}^n \tilde{A}_i(x) &= \tilde{A}(x), \forall x \in X / \epsilon < \tilde{A}(x) < 1 - \epsilon \\ \bigcup_{i=1}^n \tilde{A}_i(x) &\leq \epsilon, \forall x \in X / \tilde{A}(x) \leq \epsilon \end{aligned}$$

A lo largo de todo el trabajo, se manejarán las operaciones clásicas de unión e intersección consideradas anteriormente, así como la contención de conjuntos difusos clásica definida por:

$$\tilde{A} \subset \tilde{B} \iff \mu_{\tilde{A}}(x) \leq \mu_{\tilde{B}}(x), \forall x \in X$$

También se considerará en general que ϵ es un número menor que 0.5, para evitar problemas como los de la definición de Ruspini, en la que un mismo conjunto repetido n veces, puede ser una partición de otro conjunto totalmente distinto a él.

Cuando se están manejando referenciales finitos, puede medirse a través de la distancia de Hamming [10], la proximidad entre dos particiones de un subconjunto difuso.

Definición 2.2 [14] Sean $\Pi_{\tilde{A}} = \{\tilde{A}_1, \tilde{A}_2, \dots, \tilde{A}_n\}$ y $\Pi_{\tilde{B}} = \{\tilde{B}_1, \tilde{B}_2, \dots, \tilde{B}_n\}$, dos particiones formadas por el mismo número de elementos, dentro de un referencial finito X . Se define la distancia entre $\Pi_{\tilde{A}}$ y $\Pi_{\tilde{B}}$ como:

$$\begin{aligned} d(\Pi_{\tilde{A}}, \Pi_{\tilde{B}}) &= \min_{\sigma \in S_n} \sum_{i=1}^n d_H(\tilde{A}_i, \tilde{B}_{\sigma(i)}) = \\ &= \min_{\sigma \in S_n} \sum_{i=1}^n \sum_{x \in X} |\mu_{\tilde{A}_i}(x) - \mu_{\tilde{B}_{\sigma(i)}}(x)| \end{aligned}$$

donde hemos denotado por S_n el conjunto de todas las permutaciones del conjunto $\{1, 2, \dots, n\}$ y por d_H la distancia de Hamming entre dos subconjuntos difusos.

Puede demostrarse de forma sencilla que la aplicación d definida en 2.2 verifica los tres axiomas que definen una distancia, por lo que recibe correctamente este nombre.

2.2 RELACIONES DIFUSAS BINARIAS

Como ya se ha comentado en la introducción, las clásicas propiedades: reflexiva, simétrica y transitiva, son estudiadas en el caso difuso dando lugar a diferentes extensiones. Algunas de éstas extensiones son formuladas a continuación:

Se dice que una relación binaria $\tilde{R} \subset X \times X$ es:

- Reflexiva: Si $x\tilde{R}x = 1, \forall x \in X$
 - Reflexiva de orden δ : Si $x\tilde{R}x = \delta, \forall x \in X$
 - α -Reflexiva: Si $x\tilde{R}x \geq \alpha, \forall x \in X$
 - Fuertemente reflexiva: Si $x\tilde{R}x \geq \max_y \{x\tilde{R}y\}, \forall x \in X$
 - Absolutamente reflexiva: Si $x\tilde{R}x > 0, \forall x \in X$
 - Simétrica: $x\tilde{R}y = y\tilde{R}x, \forall x, y \in X$
 - Transitiva: $x\tilde{R}z \geq \max_y \{\min\{x\tilde{R}y, y\tilde{R}z\}\}$
 - MP-Transitiva: $x\tilde{R}z \geq \max_y \{x\tilde{R}y \cdot y\tilde{R}z\}$
 - Luk-Transitiva: $x\tilde{R}z \geq \max\{x\tilde{R}y + y\tilde{R}z - 1, 0\}, \forall y \in X$
 - T-Transitiva: $x\tilde{R}z \geq T(x\tilde{R}y, y\tilde{R}z), \forall y \in X$
- donde $x\tilde{R}y$ denota la función de pertenencia de $\tilde{R}$ en el punto (x, y) , es decir,

$$x\tilde{R}y \equiv \tilde{R}(x, y), \forall x, y \in X$$

Una de las formas más comunes de extender las relaciones de equivalencia nítidas, es mediante las relaciones de similitud.

Definición 2.3 Sea X un referencial y $\tilde{R}$ un subconjunto difuso de $X \times X$. Se dice que $\tilde{R}$ es una relación de similitud si $\tilde{R}$ es reflexiva, simétrica y transitiva.

Definición 2.4 Dada una relación difusa $\tilde{R}$, se define el λ -corte de $\tilde{R}$ como la relación nítida $\tilde{R}_\lambda$ dada por

$$\tilde{R}_\lambda = \{(x, y) \in X \times Y / x\tilde{R}y \geq \lambda\}$$

Como consecuencia de la siguiente proposición, las relaciones de similitud quedan caracterizadas por sus α -cortes.

Proposición 2.5 Sea X un referencial y $\tilde{R}$ un subconjunto difuso de $X \times X$. $\tilde{R}$ es una relación de similitud si y sólo si, $\tilde{R}_\lambda$ es una relación de equivalencia nítida para todo λ de $[0, 1]$.

3 α -EQUIVALENCIAS

Definición 3.1 Dado un referencial X , y un subconjunto difuso $\tilde{R}$ del producto cartesiano de $X \times X$, se dice que $\tilde{R}$ es una relación difusa de α -equivalencia, con $\alpha \in [0, 1]$, si

$$\tilde{R} \text{ es } \alpha\text{-reflexiva: } [21] \quad x\tilde{R}x \geq \alpha, \forall x \in X \quad (1)$$

$$\tilde{R} \text{ es simétrica: } x\tilde{R}y = y\tilde{R}x, \forall x, y \in X \quad (2)$$

$$\begin{aligned} \tilde{R} \text{ es } \alpha\text{-transitiva: } &\text{Si } x\tilde{R}y \geq \alpha \text{ e } y\tilde{R}z \geq \alpha \\ &\implies x\tilde{R}z \geq \alpha \end{aligned} \quad (3)$$

Proposición 3.2 Si $\tilde{R}$ es una relación de α -equivalencia difusa, entonces su α -corte es una relación de equivalencia nítida.

El recíproco con esta definición de α -equivalencia no es cierto, sin embargo, se verificaría sin más que modificar la condición 2, por la siguiente

$$\tilde{R} \text{ es } \alpha\text{-simétrica: } x\tilde{R}y \geq \alpha \iff y\tilde{R}x \geq \alpha, \forall x, y \in X \quad (4)$$

Definición 3.3 Dado un referencial X , y un subconjunto difuso $\tilde{R}$ del producto cartesiano de $X \times X$, se dice que $\tilde{R}$ es una relación difusa de α -equivalencia débil, con $\alpha \in [0, 1]$, si

$$\tilde{R} \text{ es } \alpha\text{-reflexiva: } x\tilde{R}x \geq \alpha, \forall x \in X \quad (5)$$

$$\tilde{R} \text{ es } \alpha\text{-simétrica: } x\tilde{R}y \geq \alpha \iff y\tilde{R}x \geq \alpha, \forall x, y \in X \quad (6)$$

$$\tilde{R} \text{ es } \alpha\text{-transitiva: Si } x\tilde{R}y \geq \alpha \text{ e } y\tilde{R}z \geq \alpha \implies x\tilde{R}z \geq \alpha \quad (7)$$

En todo el trabajo se han considerado las α -equivalencias en lugar de las α -equivalencias débiles, pero todo el desarrollo podría hacerse de forma totalmente análoga para éstas últimas.

En la siguiente proposición, se relacionan estas dos definiciones con la habitualmente utilizada de similitud.

Proposición 3.4 Dado un referencial X , y un subconjunto difuso $\tilde{R}$ del producto cartesiano de $X \times X$, entonces:

$$\tilde{R} \text{ similaridad} \implies \tilde{R} \text{ } \alpha\text{-equivalencia} \implies \tilde{R} \text{ } \alpha\text{-equivalencia débil}$$

La demostración de esta proposición es inmediata, sin más que considerar las caracterizaciones de éstas relaciones mediante sus α -cortes y las propiedades de éstos.

El concepto de "clase de equivalencia" que va a estudiarse a continuación, será muy importante en el apartado 5, en el que se relacionarán las α -clases de equivalencia con las ε -particiones.

Definición 3.5 Dado un referencial X y una α -relación de equivalencia difusa sobre él, se define la clase de α -equivalencia de cualquier elemento $a \in X$ como el conjunto difuso $\tilde{[a]}$ definido por

$$\tilde{[a]}(x) = \max_{\{b \in X / a\tilde{R}b \geq \alpha\}} (b\tilde{R}x)$$

4 ε -IGUALDADES

A continuación se va a definir la "casi" igualdad de subconjuntos, particiones y relaciones difusas, definiciones necesarias para el siguiente apartado del trabajo.

Definición 4.1 Sea X el referencial y sean $\tilde{A}$ y $\tilde{B}$ subconjuntos difusos de él, se dice que:

1) $\tilde{A}$ es ε -igual al subconjunto difuso $\tilde{B}$, y se denotará por $\tilde{A} \equiv_{\varepsilon} \tilde{B}$, si y sólo si

$$(\tilde{A})_{\delta} = (\tilde{B})_{\delta}, \forall \delta > \varepsilon \quad (8)$$

2) $\tilde{A}$ es ε -débil igual al subconjunto difuso $\tilde{B}$, y se denotará por $\tilde{A} \equiv_{\varepsilon} \tilde{B}$, si y sólo si

$$(\tilde{A})_{\delta} = (\tilde{B})_{\delta}, \forall \delta \in (\varepsilon, 1 - \varepsilon) \quad (9)$$

Por lo tanto se dirá que dos subconjuntos difusos son ε -iguales, si sólomente se diferencian en los puntos que "casi no pertenecen" a ellos. En el caso en el que se les permita una cierta libertad (ε) tanto en los puntos que "casi pertenecen" a los conjuntos, como en los que "casi no pertenecen", se estará entonces ante una ε -igualdad débil.

Definición 4.2 Dados dos subconjuntos difusos $\tilde{A}$ y $\tilde{B}$ del referencial X , se dice que son ε -distintos (ε -débil distintos) y se denota $\tilde{A} \not\equiv_{\varepsilon} \tilde{B}$ ($\tilde{A} \not\equiv_{\varepsilon} \tilde{B}$) si no verifican la condición 8 (9).

Proposición 4.3 Dado un referencial X y el conjunto $\tilde{P}(X)$ formado por todos los subconjuntos difusos de X , entonces tanto $\equiv_{\varepsilon}$ como $\equiv_{\varepsilon}$ son relaciones de equivalencia clásicas sobre $\tilde{P}(X)$, es decir, verifican la propiedad reflexiva, simétrica y transitiva.

A partir de la igualdad entre conjuntos, se puede definir una igualdad entre particiones.

Definición 4.4 Sea X el referencial y sean Π y Π' particiones difusas de él, se dice que:

1) Π está ε -contenida en la partición difusa Π' , y se denotará por $\Pi \subseteq_{\varepsilon} \Pi'$, si y sólo si

$$\forall \tilde{A}_i \in P \text{ con } \tilde{A}_i \not\equiv_{\varepsilon} \emptyset, \exists \tilde{B}_j \in \Pi' / \tilde{A}_i \equiv_{\varepsilon} \tilde{B}_j$$

2) Π está ε -débil contenida en la partición difusa Π' , y se denotará por $\Pi \subseteq_{\varepsilon} \Pi'$, si y sólo si

$$\forall \tilde{A}_i \in P \text{ con } \tilde{A}_i \not\equiv_{\varepsilon} \emptyset, \exists \tilde{B}_j \in \Pi' / \tilde{A}_i \equiv_{\varepsilon} \tilde{B}_j$$

Definición 4.5 Sea X el referencial y sean Π y Π' particiones difusas de él, se dice que:

1) Π es ε -igual a la partición difusa Π' , y se denotará por $\Pi \simeq_{\varepsilon} \Pi'$, si y sólo si

$$\Pi \subseteq_{\varepsilon} \Pi' \text{ y } \Pi' \subseteq_{\varepsilon} \Pi$$

2) Π es ε -débil igual a la partición difusa Π' , y se denotará por $\Pi \simeq_{\varepsilon} \Pi'$, si y sólo si

$$\Pi \subseteq_{\varepsilon} \Pi' \text{ y } \Pi' \subseteq_{\varepsilon} \Pi$$

A partir de la proposición 4.3 formulada para conjuntos, puede establecerse un resultado similar para las particiones.

Corolario 4.6 Las relaciones $\simeq_\varepsilon$ y $\cong_\varepsilon$ definidas en 4.5 verifican las propiedades reflexiva, simétrica y transitiva, por lo que son una verdadera relación de equivalencia en el sentido clásico, sobre el conjunto de particiones difusas del referencial.

Por último, se dirá que dos relaciones binarias son "iguales" si:

Definición 4.7 Sea X el referencial y sean $\tilde{R}$ y $\tilde{R}'$ relaciones binarias difusas sobre él, se dice que:

1) $\tilde{R}$ es ε -igual a la relación difusa $\tilde{R}'$, y se denotará por $\tilde{R} \cong_\varepsilon \tilde{R}'$, si y sólo si

$$(\tilde{R})_\delta = (\tilde{R}')_\delta, \forall \delta > \varepsilon$$

2) $\tilde{R}$ es ε -débil igual a la relación difusa $\tilde{R}'$, y se denotará por $\tilde{R} \cong_\varepsilon \tilde{R}'$, si y sólo si

$$(\tilde{R})_\delta = (\tilde{R}')_\delta, \forall \delta \in (\varepsilon, 1 - \varepsilon)$$

es decir, que $\{a, b \in X / a\tilde{R}b \geq \alpha\} = \{a, b \in X / a\tilde{R}'b \geq \alpha\}$ para cualquier $\delta > \varepsilon$ y $\delta \in (\varepsilon, 1 - \varepsilon)$ respectivamente.

De la misma forma que ocurría con los conjuntos y las particiones difusas:

Corolario 4.8 Las relaciones $\cong_\varepsilon$ y $\cong_\varepsilon$ definidas en 4.7 son relaciones de equivalencia en el sentido clásico, sobre el conjunto de relaciones binarias difusas del referencial.

5 α -EQUIVALENCIAS Y ε -PARTICIONES

En el siguiente resultado se relacionan las α -equivalencias con las ε -particiones difusas del referencial.

Teorema 5.1 Dado un referencial X , y una ε -partición de dicho conjunto formada por los conjuntos $\{\tilde{A}_1, \tilde{A}_2, \dots, \tilde{A}_n\}$. El subconjunto difuso $\tilde{R} \subseteq X \times X$ definido por

$$x\tilde{R}y = \max_{\tilde{A}_i} \{\min\{\tilde{A}_i(x), \tilde{A}_i(y)\}\}$$

es una $1 - \varepsilon$ -equivalencia.

Puesto que este teorema relaciona las $(1 - \varepsilon)$ -equivalencias con las ε -particiones, al igual que ocurre con las particiones y relaciones de equivalencia nítidas, cabe preguntarse que ocurrirá con la relación nítida que genera $\tilde{R}$ mediante su $(1 - \varepsilon)$ -corte. La solución a esta cuestión es dada en la siguiente proposición:

Proposición 5.2 Dado un referencial finito X , y una ε -partición de dicho conjunto formada por los conjuntos $\Pi = \{\tilde{A}_1, \tilde{A}_2, \dots, \tilde{A}_n\}$. Si se considera la relación de $(1 - \varepsilon)$ -equivalencia obtenida a partir de Π en el teorema anterior, y a partir de ella se obtiene la relación de equivalencia nítida $R_{1-\varepsilon}$, la partición nítida formada por las clases de equivalencia de $\tilde{R}_{1-\varepsilon}$, es la partición nítida más próxima a Π , midiendo esta proximidad a través de la distancia de Hamming 2.2.

Si a la partición difusa se le añade un elemento $\tilde{A}_{i_0}$ que tome el valor ε en todo elemento x del referencial, la nueva familia sigue siendo una ε -partición, pero la relación resultante es más restrictiva.

Proposición 5.3 Dado un referencial X , y una ε -partición $\Pi = \{\tilde{A}_1, \tilde{A}_2, \dots, \tilde{A}_n\}$ de dicho conjunto, que verifica que

$$\exists i_0 \in \{1, 2, \dots, n\} / \tilde{A}_{i_0}(x) = \varepsilon, \forall x \in X \quad (10)$$

entonces la relación de $1 - \varepsilon$ equivalencia definida en 5.1 es además transitiva.

Cuando además de la condición 10 se verifique que $\bigcup_{i=1}^n \tilde{A}_i = X$, entonces la relación de $(1 - \varepsilon)$ -equivalencia que genera es una similaridad.

En este apartado se ha estudiado como una ε -partición genera una $(1 - \varepsilon)$ -relación de equivalencia. Recíprocamente, si se tiene una α -relación de equivalencia, se puede generar a partir de ella una partición difusa.

Proposición 5.4 Dado un referencial X y una relación de α -equivalencia $\tilde{R}$ definida sobre él, verificando la condición

$$x\tilde{R}y \geq \alpha \text{ ó } x\tilde{R}y \leq 1 - \alpha, \forall x, y \in X \quad (11)$$

entonces la familia de subconjuntos de X formada por las distintas clases de α -equivalencia es una $(1 - \alpha)$ -partición de X .

Como consecuencia del teorema 5.1 y de la proposición 5.4, junto con las definiciones de particiones y relaciones ε -iguales dadas en el apartado 4, se obtiene el siguiente teorema.

Teorema 5.5 Dado un referencial X , el conjunto $P = \{\text{conjunto de las particiones difusas de } X\}$ y el conjunto $R = \{\text{conjunto de las } \alpha\text{-equivalencias sobre } X \text{ que verifican la condición 11}\}$. Bajo estas hipótesis, la aplicación

$$\begin{array}{ccc} f: P & \longrightarrow & R \\ & \Pi & \longrightarrow & \tilde{R}_\Pi \end{array}$$

donde $\tilde{R}_\Pi$ es la relación binaria difusa definida en el teorema 5.1, es inyectiva.

Corolario 5.6 Si en el teorema 5.5 se considera en lugar de la ε -igualdad, la relación de ε -igualdad débil, la aplicación f es una biyección.

6 CONCLUSIONES

En este trabajo se ha intentado encontrar la clase de relaciones binarias difusas que generan a las ε -particiones, viéndose como en realidad éstas son relaciones de "equivalencia", y como se obtiene una identificación entre "relaciones de equivalencia" y "particiones" análoga a la del caso nítido. Evidentemente, éste es sólo un primer acercamiento al problema, en futuros estudios intentaremos analizar no sólo las relaciones que generan a las particiones del referencial, sino de cualquier subconjunto de éste. También está entre nuestros propósitos el estudiar la relación resultante de un refinamiento, de un producto de particiones, la relación que une a las relaciones generadas por particiones independientes [15], etc. En sentido inverso, también nos gustaría estudiar la partición resultante de la composición de dos relaciones, de la unión, de la intersección, etc.

Agradecimientos

La investigación de este trabajo está subvencionada en parte a través del Proyecto DGICYT PB94-1328. Queremos agradecer al Ministerio de Educación y Ciencia dicha ayuda económica.

Referencias

- [1] J.C. Bezdek and J.D. Harris (1978). Fuzzy partitions and relations: An axiomatic basis for clustering. *Fuzzy Sets and Systems* 1:111-127.
- [2] J. Bezdek (1981). Pattern Recognition with Fuzzy Objective Functions Algorithms. *Plenum Press, New York*.
- [3] S.K. Bhakat and P. Das (1992). q-Similitudes and q-fuzzy partitions. *Fuzzy Sets and Systems* 51: 195-202.
- [4] S. Bodjanova (1993). Some binary relations and operations on the set of fuzzy partitions. *Fuzzy Sets and Systems* 53:305-317.
- [5] B. Bouchon and G. Cohen (1983). On fuzzy relations and partitions. *Avances in Fuzzy Sets, Possibility Theory and Applications*:97-105.
- [6] M.K. Chakraborty and M. Das (1983). Studies in fuzzy relations over fuzzy subsets. *Fuzzy Sets and Systems* 9:79-89.
- [7] M.K. Chakraborty and M. Das (1983). On fuzzy equivalence I. *Fuzzy Sets and Systems* 11:185-193.
- [8] D. Dubois and H. Prade (1980). Fuzzy Sets and Systems. Theory and Applications. *Academic Press, Inc.*
- [9] F. Esteva (1991). Cálculo con relaciones borrosas. *Lógica borrosa y sus aplicaciones. Curso de verano de la Universidad Complutense*.
- [10] A. Kaufmann (1975). Introduction to the theory of Fuzzy Subsets. *Academic Press, New York*. Vol 1.
- [11] J. Jacas and L. Valverde (1989). On Fuzzy relations, Metrics and Cluster Analysis. en J.L. Verdegay et al. *Aproximate Reasoning Tools for Artificial Intelligence*, Verlag TUV. Rheinland:21-38.
- [12] G.J. Klir and T.A. Folger (1988). Fuzzy Sets, Uncertainty, and Information. *Prentice-Hall International (UK) Limited*
- [13] R. López de Mántaras and L. Valverde (1984). New results in fuzzy clustering based on the concept of indistinguishability relations. *Proc. 7th. Intern. Conf. on Pattern Recognition, Montreal*.
- [14] S. Montes, M.J. López-Palomo and P. Gil (1995). Las ε -particiones. *V Congreso Estylf '95*. 1:96-99.
- [15] S. Montes, A. Bodini and P. Gil (1996). Fuzzy partitions and classifications. *IPMU'96:en proceso*.
- [16] S. Ovchinnikov (1981). Structure of fuzzy binary relations. *Fuzzy Sets and Systems* 6:169-195.
- [17] S. Ovchinnikov (1991). Similarity relations, fuzzy partitions, and fuzzy orderings. *Fuzzy Sets and Systems* 40:107-126.
- [18] E.H. Ruspini (1969). A new approach to clustering. *Inf. Control* 15:22-32.
- [19] E. Trillas and J.L. Castro (1989). Logic and Fuzzy relations. en J.L. Verdegay et al. *Aproximate Reasoning Tools for Artificial Intelligence*. Verlag TUV, Rheinland.
- [20] E. Trillas (1982). Assaig sobre les relacions d'indistingibilitat. *Actes del 1er. Congrés Català de Lògica Matemàtica*:51-59.
- [21] R.T. Yeh and S.Y. Bang (1975). Fuzzy relations, fuzzy graphs, and their applications to clustering analysis. *Fuzzy Sets and their Applications to Cognitive and Decision Processes*. *Academic Press*:125-149.
- [22] L.A. Zadeh (1965). Fuzzy sets. *Information and Control* 8:338-353.
- [23] L.A. Zadeh (1971). Similarity relations and fuzzy orderings. *Inform. Sci.* 3:177-200.

APLICACIÓN DE TÉCNICAS DIFUSAS AL DISEÑO DE ALGORITMOS PARA VISIÓN POR COMPUTADOR

Eduard Montseny

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

Pilar Sobrevilla

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

Resumen

En este trabajo se presenta un método para el diseño de algoritmos en los que se utilizan técnicas difusas, lo cual permite un mejor tratamiento de la vaguedad.

Para el diseño de estos algoritmos, se propone desarrollar una base de reglas que definan su comportamiento. El problema de la fuzzificación de los datos se resuelve a partir de las funciones de densidad y distribución de probabilidad, mientras que su análisis se plantea asociando, a cada una de las "reglas de análisis", un conjunto difuso, el cual se obtendrá aplicando una función de agregación, definida mediante un operador OWA.

El diseño propuesto aporta una forma de fuzzificar los valores de los datos que han de ser analizados, siendo éste un problema que ya está bastante bien resuelto en el caso de algoritmos aplicados a control, pero que, hasta el momento, presentaba grandes dificultades en los de visión.

Por otra parte, el método de análisis de los datos que se propone, aporta una solución a los problemas, no intrínsecos, presentes en los algoritmos de visión.

Palabras clave: Visión por computador, tratamiento de imágenes, técnicas difusas, funciones de agregación, funciones de pertenencia, operadores OWA.

1. INTRODUCCIÓN

La gran mayoría de algoritmos que implementan tratamientos para imágenes a bajo nivel presentan el problema del tratamiento de la vaguedad, la cual está presente en las imágenes por diversos factores, difícilmente evitables con las técnicas que se utilizan hoy para la captación y digitalización de imágenes. Así, desde que Li, Yang y Pao [9], [13] introdujeron la existencia de una componente de vaguedad, no aleatoria, en las variables que intervienen en los sistemas de visión por computador, muchos científicos han tratado de resolver estos problemas mediante técnicas difusas.

Prewitt, en 1970, ya sugirió que los resultados de la segmentación eran subconjuntos difusos del plano. Pero, las soluciones que buscó fueron soluciones crisp. También Haralick [8], en 1985, afirmó que cualquier algoritmo de extracción de contornos funciona bien en

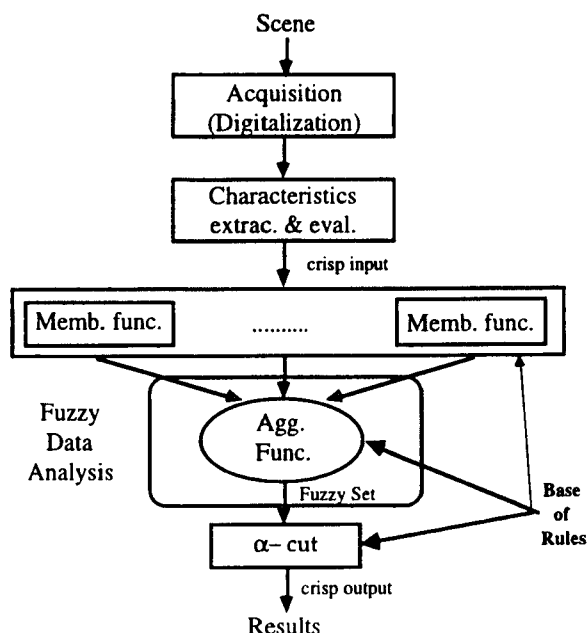
ausencia de ruido. Como es sabido el ruido es uno de los factores que introduce más vaguedad en las imágenes.

En la década de los noventa se han diseñado numerosos algoritmos que introducen, en el procesado a bajo nivel, un tratamiento de la vaguedad presente en las imágenes. Si bien en todos ellos se pueden intuir los módulos básicos que deben constituir un algoritmo basado en técnicas difusas, se echa en falta un modelo que permita definir la fuzzificación de los datos, así como una forma de analizar los mismos. Así pues en los algoritmos aparecen estos procesos implementados mediante unas ecuaciones difícilmente justificables.

Son muchos los trabajos, que utilizan técnicas difusas, en los que el algoritmo se formaliza mediante reglas, para posteriormente implementar dichas reglas mediante un conjunto de ecuaciones. Por ello, en el presente trabajo, se ha partido de esta forma de diseño y el método, que a continuación se expone, propone una formalización del algoritmo mediante reglas e introduce unas pautas a fin de facilitar la implementación de las mismas..

2. ESTRUCTURA DE LOS ALGORITMOS

El método que se propone plantea el diseño de algoritmos de visión por computador que permiten un tratamiento de la vaguedad. Estos constarán de cinco módulos, tal como puede observarse en el diagrama.



A continuación se describe la función que realiza, de cada uno de estos módulos en el algoritmo.

2.1. BASE DE REGLAS Y DEFINICIÓN DE CARACTERÍSTICAS.

En este módulo se determinan las características que se van a analizar en el algoritmo, con el fin de alcanzar los objetivos del mismo. También se determina la forma de evaluar dichas características, así como su fuzzificación. Otro conjunto de reglas describen el análisis que se debe realizar a fin de obtener los resultados deseados.

Tal como se verá más adelante las reglas deberán escribirse en un determinado formato, a fin de facilitar su implementación en el módulo correspondiente.

2.2. EXTRACCIÓN Y EVALUACIÓN DE CARACTERÍSTICAS.

En este módulo, a partir de la información contenida en la imagen digital de la escena, se realizan, para cada pixel de la imagen, diversas evaluaciones que permitirán, en el tercer módulo, inferir con qué grado se cumplen las condiciones asociadas a cada una de las características definidas en el módulo anterior.

2.3. FUZZIFICACIÓN Y REGLAS DE FUZZIFICACIÓN.

A partir de las evaluaciones realizadas para cada pixel en el módulo anterior, se obtienen las medidas difusas asociadas a cada una de las características analizadas.

Con este propósito se define, para cada característica, una aplicación

$$fc_i : R \rightarrow [0,1]$$

que viene dada a partir de las reglas correspondientes del módulo "base de reglas" y de la información obtenida en el análisis de unas imágenes patrón.

2.4. ANÁLISIS DE DATOS Y REGLAS DE AGREGACIÓN.

En este módulo se implementan las reglas relacionadas con el análisis de las características, mediante funciones de agregación. Así, para cada una de las citadas reglas, se define una aplicación

$$fa_j : [0,1]^n \rightarrow [0,1]$$

Estas aplicaciones quedarán definidas mediante la utilización de un operador OWA, obtenido a partir del análisis de la información contenida en las imágenes patrón.

2.5. DEFUZZIFICACIÓN.

A partir de la defuzzificación de los conjuntos difusos obtenidos en el módulo anterior se obtienen los resultados

objetivo del algoritmo, , para ello se define, para cada resultado, una aplicación

$$fr_i : [0,1] \rightarrow \{0,1\}$$

que viene dada mediante un α -corte aplicado al conjunto difuso correspondiente, donde el valor de α es 0.5. De este modo se clasifica cada uno de los elementos (pixels ó conjunto de pixels).

3. BASE DE REGLAS

Éste es el módulo más importante, y en él se va a formalizar el algoritmo mediante un conjunto de reglas, que contienen información sobre:

- Las características que se van a analizar.
- La fuzzificación de las medidas de la características.
- El análisis de las características.

Se deberán especificar por tanto las características que se analizarán, así como el modo de su evaluación, fuzzificación y el proceso que se seguirá para su análisis. Para especificar la fuzzificación de las características se emplearán reglas, las cuales se enunciarán de acuerdo con el siguiente formato:

Si la desviación del valor de la "característica" respecto del valor "normal-medio" es A
entonces el pixel/entorno de pixels B al "conjunto difuso".

donde

A = { pequeña, mediana, grande }

B = { no pertenece, pertenece poco, pertenece, pertenece mucho }

el cual permitirá obtener las medidas difusas asociadas a cada una de las características en el módulo de fuzzificación.

De forma análoga se procederá a enunciar un conjunto de reglas que determinarán el análisis de las características. Estas reglas se especificarán con el formato:

Si un elemento **cumple** (pertenece) las "caraterísticas" (a los "conjuntos difusos") $C_1, C_2, \dots, C_k$ entonces este elemento pertenece al "conjunto difuso" C_n .

donde: $C_1, C_2, \dots, C_k$ son características ó conjuntos difusos conocidos.

C_n es un nuevo conjunto difuso.

Las características o los conjuntos difusos a agregar, deberán tener todos la misma relevancia.

4. FUZZIFICACIÓN

A fin de fuzzificar las medidas asociadas a cada característica, se define una aplicación

$$fc_j : R \rightarrow [0,1]$$

Estas aplicaciones (aplicaciones de fuzzificación) se calculan a partir de la función de densidad de probabilidad de los datos obtenidos al evaluar dicha característica en unos determinados pixels o en unas determinadas regiones de una imagen patrón, en los que se da dicha característica. La utilización de funciones de densidad de probabilidad está motivado por el tipo de características asociadas, de manera intrínseca, a los valores de las medidas de las características a evaluar, ya que dichos valores se ven afectados por características de distribución espacial aleatoria, procedente de:

1. La presencia de ruido blanco en las imágenes.
2. El proceso de digitalización.

De este modo, a fin de obtener las aplicaciones de fuzzificación se procede de acuerdo con las siguientes reglas:

- **R1:** $\mu(fc_n(i,j))$ crece cuando el valor de la característica n para el pixel i,j , tiene un valor en la función de densidad de probabilidad que se acerca al máximo.
- **R2:** $\mu(fc_n(i,j))$ crece cuando el valor, de la característica n para el pixel i,j , tiene un valor en la función de distribución de probabilidad que se acerca al punto de corte de las curvas de dicha función.

Estas dos reglas se implementan:

R1: $\mu_1(fc_n(i,j)) = (p_n(fc_n(i,j))) / \max(p_n(k))$ donde:

$\mu_1(fc_n(i,j))$ es un valor difuso correspondiente al elemento i,j asociado a esta primera regla.

$fc_n(i,j)$ es el valor evaluado de la característica n para el elemento i,j

$p_n(fc_n(i,j))$ es el valor de la función de la densidad de probabilidad, de la característica n , para el valor $fc_n(i,j)$.

$\max(p_n(k))$ es el valor máximo que toma la función densidad de probabilidad de la característica n .

R2:

a) $\mu_2(fc_n(i,j)) = \min\{1, (1 - P_n(fc_n(i,j))) + p_n(fc_n(i,j)) / 0.5\}$

b) $\mu_2(fc_n(i,j)) =$

$= \min\{1, P_n(fc_n(i,j)) / 0.25, (1 - P_n(fc_n(i,j))) + p_n(fc_n(i,j)) / 0.25\}$

donde:

$\mu_2(fc_n(i,j))$ es un valor difuso correspondiente al elemento i,j asociado a esta segunda regla.

$P_n(fc_n(i,j))$ es el valor de la función distribución de probabilidad, de la característica n , para el valor $fc_n(i,j)$.

La segunda regla se evalúa según **a** en caso que $fc_n(i,j)$ es un error que sólo se toma valores positivos. Y se evalúa según **b** en el resto de los casos.

Finalmente $\mu(fc_n(i,j))$ se evalúa de la siguiente forma:

$$\mu(fc_n(i,j)) = 0.5 * \mu_1(fc_n(i,j)) + 0.5 * \mu_2(fc_n(i,j))$$

5. ANÁLISIS DE DATOS Y REGLAS DE AGREGACIÓN.

En este módulo se realiza el análisis de los datos, de acuerdo con la información contenida en la base de reglas. Cada una de las reglas, con información del análisis de las características, contenida en dicha base, dará lugar a un conjunto difuso C_n , el cual se obtendrá de la agregación de las medidas y/o conjuntos difusos ($C_1, C_2, \dots, C_k$) indicados en la regla correspondiente, siguiendo las directrices indicadas en la misma.

Dada la naturaleza del problema que se está analizando, se acepta que el grado de pertenencia de cada uno de los elementos i,j , al nuevo conjunto difuso, cumplirá:

$$\min\{\mu_k(i,j)\} \leq \mu(i,j) \leq \max\{\mu_k(i,j)\}$$

donde:

$\mu_k(i,j)$ son los valores difusos de las medidas de las características, o los grados de pertenencia, del elemento i,j , a agregar.

$\mu(i,j)$ es el grado de pertenencia del elemento i,j al nuevo conjunto difuso.

Tal como se ha descrito en el apartado de la base de reglas, cada una de las características o conjuntos difusos a agregar son igualmente relevantes, de este modo una buena solución para definir las funciones de agregación es la utilización de los operadores OWA introducidos por Yager.

El problema que se plantea, al utilizar los operadores OWA, es la elección de los pesos. Para la obtención de estos pesos se calcula el grado de pertenencia, o las medidas difusas de las características a agregar, en unos determinados pixels o en unas determinadas regiones de una imagen patrón, en los que los elementos i,j pertenecen al nuevo conjunto difuso.

Estos valores se representarán en una tabla, para ello se realizará la siguiente transformación. Se clasificará como:

nulo si el valor está comprendido entre 0 y 0.25

algo si está comprendido entre 0.25 y 0.5

casi si está comprendido entre 0.5 y 0.75

si si está comprendido entre 0.75 y 1.

y se incorporarán a la tabla ordenándolos de acuerdo con el número de **si** que presenta la evaluación del elemento, en caso de igualdad se compararán el número de **casi**, si la igualdad continúa se tendrán en cuenta el número de **algo**, y si aún continuase la igualdad los elementos pertenecen al mismo grupo. Una vez representadas en la tabla, de forma ordenada, las distintas evaluaciones, se incorporará en otra columna el porcentaje de elementos analizados que presentan cada una de ellas.

Finalmente se calcularán los pesos teniendo en cuenta que, al menos, un 85% de los elementos, clasificados a priori como que deben pertenecer al nuevo conjunto difuso, deberán tener un grado de pertenencia igual o superior a 0.5.

Así, en el caso de una regla que debe agregar tres medidas difusas se debería obtener una tabla del tipo

CODE	% PIXELS	CODE	% PIXELS
111	7.95	222	2.66
112	15.17	223	4.65
113	6.30	224	2.30
114	1.85	233	3.44
122	10.30	234	4.65
123	10.08	244	2.66
124	3.71	333	1.11
133	2.77	334	3.11
134	2.81	344	5.47
144	1.52	444	7.52

donde en la columna code se ha representado la valoración obtenida para los elementos siendo que: 1 representa si, 2 casi, 3 algo y 4 nulo.

6. RESULTADOS.

El método propuesto en el presente trabajo se ha aplicado al diseño de tres clases de algoritmos que analizan las características de imágenes a bajo nivel, dichos algoritmos són:

- Detección de contornos.
- Detección de texturas.
- Análisis de texturas naturales.

En los tres casos los resultados obtenidos han sido muy satisfactorios, tanto por la facilidad en el diseño del algoritmo, como por los resultados que se han obtenido.

Si bien es difícil poder evaluar los resultados obtenidos y contrastarlos con los que proporcionan los algoritmos existentes, de todas formas hay que se ha de destacar que:

a) En el caso de la detección de texturas, no se han encontrado resultados publicados ni del tipo de texturas detectadas ni del tipo de características analizadas. Hay que destacar que se han detectado correctamente el 90% de la textura (se debe tener en cuenta que las texturas analizadas presentan una gran variabilidad de características) y que, cuando se ha añadido ruido a la imagen, se ha observado un comportamiento muy robusto del algoritmo, de forma que la eficiencia de éste sólo se ha reducido hasta el 83%.



Figura 1: Imagen original.



Figura 2: Imagen resultado.

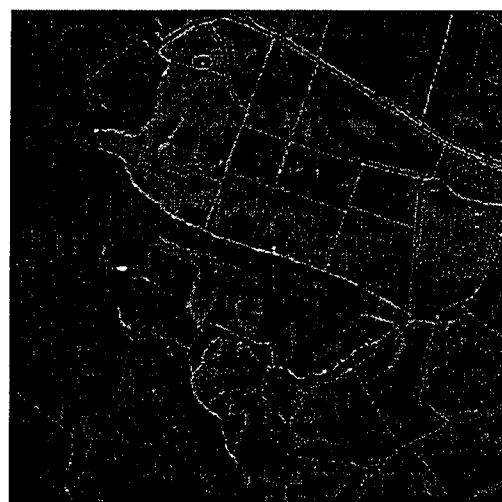


Figura 3: Imagen original



Figura 4: Imagen resultado.

b) En el caso de análisis de texturas naturales, se ha comprobado que el algoritmo trata muy bien la variabilidad de las características de este tipo de texturas, siendo la eficiencia de los resultados obtenidos del 90%, por lo que estos se deben considerar como muy buenos.

c) En el caso de la extracción de contornos se ha podido observar que, con el algoritmo diseñado, se han podido detectar contornos que presentan menor contraste, pudiéndose localizar los contornos con mayor precisión. Es decir, los que eran difícilmente detectables ahora se han podido detectar y, al utilizar operadores de dimensiones más reducidas, se mejora notablemente la precisión en la localización de los mismos.



Figura 5: Imagen original

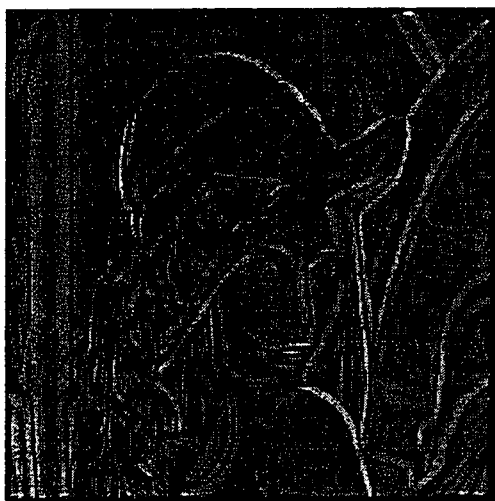


Figura 6: Imagen resultado

7. CONCLUSIONES.

Como conclusiones hay que señalar que, con el método presentado en este trabajo, se facilita el diseño de algoritmos que utilizan técnicas difusas para el tratamiento de la vaguedad. Son de destacar dos aportaciones, la primera es la introducción de unas pautas, para la fuzificación de los valores de las características, que se adecuan al tipo de vaguedad presente en las imágenes y, en segundo lugar, se presenta un método para realizar el análisis de los datos, utilizando operadores OWA, que implementen las funciones descritas en cada una de las distintas reglas que hay que aplicar en el tratamiento de los datos.

Agradecimientos

Este trabajo ha sido, en parte, financiado por la CICYT, proyecto: BIO95-0916-C02-01.

Referencias

- [1] Alsina, C., Trillas, E., Valverde, L., On Some Logical Connectives for Fuzzy Sets Theory. *Journal of Mathematical Analysis and Applications*, Vol. 93, nº 1, pag. 15-26, 1983.
- [2] Alsina, C., Fank, M.J., Schweizer, B., *Associative Functions on Intervals*, 1987.
- [3] Bajcsy, R., Lieberman, L., Texture gradient as a depth cue. *Computer Graphics and Image Processing*, Vol 5, nº 1, pag. 52-67, 1976.
- [4] du Buf, J.M.H., Kardan, M., Spann, M., Texture feature performance for image segmentation. *Pattern Recognition*, vol. 23, nº 3/4, pag. 291-309, 1990.
- [5] García-Barroso, C., Sobrevilla, P., Montseny, E., Step edge detection using fuzzy algorithms, *Proc. VI IFSA*, Sao Paulo (Brazil), pag. 269-272, 1995.
- [6] Gupta, M.M., Knopf, G.K., Fuzzy logic and uncertainty in image processing, *Proceedings SPIE*, vol. 1902, pag. 210-223, 1993.
- [7] Haralick, R.M., Shanmugam, K., Dinstein, I., Textural Features for Image Classification, *IEEE Trans. SMC-3*, nº 6, 1973.
- [8] Haralick, R.M., Author's Reply to Comments on Digital Step Edges from Zero Crossing of Second Directional Derivatives, *IEEE Trans. on PAMI* Vol 7 No. 1, pag 127-129, January, 1985.
- [9] Li, H., Yang, H.S. Fast and Reliable Image Enhancement Using Fuzzy Relaxation Technique, *Proc. of the IVth Int. Conference on Pattern Recognition*, Cambridge (U.K.) pag. 577-586, 1988.
- [10] Lumia, R., Haralick, R.M., Zuniga, O., Shapiro, L.G. Texture Analysis of Aerial Photographs, *Pattern Recognition*, vol. 16, nº1, pag. 39-46, 1983.
- [11] Pal, K. et al., Image Description and Primitive Extraction Using Fuzzy Sets, *IEEE Trans. SCM* 13, 1983.
- [12] Palubinskas, G., A comparison of texture-based and fuzzy classification approaches for regenerating tropical forest mapping using Landsat TM, *Proc. ISPRS*, Munich (Germany), pag. 640-647, 1994.
- [13] Pao, Y.H., Vague Features and Vague Decision Rules: The Fuzzy-Set Approach, *Adaptive Pattern*

Recognition and Neural Networks, Ed. Addison Wesley, 1989.

- [14] Sobrevilla, P., Puig, D., Montseny, E., Segmentation of Textured Aerial Images Using OWA Operators, *Proc. of the Int. Conference of Information Processing and Management of Uncertainty in Knowledge Based Systems*. Granada, (Spain). 1996.
- [15] Vilnrotter, F.M., Nevatia, R., Price, K.E., Structural analysis of natural textures, *IEEE Trans. PAMI-8*, n° 1, pag. 76-89, 1986.
- [16] Thompson, W., Kearney, J., Inexact Vision, *Proc. Workshop on Motion*, Kiawah Island (SC), 1986.
- [17] Yager, R.R., Families of OWA operators, *Fuzzy Sets and Systems*, n°59, pag. 125-148, 1993.
- [18] Zadeh, L.A., Fuzzy Sets, *Information and Control*, n°8, pag. 338-353, Academic Press, New York (USA), 1965.

FUZZY-LISP: UN MODELO DE CALCULO FUNCIONAL BORROSO

J.L. Pérez-de-la-Cruz R. Morales-Bueno R. Conejo

Dpto. Lenguajes y Ciencias de la Computación

Universidad de Málaga.

Apdo. 4114.

MÁLAGA 29080

E-mail: cruz@tecma1.ctima.uma.es

Resumen

Se define teóricamente una extensión del modelo de evaluación Lisp básico, de forma que, mediante el empleo sistemático del principio de extensionalidad, se puedan evaluar tanto expresiones difusas como expresiones de argumentos difusos. Se presenta una sencilla implementación de este modelo, realizada en Common Lisp.

Palabras Clave: Programación, Lenguaje, Función.

1 INTRODUCCION

La definición de lenguajes de programación capaces de procesar datos o construcciones difusas tiene gran interés, tanto para la ejecución de algoritmos difusos como para la ejecución de algoritmos nítidos aplicados a datos difusos. Por ello se han propuesto en la literatura varios lenguajes de esta clase: LPL (Adamo 80), FRIL (Baldwin 84), HALO (Clark 91), L (Morales 93), XL (Morales 96). Sin embargo, la definición e implementación de tales lenguajes no es tarea exenta de dificultades, tanto teóricas como prácticas.

En este trabajo proponemos tomar como base el modelo aplicativo del lenguaje Lisp, dado por las funciones EVAL y APPLY (McCarthy 59), para definir a partir de él, mediante el uso sistemático del principio de extensión, el correspondiente modelo de cálculo difuso. Lisp se puede definir como un sistema que aplica la función EVAL a cada expresión que se le propone. Para ello, EVAL procesa la expresión; en particular, si ésta se puede interpretar como $f(a_1, \dots, a_n)$, EVAL aplica la función f con argumentos $a_1, \dots, a_n$. Dedicaremos la sección 2 a exponer muy brevemente el modelo de cálculo dado por EVAL, es decir, una versión simplificada del lenguaje Lisp.

Como es notorio, EVAL está definida únicamente para expresiones nítidas. Ahora bien, si los argumentos $a_1, \dots, a_n$ son difusos, la aplicación del principio de extensión (Zadeh 75) permite definir el significado de f para argumentos difusos, de una forma natural. Concretamente, sea $f(x, y)$ una función nítida de argumentos nítidos. Si x es un argumento difuso dado por la función de pertenencia $\mu_x(w)$ e y es un argumento difuso dado por la función de pertenencia $\mu_y(w)$, entonces la aplicación de f con argumentos difusos x e y produce el conjunto difuso $f(x, y)$ dado por la función de pertenencia

$$\mu_{f(x,y)}(w) = \begin{cases} \sum_{f(u,v)=w} \mu_x(u) \times \mu_y(v) \\ 0 & \text{si } f^{-1}(w) = \emptyset \end{cases}$$

donde se emplean los símbolos $+$ y $\times$ para denotar las operaciones de unión (conorma) e intersección (norma) difusas.

En la sección 3, que constituye el núcleo del trabajo, definimos una nueva función EVAL-F1 que extiende la semántica de manera que, siguiendo las reglas deducidas del principio de extensión, se pueden procesar también expresiones de argumento difuso. En la sección 4 definimos otra variante EVAL-F2 que procesa además expresiones difusas más generales. En la sección 5 presentamos algunos ejemplos resueltos con nuestra propuesta, y finalmente extraemos algunas conclusiones y nos referimos a posibles trabajos futuros.

2 EL MODELO NANO-LISP

Definimos a continuación el modelo de cálculo Nano-Lisp, que corresponde a la esencia de la familia de lenguajes Lisp. Sea C un conjunto cualquiera. Representaremos como $L(C)$ al conjunto de listas sobre C , es decir, al conjunto definido por la siguiente inducción:

() es una lista

Si $c_1, \dots, c_n \in C \cup L(C)$, $(c_1 \dots c_n) \in L(C)$

Sea N un conjunto de números. Sea S un conjunto de objetos, al que llamaremos símbolos. Supondremos que S contiene al menos los elementos T y NIL . Sea el conjunto de átomos $A = N \cup S$. El conjunto de expresiones nano-Lisp es $E = A \cup L(A)$. Por definición, NIL y $()$ son expresiones idénticas.

Llamemos $F(E)$ a la unión de los conjuntos de funciones parciales de E_n en E , $n = 0, 1, \dots$. Una ligadura es un par $(s, v) \in S \times E$. Un significado funcional es un par $(s, f) \in S \times F(E)$. Un entorno es un conjunto de ligaduras y significados funcionales, $Env \subset S \times (E \cup F(E))$, tal que no hay dos ligaduras distintas $(s, v_1), (s, v_2)$ para el mismo símbolo. Un entorno Env determina pues una función parcial de E en E , y otra función parcial de E en $F(E)$. Llamemos $\mathcal{E}$ al conjunto de todos los entornos. Llamemos $(s, a) * Env$ al entorno resultante de añadir (s, a) a Env , retirando si existe la ligadura anterior de s (si a es un símbolo) o el significado funcional anterior de s (si a es una función).

$EVAL$ es una función parcial, $EVAL: E \times \mathcal{E} \rightarrow E$, definida de la siguiente manera:

- si $e \in N$, $EVAL(e, Env) = e$
- si e es la lista vacía o el símbolo T , $EVAL(e, Env) = e$
- si $e \in S$ y $(s, v) \in Env$, $EVAL(e, Env) = v$; en otro caso, queda indefinido.
- si e es la lista $(QUOTE\ e_2)$, $EVAL(e, Env) = e_2$
- si e es la lista $(COND\ (c_1\ a_1) \dots (c_n\ a_n))$, $EVAL(e, Env)$ se determina de la siguiente forma: sean $C_1 = EVAL(c_1, Env)$, ..., $C_n = EVAL(c_n, Env)$, $A_1 = EVAL(a_1, Env)$, ..., $A_n = EVAL(a_n, Env)$ (existentes o indefinidas). Sea C_a la primera distinta de NIL , sea C_b la primera indefinida. Si $a < b$, o existe a pero no existe b , entonces $EVAL(e, Env) = A_a$. Si $b < a$, o existe b pero no existe a , entonces $EVAL(e, Env)$ queda indefinida. Si no existen a ni b , entonces $EVAL(e, Env) = NIL$. Quizás se comprenda mejor lo anterior llamando "falso" a NIL , "verdadero" a cualquier expresión no NIL , y diciendo que $COND$ evalúa las c_i hasta encontrar una verdadera, en cuyo caso devuelve el valor de la A_i asociada, y si no hay ninguna verdadera, $COND$ devuelve "falso".

Por ejemplo (introduciremos el argumento de $EVAL$ tras la señal $>$, y el valor evaluado en la línea siguiente):

```
> (COND (1 2) (KK KK))
```

2

```
> (COND (KK KK) (1 2))
```

Indefinido: KK indefinido

-si e es la lista $(e_1 \dots e_n)$, e_1 es un símbolo, existe una función f tal que $(e_1, f) \in Env$, y para los restantes elementos e_i ($i > 1$) de la lista existe $EVAL(e_i, Env) = v_i$, entonces $EVAL(e, Env) = f(v_2, \dots, v_n)$; en otro

caso, si e es una lista y s es un símbolo, $EVAL(e, Env)$ queda indefinido.

Por ejemplo, si suponemos el significado habitual de los operadores aritméticos,

```
> (+ (+ 1 1) 2)
```

4

-si e es la lista $(e_1 \dots e_n)$, e_1 es una lista $(LAMBDA\ l\ e_l)$, l es una lista de n símbolos $p_1, \dots, p_n$ llamados parámetros, entonces $EVAL(e, Env) = f^*(e)$, donde f^* es la función de aridad n definida de la siguiente manera: $f(a_1, \dots, a_n)$ es $EVAL(e, Env_1)$ donde Env_1 es $(p_1\ v_1) * \dots * (p_n\ v_n) * Env$.

Por ejemplo,

```
> ((LAMBDA (X Y) (+ X Y)) 1 2)
```

3

$EVAL^*$ es una función parcial $EVAL^*: E \times \mathcal{E} \rightarrow E \times \mathcal{E}$, definida de la siguiente manera:

-todos los casos anteriores se extienden de manera trivial, dejando inalterado el entorno.

-si e es $(SETQ\ s\ e)$, s es un símbolo y $EVAL(e, Env) = v$, entonces $EVAL^*(e, Env) = (s, (s, v) * Env)$

-si e es $(DEFUN\ s\ l\ e)$, s es un símbolo, l es una lista de n símbolos $p_1, \dots, p_n$ llamados parámetros, entonces $EVAL^*(e, Env) = (s, (s, f) * Env)$, donde f es la función de aridad n definida de la siguiente manera: $f(a_1, \dots, a_n)$ es $EVAL(e, Env_1)$ donde Env_1 es $(p_1\ v_1) * \dots * (p_n\ v_n) * Env$.

Un programa es una sucesión de expresiones, que se evalúan según las reglas anteriores. Al principio las expresiones se evalúan en un entorno inicial dado por el sistema, donde existen significados predefinidos (por ejemplo, los operadores aritméticos, CAR para la función que devuelve la cabeza de una lista, CDR para la que devuelve la cola, $CONS$ para el constructor de listas a partir de cabeza y cola, etc). Cada expresión e devuelve un valor $EVAL(e)$ y posiblemente modifica el entorno donde se evaluó. Estas modificaciones tienen vigencia:

-en todas las evaluaciones que sigan (ligaduras globales). Este es el caso de las modificaciones establecidas por $DEFUN$, y por $(SETQ\ s \dots)$ si s no tenía ninguna ligadura.

-en las evaluaciones de las subexpresiones de la expresión que introdujo la ligadura que ahora se modifica (ligaduras léxicas). Este es el caso de las modificaciones establecidas por $(SETQ\ s \dots)$, si s tenía ya una ligadura.

Por ejemplo,

```
> (SETQ X (* 3 4))
```

12

```
> (SETQ Y (CONS X (QUOTE (B C))))
```

```
(12 B C)
```

3 EL MODELO NANO-FLISP1

En este apartado se describe la aportación fundamental de este trabajo: la extensión del modelo de cálculo Nano-Lisp a las funciones y expresiones difusas.

Sean N , S , A y E como antes. Consideremos como conjunto de grados el intervalo real $[0,1]$. Definimos $FUZZ(E)$ como el conjunto de las partes difusas de E con soporte finito.

El conjunto de expresiones Nano-Flispl es $E' = E \cup FUZZ(E) \cup L(E) \cup L(FUZZ(E)) \cup FUZZ(L(FUZZ(E)))$. Identificaremos el conjunto difuso $1/1 + .2/2$ mediante la notación $\#S(CFUZZY ALIST ((1 . 1) (2 . 0.2)))$. Por ejemplo, son expresiones difusas

```
A
(A 1)
#S(CFUZZY ALIST ((1 . 1) (2 . 0.2)))
(+ #S(CFUZZY ALIST ((1 . 1)
                    (2 . 0.2)))
   #S(CFUZZY ALIST ((1 . 1)
                    (2 . 0.2))))
```

Basándonos en la descripción anterior de EVAL y EVAL*, definimos ahora nuestras funciones parciales EVAL-F1: $E' \rightarrow E'$, y EVAL-F1*: $E' \times E' \rightarrow E'$. Informalmente, el modelo de cálculo queda descrito de la forma siguiente:

-se introduce como predefinida la función de construcción de conjuntos difusos HAZ-FUZZY.
Por ejemplo, (introduciremos el argumento de EVAL-F1 tras la señal EVAL-FUZZY1>, y el valor evaluado en la línea siguiente):

```
EVAL-FUZZY1> (SETQ F1 (HAZ-FUZZY 1 0.8
3 0.8 5 0.5))
Valor: 1; Grado: 0.8
Valor: 3; Grado: 0.8
Valor: 5; Grado: 0.5
(Se construye un conjunto difuso y se liga a F1)
```

-los conjuntos difusos se evalúan a sí mismos.

Por ejemplo,
EVAL-FUZZY1> $\#S(CFUZZY ALIST ((1 . 1) (2 . 0.3) (3 . 0.2)))$
Valor: 1; Grado: 0.1
Valor: 2; Grado: 0.3
Valor: 3; Grado: 0.2

-la expresión condicional se evalúa de la siguiente forma: se evalúan en todo caso todas las condiciones c_i a sus valores C_i . Para cada C_i se calcula un número nítido V_i componiendo -mediante la operación de unión o conorma- los grados de sus elementos distintos

de NIL, es decir, $V_i = \sum_{x \neq NIL} \mu_{C_i}(x)$. Se evalúan en todo

caso todas las acciones a_i , dando los valores A_i . Cada uno de ellos se modula por el correspondiente V_i obteniendo A_i' , es decir, $\mu_{A_i'}(x) = V_i \times \mu_{A_i}(x)$. La forma condicional se evalúa a la unión difusa U de todos los A_i' , es decir, $\mu_U(x) = \sum_i \mu_{A_i'}(x)$. La justificación de esta regla de

evaluación es clara si pensamos en los valores difusos como distribuciones de posibilidad, análogas a las de probabilidad.

Por ejemplo,

```
EVAL-FUZZY1> (COND (F1 1) (T F1))
Valor: 1; Grado: 0.3
Valor: 2; Grado: 0.3
Valor: 3; Grado: 0.2
```

-las listas cuyo primer elemento es un símbolo cuyo significado funcional es f , y cuyos restantes argumentos se evalúan a valores difusos, se evalúan de la siguiente forma: se aplica f a todas las combinaciones posibles de valores nítidos formados a partir de los argumento, y se componen los resultados según el principio de extensionalidad.

-análogamente se evalúan las listas cuyo primer elemento es una lista (LAMBDA ...)

Nótese que todas las funciones, predefinidas o definidas por el usuario mediante DEFUN, pasan de esta manera a procesar también argumentos difusos. Nótese además que DEFUN y SETQ siguen disponibles en EVAL-F1.

Por ejemplo,

```
EVAL-FUZZY1> (SETQ F2 (+ F1 F1))
Valor: 2; Grado: 0.1
Valor: 3; Grado: 0.1
Valor: 4; Grado: 0.3
Valor: 5; Grado: 0.2
Valor: 6; Grado: 0.2
```

(Se ha aplicado la versión extensional de + para calcular el valor)

```
EVAL-FUZZY1> (DEFUN FOO (X) (* X X))
FOO
EVAL-FUZZY1> (FOO 4)
16
```

(Se da un significado funcional nítido a FOO)

```
EVAL-FUZZY1> (FOO F1)
Valor: 1; Grado: 0.1
Valor: 4; Grado: 0.3
Valor: 9; Grado: 0.2
```

(Se aplica la versión extensional de FOO)

```
EVAL-FUZZY1> (* F1 F1)
Valor: 1; Grado: 0.1
```

Valor: 2; Grado: 0.1
 Valor: 3; Grado: 0.1
 Valor: 4; Grado: 0.3
 Valor: 6; Grado: 0.2
 Valor: 9; Grado: 0.2

(Nótese que el resultado difiere del anterior, contra lo que ocurre en el caso nítido)

4 EL MODELO NANO-FLISP2

El conjunto de expresiones Nano-Flisp1 es también

$E' = E \cup \text{FUZZ}(E) \cup L(E) \cup L(\text{FUZZ}(E)) \cup \text{FUZZ}(L(\text{FUZZ}(E)))$.

Pero ahora consideramos de otra forma los conjuntos difusos de listas. En el modelo de cálculo anterior, los conjuntos difusos se evalúan a sí mismos. Aplicando este mismo criterio, un conjunto difuso de listas se evaluaría a sí mismo. Sin embargo, es posible definir una función EVAL-F2 que siga una regla distinta:

-si e es un conjunto difuso #S(CFUZZY ALIST ((v1 . g1) ... (vn . gn))), se evalúa cada v_i dando V_i . EVAL(e, Env) es la unión U de todos los valores resultantes, modulados multiplicativamente por los correspondientes grados g_i . Es decir,

$$\mu_U(x) = \sum_i g_i \times \mu_{V_i}(x).$$

De esta forma

```
EVAL-FUZZY2> #S(CFUZZY ALIST
  (((CAR '(A B)) . 0.3)
   ((CAR '(E F)) . 1)))
#S(CFUZZY ALIST ((A . 1) (E . 0.3)))
pero
EVAL-FUZZY1> #S(CFUZZY ALIST
  (((CAR '(A B)) . 0.3)
   ((CAR '(E F)) . 1)))
#S(CFUZZY ALIST (((CAR '(A B)) . 0.3)
  ((CAR '(E F)) . 1)))
```

5 EJEMPLOS

Consideremos el sencillo ejemplo de (Dubois y Prade 1988), pg. 57-60 acerca de la estimación de la financiación disponible para un proyecto. A es una cantidad nítida, B, C y D son distribuciones de posibilidad, y la financiación esperada es la suma de las cuatro cantidades. En nuestro sistema se mantiene el siguiente diálogo:

```
EVAL-FUZZY1> (SETQ A 100)
100
EVAL-FUZZY1> (SETQ B
  (HAZ-FUZZY 50 1 60 1 70
```

```
1 80 0.66 90 0.33))
Valor: 50; Grado: 1
Valor: 60; Grado: 1
Valor: 70; Grado: 1
Valor: 80; Grado: 0.66
Valor: 90; Grado: 0.33
EVAL-FUZZY1> (SETQ C
  (HAZ-FUZZY 0 0.5 100 1
    110 1))
Valor: 0; Grado: 0.5
Valor: 100; Grado: 1
Valor: 110; Grado: 1
EVAL-FUZZY1> (SETQ D
  (HAZ-FUZZY 0 1
    20 0.8))
Valor: 0; Grado: 1
Valor: 20; Grado: 0.8
EVAL-FUZZY1> (SETQ TOTAL (+ A B C D))
Valor: 150; Grado: 0.5
Valor: 160; Grado: 0.5
Valor: 170; Grado: 0.5
Valor: 180; Grado: 0.5
Valor: 190; Grado: 0.5
Valor: 200; Grado: 0.5
Valor: 210; Grado: 0.33
Valor: 250; Grado: 1
Valor: 260; Grado: 1
Valor: 270; Grado: 1
Valor: 280; Grado: 1
Valor: 290; Grado: 0.8
Valor: 300; Grado: 0.8
Valor: 310; Grado: 0.66
Valor: 320; Grado: 0.33
```

El resultado coincide, salvo la discretización realizada, con el dado en las páginas citadas.

Consideremos ahora un ejemplo de algoritmo difuso propuesto en (Zadeh 68) y ya considerado en (Morales 93): Si x es grande, incrementar y en varias unidades; si x es pequeño, decrementar y en varias unidades; en otro caso, dejar y sin modificar. En primer lugar, establezcamos los valores difusos de "varios", "límite-grande" y "límite-pequeño":

```
(SETQ VARIOS
  (HAZ-FUZZY 2 0.8 3 0.9 4 1 5 1 6 1
    7 1 8 0.8 9 0.6 10 0.4 11 0.2))
(SETQ L-G
  (HAZ-FUZZY 20 0.2 40 0.4 60 0.6
    80 0.8 100 1))
(SETQ L-P
  (HAZ-FUZZY 20 0.2 40 0.4 60 0.6
    80 0.8 100 1))
```

Consideramos que es verdadero que un número es grande si es mayor o igual que "límite-grande, y falso si es menor. Lo mismo para el predicado "pequeño":

```
(DEFUN GRANDE-P (X)
  (COND ((>= X L-G) T)
        ((< X L-G) NIL)))
(DEFUN PEQUENO-P (X)
```

```
(COND ((<= X L-P) T)
      (> X L-P) NIL)))
```

El algoritmo propuesto es implementado por la función ZADEH:

```
(DEFUN ZADEH (X Y)
  (COND ((GRANDE-P X) (+ Y VARIOS))
        ((PEQUENO-P X) (- Y VARIOS))
        ((NOT (GRANDE-P X)) (COND
          ((NOT (PEQUENO-P X)) Y))))))
```

Y ahora tenemos

```
EVALFUZZY1> (ZADEH 10 10)
Valor: -1; Grado: 0.2
Valor: 0; Grado: 0.4
Valor: 1; Grado: 0.6
Valor: 2; Grado: 0.8
Valor: 3; Grado: 1
Valor: 4; Grado: 1
Valor: 5; Grado: 1
Valor: 6; Grado: 1
Valor: 7; Grado: 0.9
Valor: 8; Grado: 0.8
```

Daremos un último ejemplo, propuesto en (Ostasiewicz 82) y también considerado en (Morales 93): Sea la sucesión [2, 4, 5, 3, 6, 7, 1, 87]. Sumar unos cuantos de estos números. Ligamos L a la sucesión propuesta (SETQ L '(2 4 5 3 6 7 1 87))

y establecemos el valor difuso de "unos-cuantos":

```
(SETQ UNOS-CUANTOS
  (HAZ-FUZZY 3 .8 4 1 5 0.8))
```

La función es sencillamente la aplicación de + a la subsucesión que va del elemento 0-ésimo (incluido) al UNOS-CUANTOS-ésimo (excluido):

```
(DEFUN OST (L)
  (APPLY #' +
    (SUBSEQ L 0 UNOS-CUANTOS)))
```

Y ahora tenemos

```
EVALFUZZY1> (OST L)
Valor: 11; Grado: 0.8
Valor: 14; Grado: 1
Valor: 20; Grado: 0.8
```

6 CONCLUSIONES

Los ejemplos anteriores muestran la potencia del enfoque adoptado para definir y manipular datos y algoritmos difusos. Por otra parte, la definición teórica del modelo de cálculo es notablemente sencilla: basta con modificar EVAL para que aplique sistemáticamente el principio de extensión. Se ha implementado un prototipo que se comporta de la manera indicada, de forma que un intérprete Common Lisp (concretamente, CLISP) pase a funcionar de la manera descrita, sobrecargando todas las funciones -ya sean predefinidas, definidas por el usuario

anteriormente, o definidas en lo sucesivo- con un significado difuso. La tarea siguiente consistirá en la extensión de EVAL-FUZZY a otros macros y operadores especiales (no funciones propiamente dichas) de Lisp: LET, BLOCK, DO, ...

Existen numerosas propuestas prácticas que consideran el procesamiento de conjuntos de soporte infinito. Concretamente, estas propuestas suelen limitarse a realizar la implementación de algunos operadores especiales para intervalos reales difusos. Sin embargo, nuestro enfoque no trata ningún conjunto de soporte infinito. Ello se debe a que el principio de extensión, aplicado directamente a estos conjuntos, lleva a computaciones también infinitas. Creemos que, para llegar a una solución más general de carácter teórico, podrían intentarse varios caminos: por ejemplo, la evaluación perezosa del soporte representado por una corriente, su discretización automática o su muestreo.

REFERENCIAS.

- Adamo, J. M. (1980): A fuzzy programming language. *Fuzzy Sets and Systems* 3 151-179; 3 262-289.
- Baldwin, J. F. y Zhou, S. Q. (1984): A fuzzy relational inference language. *Fuzzy Sets and Systems* 14 155-174.
- Clark, F. y Kandel, A. (1991): HALO -a fuzzy programming language. *Fuzzy Sets and Systems* 44 199-208.
- Dubois, D. y Prade, H. (1988): *Possibility Theory*. Plenum Press, New York.
- McCarthy, J. (1960): Recursive functions of symbolic expressions. *Communications ACM* 3(4) 184-195.
- Morales, R. et al. (1993): An elementary fuzzy programming language. *Fuzzy Sets and Systems* 57 55-74.
- Morales, R. et al. (1996): A family of fuzzy programming languages. *Fuzzy Sets and Systems* (to appear).
- Ostasiewicz, W. (1982): A new approach to fuzzy programming. *Fuzzy Sets and Systems* 7 139-152.
- Zadeh, L. (1968): Fuzzy algorithms. *Information and Control* 12 94-102.
- Zadeh, L. (1975): The concept of a linguistic variable and its application to approximate reasoning. *Information Science* 8 199-249; 8 301-357; 9 43-80

Una Arquitectura para Bases de Datos Relacionales Difusas Deductivas*

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

Resumen

Este artículo presenta la arquitectura de un Sistema Gestor de Bases de Datos Relacionales Difusas (SGBDRD) con capacidad de deducción, cuyas principales características son: 1) está construido sobre la base de un modelo teórico de bases de datos difusas y sobre un modelo lógico de bases de datos difusas; 2) Se implementa completamente sobre SGBDR clásicos, utilizando los recursos que estos proporcionan; 3) Conserva todas las operaciones nativas del SGBDR anfitrión potenciándolo con capacidad para tratar información difusa y deductiva; 4) Proporciona una extensión de SQL (DFSQ) que permite expresar sentencias deductivas y difusas y un procesador para este lenguaje que traduce estas sentencias a sentencias SQL ejecutables por el SGBDR anfitrión.

Palabras clave: Base de datos deductiva, bases de datos difusa.

1 Introducción

Desde la aparición del modelo Relacional de Base de Datos (BDR) propuesto por Codd, han surgido varias propuestas para proporcionar capacidad al modelo para representar y manipular información difusa. Las más importantes son las de Buckles & Petry [3, 11], Umano [12], Prade & Testemale [10] and Zemankova & Kandel [16]. En [6], presentamos un modelo, denominado GEFRED, que trata de aglutinar en un marco común los aspectos más importantes de las anteriores propuestas.

Los modelos teóricos de bases de datos difusas precisan mecanismos para construir sistemas relacionales difusos (SGBDRD) que funcionen de acuerdo con esos principios. En [2, 7, 10, 12, 16] hay algunas propuestas al respecto. En [5, 14, 15] mostramos algunas ideas de como tratar información difusa utilizando SGBDR convencionales. De otro lado, nos planteamos la posibilidad de introducir información mediante reglas. Esto nos permite definir conceptos definidos recursivamente. En [9, 13] hemos presentado una aproximación lógica a las bases de datos difusas que nos permite deducir información difusa (o no) a partir de un conjunto de reglas utilizando "hechos difusos".

Integrando todo esto, la arquitectura que presentamos está basada en los siguientes aspectos:

1. Un modelo teórico adecuado de bases de datos relacionales difusas.
2. Un modelo teórico adecuado de bases de datos lógicas difusas.
3. El empleo de los recursos proporcionados por SGBDR convencionales para implementar representación de datos difusos y el tratamiento de los mismos.
4. El empleo de los recursos proporcionados por sistemas deductivos convencionales (Prolog en este caso) para implementar deducción difusa.

Por tanto, lo que pretendemos es posibilitar resolver consultas que implican procesos deductivos no resolubles por los SGBDR convencionales. Nuestro SGBDR Deductivo-Difuso (SGBDRDD) podrá trabajar con dos tipos de tablas: *Tablas Extensivas*: cuyos datos se encuentran almacenados físicamente en la base de datos o pueden ser calculados por métodos relacionales (p.e. vistas) y *Tablas Intensivas*: cuyos datos han de ser calculados mediante un módulo de deducción externo.

Para ilustrar la capacidad de esta arquitectura, consideremos una base de datos con el siguiente esquema:

PADRES(Nombre,Padre,Madre)
PERSONAS_FALLECIDAS(Nombre,Edad,Fecha)
PERSONAS_VIVAS(Nombre,Dirección)

y la consulta "Encontrar los nombres de personas junto con los de sus antecesores tal que el antecesor haya muerto joven". Nótese que no existe información explícita en la BD acerca de los *antecesores* y que esta información no puede ser obtenida en términos relacionales, y que existe un valor de atributo etiquetado *joven*, lo cual implica un tratamiento difuso de la información.

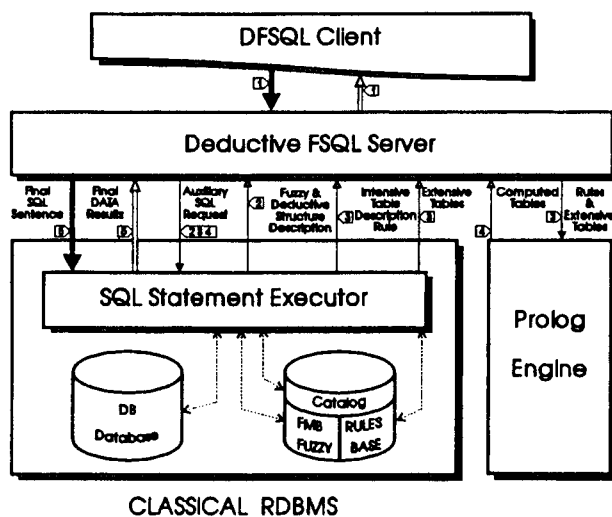
Para resolver este tipo de problemas, necesitamos, primero una implementación relacional de la regla que define la tabla intensiva *antecesor*, para representarla en la bases de datos. Una vez determinada

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

la representación, el SGBDRDD tiene que construir la regla en formato lógico y enviarlo al módulo de deducción. Todo este proceso será descrito en los próximos apartados. Una vez que el módulo de deducción ha recibido la regla, calcula la tabla *antecesor* que es usada entonces por el SGBDRDD para resolver el resto de la consulta.

El artículo está organizado como sigue: En la sección 2 se muestran los aspectos más importantes de la arquitectura. Describimos sus módulos, prestando una atención especial al Servidor de SQL Deductivo-Difuso. En la sección 3 explicamos como se implementa en el SGBDRDD la información intensiva y la información difusa y los ilustramos mediante un ejemplo. Finalmente, la sección 4 resume las contribuciones más importantes de este trabajo así como las líneas futuras de investigación.

2 Arquitectura del SGBDRDD



CLASSICAL RDBMS

Figura 1: Arquitectura General

La figura 1 muestra la arquitectura general del sistema; éste supone una extensión deductiva y difusa de un SGBDR convencional, adoptando una estructura modular que permite una organización Cliente/Servidor. El módulo principal es el SGBDR, el cual actúa como anfitrión; alrededor de él se desarrollan los módulos encargados de realizar el tratamiento deductivo y difuso de la información.

2.1 Descripción de los Módulos

- El **SGBDR** es el sistema empleado para implementar esta arquitectura. Proporciona recursos para la gestión del esquema relacional y herramientas de programación para ayudar en la construcción del SGBDRDD. Le acoplaremos una extensión de SQL denominada DFSQL que permitirá expresar sentencias deductivas y difusas
- **BD** almacena toda la información extensiva, sea o no difusa.
- **FMB**, la Base de Metaconocimiento Difuso, es una extensión del catálogo del SGBDR que almacena las definiciones de las estructuras y datos di-

fusos contenidos en la BD, también referencia las tablas intensivas presentes en la BD. Sigue un esquema de organización coherente con el catálogo del SGBDR.

- **RB**, la Base de Reglas es otra extensión del catálogo del SGBDR anfitrión que contiene toda la información relativa a la tablas intensivas definidas en la BD. Su organización se muestra en la sección 3.3.
- El módulo **Servidor de SQL Deductivo-Difuso** procesa las peticiones deductivo-difusas al sistema expresadas en DFSQL. En la sección 2.2 describiremos la sintaxis, su organización y funcionamiento.
- La **Máquina Prolog**, permite obtener la información intensiva requerida. El Servidor de DFSQL tiene que determinar que reglas, tablas extensivas y definiciones son necesarias enviarle para que obtenga esa información.

2.2 El Servidor de SQL Deductivo-Difuso

Este módulo atiende las peticiones de aplicaciones clientes expresadas en DFSQL. La figura 1 ilustra como opera este módulo en el proceso de resolución de una consulta. Los pasos que sigue son:

1. Acepta una petición DFSQL (flecha negra 1) y separa las cláusulas que implican un tratamiento difuso o intensivo. Para hacer esto explora la FMB mediante consultas SQL auxiliares (flechas negras 2 3 4).
2. De la FMB obtiene la estructura de las cláusulas difusas, de la RB obtiene las reglas que definen las tablas intensivas y, de la BD, las tablas extensivas implicadas (flechas blancas 2 y 3, respectivamente).
3. Envía toda la información implicada en el proceso de deducción a la máquina Prolog (flecha negra 3), obtiene los resultados (flecha blanca 4) y actualiza la BD con los mismos (flecha negra 2 3 4).
4. Traslada las cláusulas difusas a cláusulas SQL y compone una sentencia SQL final (flecha negra 5).
5. Ejecuta la sentencia y obtiene los resultados (flecha blanca 5) y la procesa antes de enviarla a la aplicación cliente (flecha blanca 1).

2.2.1 Sintaxis de DFSQL

Como hemos visto, el funcionamiento del sistema está basado en una versión extendida del SQL proporcionado por el SGBDR anfitrión. A continuación, mostramos en formato YACC algunas de las cláusulas nuevas que incorpora DFSQL.

• Lenguaje de Definición de Datos

Este sublenguaje incorpora sentencias SQL nuevas para definir tablas que contienen atributos difusos y para definir tablas intensivas. El sistema soporta tablas intensivas con atributos clásicos y difusos. Por

brevedad sólo vamos a reflejar aquí la sintaxis adoptada para la sentencia de definición de tablas intensivas.

```
create_table : CREATE INTENSIVE TABLE table_item
              '(' column_list ')' RULE rule_descpt ;

table_item   : id_user '.' id_table | id_table ;

column_list  : column_list ',' column_list
              | column_descpt ;

column_descpt : column_id datatype ;

data_type    : classical_datatype | fuzzy_datatype ;

fuzzy_datatype : crisp | possibilistic | nearness ;

nearness     : SCALAR '(' NATURAL_NUMBER ')'
              | SCALAR ;

rule_descpt  : '(' rule_list ')' ;

rule_list    : rule_list ';' rule_list | rule_item ;

rule_item    : rule_item AND rule_item | predicate ;

predicate    : table_descpt | condition ;

table_descpt : table_item '(' table_arg ')' ;

table_arg    : table_arg ',' table_arg | arg_item ;

arg_item     : '_' | var_id ;

var_id       : 'X' NATURAL_NUMBER ;

condition    : var_id comp var_id ;

comp         : '=' | '<' | '>' | '<=' | '>=' | '<>' ;
```

Los símbolos no terminales *user_id*, *table_id* y *column_id* identifican los elementos usuario, tabla y atributo, respectivamente, y siguen las reglas de formación establecidas en la sintaxis del SGBDR anfitrión. Los símbolos no terminales *crisp*, *possibilistic* y *nearness* identifican tipo de datos para atributos "difusos". El símbolo terminal *NATURAL_NUMBER* identifica un número natural. Los símbolos entre comillas simples son literales y el resto de los símbolos son palabras reservadas de la sintaxis.

• Lenguaje de Manipulación de Datos

La consulta es la operación que implica toda la capacidad expresiva de DFSQL, vamos a detallar las nuevas cláusulas incorporadas a la sentencia *SELECT* para poder hacer consultas que impliquen información intensiva y difusa.

```
fcond_simp   : fcond_wtout threshold | fcond_wtout ;

fcond_wtout  : column_item fcomp fuz_constant ;

threshold    : THOLD NUMBER | THOLD '$' ID ;

fuz_constant : '$' ID
              | number
              | '#' NUMBER
```

```
| '[' NUMBER ',' NUMBER ']'
| '$' '[' NUMBER ',' NUMBER ',' NUMBER ','
NUMBER ']' ;

fcomp        : FEQ | FGT | FLT | FGEQ | FLEQ ;

comp_deg     : CDEG '(' fuz_arith_op ')' ;

agre_funct   : FMAX '(' column_item ')' | FMIN '('
              column_item ')'
              | FSUM '(' column_item ')' | FAVG '('
              column_item ')' ;

fuz_arith_op : fuz_arith_op '+' fuz_arith_op
              | fuz_arith_op '-' fuz_arith_op
              | fuz_arith_op '*' fuz_arith_op
              | fuz_arith_op '/' fuz_arith_op
              | '(' fuz_arith_op ')'
              | column_item
              | fuz_constant ;

column_item  : table_id '.' column_id | table_alias '.'
              column_id
              | column_id ;
```

El símbolo terminal *NUMBER* identifica un dato numérico. Los símbolos restantes son palabras reservadas de la nueva sintaxis, con la que podemos expresar condiciones, modificadores y funciones "difusos". El símbolo no terminal *fcomp* representa un conjunto de comparadores difusos. Las condiciones compuestas se obtienen mediante el uso de los conectivos *NOT*, *AND* y *OR* sobre condiciones atómicas que pueden ser de naturaleza difusa o no.

Las siguientes podrían ser sentencias válidas de la sintaxis de DFSQL adoptada, (asumiendo la existencia de las tablas y etiquetas referenciadas).

```
CREATE INTENSIVE TABLE antecesor (X char(30), Y char(30))
RULE(
  padres(X,Y,_);
  padres(X,_,Y);
  padres(X,Z,_) and antecesor(Z,Y);
  padres(X,_,Z) and antecesor(Z,Y)).
```

```
SELECT FACOUNT,FAVG(Sal),FMAX(Sal),FMIN(Sal)
FROM empleados
GROUP BY edad THOLD 0.8;
```

```
SELECT Nombre,Y FROM padres,antecesor WHERE Nombre=X
and Y in (SELECT Nombre FROM personas_fallecidas
WHERE edad='Joven');
(consulta de la seccion 1)
```

3 Implementación de Información Intensiva en el DFRDBMS

Esta sección describe principalmente los criterios y técnicas empleadas para implementar información intensiva en el SGBDR anfitrión. Los detalles relativos a la implementación de información difusa en el SGBDR pueden encontrarse en [5].

En cualquier caso, los criterios de implementación se aplican sobre tres elementos principalmente: La base de datos, el catálogo del sistema y el servidor de DFSQL.

En la base de datos adoptamos un esquema de representación relacional mostrado en [5] que proporciona capacidad para albergar datos difusos de diferentes tipos.

En el catálogo del sistema incorporamos una extensión denominada Base de Metaconocimiento Difuso en la que recogemos toda la información necesaria para identificar los elementos del esquema relacional que han de recibir tratamiento "difuso" y/o intensivo. Una parte importante es la *base de reglas* cuya estructura estudiaremos con más detalle en el próximo subapartado.

El servidor de DFSQL implementa los procedimientos difusos y es el motor de todas las operaciones destinadas a resolver las peticiones al sistema.

3.1 Representación de las Reglas

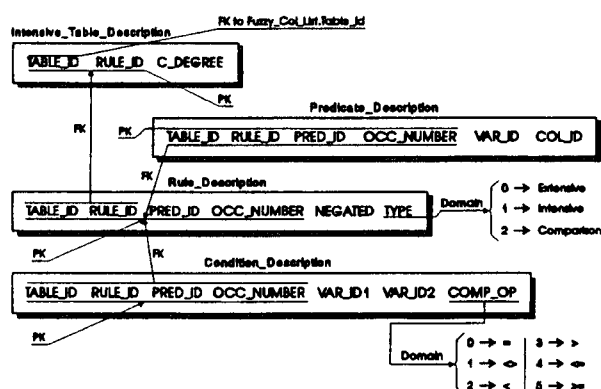


Figura 2: Representación relacional de las reglas.

La arquitectura que proponemos apuesta por representar las reglas que definen el comportamiento intensivo de la base de datos utilizando el esquema relacional. Para ello hemos ideado una estructura de tablas que nos permite organizar toda la información relativa a dichas reglas. Esta estructura, que constituye la *base de reglas* (RB), se muestra en la figura 2 y su descripción es la siguiente:

• **Intensive_Table_Description**: Esta tabla contiene una descripción de todas las tablas intensivas que se encuentran definidas y los identificadores de las reglas empleadas para su definición. Los atributos son:

- **TABLA_ID**: Contiene el identificador de tabla intensiva, es una referencia al valor asignado a esta tabla en el catálogo de la BD. Las tablas intensivas se encuentran definidas en la BD de igual forma que las tablas intensivas, lo que no se encuentra de forma extensiva son las tuplas que contienen.
- **RULE_ID**: Contiene un identificador para cada una de las reglas que se emplean para definir intensivamente el contenido de la tabla intensiva. Estos identificadores son asignados automáticamente por el SGB-DRDD.

• **Rule_Description**: Esta tabla contiene la descripción de cada regla, esto es, los predicados implicados. Sus atributos son:

- **TABLA_ID, RULE_ID**: Estos atributos identifican externamente a qué regla de qué tabla, contenida en la tabla Intensive_Table_Description, corresponde el predicado.
- **PRED_ID**: Este atributo contiene el identificador del predicado implicado en una regla identificada mediante los dos atributos anteriores.
- **OCC_NUMBER**: Si un predicado aparece varias veces en el cuerpo de una regla este atributo se encarga de recoger un valor que identifica cada ocurrencia del mismo.
- **NEGATED**: Este atributo contiene un valor 1, si la ocurrencia de un predicado en la regla aparece negada, y 0 si no.
- **TYPE**: Este atributo contiene un valor tomado del conjunto {0,1,2} con el siguiente significado:
 - 0 → Extensivo, esto es, el identificador de predicado corresponde a una tabla que se encuentra almacenada en la base de datos.
 - 1 → Intensivo, osea, el identificador de predicado corresponde a una regla que se encuentra descrita en la tabla Intensive_Table_Description.
 - 2 → Comparación, osea, no se identifica un predicado sino una expresión de comparación entre variables presentes en la regla.

• **Predicate_Description**: Esta tabla contiene la descripción de cada ocurrencia de un predicado, esto es, las variables empleadas, el orden en que aparecen, etc. Sus atributos significan:

- **TABLA_ID, RULE_ID, PRED_ID y OCC_NUMBER**: Estos atributos identifican externamente a un predicado de la tabla Rule_Description.
- **VAR_ID**: Este atributo contiene los identificadores de las variables que aparecen en la expresión del predicado.
- **COL_ID**: Este atributo indica la posición en que la variable está situada dentro del predicado (si X_i está en el primero, segundo, ... lugar en la expresión del predicado P).

• **Condition_Description**: Esta tabla contiene la descripción de las operaciones de comparación que aparezcan en las reglas. Sus atributos son:

- **TABLA_ID, RULE_ID, PRED_ID y OCC_NUMBER**: Identifican externamente a una ocurrencia concreta de un predicado en una regla que define una tabla intensiva. Constituye, por tanto, una clave externa de la tabla Rule_Description.

- VAR_ID1 y VAR_ID2: Son identificadores de las variables implicadas en la operación de comparación.
- COMP_OP: Este atributo indica que operador de comparación emplea la expresión. Toma valores del conjunto {0,1,2,3,4,5} significando =, <>, <, >, <= y >=, respectivamente.

• Ejemplo.

El siguiente ejemplo ilustra como se representa en la base de reglas la tabla intensiva dada por la siguiente definición:

$P(X_1, X_2, X_3)$ if $Q_1(X_1, -)$ and not($Q_2(X_2, X_3)$) and ($X_1 < X_3$).

$P(X_1, X_2, X_3)$ if $Q_1(X_1, X_4)$ and $P(X_4, X_2, X_1)$ and $Q_3(X_3, -)$.

con las siguientes características:

- El predicado P ha sido definido utilizando dos reglas.
- Los predicados extensivos Q_1 , Q_2 y Q_3 que aparecen en la definición de P son tablas de la base de datos.
- En la definición de P aparece P , esto significa que se trata de una definición recursiva.
- En la definición de P , se considera una expresión de comparación ($X_1 < X_3$).

Las tablas 1, 2, 3 y 4 muestran como queda representada la definición de esta tabla intensiva en la base de reglas RB.

TABLE_ID	RULE_ID
P	P_1
P	P_2

Tabla 1: Tabla Intensive_Table_Description.

TAB_ID	RUL_ID	PRED_ID	OCC_NUM	NEG	TYPE
P	P_1	Q_1	1	0	0
P	P_1	Q_2	1	1	0
P	P_1	compl	1	0	2
P	P_2	Q_1	1	0	0
P	P_2	P	1	0	1
P	P_2	Q_3	1	0	0

Tabla 2: Tabla Rule_Description

4 Conclusiones y Trabajos Futuros

Este artículo se inscribe en una línea de trabajo que persigue proporcionar los elementos necesarios para el desarrollo de SGBDRDD. Para conseguir esto, hemos integrado un modelo teórico de base de datos relacionales difusas con un modelo lógico, aprovechando la capacidad del primero para manipular información

TAB_ID	RUL_ID	PRED_ID	OCC_NUM	VAR_ID	COL_ID
P	P_1	Q_1	1	1	1
P	P_1	Q_2	1	2	1
P	P_1	Q_2	1	3	2
P	P_2	Q_1	1	1	1
P	P_2	Q_1	1	4	1
P	P_2	P	1	4	1
P	P_2	P	1	2	2
P	P_2	P	1	1	3
P	P_2	Q_3	1	3	1

Tabla 3: Tabla Predicate_Description

TAB_ID	RUL_ID	PRED_ID	OCC_NUM	V_ID1	V_ID2	COMP_OP
P	P_1	compl	1	1	3	2

Tabla 4: Tabla Condition_Description

extensiva y la capacidad del segundo para tratar con información intensiva.

Este modelo puede ser contemplado como una extensión del modelo relacional clásico con capacidad para tratar con información imprecisa y deductiva. Por este motivo, hemos mostrado que es posible partir de SGBDR convencionales y construir extensiones de los mismos que operen con este tipo de información.

Futuras líneas de trabajo van encaminadas en a profundizar en los siguientes aspectos de implementación

- Desarrollar un prototipo completo de SGBDRDD que incorpore todos los módulos.
- Desarrollar una extensión completa de DFSQL, incluyendo todas las operaciones de los sublenguajes DDL, DML y DCL.
- Incorporar herramientas para el desarrollo de aplicaciones sobre Bases de Datos Deductivo-Difusas.
- Adaptar la arquitectura a los SGBDR más extendidos.
- En el caso de SGBDR con extensiones procedurales de SQL, utilizar esta característica para construir un motor de inferencia "ad hoc" que permita procesar las consultas sin necesidad de recurrir a un módulo Prolog externo.

Referencias

- [1] : J. F. Baldwin, FRIL-A Fuzzy Relational Inference Language, Fuzzy Sets and Systems, vol. 14, pp. 155-174. (1984).
- [2] : P. Bosc, M. Galibourg, G. Hamon. Fuzzy Querying with SQL: Extensions and Implementation Aspects, Fuzzy Sets and Systems. v.28 pp. 333-349. (1988)
- [3] : B.P. Buckles, F.E. Petry. A Fuzzy Representation of Data for Relational Databases, Fuzzy Sets and Systems, 7. 213-226. (1982)
- [4] : J. C. Cubero, M. A. Vila. A New Definition of Fuzzy Functional Dependencies in Fuzzy Relational Databases, International Journal of Intelligent Systems, 9(5), pp 441-448. (1994)

- [5] : J. M. Medina, M. A. Vila, J. C. Cubero, O. Pons. Towards the Implementation of a Generalized Fuzzy Relational Database Model, *Fuzzy Sets & Systems* 75 pp 273-289 (1995).
- [6] : J. M. Medina, O. Pons, M. A. Vila. GE-FRED. A Generalized Model of Fuzzy Relational Databases, *Information Sciences*, 76, 1-2, pp 87-109. (1994)
- [7] : Nakajima Hiroshi, Sogoh Taiji, Arao Masaki. Fuzzy Database Language and Library - Fuzzy Extension to SQL -, *Proceedings of Second IEEE International Conference on Fuzzy Systems*, pp. 477-482. (1993)
- [8] : O. Pons, M. A. Vila, J. M. Medina. Handling Imprecise Medical Information in the Framework of Logic Fuzzy Databases, *Fuzzy Systems & A. I. Vol. III. Nr. 1/1994*. Ed. Academiei Romane. (1994)
- [9] : O. Pons, J. M. Medina, M. A. Vila. Inference from a Fuzzy Database using Fuzzy Rules, Submitted to *Fuzzy Sets & Systems*. (1994)
- [10] : H. Prade, C. Testemale. Generalizing Database Relational Algebra for the Treatment of Incomplete/Uncertain Information and Vague Queries, *Information Sciences*, 34. 115-143. (1984)
- [11] : S. Sheno, A. Melton. An Extended Version of the Fuzzy Relational Database Model, *Information Sciences*, 52. 35-52. (1990)
- [12] : M. Umamo. Freedom-0 : A Fuzzy Database System, *Fuzzy Information and Decision Processes*. Gupta-Sanchez edit. North-Holland Pub. Comp. (1982)
- [13] : M. A. Vila, J. C. Cubero, J. M. Medina, O. Pons. Logic and Fuzzy Relational Databases: A New Language and a New Definition. In *Fuzzy Sets and Possibility Theory in Databases Management Systems*. P. Bosc and J. Kacprzyk Eds. Physica-Verlag.
- [14] : M. A. Vila, J. C. Cubero, J. M. Medina, O. Pons. Some Recent Advances in Fuzzy Relational and Fuzzy Deductive Databases, *European Research Consortium for Informatics and Mathematics*, pp.161-176, Barcelona, 1-2 November. (1994).
- [15] : M. A. Vila, J. C. Cubero, J. M. Medina, O. Pons. Towards the Computer Implementation of a Fuzzy Relational and Deductive Database System, *Proceedings of the FUZZ-IEEE/IFES'95 workshop on Fuzzy Relational Systems and Information Retrieval*, Yokohama, Japan. March (1995).
- [16] : M. Zemankova, A. Kandel. *Fuzzy Relational Data Bases - A Key to Expert Systems*, Verlag TUV Rheinland. (1984)

ID3f+A

ALGORITMO DE APRENDIZAJE INDUCTIVO BORROSO CON DIVISIÓN INTERVALAR AUTOMÁTICA DE LOS ATRIBUTOS

Gonzalo Ramos Jiménez
Dpto. Lenguajes y Ciencias de la Computación
E.T.S. de Ingeniería Informática
Universidad de Málaga
Aptdo. 4114, 29080-Málaga (SPAIN)
Tlfno: 95-2132725, FAX: 95-2131397
e-mail: ramos@lcc.uma.es

Javier López Muñoz
Dpto. Lenguajes y Ciencias de la Computación
E.T.S. de Ingeniería Informática
Universidad de Málaga
Aptdo. 4114, 29080-Málaga (SPAIN)
Tlfno: 95-2131327, FAX: 95-2131397
e-mail: jlm@lcc.uma.es

Resumen

Uno de los campos más prometedores dentro del estudio de la ambigüedad es el del aprendizaje, tanto por su importancia consustancial como por su relación con la Inteligencia Artificial. Esta relación se hace evidente cuando intentamos resolver, desde una perspectiva borrosa, el problema de la adquisición automática del conocimiento en los sistemas expertos. El algoritmo ID3, el más relevante de los utilizados para la inducción de árboles de decisión, no es utilizable tal cual con un enfoque borroso del concepto de pertenencia. Además se muestra ineficiente cuando no existe un experto humano que defina correctamente los subrangos de actuación para los atributos, que junto a las clases expresan las relaciones entre situaciones que este algoritmo de aprendizaje intenta descubrir. Proponemos como solución un nuevo algoritmo, el ID3f+A, que posee la capacidad de tratamiento borroso del concepto de pertenencia, gracias a una modificación del concepto de entropía, y además realiza la división intervalar automática de los atributos, merced al control del proceso inductivo por medio de la utilización de experiencias de control.

Palabras Clave: fuzzy, aprendizaje, inducción, ID3, ID3f+A, árbol de decisión.

1. INTRODUCCIÓN

El aprendizaje empírico a partir de experiencias ha recibido una considerable atención en términos de investigación y aplicaciones industriales. Los programas que aprenden de ejemplos preclasificados (experiencias) intentan evitar el cuello de botella que supone la adquisición de conocimiento en el desarrollo de sistemas expertos [3]. Los algoritmos inductivos de aprendizaje [7][9] intentan descubrir las relaciones (reglas) entre las situaciones, expresándolas en términos de un conjunto de atributos, y las clases. El inconveniente que surge es que la ambigüedad es inherente a la realidad [13][16], haciendo que sea difícil una representación correcta de las experiencias y de las reglas en términos no ambiguos.

El clásico algoritmo ID3 presentado por Quinlan [10][11] es un método simple y eficiente para inducir *Árboles de Decisión* a partir de conjuntos de experiencias previamente asignadas a determinadas clases. A pesar de las indiscutibles ventajas y de la facilidad de uso del ID3, este presenta dos limitaciones. La primera, que los posibles valores de todos los atributos que el algoritmo vaya a utilizar han de estar perfectamente definidos antes de que dicho algoritmo empiece a funcionar. Por ello es de suponer que existe un experto humano que basándose en su conocimiento y, a partir del conjunto de experiencias existente, va a delimitar los valores de cada uno de los atributos; es decir, va a indicar cuáles son los valores significativos de cada atributo dentro del rango de valores discretos que estos pueden adquirir. Pero no siempre es posible disponer de un experto humano que realice dicha valoración. La segunda, que el ID3 obliga a que cada experiencia pertenezca por completo a una sola clase, tanto las experiencias clasificadas de partida, como la predicción realizada

para una experiencia nueva sin clasificar; es decir, el ID3 no admite ambigüedad.

El algoritmo ID3f+A que presentamos amplía al ID3 de forma que soluciona estas dos limitaciones ("f", "+A"). Para conseguir que las experiencias iniciales utilizadas en el proceso de aprendizaje inductivo puedan pertenecer simultáneamente, con cierto grado borroso [14], a distintas clases; y además las reglas obtenidas tras la inducción sean borrosas, es decir, realicen predicciones sobre el grado de pertenencia a las distintas clases para experiencias no clasificadas, utilizamos una modificación del concepto de entropía [12]. Por otra parte, respecto a la segunda limitación, el ID3f+A además de identificar el árbol de clasificación a partir de las citadas experiencias borrosas es capaz, también, de encontrar los valores que delimitan los subrangos en los que dividir el espectro de valores de cada uno de los atributos del conjunto de experiencias. Ello es posible gracias a la división (exclusiva de nuestro algoritmo) del conjunto de experiencias en otros dos subconjuntos, un subconjunto de *clasificación* y un subconjunto de *control*. El primero lo usaremos para construir el árbol de decisión y el segundo para controlar dicho proceso de construcción.

En un trabajo anterior [1] L.I. Arranz presentó una modificación al ID3, pero sin alterar el concepto de entropía, por lo que sus grados de pertenencia debían sumar uno, lo cual era un enfoque más probabilístico que borroso. Además no realizaba la división intervalar automática de los atributos propia del ID3f+A.

En la Sección 2 se realiza la definición del problema de búsqueda de un árbol de clasificación y se definen los conceptos que necesitaremos, y en la Sección 3 se presenta nuestro algoritmo ID3f+A que da solución a las limitaciones mencionadas.

2. DEFINICIÓN DEL PROBLEMA

Un árbol de decisión [2][15] es una representación de un procedimiento de decisión para determinar la clase de una determinada instancia. Cada nodo hoja de dicho árbol de decisión tiene asociada una regla, definida por el recorrido por el árbol desde el nodo raíz hasta el nodo hoja que estamos considerando. Un árbol de este tipo es el que deseamos construir, para lo cual primero debemos delimitar el problema.

Disponemos de un conjunto de experiencias, E , cada una de las cuales viene definida por unos valores para unos atributos, $A, B, C, \dots$, y un grado de pertenencia [4][17] a cada una de las p clases posibles a priori. De esta forma, para una experiencia concreta j perteneciente a E , tendremos:

$$j: A_m, B_n, C_r, \dots / X_1, X_2, \dots, X_p$$

donde $A_m, B_n, C_r, \dots$ son los valores de los atributos $A, B, C, \dots$ en la experiencia j , y $X_i \in [0,1]$ es el grado de pertenencia de la experiencia j a la clase i .

Los valores de los atributos han de ser discretos y finitos, y el conjunto de las clases ha de ser finito. El problema consiste en formular un conjunto de reglas que tengan como antecedente valores o intervalos de valores de los atributos y como consecuente grados de pertenencia a cada una de las clases, de forma que este conjunto de reglas sea un "buen" predictor de a qué clases pertenecen futuras experiencias no clasificadas.

Para nuestro algoritmo ID3f+A dividiremos el conjunto de experiencias E en dos de tamaño similar, experiencias de clasificación, ECL , y experiencias de control, ECO .

También necesitamos dos índices de control, I_A e I_R , absoluto y relativo respectivamente, que nos indican, para el árbol de clasificación que en cada momento estamos construyendo, el grado de pronóstico acertado que dicho árbol tiene asociado para el conjunto ECO . Para obtener los índices, calculamos, para el absoluto, cuantas experiencias de ECO se clasificarían bien con dicho árbol si consideráramos que cada experiencia es de la clase de mayor grado de pertenencia y tomáramos como pronóstico la clase con mayor grado de pertenencia de la regla asociada a cada experiencia; para el relativo, también consideramos que cada experiencia es de la clase de mayor grado de pertenencia pero en lugar de contabilizar experiencias bien clasificadas sumamos los grados de pertenencia asociados a la clase de cada experiencia que indican sus reglas correspondientes; y en ambos casos dividimos por el cardinal de ECO . Así:

Sea $CEB = |\{\text{experiencias de } ECO \text{ bien clasificadas por mayor grado de pertenencia}\}|$

Sea $SG = \sum (\text{grados de pertenencia de } ECO \text{ asociados a reglas})$

$$I_A = \frac{CEB}{|ECO|}$$

$$I_R = \frac{SG}{|ECO|}$$

Por supuesto, al ser el ID3f+A un algoritmo que engloba al ID3, utiliza la idea de la entropía de la información para la creación del árbol de decisión, pero usando un concepto modificado de entropía. Hartley (1928) [6] fue el antecedente del concepto de

entropía de Shannon (1948) [12], y desde entonces este concepto de entropía ha sido modificado en mayor o menor grado con el fin de obtener distintas medidas de la incertidumbre (funciones U, E, C, V, etc...) [8]. Nuestra modificación, aún manteniendo la forma general de una medida de incertidumbre, es distinta de otras anteriores y permite la utilización de grados de pertenencia.

Para su cálculo usaremos la función Σ_g que aplicada a un conjunto de experiencias nos devuelve la suma de todos los grados de pertenencia de las experiencias del conjunto. Esta función la utilizaremos como factor de normalización. Así, nuestra entropía H , se define como:

$$H = -\sum_i P_i \log_2 P_i$$

donde

$S1 = \Sigma_g(\{\text{conjunto de experiencias de clase } i \text{ asociadas al nodo}\})$

$S2 = \Sigma_g(\{\text{conjunto de experiencias asociadas al nodo}\})$

$$P_i = \frac{S1}{S2}$$

Si se clasifica en ese nodo según un atributo A , se producirán h subconjuntos de experiencias asociadas a los h nodos hijo, en cada uno de los cuales podemos calcular la entropía parcial H_i ($i=1, \dots, h$) y a partir de aquí calcular la nueva entropía en el nodo padre gracias a la media ponderada:

$$H(A) = \sum_{i=1}^h \text{prob}(A_i) \cdot H_i$$

por tanto, el incremento de información aportado por la clasificación según A se evalúa por la disminución producida de nuestra entropía respecto de la inicial del nodo, es decir:

$$\Delta(A) = H - H(A)$$

Por cada nodo hoja del árbol obtendremos una regla que tendrá como antecedente el recorrido por el árbol desde el nodo raíz hasta dicho nodo y como consecuente grados de pertenencia a cada clase que serán la media de los grados respectivos de las experiencias asociadas al nodo.

3. EL ALGORITMO ID3f+A

Con la modificación realizada al concepto de entropía hemos incorporado la ambigüedad a nuestro algoritmo,

solucionando una de las limitaciones mencionadas. Para resolver la segunda no se puede aplicar directamente la generación del árbol de decisión dividiendo los valores de los atributos, ya que esto provoca una gran cantidad de reglas con muy pocas experiencias (a veces solo una) que la respalden, es decir, reglas muy poco significativas. Por ello surge la necesidad de un método de control del proceso de división intervalar que impida la generación de las reglas poco significativas antes mencionadas. Se propone como método de control la separación en dos conjuntos distintos de experiencias, utilizando uno de ellos para la generación del árbol y el otro para supervisar dicha generación. Al algoritmo surgido a partir de la utilización conjunta de estas dos metodologías, la modificación mostrada del concepto de entropía y el control del proceso de división intervalar de los atributos, lo hemos denominado ID3f+A.

Entrada: Un conjunto E de experiencias borrosas dividido en dos de tamaño similar, ECL y ECO.

Salida: Un conjunto de reglas para la predicción borrosa de futuras experiencias no clasificadas.

ID3f+A:

- 1/ El nodo raíz tiene asociado los conjuntos ECL y ECO completos.
- 2/ Para cada nodo del árbol no marcado como hoja hacer:
 - 2-1/ ELEGIR_ATRIBUTO
 - 2-2/ EXPANDIR_NODO

ELEGIR_ATRIBUTO:

- 1/ Para cada atributo libre del nodo hacer:
 - 1-1/ ELEGIR_PARTICIÓN
- 2/ Si no hay ningún atributo-partición generado marcar nodo como hoja.
- 3/ Si hay algún atributo-partición generado elegimos el que más disminuye la entropía (ECL).

ELEGIR_PARTICIÓN:

- 1/ Calcular I_A e I_R (ECO).
- 2/ Para cada posible partición dicotómica del intervalo del atributo calcular nueva entropía (ECL).
- 3/ Si no hay partición dicotómica que disminuya la entropía no se genera atributo-partición.
- 4/ Si hay, elegir la partición dicotómica que más disminuye la entropía (ECL).
- 5/ Si hay varias particiones con la misma disminución máxima, elegir la partición intermedia.
- 6/ Calcular I_A e I_R con dicha partición (ECO).
- 7/ Si ninguno aumenta o alguno disminuye no se genera atributo-partición.
- 8/ Si ninguno disminuye y alguno aumenta hacer E_P de los intervalos de la partición dicotómica.

E_P:

- 1/ Calcular I_A e I_R (ECO).
- 2/ Para cada posible partición dicotómica del intervalo calcular nueva entropía (ECL).
- 3/ Si no hay partición dicotómica que disminuya la entropía, fin de E_P que devuelve el intervalo.
- 4/ Si hay, elegir la partición dicotómica que más disminuye la entropía (ECL).
- 5/ Si hay varias particiones con la misma disminución máxima, elegir la partición intermedia.
- 6/ Calcular I_A e I_R con dicha partición (ECO).
- 7/ Si ninguno aumenta o alguno disminuye, fin de E_P que devuelve el intervalo.
- 8/ Si ninguno disminuye y alguno aumenta hacer E_P de los intervalos de la partición dicotómica.

EXPANDIR_NODO:

- 1/ Si nodo no marcado como hoja,
GENERAR_NODOS
- 2/ Si nodo marcado como hoja,
GENERAR_REGLA

GENERAR_NODOS: Para el atributo-partición elegido generamos tantos nodos hijos como intervalos tenga la partición, asociándoles a cada uno los subconjuntos correspondientes de ECL y ECO del nodo.

GENERAR_REGLA: Generamos una regla que tiene como antecedentes los pares atributo-intervalo que llevan desde el nodo raíz hasta el nodo actual, y como consecuente la media de los grados de pertenencia a cada clase del conjunto de experiencias ECL asociadas al nodo. También generamos asociado a la regla el cardinal del conjunto ECL del nodo, que nos indica cuantas experiencias respaldan la regla generada.

En el algoritmo todas las referencias al concepto de entropía se refieren a nuestro concepto de entropía modificada explicada en la sección anterior. Además se indica entre paréntesis qué subconjunto de experiencias, de las asociadas al nodo, utilizamos en cada caso, si las ECL o las ECO. La división de E en estos dos subconjuntos, ECL y ECO, es, como indicamos, exclusiva del algoritmo ID3f+A; así como en el nivel de procedimientos el de ELEGIR_PARTICIÓN, con todas las operaciones que engloba, también es propio del ID3f+A, siendo en dicho procedimiento donde realizamos la división intervalar inteligente de los atributos.

Siguiendo el algoritmo se observan claramente las funciones diferenciadas que tienen ECL y ECO dentro del ID3f+A. Las primeras se utilizan en la creación del árbol de decisión, siempre basada esta creación en nuestro concepto de entropía de la información. El

problema que surgiría si utilizásemos el ID3, además de la fundamental imposibilidad de utilizar grados de pertenencia, es que tiende a generar reglas con pocas experiencias que las respalden, ya que busca la completa ordenación de las experiencias disponibles, sin preocuparse de hasta qué punto las reglas generadas son realmente abstracciones útiles para la predicción futura o simplemente rozan los casos puntuales sin validez predictiva. Dicho problema se acentúa, como comentamos previamente, cuando intentamos realizar una división intervalar automática de los atributos sin el control que proponemos. En este caso el ID3 tendería a realizar una división de los atributos prácticamente total, lo que no supone realmente ninguna aportación en lo que a la división intervalar automática de los atributos se refiere.

Todos estos problemas, la generación de reglas con pocas experiencias que las respalden, así como la excesiva división de los atributos, se solucionan en el ID3f+A mediante el uso de las ECO que controlan tanto la división de los atributos como la generación de reglas no significativas (esto último gracias al control de la división) permitiendo solamente dividir cuando realmente ganamos poder predictivo, e impidiendo dicha división si no se produce la mencionada ganancia. Por lo tanto para nuestro algoritmo no importa que tengamos muchos valores posibles para cada atributo, lo que nos lleva a poder utilizarlo con atributos continuos [5], siempre que estén acotados y discretizados con la precisión que deseamos.

A todo esto hay que unir la capacidad de tratamiento borroso del concepto de pertenencia por parte del ID3f+A gracias a nuestro concepto de entropía modificada.

Destacar el hecho de que, a diferencia del ID3, en el ID3f+A un mismo atributo en distintos nodos puede tener divisiones intervalares diferentes, lo que aumenta su flexibilidad y por tanto su capacidad de inducción.

A destacar también que las reglas que generamos, en GENERAR_REGLA, llevan asociadas el cardinal del conjunto ECL de experiencias que respaldan dicha regla. Esto nos proporciona un "peso" de la regla, un indicativo de su "fiabilidad", que usamos cuando intervienen en una predicción más de una regla. Un ejemplo de esto último se produce cuando el valor de alguno de los atributos de una instancia que queremos clasificar nos es desconocido. En este caso ignoramos dicho atributo suponiendo que puede tener cualquier valor, lo que generalmente conlleva que más de una regla se corresponda con nuestra experiencia a clasificar. Cada una de estas reglas nos da un pronóstico, y para obtener un único pronóstico a partir de ellas realizamos una media ponderada de dichos

pronósticos en base a los pesos ya comentados de las reglas. Otro caso en el que necesitamos los pesos asociados a las reglas es si aplicamos el algoritmo ID3f+A más de una vez a un conjunto inicial E de experiencias. Esto puede ser conveniente debido a que la división de E en ECL y ECO es aleatoria, lo que puede introducir algún sesgo. Hasta que punto se produce dicho sesgo, cuanto puede influir en variar el árbol de decisión generado, y cuantas veces convendría repetir el ID3f+A para corregir dicho sesgo, dependerá del tipo de problema tratado y del tamaño de E . En las simulaciones que estamos realizando pretendemos obtener respuestas heurísticas a estas cuestiones.

Debido al espacio disponible, no es posible presentar aquí un ejemplo concreto de aplicación paso a paso de nuestro algoritmo. Las pruebas que hemos hecho han sido con muestras de 20 a 100 experiencias, con el objeto de verificar el correcto funcionamiento del algoritmo, y dichas pruebas han resultado satisfactorias. Un resultado relevante a destacar de dichas pruebas es la importancia que adquiere, cuando hay pocas experiencias, el hecho de elegir la partición intermedia cuando hay varias particiones con la misma disminución máxima de entropía. Esto hace que los intervalos seleccionados sean mucho más coherentes que si la elección fuera azarosa.

Respecto a la aplicación del ID3f+A a conjuntos grandes de experiencias hay dos aspectos importantes a señalar. El primero es referente al costo, en tiempo de cómputo, de la división intervalar automática, ya que si bien este costo existe, hay que tener en cuenta que dicho costo se produce solamente una vez, cuando el sistema está aprendiendo, o sea, induciendo las reglas. Es claro que este costo no afecta a lo que después será el uso del sistema una vez que ha aprendido, por lo que es perfectamente asumible dicho costo si sólo se ha de producir una vez, durante el aprendizaje inductivo borroso. El segundo aspecto hace referencia a como se produce la división intervalar automática conforme vamos bajando por el árbol de decisión generado cuando la muestra de experiencias es grande. Así, para los atributos superiores del árbol, al tener muchas experiencias asociadas, se tenderá a producir una partición más fina, con intervalos menores, mientras que en los últimos atributos del árbol, estos intervalos serán mayores, agrupando si es posible las pocas experiencias que queden. Esto es razonable ya que los atributos superiores del árbol indican que son más importantes, y por lo tanto una variación mínima en sus valores puede influir mucho, mientras que los atributos menos importantes han de variar sensiblemente para que afecten.

Así pues, hemos resuelto satisfactoriamente los problemas de como incorporar la ambigüedad dentro

del proceso inductivo y de como incluir dentro del proceso de aprendizaje la división intervalar automática de los atributos, o dicho en otras palabras, hemos conseguido un algoritmo, el ID3f+A, que tiene como objetivo principal, no clasificar mejor un conjunto de experiencias no borrosas dadas, sino predecir mejor experiencias futuras (borrosas o no), que es al fin y al cabo lo que buscamos, predecir.

4. CONSIDERACIONES FINALES

Creemos que la capacidad de tratamiento borroso del concepto de pertenencia, gracias a la modificación introducida del concepto de entropía, y la división intervalar automática de los atributos, merced al control del proceso inductivo por medio de la utilización del conjunto ECO, convierten a nuestro algoritmo ID3f+A en el primer algoritmo del tipo TDIDT (*top-down induction of decision trees*) con estas características, lo que le confieren una capacidad predictiva y flexibilidad superiores.

El ID3f+A ha sido probado satisfactoriamente con pequeñas muestras de experiencias con objeto de comprobar su correcto funcionamiento. Actualmente estamos trabajando en aplicarlo a varios grupos grandes de experiencias reales para realizar una comparación de resultados entre el ID3f+A, el ID3 y otros algoritmos.

Agradecimientos

Queremos agradecer a los Dres. Rafael Morales Bueno y José Luis Pérez de la Cruz Molina sus consejos y apoyo durante la realización de este artículo.

Referencias

- [1] Arranz, L. I. (1992). Modificaciones sobre ID3 para el tratamiento de incertidumbre. *II Congreso Español sobre Tecnologías y Lógica Fuzzy*: 193-202. Boadilla del Monte, Madrid.
- [2] Buntine, W. y Nibblett, T. (1992). A Further Comparison of Splitting Rules for Decision-Tree Induction. *Machine Learning* 8: 75-85
- [3] Cuenca, J. (1987). Inteligencia Artificial: Sistemas Expertos. Alianza Editorial.
- [4] Dubois, D., Prade, H. (1980). Fuzzy Sets and Systems: Theory and Applications. Academic Press.
- [5] Fayyad, U. M. and Irani, K. B. (1992). On the Handling of Continuous-Valued Attributes in Decision Tree Generation. *Machine Learning* 8: 87-102.

- [6] Hartley, R. V. L. (1928). Transmission of information. *The Bell Systems Technical Journal* 7: 535-563.
- [7] Hunt, E. B., Marin, J. and Stone, P. T. (1996). Experiments in induction. Academic Press.
- [8] Klir, G. J. and Folger, T. A. (1988). Fuzzy Sets, Uncertainty, and Information. Prentice-Hall International Limited.
- [9] Michalski, R. (1983). Unifying principles and a methodology of Inductive learning. *Artificial Intelligence*.
- [10] Quinlan, J. R. (1979). Discovering rules from large collections of examples: a case study. Expert Systems in the Micro Electronic Age. Edinburgh University Press.
- [11] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning* 1: 81-106.
- [12] Shannon, C. E. (1948). The mathematical theory of communication. *The Bell Systems Technical Journal* 27: 379-423, 623-656.
- [13] Trillas, E. (1980). Conjuntos Borrosos. Vicens-Vives.
- [14] Trillas, E. y Gutiérrez Ríos, J. (Editores) y otros (1992). Aplicaciones de la Lógica Borrosa. Consejo Superior de Investigaciones Científicas.
- [15] Utgoff, P. E. (1989). Incremental Induction of Decision Trees. *Machine Learning* 4: 161-186.
- [16] Zadeh, L. A. (1965). Fuzzy Sets. En Yager, R. y col. (Eds.), 29-44 (1987).
- [17] Zadeh, L. A. Fuzzy Sets and Applications. (1987). Selected Papers, editado por R. R. Yager, S. Dvchinnikov, R. M. Tong, H. T. Nguyen John Wiley, Nueva York.

SEGMENTACION DIFUSA DE CLIENTES BANCARIOS USANDO REDES NEURONALES ARTIFICIALES *

Ignacio Requena
Dpto. Ciencias Computación e I.A.
Universidad de Granada
e-mail: requena@robinson.ugr.es

Enrique López
Dpto. Dirección y Economía de
la Empresa. Universidad de León

Resumen

Presentamos un trabajo sobre la segmentación de clientes de entidades bancarias, utilizando técnicas de Redes Neuronales Artificiales y de Clustering con información difusa. Este trabajo se enmarca en la realización de un proyecto de investigación, que está ahora mismo en su fase inicial.

Palabras Clave: Segmentación de clientes. Redes Neuronales. Clustering difuso.

1 Introducción.

La interrelación entre un banco y los clientes es seguramente una de las componentes más importantes, sino la más importante para el éxito de una entidad en el sector financiero-bancario. Esta relación, y el grado de satisfacción de los clientes determinan el objetivo central y fundamentan las estrategias de la banca. Las necesidades de cada cliente han de estar en la mente de los expertos bancarios al realizar su planificación estratégica.

En un mercado evolucionado como el bancario, donde hay un alto nivel de competencia y gran variedad de productos financieros, y dada la complejidad de la propia actividad y de la confrontación con otras entidades, parece claro que la mejora de sus prestaciones supone la necesidad de ubicar a los clientes en grupos o segmentos diferen-

ciados (con distintos objetivos) que incluyan clientes similares. Así, se pueden ofrecer productos y servicios adecuados a cada segmento, en la medida en que los clientes del segmento esperan ser atendidos (y están dispuestos a pagarlos).

Sin duda, un aspecto tan importante en una entidad financiera, como es la concesión de créditos, puede verse ayudado, a la hora de evitar riesgos innecesarios, por el conocimiento del comportamiento del cliente, ajustado en la medida de lo posible a parámetros estándares del segmento reconocido. Además, la segmentación provee de una importante ayuda en la toma de decisiones sobre aspectos como la dirección estratégica de la entidad, o el lanzamiento de nuevos productos financieros o bancarios.

Puesta de relieve la importancia de la segmentación, tenemos que considerar los métodos idóneos para identificar y/o ajustar los segmentos o grupos de clientes, de acuerdo con las tecnologías disponibles actualmente, y con las posibilidades de información que la entidad financiera disponga.

Al tomar decisiones en el mundo real (y en concreto en banca y finanzas), el experto maneja información imprecisa o difusa, que puede ser presentada mediante variables no necesariamente numéricas. Además, el experto utiliza un criterio propio para seleccionar la mejor alternativa, que casi nunca se puede reproducir mediante un proceso o algoritmo, pero que sí puede ser aprendido con redes neuronales ([1,2]). También se podría aprender, con redes neuronales, el consenso alcanzable por varios expertos.

La existencia de múltiples discriminadores (umbrales) en el comportamiento del cliente, y

*Este trabajo ha sido financiado por Caja Asturias, dentro del proyecto SEDI

el grado de incertidumbre o imprecisión de la información disponible, ocasionado generalmente por la dificultad para obtenerla, propician la utilización de metodologías capaces de manejar esta imprecisión y/o de extraer los aspectos fundamentales subyacentes en el comportamiento de los clientes, en vez de considerar clasificaciones rígidas y excluyentes.

Las segmentaciones existentes en la mayoría de las entidades financieras, están realizadas de forma rígida y excluyente, lo que en muchas ocasiones producen insatisfacciones en los usuarios directos de las mismas. Pensamos que la utilización de nuevas tecnologías de la Inteligencia Artificial, como el clustering difuso y las redes neuronales artificiales, deben ser la herramientas a utilizar para la obtención de una segmentación más adecuada a la realidad actual.

El trabajo que se presenta aquí, se enmarca dentro del proyecto de investigación SEDI, financiado por Caja Asturias, y en el que participan investigadores de las universidades de Granada y León, junto con expertos de Caja Asturias, de las áreas de Economía Financiera y Contabilidad y de Ciencias de la Computación e I.A.. El proyecto, que actualmente se encuentra en su fase inicial, pretende abordar el problema de la segmentación de clientes en la referida entidad, utilizando metodologías derivadas de la fusión de técnicas difusas y redes neuronales, y tomando como punto de partida la segmentación actual de la propia entidad, realizada de forma crisp y excluyente en el sentido de que cada cliente se incluye en un solo segmento.

Los datos a manejar en el proyecto son solo de tipo económico, con un código opaco para cada registro, de forma que solo la entidad conoce la identidad del cliente y por tanto, las normativas de confidencialidad quedan perfectamente salvaguardadas. Por ello, hablaremos de registros y no de clientes.

2 Segmentación con Redes Neuronales Feedforward.

La segmentación que actualmente se usa está realizada en base a una serie de variables directas,

en el sentido de que los valores para cada registro se obtienen de forma directa a partir de los datos económicos que la entidad dispone, y asignando criterios de prioridad a las variables, ya que no todos los registros tienen datos en todas las variables. Cuando no se disponen de datos para las variables directas, se usan otras variables, que llamaremos indirectas, y criterios adecuados a las mismas.

La segmentación se realiza de acuerdo con unos segmentos determinados previamente, en número y tipo, de acuerdo con los valores de las variables indicadas y la prioridad asignada. La segmentación obtenida es crisp y excluyente (cada registro pertenece con grado uno a uno solo de los segmentos).

La primera tarea que abordamos, sigue dos líneas diferenciadas pero con el mismo origen.

- a) Partiendo de los criterios actuales para la segmentación de clientes, y las clases actuales consideradas, pretendemos usar la información disponible para proporcionar una segmentación difusa, de forma que cada ítem (registro) pueda estar en más de una clase final, con distintos grados de pertenencia.
- b) A partir de la segmentación realizada mediante las variables directas, se trata de obtener una herramienta de clasificación para los clientes de los que solo se dispone de datos en las variables indirectas.

Usamos tecnologías de Redes Neuronales Artificiales (RNA) feedforward, tipo Backpropagation. En primer lugar hemos obtenido una estructura de la RNA adecuada al problema, y un conjunto de ejemplos de entrenamiento válido, para obtener una RNA capaz de reproducir el actual sistema de segmentación, considerando todas las variables involucradas.

En esta fase usamos las variables con valores crisp, debidamente normalizados, a partir de una muestra representativa de los datos económicos de los clientes. Parte de la muestra la utilizamos como test de comprobación de resultados. La capa de entrada de la RNA tiene un número fijo de neuronas, igual al número de variables a considerar, y

la capa de salida tiene tantas neuronas como segmentos predeterminados. De acuerdo con los experimentos ya realizados, la estructura de las redes que vamos a usar son las siguientes:

- Si consideramos solo los datos de las variables directas, la red tiene 7 entradas y 8 salidas, y usaremos para la capa oculta 10, 12 o 16 neuronas, ya que con estos casos hemos obtenido unos resultados superiores al 95 por ciento sobre el conjunto de tests.
- Si consideramos datos de las variables directa y de las indirectas, las estructuras son las mismas, excepto en la capa de entrada, donde hay ahora 12 neuronas.

Con la estructura de la red obtenida, y con los ejemplos de entrenamiento de clientes con datos en las variables directas, entrenamos la RNA. Una vez entrenada, la usamos para clasificar a los clientes con datos solo en las variables indirectas. Esto resolvería el punto b) citado más arriba.

Para el punto a), partimos de la misma estructura de la RNA, en cuanto a las capas ocultas, y los ejemplos de entrenamiento, y realizamos los cambios adecuados para procesar información difusa, en la forma utilizada en [1, 2]. Los valores de entrada en el conjunto de entrenamiento tendrán ahora valores que son números difusos triangulares, y las salidas esperadas expresarán grados de pertenencia a cada segmento, de acuerdo con lo indicado por los expertos de la entidad.

Una vez entrenada la RNA, se utiliza normalmente, para obtener una segmentación con grados de pertenencia a cada segmento, llevando a cero los que no sean significativos. El análisis de la información obtenida por parte de los expertos bancarios será fundamental para validar los resultados obtenidos en esta primera fase, y para continuar y/o reconducir los criterios para las siguientes tareas.

Hemos obtenido ya una clasificación difusa de la muestra de clientes, que está siendo analizada por los expertos de la entidad bancaria, y cuyas opiniones esperamos poder presentar en el congreso.

3 Segmentación sin Clusters previos.

La segunda tarea que se pretende abordar en el proyecto, consiste en la realización de la segmentación, sin número ni tipo de clases preestablecidos. También consideramos dos líneas a seguir:

1. Utilizando la misma información que se utiliza actualmente, pretendemos obtener una clasificación alternativa (sin número ni tipo de clases preestablecido) de los items, que pueda ser comparada con la actual y pueda servir de información adicional a los expertos de la entidad bancaria. Para ello utilizaremos técnicas de clasificación jerárquicas y difusas, junto con RNA con aprendizaje no supervisado e información difusa.

Adicionalmente, y usando otras técnicas difusas, basadas en la integral de Sugeno, se podrá obtener una valoración de la importancia relativa de cada cliente para la entidad bancaria.

2. Realizar un estudio similar al apartado anterior, y con las mismas técnicas, pero utilizando un conjunto de variables alternativo al actual, basado también en información disponible por la entidad bancaria, y seleccionado por los expertos economistas y bancarios del proyecto.

4 Variables relevantes en la Segmentación.

Como consecuencia de las tareas indicadas en la sección anterior, y tras el correspondiente análisis de los resultados por los expertos bancarios, vamos a obtener un conjunto de variables (información) posiblemente interesantes para realizar una clasificación automatizada. Podremos entonces, y mediante las técnicas señaladas en [3,4], abordar el problema de detectar las variables más relevantes para la realización de la segmentación, lo que evitaría utilizar información no necesaria o redundante. También sería factible entonces obtener un conjunto de reglas del tipo

SI A es a, B es b ... ENTONCES el item esta en la clase I

que permitan comprender mejor a usuarios no especializados en las nuevas metodologías, como se realiza la segmentación.

Para ello usaremos técnicas de Redes Neuronales y Lógica Difusa, como las utilizadas en [3,4].

5 Conclusiones y Comentarios Finales.

Hemos presentado las líneas generales de un proyecto de investigación, que está en su fase inicial, y cuyos objetivos básicos podemos resumir en obtener:

1. Un desarrollo informático para obtener una segmentación difusa de la clientela de una entidad bancaria en ambiente de incertidumbre.
2. Un desarrollo informático para obtener segmentaciones alternativas, con los mismos o diferentes criterios que actualmente se utilizan.
3. Detectar las variables relevantes en la segmentación, tomando como posibles inicialmente, las usadas actualmente y las detectadas en las tareas previas.

Hasta ahora hemos obtenido la estructura inicial de la RNA y un conjunto de entrenamiento que consideramos adecuado. En el congreso podremos presentar ya resultados concretos, validados con la opinión de los expertos bancarios.

Las técnicas que utilizamos se pueden aplicar también a otros problemas interesantes tales como el análisis de inversiones, la clasificación de cuentas corrientes o la localización óptima de oficinas. Algunos de estos problemas se pueden abordar también con otras técnicas, como son la obtención de las reglas difusas que describen o gobiernan el sistema, utilizando también Redes Neuronales.

Estas técnicas están contrastadas científicamente e internacionalmente, y utilizadas para resolver variados tipos de problemas, pero el proyecto no

está exento de dificultades. En concreto, las dificultades para el proyecto se pueden derivar fundamentalmente del uso de grandes cantidades de información, que no son tan habituales en los estudios científicos, y de la dificultad inherente que tienen la toma de decisiones en la gestión de empresas financieras..

6 REFERENCIAS

- 1.- I. Requena, M. Delgado, J.L. Verdegay. (1993). Redes Neuronales Artificiales para algunos problemas de decisión con ambiente difuso. Informática y Automática. vol 26. num 4. pg 24-36
- 2.- I. Requena, M. Delgado, J.L. Verdegay. (1994). Automatic ranking of fuzzy numbers with the criterion of a decision-maker learnt by an artificial neural network. Fuzzy Set and Systems 64. pg 1-19.
- 3.- J.M. Benítez, A. Blanco, M. Delgado, I. Requena. (1995). Redes neuronales para obtener reglas difusas. Actas ESTYLF'95. Murcia.
- 4.- J.M. Benítez, A. Blanco, M. Delgado, I. Requena. (1996). Obtaining fuzzy rules by means of neural network. Aceptado en MATHWARE.
- 5.- I. Requena. Redes Neuronales para algunos problemas de decisión en ambiente difuso. Tesis Doctoral. Universidad de Granada. 1992

GRAMÁTICAS FORMALES PARA EL PROCESAMIENTO DE LENGUAJE NATURAL VAGO

Alejandro Sobrino

Área de Lógica y Filosofía de la Ciencia. Campus Universitario, s.n. 15706 Santiago de Compostela.

e-mail: lflgalex@usc.es

Resumen

Apoyándose en el concepto clásico (preciso) de gramática formal, en este trabajo se pretende mostrar, a) como desarrollar una gramática parcial del castellano que genere oraciones de significado vago, b) como obtener una gramática lógica que produzca (algunas de las) fórmulas bien formadas de la lógica fuzzy y c) como obtener una semántica lógica que, asociada a la gramática lógica, y expresada con una sintaxis similar a la del Prolog, permita generar valoraciones para las fórmulas bien formadas. Con ello se pretende mostrar la adecuación de las gramáticas formales para expresar la sintaxis y semántica lógica de las expresiones de significado vago de una manera asociada o conjunta.

Palabras clave: Procesamiento del lenguaje natural, gramática formal, semioración.

1. INTRODUCCIÓN

Las gramáticas formales han sido inicialmente extendidas para las oraciones de significado vago por Lee y Zadeh [3], atribuyendo grados a las diferentes reglas de reescritura. Recientemente, se ha desarrollado el concepto de semicadena y se han caracterizado las gramáticas formales que permiten su generación, útiles sobre todo en el reconocimiento vago de patrones. Estas gramáticas se distinguen por tener reglas 'infradefinidas', en el sentido de que la cadena por la que se reescribe un elemento no

terminal puede tener sin caracterizar alguna de sus posiciones [4].

En este estudio intentaremos prologar la tradición de aplicación de la metodología de las gramáticas formales al ámbito del lenguaje vago desde una perspectiva diferente. Utilizaremos la noción clásica de gramática formal [1] para generar a) oraciones vagas gramaticalmente correctas, b) fórmulas bien formadas (fbf) de la lógica fuzzy (LF) y c) una semántica de valores graduados asociados a estas fórmulas. Empezamos generando oraciones de significado vago con un fragmento de la gramática del castellano. A continuación, con una gramática formal producimos (algunas de) las expresiones bien formadas de la LF. Por fin, la asociación a esta gramática de una semántica formal, expresada con una sintaxis similar a la del Prolog [2], permitirá hallar el valor de verdad de una de las fbf.

La asociación natural, a través del concepto de gramática formal, de una oración de significado vago con la fbf que constituye su reescritura formal y con la semántica asociada a esa fórmula, le da un protagonismo interesante a esta metodología de análisis en el intento de acercar la sintaxis a la semántica y de ambas al lenguaje natural. Constituye, por tanto, un medio de análisis adecuado para el procesamiento del lenguaje natural vago.

2. FRAGMENTO DE LA GRAMÁTICA DEL CASTELLANO PARA ALGUNAS ORACIONES DE SIGNIFICADO VAGO.

Una buena parte de las oraciones del lenguaje castellano (como las de cualquier lenguaje natural), son de significado aproximado o vago. Constituyen la expresión del uso cauteloso, prudente y adaptativo que hacemos los humanos de nuestro lenguaje, como una forma de aprehender un medio en continuo cambio y evolución. La sintaxis del castellano provee reglas que permiten generar este tipo de oraciones. A continuación se ofrece una gramática parcial del castellano que genera oraciones típicamente vagas, sean atómicas, como *pocos jugadores son muy bajos*, *pocos jugadores son bastante fuertes* o moleculares, como la conjunción de las dos anteriores.

1. <oración> → <frase nominal> <frase verbal>
2. → <oración> y <oración>
3. <frase nominal> → <adv. de cantidad> <nombre>
4. <frase verbal> → <verbo> <atributo>
5. <atributo> → <modificador> <adjetivo>
6. <adverbio de cantidad> → pocos
7. <nombre> → jugadores
8. <verbo> → son
9. <modificador> → muy
10. → bastante
11. <adjetivo> → bajos
12. → fuertes

Como es usual, la generación de una frase consiste en sucesivas aplicaciones de las reglas de tal manera que transformen la variable <oración> en una cadena de símbolos terminales. Por ejemplo, *pocos jugadores son muy bajos* se genera por las siguientes transformaciones:

DERIVACIÓN

REGLA APLICADA

<oración>

- | | |
|----------------------------------|---|
| ⇒ <frase nominal> <frase verbal> | 1 |
| ⇒ <frase nominal> <frase verbal> | 2 |

- | | |
|--|----|
| ⇒ <adv. de cantidad> <nombre común> <frase verbal> | 3 |
| ⇒ pocos <nombre común> <frase verbal> | 6 |
| ⇒ pocos jugadores <frase verbal> | 7 |
| ⇒ pocos jugadores <verbo> <atributo> | 4 |
| ⇒ pocos jugadores son <atributo> | 8 |
| ⇒ pocos jugadores son <modificador> <adjetivo> | 5 |
| ⇒ pocos jugadores son muy <adjetivo> | 9 |
| ⇒ pocos jugadores son muy bajos | 11 |

donde el símbolo ⇒ indica la aplicación de una regla de reescritura →. Este mismo proceder puede seguirse con las oraciones *pocos jugadores son bastante fuertes* o con *pocos jugadores son muy bajos* y *pocos jugadores son bastante fuertes*, cuya generación omitimos por ser similar a la anterior.

La producción de estas frases no presenta, desde el punto de vista de la gramática del español, una complejidad mayor que la generación de otras oraciones; en particular, de aquellas de significado preciso. Ello se debe a que el deseo de representar la vaguedad responde en este caso a una extensión del *léxico* considerado, no a tener en cuenta aspectos de tipo gramatical (implícitos, p. ej., en el concepto de semioración, del cual no nos ocuparemos en este trabajo). En efecto, el tener presente a la vaguedad hace que notemos en la oración *pocos jugadores son muy bajos* la presencia distintiva, respecto a una oración precisa como *Juan mide 1.67 m.*, del siguiente léxico [6]: a) adverbios de cantidad, como *pocos*; b) modificadores, como *muy* y c) adjetivos (predicados vagos) como *bajos*.

Oraciones como las del ejemplo de antes pueden ser formalizadas por una lógica de predicados vagos. En la siguiente sección se muestra como representar oraciones de significado vago en el lenguaje formal de la LF y como, mediante una gramática formal, es posible generar algunas de estas fbf.

3. UNA GRAMÁTICA FORMAL PARA GENERAR fbf DE LA LÓGICA DE LOS PREDICADOS VAGOS (sólo un fragmento).

En el vocabulario de la gramática formal que exponemos a continuación sólo se incluyen tres operadores: $\wedge$, $\vee$ y $\neg$, respectivamente. Las letras predicativas son simbolizadas por P, Q, P', Q',... y la variable de individuos por x, (sobre un dominio D). Finalmente, se incluyen tres tipos de cuantificadores, los clásicos $\forall$ y $\exists$ (que no se usarán en el ejemplo) y uno que representa al cuantificador aproximado 'pocos' y que, convencionalmente denotaremos por $\%$. También se introducen dos símbolos nuevos, para los modificadores 'muy' y 'bastante', que representaremos respectivamente y de modo convencional por Δ y ∇ . La gramática queda definida por la siguiente tupla, $G = \langle V_{NT}, V_T, S, P \rangle$, donde

V_T (Vocabulario terminal) = $\{ \wedge, \vee, \neg, \forall, \exists, \%, P, Q, R, \Delta, \nabla, ', x, (,) \}$.

V_{NT} (Vocabulario no terminal) = $\{ S, A \}$

P:

1. $S \rightarrow (S) \wedge (S)$
2. $\rightarrow (S) \vee (S)$
3. $\rightarrow \neg(S)$
4. $\rightarrow \forall x (Sx)$
5. $\rightarrow \exists x (Sx)$
6. $\rightarrow \%x (Sx)$
7. $\rightarrow A$
8. $A \rightarrow \Delta A$
9. $\rightarrow \nabla A$
10. $\rightarrow A'$
11. $\rightarrow P$
12. $\rightarrow Q$

Esta gramática permite generar fórmulas como la siguiente, $\%x (\Delta Px)$ ($x \in D$), que traduce la expresión,

pocos jugadores son muy bajos, admitido que $\%$ está por *pocos*, Δ por *muy*, P por *bajos* y x por los individuos del dominio. El proceso de generación de esta fórmula sería el siguiente:

$S \Rightarrow \%x (Sx)$	6
$\Rightarrow \%x (Ax)$	7
$\Rightarrow \%x (\Delta Ax)$	8
$\Rightarrow \%x (\Delta Px)$	11

De modo similar, se puede producir la fórmula correspondiente a *pocos jugadores son muy bajos y pocos jugadores son bastante gruesos*, - $\%x (\Delta Px) \wedge \%x (\nabla Px)$ -, cuya generación, sería así:

$S \Rightarrow (S) \wedge (S)$	1
$\Rightarrow \%x (Sx) \wedge (S)$	6
$\Rightarrow \%x (Sx) \wedge \%x (Sx)$	6
$\Rightarrow \%x (A) \wedge \%x (Sx)$	7
$\Rightarrow \%x (\Delta A) \wedge \%x (Sx)$	8
$\Rightarrow \%x (\Delta Px) \wedge \%x (Sx)$	11
$\Rightarrow \%x (\Delta Px) \wedge \%x (Ax)$	7
$\Rightarrow \%x (\Delta Px) \wedge \%x (\nabla Ax)$	9
$\Rightarrow \%x (\Delta Px) \wedge \%x (\nabla Qx)$	12

Como ya se indicó al principio de la sección, esta gramática sólo genera algunas fbf de una lógica de los predicados vagos, no todas las posibles. Para ello habría que incluir un listado exhaustivo de los cuantificadores aproximados y modificadores, cuestión lingüísticamente problemática. Pero aún haciendo esto surgirían algunos problemas, de los que en este trabajo simplemente daremos breve noticia. Estos problemas se apoyan en la convicción de que las fbf de un lenguaje lógico deben tener que ver con la corrección gramatical de las oraciones de un lenguaje ordinario, en el supuesto de que si una lógica aspira a facilitar la resolución de problemas relacionados

con la vida cotidiana, debe ser receptiva a la forma en que los humanos verbalizamos esos problemas. Y en esto, la corrección es importante. Bajo estos presupuestos presentamos las siguientes dificultades:

(i) No para todos los modificadores es gramatical y aceptable el uso repetido de dos o más de ellos. *Muy*, p. ej., admite esta iteración, como ocurre en *pocos jugadores son muy muy bajos*. En cambio, la repetición de *suficientemente* origina una oración agramatical, como pasa en *pocos jugadores son suficientemente suficientemente bajos*. Es decir, algunas reglas de producción, como es el caso de la número 8 de la gramática antes esbozada, podría presentar este aspecto: $A \rightarrow \Delta\Delta A$. Pero otras reglas, que incluyan otros modificadores, no.

(ii) Vinculado con el problema anterior: hay algunos modificadores tal que su inversión en la oración no provocan la pérdida completa de la gramaticalidad, como p. ej., *completamente* en *es verdad completamente* frente a la frase gramatical *es completamente verdad*. En otros casos se pierde esa gramaticalidad, como ocurre con *muy* o con *una especie de* (*es muy alto* frente a *es alto muy*). Para los casos en los que no lleva a una pérdida de gramaticalidad, una gramática generadora de fbf de la LF debería admitir, además de reglas de reescritura del tipo $A \rightarrow mA$, reglas que conmutan los elementos de la parte derecha de la producción, como $A \rightarrow Am$, si *m* es un modificador que, como *completamente*, admite esta inversión [5].

Retomemos ahora los ejemplos de las fórmulas generadas. Estas fórmulas son susceptibles de valoración. En el siguiente apartado se propone una semántica formal que, asociada a la gramática, permite generar estos valores.

4. SEMÁNTICA LÓGICA ASOCIADA A (UN FRAGMENTO DE) LA GRAMÁTICA QUE GENERA fbf DE LA LF.

A la gramática formal que genera fbf de la lógica fuzzy se le puede asociar una semántica lógica, de tal forma que las reglas gramaticales queden 'interpretadas' por reglas semánticas. A continuación, sólo se describen las reglas necesarias para generar la frase *pocos jugadores son muy bajos*. Estas se expresan con una sintaxis parecida a la del Prolog y presentarían la siguiente formulación general -la numeración que aparece destacada en el margen derecho señala con qué reglas de la gramática formal se relacionan las distintas especificaciones semánticas-:

(a) Especificación de *es predicado vago*

es_predicado_vago(individuo, grado). 11

donde esta cláusula representa en el grado en el que es compatible el individuo del dominio con lo que significa el predicado vago.

(b) Especificación de *es modificador predicado vago*

es_modificador_predicado_vago(Individuo, Verdad):- 8

es_predicado_vago(individuo, Grado),

Verdad is #Grado.

donde esta cláusula sirve para representar en el grado en el que es compatible cada individuo del dominio con el significado del predicado vago afectado por un modificador o cerca (la operación que induce el modificador la representamos genéricamente por *#Grado*).

(c) Especificación de *cuantificador aproximado individuos modificador predicado vago* (esto es, del conjunto de los elementos lógicos que son relevantes en la generación del valor de la frase). Distinguimos,

(c1) especificación del significado de *es cuantificador aproximado*

es_cuantificador_aproximado(unidad, grado). 6

donde esta cláusula representa el grado en el que cada una de las unidades del dominio es compatible con el significado vago de *poco*.

(c2) Especificación de *cuantificador aproximado individuos modificador predicado vago*

cuantificador_aproximado_es_modificador_predicado_vago (X, Y):-
6

proporción_predicado_vago(X, Z), es_poco(Z, Y).

Esta especificación requiere definir *proporción(X, Z)*,

proporción_predicado_vago(X, Z):-

total_es_modificador_predicado_vago(X, Z),

*Y is (Z*10) div |D|.*

Por último, hay que definir *total_es_predicado_vago(X, Z)*

total_es_modificador_predicado_vago(X, Y):-

es_modificador_predicado_vago(individuo, A),

es_modificador_predicado_vago(individuo, B),

...

es_modificador_predicado_vago(individuo, N),

Y is A+B+...+N

En la secuencia (c2) se especifica, de abajo arriba, en qué grado tienen todos los individuos el predicado vago modificado. A continuación, se halla la proporción con la que los individuos satisfacen el predicado vago modificado, multiplicando el grado resultado de la operación anterior por 10 (se supone que *pocos* esta definido para un dominio de enteros) y se divide todo ello (con la operación *div* para que dé un cociente sin decimales) por la cardinalidad del dominio, (|D|). Por último, queda hallar en qué grado ese valor proporcional es compatible con el significado de *pocos*.

Para oraciones complejas, como *pocos jugadores son muy bajos* y *pocos jugadores son bastante fuertes* tendríamos que ampliar las reglas semánticas a fin de incorporar definiciones de conectivas, pero por razones de espacio, este aspecto no será tratado en este trabajo.

Veamos un ejemplo, enmarcado en el contexto de jugadores de un equipo español de baloncesto. Contamos con los siguientes datos:

Dominio = 6 jugadores

Talla jugador 1:1.65 m.

Talla jugador 2:1.60 m.

Talla jugador 3:2.07 m.

Talla jugador 4:1.95 m.

Talla jugador 5:1.90 m.

Talla jugador 6:2.05 m.

Para ellos proponemos un conjunto de valoraciones, que esperamos parezcan plausibles, así como el programa que calcula el valor de verdad de la oración *pocos jugadores son muy bajos* (en el margen derecho se especifican las reglas de la gramática formal con las que se corresponden las respectivas cláusulas Prolog).

```
/*-----
Un programa para calcular el valor de
verdad de la expresión 'Pocos jugadores
son muy bajos'
*-----*/
```

```
/*grado en qué cada jugador es bajo 11*/
```

```
es_bajo(jug1, 0.85).
```

```
es_bajo(jug2, 1).
```

```
es_bajo(jug3, 0).
```

```
es_bajo(jug4, 0.30).
```

```
es_bajo(jug5, 0.40).
```

```
es_bajo(jug6, 0.15).
```

```
es_poco(1, 0.8). %significado de poco 6
```

```
es_poco(2, 0.9).
```

```
es_poco(3, 0.3).
```

```
es_poco(4, 0.1).
```

```
es_poco(5, 0).
```

```
es_poco(6, 0).
```

```
/*significado de muy bajo 8*/
```

```
es_muy_bajo(X, Y):-
```

```
es_bajo(X, Z),
```

```
Y is Z*Z.
```

```
/*procedimientos para calcular el valor
de la frase 6
```

```
a) cálculo del grado en el que todos
losjugadores son muy 'bajos'*/
```

```
total_es_muy_bajo(X, Y):-
```

```
es_muy_bajo(jug1, A),
```

```
es_muy_bajo(jug1, A),
```

```
es_muy_bajo(jug2, B),
```

```
es_muy_bajo(jug3, C),
```

```
es_muy_bajo(jug4, D),
```

```
es_muy_bajo(jug5, E),
```

```
es_muy_bajo(jug6, F),
```

```
Y is A+B+C+D+E+F.
```

```
/*proporción en que cada jugador es
muy bajo*/
```

```
proporcion_bajo(X, Y):-
    total_es_muy_bajo(X, Z),
    Y is (Z*10) div 6.
```

```
/*compatibilidad del resultado de la
proporción con 'pocos'*/
```

```
pocos(X, Y):-
    proporcion_bajo(X, Z), es_poco(Z,Y).
```

Sesión de consulta:

```
? es_muy_bajo(X, Y).
X = jug1,
Y = 7.22500000000e-01 ;
X = jug2,
Y = 1 ;
X = jug3,
Y = 0 ;
X = jug4,
Y = 9.00000000000e-02 ;
X = jug5,
Y = 1.60000000000e-01 ;
X = jug6,
Y = 2.25000000000e-02 ;
```

No

```
? total_es_muy_bajo(X, Y).
X = _114,
Y = 1.99500000000e+00 ;
```

No

```
? proporcion_bajo(X, Y).
X = _114,
Y = 3 ;
```

No

```
? pocos(X, Y).
X = _124,
Y = 3.00000000000e-01 ;
```

No

5. A MODO DE CONCLUSIÓN

En este trabajo se ha pretendido mostrar la adecuación del concepto clásico de gramática formal para analizar, de modo unificado, tanto oraciones del lenguaje castellano de significado vago como aspectos sintácticos y semánticos

de la LF como lenguaje lógico formal, lógica que ha mostrado su utilidad para formalizar y valorar a estas oraciones.

Se han apuntado además algunos problemas que es este cometido permanecen, relacionados con las diferentes propiedades de los modificadores lingüísticos respecto a su repetición o inversión en las reglas, bajo el supuesto de que las formas lógicas deben atender a las formas gramaticales del lenguaje común.

Agradecimientos.

El autor agradece la ayuda económica prestada por el proyecto XUGA20501A95 de la Xunta de Galicia.

Referencias.

- [1] CHOMSKY, N. & MILLER, G. A. 'Formal properties of grammars'. *Handbook of Mathematical Psychology* (ed. D. Luce, E. Bush and E. Galanter). John Wiley and Sons, Inc. 1963
- [2] DAHL, V. & SOBRINO, A., *Estudios sobre programación lógica y sus aplicaciones*. Publicaciones de la Universidad de Santiago de Compostela, 1966.
- [3] LEE, E. T. & ZADEH, L. A. 'Note on Fuzzy Languages', *Information Sciences*, 1 (1969): 421-434.
- [4] SCHNEIDER, M. & LIM, H. & SHOAFF, W., 'The utilization of fuzzy sets in the recognition of imperfect strings', *Fuzzy Sets and Systems*, 49 (1992):331-337.
- [5] SOBRINO, A. y OLIVAS, J. A. 'Gramáticas de estructura de frase y expresividad de los conceptos vagos', en R. Marín et al. (eds.), *Actas del Congreso ESTYLF'95*. Servicio de Publicaciones de la Universidad de Murcia, 1995 : 129-134.
- [6] ZADEH, L. A., 'Quantitative fuzzy semantics', *Information Sciences*, 3: 159-176.

INTEGRACIÓN DIFUSA PARA MEDIDAS DEL TIPO S Y NSA. TEOREMA DE RADON-NIKODYM

E. Suárez Díaz
Universidad de Oviedo
e-mail: emilio@etsiig.uniovi.es

F. Suárez García
Universidad de Oviedo
e-mail: fermin@etsiig.uniovi.es

Resumen

El principal objetivo de este trabajo es dar una versión del teorema de Radon-Nikodym para integrales difusas. Para ello, definimos este tipo de integrales a partir de medidas difusas del tipo S y NSA, definidas por Weber en [17] y estudiamos algunas de sus principales propiedades, obteniendo versiones de teoremas clásicos como el teorema de la convergencia monótona, convergencia mayorada o el lema de Fatou, finalizando con una versión del teorema de Radon-Nikodym para integrales difusas respecto de medidas del tipo S. En [4], Kruse da una versión de dicho teorema para medidas de Sugeno, que no son más que un caso particular de medidas NSA, pero la demostración ignora, a nuestro entender, un detalle que puede ser de gran importancia para la validez del resultado, como se ve en el ejemplo 3.1

Introducción

El trabajo está organizado como sigue: El apartado 1 trata de recoger los conceptos previos y necesarios para seguir el resto del trabajo. En el apartado 2 es donde se introducen las definiciones de integrales difusas para medidas del tipo S y NSA, dando como resultado una generalización de la integral definida por Kruse en [4]. En el teorema 2.3 de este apartado condensamos las principales propiedades de estas integrales. Por último, en el apartado 3 damos la versión del teorema de Radon-Nikodym para medidas del tipo S y ponemos de manifiesto el error que contiene la demostración hecha por Kruse en [4] para integrales difusas respecto de medidas de Sugeno, caso particular de medidas NSA.

1 Conceptos preliminares

En adelante, llamaremos $I = [0, 1]$

DEFINICIÓN 1.1 Una operación binaria, continua $\perp: I \times I \rightarrow I$ diremos que es una *t-conorma Arquimediana* si, y sólo si:

- 1) Es no decreciente respecto de cada variable.
- 2) $\perp(x, 0) = x$, para todo $x \in I$.
- 3) $\perp$ es asociativa.
- 4) $\perp(x, x) > x$, para todo $x \in (0, 1)$.

$\perp$ diremos que es *estricta* si es estrictamente creciente. También denotaremos $x \perp y$, en vez de $\perp(x, y)$. Indicamos, a continuación, algunas propiedades de las t-conormas arquimediana que necesitaremos más adelante.

TEOREMA 1.1 (C.H. Ling) [5] Una operación binaria $\perp: I \times I \rightarrow I$ es una *t-conorma arquimediana* si, y sólo si, existe una función $g: I \rightarrow [0, \infty]$, creciente y continua, con $g(0) = 0$ tal que:

$$x \perp y = g^{(-1)}[g(x) + g(y)] ; \text{ para todo } x, y \in I$$

donde $g^{(-1)}$ denota la función pseudoinversa de g , definida por:

$$g^{(-1)}(y) = \begin{cases} g^{-1}(y) & \text{if } y \in [0, g(1)] \\ 1 & \text{if } y \geq g(1) \end{cases}$$

O, equivalentemente: $g^{(-1)}(y) = g^{-1}(\min\{y, g(1)\})$. Además $\perp$ es estricta si, y sólo si $g(1) = \infty$.

La función g se dice que es un generador aditivo de $\perp$, y es única excepto para multiplicaciones por números positivos. En el caso no estricto, se suele considerar el generador aditivo normalizado $g(1) = 1$. Para el objeto de este trabajo, necesitamos definir

un producto en I . Como muestra Weber en [18], y para la construcción de lo que se verá en la sección 2, definiremos tal producto como:

$$x \otimes_g y = g^{-1}(g(x) \cdot g(y))$$

Observemos como este producto depende del generador aditivo g de la t -conorma. No obstante es inmediato comprobar que tal producto es único salvo producto por constantes positivas. Se sigue, mediante una comprobación rutinaria, que $\otimes_g$ es una t -norma [8]

• Sea

$$D = \{(x, y) \in I \times I : g(x) + g(y) \leq g(1)\} \quad (1)$$

Es fácil ver que se para todo $x \in I$ e $(y, z) \in D$ cumple:

$$x \otimes_g (y \perp z) = (x \otimes_g y) \perp (x \otimes_g z) \quad (2)$$

Esta "distributividad" of $\otimes_g$ con respecto a $\perp$ será esencial en el presente trabajo. La propiedad anterior, (2) es cierta para todo $x, y, z \in I$ si, y sólo si $\perp$ es estricta. Para más detalles, ver Weber [18]

1.1 Medidas descomponibles

Sea un espacio medible $(\Omega, \mathcal{A})$ y $\perp$ una t -conorma Arquimediana.

DEFINICIÓN 1.2 Una función $m : \mathcal{A} \rightarrow [0, 1]$ con $m(\emptyset) = 0$; $m(\Omega) = 1$ se dice que es una medida difusa descomponible, con respecto de la t -conorma arquimediana $\perp$ si, y sólo si, para todo $A, B \in \mathcal{A}$ con $A \cap B = \emptyset$ se tiene que:

$$m(A \cup B) = m(A) \perp m(B)$$

DEFINICIÓN 1.3 Una medida $\perp$ -descomponible m se dice que es σ - $\perp$ -descomponible si, y sólo si, es continua inferiormente.

Esto equivale a afirmar que para cualquier familia numerable $\{A_k\}_{k \in \mathbb{N}}$ con $A_k \cap A_j = \emptyset$, se tiene:

$$m(\bigcup_{k=1}^{\infty} A_k) = \bigwedge_{k=1}^{\infty} m(A_k)$$

donde $\bigwedge_{k=1}^n x_k$ se define de forma recursiva como $\bigwedge_{k=1}^n x_k = \bigwedge \left[\left(\bigwedge_{k=1}^{n-1} x_k, x_n \right) \right]$ y $\bigwedge_{k=1}^{\infty} x_k = \lim_{n \rightarrow \infty} \bigwedge_{k=1}^n x_k$

PROPOSICIÓN 1.1 Si m es una medida difusa σ - $\perp$ -descomponible en $(\Omega, \mathcal{A})$ y g es un generador aditivo de $\perp$, entonces:

$$g \circ m \left(\bigcup_{k=1}^{\infty} A_k \right) = \min \left\{ \sum_{k=1}^{\infty} g \circ m(A_k), g(1) \right\}$$

Para más detalles, ver Weber [18].

De la proposición anterior, Weber distingue tres casos con la siguiente notación:

S: Cuando $\perp$ es estricta. Entonces la función $m^* = g \circ m$ es una medida σ -aditiva infinita.

NSA: Cuando $\perp$ no es estricta, y para todo $A, B \in \mathcal{A}$ tal que $A \cap B = \emptyset$ se tiene:

$$(m(A), m(B)) \in D$$

donde D se definió en (1). Entonces la función $m^* = g \circ m$ es una medida de probabilidad, una vez considerado el generador aditivo normalizado $g(1) = 1$.

NSP: Cuando $\perp$ no es estricta y existen $A, B \in \mathcal{A}$ tales que $A \cap B = \emptyset$ y $(m(A), m(B)) \notin D$.

A partir de este punto, consideraremos medidas del tipo S y NSA.

2 Integrales difusas para medidas S y NSA

Sea $(\Omega, \mathcal{A})$ un espacio medible y consideremos una función simple $\mathcal{A}$ -medible $f : \Omega \rightarrow [0, 1]$, i.e.

$$f(\omega) = \sum_{k=1}^n a_k \cdot I_{A_k}(\omega)$$

donde $A_k \in \mathcal{A}$; $k = 1, 2, \dots, n$ es una partición de Ω . Sea m una NSA o S medida difusa con respecto de la t -conorma arquimediana $\perp$. Definimos:

$$\int_{\Omega}^{(g)} f \perp m = \bigwedge_{k=1}^n a_k \otimes_g m(A_k) \quad (3)$$

Para el caso NSA, podemos considerar el generador aditivo normalizado y entonces, si llamamos $\otimes_g = \otimes$, simplificamos la notación de la integral:

$$\int_{\Omega}^{(g)} f \perp m = \int_{\Omega} f \perp m$$

Esta integral puede considerarse como una generalización de la integral de Lebesgue-Stieljes.

TEOREMA 2.1 La integral anterior (3) está bien definida.

DEMOSTRACIÓN Supongamos que $f = \sum_{i=1}^n a_i \cdot I_{A_i}$ y consideremos otra representación de f , de forma que $f = \sum_{j=1}^m b_j \cdot I_{B_j}$. Si tomamos la partición $\{A_i \cap B_j\}_{\substack{i=1,2,\dots,n \\ j=1,2,\dots,m}}$ podemos escribir $\int_{\Omega}^{(g)} f \perp m =$

$\perp_{i=1}^n \perp_{j=1}^m c_{ij} \otimes_p m(A_i \cap B_j)$ donde $c_{ij} = a_i$ para todo $j = 1, 2, \dots, m$ y $c_{ij} = b_j$ para todo $i = 1, 2, \dots, n$ y entonces:

$$\begin{aligned} \int_{\Omega}^{(g)} f \perp m &= \perp_{i=1}^n \perp_{j=1}^m c_{ij} \otimes_p m(A_i \cap B_j) = \\ &= \perp_{i=1}^n c_{ij} \otimes_p \left[\perp_{j=1}^m m(A_i \cap B_j) \right] = \\ &= \perp_{i=1}^n [c_{ij} \otimes_p m(A_i)] = \\ &= \perp_{i=1}^n a_i \otimes_p m(A_i) \end{aligned}$$

De una forma similar, se prueba:

$$\begin{aligned} \perp_{i=1}^n \perp_{j=1}^m c_{ij} \otimes_p m(A_i \cap B_j) &= \\ &= \perp_{j=1}^m b_j \otimes_p m(B_j). \end{aligned}$$

□

En el siguiente teorema, veremos que la integral anterior (3) se puede escribir en término de una integral de Lebesgue-Stieljes:

TEOREMA 2.2 La integral difusa (3) se relaciona con la integral de Lebesgue-Stieljes mediante:

$$\int_{\Omega}^{(g)} f \perp m = g^{-1} \left[\int_{\Omega} (g \circ f) d(g \circ m) \right] \quad (4)$$

para cualquier función simple f .

DEMOSTRACIÓN Sea $f = \sum_{i=1}^n a_i \cdot I_{A_i}$ entonces:

$$\begin{aligned} \int_{\Omega}^{(g)} f \perp m &= \perp_{i=1}^n a_i \otimes_p m(A_i) = \\ &= \perp_{i=1}^n g^{-1} [g(a_i) \cdot g(m(A_i))] = \\ &= g^{(-1)} \left[\sum_{i=1}^n (g \circ g^{-1}) (g(a_i) \cdot g(m(A_i))) \right] = \\ &= g^{(-1)} \left[\int_{\Omega} (g \circ f) d(g \circ m) \right] = \\ &= g^{-1} \left[\int_{\Omega} (g \circ f) d(g \circ m) \right] \end{aligned}$$

La última igualdad se debe a que $g \circ f \leq 1$ en el caso NSA y, por tanto, $g^{(-1)} = g^{-1}$, igualdad que se produce siempre en el caso S. □

Con objeto de extender la definición de fuzzy integral para cualquier función $\mathcal{A}$ -medible f consideramos el hecho de que existe alguna sucesión de funciones simples $\{f_n\}$ tal que $f_n \uparrow f$ y entonces, tendremos:

DEFINICIÓN 2.1 Sea $f : \Omega \rightarrow [0, 1]$ una función $\mathcal{A}$ -medible, definimos:

$$\int_{\Omega}^{(g)} f \perp m = \lim_{n \rightarrow \infty} \int_{\Omega}^{(g)} f_n \perp m \quad (5)$$

donde $\{f_n\}$ es cualquier sucesión de funciones simples tal que $f_n \uparrow f$.

Es fácil comprobar que la definición anterior es independiente de la sucesión $\{f_n\}$ elegida.

Veamos ahora que la relación (4) se verifica para cualquier función $\mathcal{A}$ -medible f :

PROPOSICIÓN 2.1 Se tiene que:

$$\int_{\Omega}^{(g)} f \perp m = g^{-1} \left[\int_{\Omega} (g \circ f) d(g \circ m) \right] \quad (6)$$

para cualquier función f que sea $\mathcal{A}$ -medible.

DEMOSTRACIÓN

$$\begin{aligned} \int_{\Omega}^{(g)} f \perp m &= \lim_{n \rightarrow \infty} \int_{\Omega}^{(g)} f_n \perp m = \\ &= \lim_{n \rightarrow \infty} g^{-1} \left[\int_{\Omega} (g \circ f_n) d(g \circ m) \right] = \\ &= g^{-1} \left[\lim_{n \rightarrow \infty} \int_{\Omega} (g \circ f_n) d(g \circ m) \right] = \\ &= g^{-1} \left[\int_{\Omega} (g \circ f) d(g \circ m) \right] \end{aligned}$$

lo cual, se deduce por la continuidad de g , y el teorema de la convergencia monótona. □

Para cualquier conjunto $A \in \mathcal{A}$ definimos $\int_A^{(g)}$ de la forma usual:

$$\begin{aligned} \int_A^{(g)} f \perp m &= \int_{\Omega}^{(g)} f \cdot I_A \perp m = \\ &= g^{-1} \left[\int_{\Omega} g(f \cdot I_A) d(g \circ m) \right] = \\ &= g^{-1} \left[\int_{\Omega} I_A \cdot (g \circ f) d(g \circ m) \right] = \\ &= g^{-1} \left[\int_A (g \circ f) d(g \circ m) \right] \end{aligned}$$

y así, obtenemos una relación análoga a la (6). Debido a este resultado, sólo consideraremos la integral extendida a Ω .

Ejemplo: En el caso de considerar la t-conorma de Sugeno, dada por:

$$\perp_{\lambda}(x, y) = \min \{x + y + \lambda xy, 1\} ; \lambda > -1$$

los generadores aditivos correspondientes son:

$$g_\lambda(x) = \frac{\ln(1 + \lambda x)}{\ln(1 + \lambda)} = \log_{1+\lambda}(1 + \lambda x)$$

y:

$$g_\lambda^{(-1)}(y) = \begin{cases} \frac{(1+\lambda)^y - 1}{\lambda} & \text{if } y \leq 1 \\ 1 & \text{if } y > 1 \end{cases}$$

Entonces si m_λ es la medida difusa NSA correspondiente, se sigue:

$$\int_A^{(g)} f \perp_\lambda m_\lambda = -\frac{1}{\lambda} + \frac{1}{\lambda}(1 + \lambda) \int_A \log_{1+\lambda}(1 + \lambda f) dg^*$$

con

$$g^* = g \circ m = \log_{1+\lambda}(1 + \lambda m)$$

Y ésta es la definición de la integral de Kruse dada en [4].

Resumimos ahora las propiedades de la integral difusa (5) mediante el teorema siguiente:

TEOREMA 2.3 La integral difusa, definida anteriormente, tiene las siguientes propiedades:

- 1) $f_1 \leq f_2 \Rightarrow \int_\Omega^{(g)} f_1 \perp m \leq \int_\Omega^{(g)} f_2 \perp m$
- 2) $A \subset B \Rightarrow \int_A^{(g)} f \perp m \leq \int_B^{(g)} f \perp m$
- 3) Para $A \cap B = \emptyset$ es $\int_{A \cup B}^{(g)} f \perp m = \left(\int_A^{(g)} f \perp m \right) \perp \left(\int_B^{(g)} f \perp m \right)$

4) Teorema de la convergencia monótona:

Si $\{f_n\}$ es una sucesión de funciones medibles de Ω en $[0, 1]$ y $(f_n) \uparrow f$ (o $(f_n) \downarrow f$) entonces $0 \leq f(\omega) \leq 1$ para todo $\omega \in \Omega$ y se cumple:

$$\int_\Omega^{(g)} f_n \perp m \uparrow \int_\Omega^{(g)} f \perp m$$

$$(o bien \int_\Omega^{(g)} f_n \perp m \downarrow \int_\Omega^{(g)} f \perp m)$$

5) Teorema de la convergencia dominada:

Supongamos que $(f_n) \rightarrow f$ y que existe una función medible F tal que $f \leq F$, entonces:

$$\int_\Omega^{(g)} f_n \perp m \rightarrow \int_\Omega^{(g)} f \perp m$$

6) Lema de Fatou:

$$\int_\Omega^{(g)} \left(\liminf_n f_n \right) \perp m \leq \liminf_n \int_\Omega^{(g)} f_n \perp m$$

DEMOSTRACIÓN i) y ii) son triviales.

$$\begin{aligned} 3) \int_{A \cup B}^{(g)} f \perp m &= g^{-1} \left[\int_{A \cup B} (g \circ f) dm^* \right] = \\ &= g^{-1} \left[\int_A (g \circ f) dm^* + \int_B (g \circ f) dm^* \right] = \\ &= g^{-1} \left[(g \circ g^{-1}) \int_A (g \circ f) dm^* + \right. \\ &\quad \left. + (g \circ g^{-1}) \int_B (g \circ f) dm^* \right] = \\ &= g^{-1} \left[g \left[\int_A^{(g)} f \perp m \right] + g \left[\int_B^{(g)} f \perp m \right] \right] = \\ &= \left(\int_A^{(g)} f \perp m \right) \perp \left(\int_B^{(g)} f \perp m \right) \end{aligned}$$

4) Como g es continua y monótona creciente, $g \circ f_n \uparrow g \circ f$ y aplicamos el teorema clásico de la convergencia monótona:

$$\begin{aligned} \lim_{n \rightarrow \infty} \int_\Omega^{(g)} f_n \perp m &= \lim_{n \rightarrow \infty} g^{-1} \left[\int_\Omega (g \circ f_n) dm^* \right] = \\ &= g^{-1} \left[\int_\Omega (g \circ f) dm^* \right] = \int_\Omega^{(g)} f \perp m \end{aligned}$$

5) Ya que g es creciente, tendremos $g \circ f_n \leq g \circ F$ y aplicamos el teorema clásico de la convergencia dominada de una forma similar.

6) Si $h_n = \inf_{k \geq n} f_k$ entonces:

$$0 \leq h_n \uparrow h = \liminf_n f_n$$

y por los apartados anteriores i) y iv) se sigue:

$$\begin{aligned} \int_\Omega^{(g)} f_n \perp m &\geq \\ &\geq \int_\Omega^{(g)} h_n \perp m \rightarrow \int_\Omega^{(g)} h \perp m \end{aligned}$$

□

TEOREMA 2.4 Sea $(\Omega, \mathcal{A}, m)$ un espacio de medida difusa. La función: $m_1: \mathcal{A} \rightarrow [0, 1]$ definida por:

$$m_1(A) = \int_A^{(g)} 1 \perp m = \int_\Omega^{(g)} I_A \perp m$$

es una medida difusa σ - $\perp$ -descomponible.

DEMOSTRACIÓN $m_1(\emptyset) = 0$ y $m_1(\Omega) = 1$ son obvias.

$m_1(A \cup B) = m_1(A) \perp m_1(B)$ es una consecuencia de iii) de 2.3

Como $h_n = I_{\bigcup_{k=1}^n A_k}$ es una sucesión monótona tal que $h_n \uparrow I_{\bigcup_{k=1}^\infty A_k}$ entonces por iv) del teorema 2.3:

$$\begin{aligned} m_1\left(\bigcup_{k=1}^\infty A_k\right) &= m_1\left(\lim_{n \rightarrow \infty} \bigcup_{k=1}^n A_k\right) = \\ &= \int_{\Omega}^{(g)} \lim_{n \rightarrow \infty} I_{A_n} \perp m = \\ &= \lim_{n \rightarrow \infty} \int_{\Omega}^{(g)} I_{A_n} \perp m = \\ &= \lim_{n \rightarrow \infty} \perp_{k=1}^n m_1(A_k) = \perp_{k=1}^\infty m_1(A_k) \end{aligned}$$

□

3 Teorema de Radon-Nikodym para medidas del tipo S

DEFINICIÓN 3.1 Una medida m , descomponible respecto de la t -conorma $\perp$, diremos que es m -alcanzable (m -achievable) cuando existe una familia numerable $\{A_k\}_{k \in \mathbb{N}}$ de conjuntos de $\mathcal{A}$ tales que $\Omega = \bigcup_{k=1}^\infty A_k$ y $m(A_k) < 1$, $\forall k \in \mathbb{N}$

Si m es una medida difusa del tipo NSA o S y es m -alcanzable, es evidente que la medida aditiva asociada m^* es σ -finita.

DEFINICIÓN 3.2 Sean m_1 y m_2 dos medidas definidas en el espacio medible $(\Omega, \mathcal{A})$, descomponibles respecto de la t -conorma $\perp$. Diremos que m_1 está dominada por m_2 ($m_1 \ll m_2$) cuando $m_1(A) = 0$, para todo $A \in \mathcal{A}$ tal que $m_2(A) = 0$.

TEOREMA 3.1 Sean m_1 y m_2 medidas descomponibles del tipo S, respecto de la t -conorma arquimediana $\perp$. Suponemos que son, respectivamente, m_1 -alcanzable y m_2 -alcanzable y que $m_1 \ll m_2$. Entonces si g es un generador aditivo de esas medidas, existe una función medible $f : \Omega \rightarrow [0, 1]$, tal que:

$$m_1(A) = \int_A^{(g)} f \perp m_2 ; \forall A \in \mathcal{A}$$

DEMOSTRACIÓN Por ser m_1 y m_2 alcanzables, sus medidas aditivas asociadas m_1^* y m_2^* son σ -finitas y m_1^* está dominada por m_2^* . En efecto si $m_2^*(A) = 0$ será $g^{-1}[m_2^*(A)] = \emptyset$; o sea $m_2(A) = 0$ y, por tanto, $m_1(A) = 0$. Entonces $m_1^*(A) = g[m_1(A)] = 0$.

Aplicamos ahora el teorema clásico de Radon-Nikodym para las medidas aditivas y σ -finitas m_1^*

y m_2^* . Existirá una función no negativa $h : \Omega \rightarrow \mathbb{R}$ tal que para cualquier $A \in \mathcal{A}$:

$$m_1^*(A) = \int_A h dm_2^* \quad (7)$$

consideremos la función $f = g^{-1} \circ h : \Omega \rightarrow [0, 1]$. Se cumple que $g \circ f = (g \circ g^{-1}) \circ h = h$ y de (7) obtenemos:

$$\begin{aligned} m_1^*(A) &= \int_A (g \circ f) dm_2^* \Rightarrow \\ m_1(A) &= g^{-1}[m_1^*(A)] = g^{-1}\left[\int_A (g \circ f) dm_2^*\right] = \\ &= \int_A^{(g)} f \perp m_2 \end{aligned}$$

□

Si quisiéramos dar una versión del teorema anterior para medidas del tipo NSA, nos encontramos con el problema de que la función de densidad h no tiene por qué tomar su recorrido en $[0, 1]$ (recordemos que para medidas NSA se toma el generador aditivo normalizado) y no podríamos considerar la función $f = g^{-1} \circ h$, como se ve en el siguiente ejemplo:

EJEMPLO 3.1 Sea $\Omega = [0, 1]$, $\mathcal{A} = \sigma$ -álgebra de Borel sobre $[0, 1]$. Supongamos que m_1 es la medida de probabilidad inducida por una variable aleatoria uniforme en $[0, 1]$ y m_2 la medida de probabilidad inducida por una variable aleatoria uniforme en $[0, \frac{1}{2}]$. Es evidente que $m_2 \ll m_1$. Entonces $h(x) = 2$, $\forall x \in [0, 1]$.

Notas finales

En la teoría de la probabilidad clásica, los teoremas de Fubini y Radon-Nikodym son esenciales para la definición y estudio de las distribuciones marginales y condicionadas. En esta línea, nuestro objetivo para posteriores trabajos es dar una versión del teorema de Radon-Nikodym para integrales difusas respecto de medidas NSA y del teorema de Fubini para integrales respecto de medidas del tipo S. Este último teorema respecto de medidas NSA ha sido presentado en [12]. También intentaremos extender los resultados anteriores para medidas NSP.

Referencias

- [1] Breiman, L.: *Probability*. Addison-Wesley Publishing Company, 1968
- [2] Choquet, G.: *Theory of Capacities*. Ann. Institut Fourier V (1953)(131-295)

- [3] Halmos, P.: *Measure Theory*. Springer-Verlag. New York Inc. 1974
- [4] Kruse, R.: *Fuzzy Integrals and Conditional Fuzzy Measures*. Fuzzy sets and Systems 10 (1983) (309-313)
- [5] Ling, C.H.: *Representations of associative functions*. Publ. Math. Debrecen 12 (189-212)
- [6] Loeve, M.: *Teoría de la Probabilidad*. Ed. Tecnos. Madrid.
- [7] Ralescu, D. and Adams, G.: *The Fuzzy Integral*. J. Math. Anal. Appl. 75 (1980) (562-570)
- [8] Schweizer, B. and Sklar, A.: *Associative functions and statistical triangle inequalities*. Pub. Math. Debrecen 12 (1965) (189-212)
- [9] Schwyhl: *About the Isomorphism between some Sugeno-Measures and Classical Measures*. in: E.P. Klement, Ed., Proc. 2nd Int. Sem. on Fuzzy Set Theory, Linz (1980) (143-174)
- [10] Shafer, G.: *A Mathematical Theory of Evidence*. Princeton University Press. Princeton, New York 1976.
- [11] Shafer, G.: *Nonadditive Probabilities in threework of Bernoulli and Lambert*. Arch. history Exact. Sci., 19 (1978) (309-370)
- [12] Suárez Díaz, E.: *Integración difusa sobre espacios producto*. V congreso español sobre tecnologías y lógica fuzzy. Murcia 1995. Pag 83-88.
- [13] Suárez García, F. and Gil Alvarez, P.: *Two Families of Fuzzy Integrals*. Fuzzy Sets and Systems 18 (1986) (67-81)
- [14] Sugeno, M.: *Theory of Fuzzy Integrals and its Applications*. Doctoral thesis. Tokyo Institute of technology. Tokyo (1974)
- [15] Sugeno, M. and Murofushi, T.: *Pseudo-additive Measures and Integrals*. Journal of Mathematical Analysis and Application. Vol. 122, No. 1 (1987) (197-222)
- [16] Szolovits, P. and Pauker, S.G.: *Categorical and Probabilistic Reasoning in Medicine*. Artificial Intelligence, 11 (1978) (115-144)
- [17] Weber, S.: *$\perp$ -Decomposable Measures and Integrals for Archimedean t -Conorms $\perp$* . J. Math. Anal. Appl. 101 (1984) (114-138)
- [18] Weber, S.: *Two Integrals and some modified versions-critical remarks*. Fuzzy Sets and Systems 20 (1986) (97-105)

Discretización de atributos cuantitativos en Sistemas Basados en el conocimiento.

Aïda Valls, Vicenç Torra
Dpt. d'Enginyeria Informàtica (ETSE)
Universitat Rovira i Virgili
Carretera de Salou, s/n
E-43006 Tarragona (Spain)
aidav@goliat.upc.es, vtorra@etse.urv.es

Resumen

Después de repasar algunos de los métodos de discretización de dominios cuantitativos existentes en la literatura, introducimos uno de nuevo con el que se obtienen mejores resultados.

Palabras Clave: Discretización.

1 INTRODUCCIÓN

La mayoría de sistemas en la inteligencia artificial trata con algún tipo de información cualitativa. Por ejemplo, en el desarrollo de Sistemas Basado en el Conocimiento (SBC) las características del dominio de aplicación pueden expresarse mediante variables cualitativas (variables que toman valores dentro de un conjunto de etiquetas lingüísticas). Entre los sistemas que usan etiquetas lingüísticas encontramos los árboles de decisión [25], los sistemas de *frames* [20], los sistemas expertos [17] o los sistemas de toma de decisión [13, 14] y algunos SBC basados en la teoría del constructo personal [15]. En estos sistemas es necesario un procedimiento para operar con los valores cualitativos. Por ejemplo, un sistema basado en la teoría del constructo personal (KSSn [11], ETS [5], AQUINAS [4, 11], EAR [23], ...) engloba varias herramientas: para combinar información de diferentes fuentes [24], para generar reglas a partir de estructuras definidas por el usuario, para analizar estas estructuras (clasificadores [12, 18], regresiones [5], ...). En este caso, cuando la matriz toma valores cualitativos para ciertos atributos, las herramientas deben poder tratar este tipo de información. En un clasificador deberá definirse una distancia para cada par de etiquetas lingüísticas y en una herramienta para síntesis de información deberá definirse una función de agregación para cada conjunto de etiquetas lingüísticas.

Podemos considerar dos opciones para operar con etiquetas lingüísticas. Podemos definir la semántica para cada etiqueta (esto es, definir un intervalo o un conjunto difuso para cada etiqueta) y operar de acuerdo con la semántica. Esta opción es la escogida en los sistemas de control difuso [8, 10] donde cada etiqueta corresponde a un conjunto difuso. Otra opción es operar sin tener en cuenta la semántica. Algunos sistemas porque no necesitan de la semántica, por ejemplo los métodos inductivos (ID3 [25]). Otros porque definen las operaciones directamente sobre las etiquetas lingüísticas. Este es el caso del operador de combinación convexa [7] o de la definición de operadores de negación y operadores y/o [3]. Estos operadores se definen sin una semántica previa de las etiquetas lingüísticas, sin embargo esta semántica está implícita en ellos.

La definición de la función de negación tal como aparece en la mayoría de sistemas [13, 14, 3] corresponde a este último caso. Las funciones de negación (sobre un conjunto ordenado de etiquetas lingüísticas $L=\{x_0, \dots, x_n\}$ (con $x_0 < \dots < x_n$)) se definen [13, 14, 19] como funciones de L en L que satisfacen:

N1) si $x < x'$ entonces $\text{Neg}(x) > \text{Neg}(x')$
para todo x, x' de L

N2) $\text{Neg}(\text{Neg}(x)) = x$
para todo x de L

Bajo estas condiciones, el conjunto ordenado L determina completamente la función de negación.

Proposición 1.1. [1] Para cada conjunto ordenado de etiquetas lingüísticas $L=\{x_0, \dots, x_n\}$ existe una única función de negación que satisface las condiciones N1 y N2. La función de negación se define como:

$$\text{N}(x_i) = x_{n-i} \quad \text{para todo } x_i \text{ de } L \quad (1.1)$$

Una función de negación de acuerdo con esta proposición corresponde a situaciones donde cada etiqueta en el par $\langle x_i, x_{n-i} \rangle$ es igualmente

informativa. Por tanto, esta condición es implícita siempre que se consideran las condiciones N1 y N2 (por ejemplo en [13, 14, 1]).

Cuando consideramos un sistema basados en el conocimiento no siempre es adecuado que las etiquetas sean igualmente informativas. Sin embargo, no es siempre posible que el experto suministre en el proceso de adquisición del conocimiento un intervalo o un conjunto difuso para cada etiqueta lingüística. Esto requiere un exceso de precisión. En este sentido, puesto que es más fácil para un experto definir una función de negación que una semántica precisa se estudiaron [28] las funciones de negación y se mostró como estas funciones inducen una semántica para cada etiqueta. Se mostró también que la nueva definición de funciones de negación no exige que las etiquetas sean igualmente informativas. De este modo el experto suministra

más información sobre la semántica de forma más natural.

Como ejemplo del interés de etiquetas que no son igualmente informativas podemos considerar un sistema de control de la temperatura al que se le exige precisión cuando la temperatura es baja. En este caso, es conveniente una sensibilidad mayor en las etiquetas correspondientes a temperaturas bajas. Por tanto la distribución de las etiquetas lingüísticas en el dominio será parecida a la figura 1 (en la figura 1, AN corresponde a Almost-Nil, VL corresponde a Very-Low, QL=Quite-Low, VH=Very-High). Cuando el conjunto de etiquetas se define de este modo, la función de negación no puede ser de la forma de la proposición 1. Debilitando las condiciones N1 y N2, y definiendo la función de negación como una función de L en $\wp(L)$ se puede solucionar este problema.

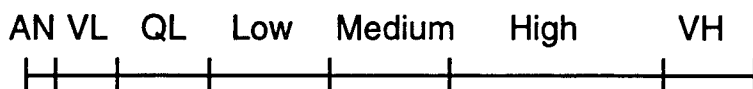


Figura 1

Hemos estudiado con detalle las funciones de negación y la semántica introducida en [28] observando que son adecuadas para capturar más información. El estudio, que se ha realizado a partir del clasificador Sedàs [29], utiliza un método de discretización. Este método construye una partición del dominio a partir de un conjunto de objetos que están evaluados de forma cuantitativa en este dominio. La construcción de particiones se ha estudiado en la literatura desde diferentes puntos de vista (por ejemplo, en reconocimiento de formas [16, 30], análisis de datos [2], aprendizaje inductivo [6]). Sin embargo, estos métodos se aplican con un objetivo diferente pues intenta detectar acumulaciones de objetos de acuerdo con sus valores. Nuestro caso es diferente, nos interesa un método que construya las etiquetas lingüísticas tal como un experto lo haría. El método debe establecer semejanzas entre objetos en la parte relevante del dominio en lugar de reunir estos objetos en una única partición. Esta diferencia dirige nuestro trabajo. El trabajo se centra en el proceso de discretización. En la sección 2, repasamos el análisis de la semántica que ha dirigido este trabajo. En la sección 3, introducimos el proceso de discretización y en la sección 4 damos algunos resultados correspondientes al análisis de la semántica. El trabajo finaliza con las conclusiones.

2. ANÁLISIS DE LA SEMÁNTICA

El estudio de las funciones de negación de L en $\wp(L)$ definidas en [28] y su adecuación para capturar información se basa en la comparación de las clasificaciones obtenidas a partir de una matriz cualitativa utilizando la semántica de [28] y utilizando la semántica clásica. Para realizar la comparación construimos una matriz cualitativa a partir de una cuantitativa. Esta clasificación al considerarse la clasificación *ideal* puede utilizarse para comparar las otras clasificaciones.

Por tanto, hemos seleccionado primero una matriz cuantitativa. Entonces, cada atributo se ha discretizado en un conjunto de etiquetas lingüísticas. Esto es, se han realizado varias particiones de cada dominio, asociando un intervalo a cada etiqueta. En el siguiente paso, y de acuerdo con las discretizaciones, hemos definido la matriz cualitativa substituyendo cada valor cuantitativo por su correspondiente valor cualitativo.

Finalmente, se ha utilizado el clasificador Sedàs [29] para clasificar la matriz cuantitativa y la cualitativa. Esta última se ha clasificado de dos formas diferentes. Por un lado, de acuerdo con la semántica clásica: esto corresponde a clasificar con la función de negación definida como

$Neg(x_i)=x_{n-i}$ para cada etiqueta x_i . Por otra parte, se ha clasificado de acuerdo con la nueva semántica: se ha definido para ello, la negación de cada etiqueta de forma que se satisficieran las condiciones de consistencia definidas en [28]. Esto es, si consideramos que $I(x)$ es la partición del dominio correspondiente a la etiqueta x , entonces la negación de x se define como:

$$Neg(x) = \{x_i | \exists p \in I(x) \text{ such that } 1-p \in I(x_i)\}$$

Una vez se dispone de las tres clasificaciones, es posible comparar las dos calculadas a partir de la matriz cualitativa y la calculada a partir de la

matriz cuantitativa. En nuestro caso, el análisis se ha realizado partiendo de una matriz cuantitativa con 90 objetos y 4 atributos (los datos pueden encontrarse en el *UCI repository* [21]). Originariamente la matriz disponía de 150 objetos pero para simplificar los cálculos estos se han reducido a 90. De este modo, la matriz de semejanzas se ha reducido de 11175 valores a 4005. Los dominios de los atributos se encuentran en la tabla I (*Min* y *Max* corresponden, respectivamente, al mínimo y al máximo de los valores del dominio).

Tabla I. Dominios de los atributos y los centros de los intervalos construidos.

	Min	Max	m_1	m_2	m_3	m_4	m_5	m_6	m_7
Sepal length	4.4	7.7	4.62 9	5.05 21	5.50 10	5.84 12	6.27 15	6.69 14	7.41 9
Sepal width	2.0	4.2	2.34 12	2.75 17	2.90 7	3.02 18	3.20 11	3.39 15	3.79 10
Petal length	1.0	6.9	1.32 13	1.60 17	3.4 4	4.26 16	4.86 20	5.71 15	6.64 5
Petal width	0.1	2.5	0.19 21	0.40 9	1.04 7	1.41 21	1.65 4	1.87 15	2.25 13

3. DEFINICIÓN DE ETIQUETAS: EL PROCESO DE DISCRETIZACIÓN

La construcción de un conjunto de intervalos en un determinado dominio corresponde a la construcción de una partición de un conjunto de valores cuantitativos. Este problema ha sido estudiado desde varios puntos de vista en la literatura (por ejemplo, en el reconocimiento de patrones y en el análisis de datos), sin embargo los métodos existentes no se adecuan a nuestro caso. En el análisis de datos, los métodos se aplican a la detección de acumulaciones de objetos, definiendo en este caso un intervalo que los recubre. Los métodos para el reconocimientos de formas [16, 30] dan resultados similares. Por ejemplo, en la figura 2 se obtienen los intervalos ($P_1=[0,7.25]$, $[7.25,13]$, $[13,21]$) en lugar de

los más naturales ($P_2=[0,3.5]$, $[3.5,10.5]$, $[10.5,21]$). Debido a que una acumulación de objetos corresponde a valores frecuentes, se necesitará en este caso precisión (esto es, se necesitarán intervalos pequeños). Si consideramos el ejemplo de la figura 1, un sistema de control de la temperatura encuentra a lo largo de su ejecución valores pequeños en su mayor parte. En pocas ocasiones, únicamente, el sistema encontrará valores grandes (errores grandes). Esto es, la frecuencia de los puntos será parecida a los de la figura 3. En este caso, aunque los valores están acumulados en el lado izquierdo del dominio, preferiremos varias etiquetas pequeñas para obtener una mayor precisión.

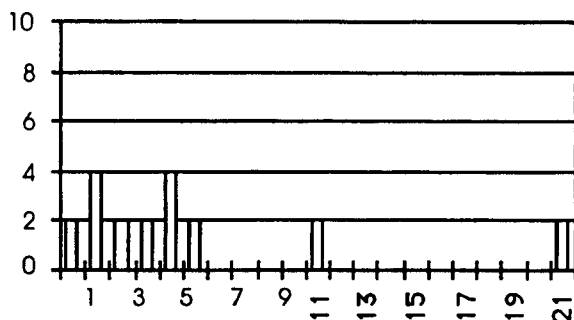


Figura 2.

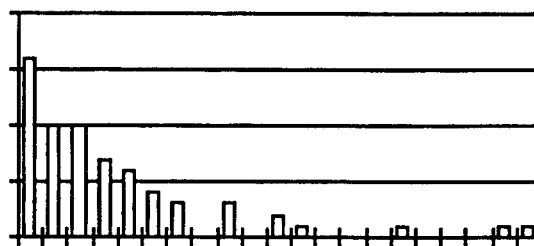


Figura 3

En la literatura sobre clasificación, el conjunto de métodos que genera particiones se conoce como técnicas de clasificación no jerárquicas [2] o

técnicas de partición [9]. Estos métodos se basan en el algoritmo clásico introducido por Forgy en 1965 [2]:

1. Selecciona k centros iniciales que inducen k clases.
2. Asigna cada objeto a la clase que tiene el centro más cercano.
3. Recalcula los nuevos centros de cada clase como el valor medio de los valores.
4. Repite los pasos 2 y 3 hasta que el centro converge.

Una variación de este método fue introducido por MacQueen [2] y es la base de otros algoritmos. En cada paso sólo se considera un único objeto y el centro de la clase más cercana se modifica para acercarlo a este objeto.

Aunque este último método tiene un coste computacional menor, tiene algunos inconvenientes. Por un lado, el resultado depende del orden de selección de los elementos. Para reducir esta dependencia se realizan varias iteraciones con el conjunto de objetos, y en cada iteración el orden de selección es aleatorio (cada iteración con todo el conjunto se llama *epoch*). Esta modificación del algoritmo provoca un incremento del coste computacional. Por otra parte, el método depende de un parámetro extra: la función que modifica el centro de una clase teniendo en cuenta el elemento. Kohonen [16] junto con [30] usan la función siguiente:

$$m_c(t+1) = m_c(t) + \eta(t) (x(t) - m_c(t)) \quad (1)$$

donde $x(t)$ es el objeto escogido de forma aleatoria, m_c el centro más cercano y $\eta(t)$ un coeficiente de aprendizaje que debe ser una función decreciente. Sin embargo, ni [16] ni [30] estudian con detalle esta función. En [22] se exige que esta función debe ser lineal decreciente de $(0,0.5)$ a $(1000,0.04)$ y entonces también lineal hasta $(10000,0)$. A partir de este punto igual a cero.

En las dos familias de métodos (los basados en los algoritmos de Forgy's y en el de MacQueen) se necesita seleccionar un centro inicial para cada clase. Dos aproximaciones se han considerado en la literatura, seleccionando los centros iniciales a partir de los datos (de forma aleatoria o los primeros k elementos) o inicializando el conjunto de centros para que estén distribuidos a lo largo de X , esto es:

$$m_i(0) = \min_{x \in X} \{x\} + (\max_{x \in X} \{x\} - \min_{x \in X} \{x\}) * (i-1) / (k-1) \quad i=1, \dots, k$$

Hemos comprobado que la selección de los centros iniciales determina el resultado final y que una mala distribución inicial provoca la obtención de *malas* particiones. De hecho, cuando el conjunto de datos no se encuentra uniformemente

repartido a lo largo del dominio, los centros no cambiarán de forma radical. Por ejemplo, si en el ejemplo 3 intentamos construir 7 intervalos y se fijan inicialmente de forma que la longitud sea igual $(m_i = (i+0.5) * 21/7, i=0, \dots, 6)$ entonces tendremos que al final del proceso algunos de los intervalos no se han modificado porque no engloban a ningún objeto (en particular m_5 continuará valiendo 16.5, como en la definición inicial, debido a que no sufre la influencia de ningún objeto). Con la selección de los centros iniciales a partir de datos (de forma aleatoria o con la selección de los k primeros elementos) se evitan los intervalos vacíos. Sin embargo, los resultados dependen de esta selección y también pueden obtenerse *malas* particiones.

En todos los métodos es necesario un criterio de convergencia. En la literatura se consideran, principalmente, dos criterios. En el análisis de datos corresponde a la variación de centros entre iteraciones.

$$D_V(M^{(k)}, M^{(j)}) = \sum_i \|m_i^{(k)} - m_i^{(j)}\|$$

A su vez, Kohonen define una medida de la distancia entre los datos originales y una partición:

$$D(X, M) = \sum_{x \in X} \min_i \|x - m_i\|$$

En este caso, el método se aplica hasta que la distancia converge.

De acuerdo con [30], el método de discretización lleva a una minimización de esta distancia. Por tanto ambos criterios de convergencia pueden utilizarse. En nuestro caso se ha utilizado el criterio de variación de centros porque es más fácil de calcular.

Para realizar el test de estos métodos de discretización, hemos creado varios conjuntos de datos artificiales (mediante el software estadístico MINITAB). Además hemos utilizado datos reales extraídos del *UCI repository* [21]. Hemos implementado los métodos de Forgy y de MacQueen, utilizando en este último caso la función de modificación (1) con $\eta(t)$ definida como $1/(8+t/2000)$ y la definida en [22]. Hemos detectado los problemas mencionados más arriba. De hecho, los malos resultados corresponden a mínimos locales de la función D definida más arriba. Para solucionar estos problemas se han introducido dos modificaciones a los métodos mencionados.

Por una parte, definiendo los centros iniciales de forma que cada uno agrupe a la misma proporción de los objetos. En un primer paso, se ordenan los objetos de acuerdo con sus valores y

a continuación se agrupan en tantas clases como intervalos se necesiten. Los centros de los intervalos se calculan como el promedio de los valores en cada clase:

$$m_i = (1/|C_i|) * \sum_{O \in C_i} V(O)$$

donde $V(O)$ es el valor asociado al objeto O , y C_i es la i -ésima clase de objetos.

Por otra parte, hemos considerado un método de eliminación de intervalos superfluos y un método de creación de nuevos intervalos. En cada paso, los intervalos con menos de un 1% de objetos se eliminan. De este modo, los elementos aislados se concentran en clases de dimensión mayor. Además, en cada *epoch*, si el número de clases es menor que el número de intervalos permitidos, la clase con un número de elementos mayor se divide en dos.

Tabla II. Muestras artificiales estudiadas.

	N. P.	Dominio	Distribuciones
S1	230	[-3.26,5.72]	30 with N(5,..5), 200 with Chisquare (with 200 degrees of freedom)
S2	200	[-2.49,10.82]	100 with Chisquare, 100 Chisquare moved 8 units to the right (100 d. of f.)
S3	210	[1.0,19.0]	30 each N(1,.5), N(4,.5), N(7,.5), N(10,.5), N(13,.5), N(16,.5), N(19,.5)
S4	180	[1.0,19.0]	30 each N(2.5,.5), N(5.5,.5), N(8.5,.5), N(11.5,.5), N(14.5,.5), N(17.5,.5)
S5	250	[2.86,20.1]	80 N(4,.5), 55 N(8,.5), 35 N(8,.5), 20 N(16,.3) 60 N(19,.5)

Con estas modificaciones hemos conseguido mejores resultados. Debe apuntarse que la eliminación de intervalos superfluos puede incrementar el valor de D . Sin embargo da resultados que son más adecuados a nuestros

intereses. Por ejemplo, considerando de nuevo el ejemplo 3, $D(X,P_1) < D(X,P_2)$ pero P_2 se prefiere a P_3 .

Tabla III. Intervalos generados con el método de Forgy.

(a) centros iniciales con distribución uniforme								(b) misma proporción de objetos en cada centro.						
	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_1	m_2	m_3	m_4	m_5	m_6	m_7
S1	-2.4	-1.0	-.05	.8	1.6	4.6	5.4	-2.5	-1.0	-.3	.3	1.0	1.8	5.2
S2	-1.6	-.3	.9	6.6	7.8	8.8	10.2	-1.6	-.4	.5	1.3	6.6	8.0	9.6
S3	1.4	4.1	6.9	10.1	12.9	16.0	18.6	1.4	4.1	6.9	10.1	12.9	16.0	18.5
S4	2.5	5.4	8.4	10.8	11.8	14.6	17.5	2.5	5.3	7.8	8.9	11.5	14.6	17.5
S5	3.5	4.3	7.7	8.5	11.0	16.0	19.0	3.5	4.3	8.0	11.0	16.0	18.5	19.3

Tabla IV. Intervalos generados con el método de MacQueen

(a) centros iniciales con distribución uniforme								(b) misma proporción de objetos en cada centro.						
	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_1	m_2	m_3	m_4	m_5	m_6	m_7
S1	-1.5	-.5	.1	.9	1.7	4.6	5.3	-1.9	-.9	-.3	.2	.9	1.6	4.9
S2	-1.7	-.3	.8	6.5	7.6	8.5	9.8	-1.6	-.1	1.0	6.4	7.6	8.6	9.8
S3	1.4	4.2	6.9	10.0	13.0	15.8	18.6	1.4	4.1	6.9	10.2	13.0	15.9	18.6
S4	2.4	5.2	7.8	8.8	11.5	14.7	17.6	2.5	5.2	7.7	8.9	11.4	14.7	17.4
S5	3.6	4.3	8.0	11.0	16.0	18.6	19.3	3.5	4.3	7.7	8.6	11.0	16.0	19.1

Para realizar el análisis de la semántica definida en [28], hemos utilizado el método de Forgy con las variaciones presentadas en este trabajo. La elección del método de Forgy se debe a que no depende del proceso de selección ni de la función η aunque sea más costoso que el método de MacQueen. Para realizar el estudio hemos fijado el número de etiquetas a 7 (vease [3, 6] sobre la selección del número de etiquetas). En la tabla I, damos el centro de cada clase para cada atributo (el valor superior en cada celda m_i , $i=1,...,7$). Junto a los centros encontramos también el número de elementos de la clase (el valor inferior en cada celda). En las tablas II, III y IV

mostramos los resultados correspondientes a los datos construidos de forma artificial. En la tabla II mostramos los conjuntos generados (número de puntos (NP), dominio y distribuciones generadas). En las tablas III y IV aparecen los centros de los siete intervalos generados con cuatro métodos distintos: métodos de Forgy y de MacQueen considerando dos formas de definir los centros iniciales (distribución uniforme o misma proporción de objetos en cada centro).

4. CONCLUSIONES

En este trabajo se ha presentado un método para la discretización de los atributos cuantitativos. Se ha justificado su necesidad en base a los principales métodos existentes. La definición presentada aquí ha sido utilizada para estudiar la semántica introducida en [28]. Se ha podido comprobar que los atributos cuantitativos tienen mayor estructura que los cualitativos, y que la nueva semántica permite representar mejor esta estructura. Como trabajo futuro apuntamos la necesidad de estudiar la definición de una función D alternativa cuya minimización corresponda a los valores obtenidos.

Agradecimientos

Los autores agradecen a David Riaño sus comentarios. Aida Valls agradece la ayuda recibida en el departamento de Lenguajes y Sistemas Informáticos de la UPC. Parte de este trabajo ha sido realizado en el Institut d'Investigació en Intel·ligència Artificial (IIIA, CSIC).

Este trabajo se enmarca dentro del proyecto VIM: CHRXC93-0401.

Referencias

- [1] Agustí Cullel, J., Esteve, F., García, P., Godó, L., Sierra, C., (1991), Combining Multiple-valued Logics in Modular Expert Systems, *Proc. 7th Conference on Uncertainty in AI*, Los Angeles, 17-25.
- [2] Anderberg, M.R., (1973), *Cluster Analysis for Applications*, Academic Press, Inc.
- [3] Bonissone, P. P., Decker, K. S., (1986), Selecting Uncertainty Granularity: An Experiment in Trading-off Precision and Complexity, in L. H. Kanal, J. F. Lemmer, Eds., *Uncertainty in Artificial Intelligence*, North-Holland, 217-247.
- [4] Boose, J. H., Bradshaw, J. M., (1987), Expertise transfer and complex problems: using AQUINAS as a knowledge-acquisition workbench for knowledge-based systems, *Int. J. of Man-Machine Studies*, 26, 3-28.
- [5] Boose, J. H., (1986), *Expertise Transfer for Expert System Design*, Elsevier Science Publishers B.V.
- [6] Ching, J. Y., Wong, A. K. C., Chan, K. C. C., (1995), Class-Dependent Discretization for Inductive Learning from Continuous and Mixed-Mode Data, *IEEE Trans. on PAMI*, 17:7, 641-651.
- [7] Delgado, M., Verdegay, J.L., Vila, M.A., (1993), On Aggregation Operations of Linguistic Labels, *Int. J. of Intelligent Systems*, 8, 351-370.
- [8] Efstathiou, J., (1995), Special Issue on Modern Fuzzy Control, *Fuzzy Sets and Systems*, 70:2-3.
- [9] Everitt, B. (1977), *Cluster analysis*, Heinemann Educational Books Ltd.
- [10] Fujieda, Sh., Tanaka, S., (1992), Fuzzy Control for the precision water bath of the heat exchange calorimeter", *J. of Thermal Analysis*, 38, 1785-1792.
- [11] Gaines, B. R., Shaw, M. L. G., (1993), Knowledge Acquisition Tools Based on Personal Construct Psychology, *The Knowledge Eng. Review*, 8:1 49-85.
- [12] Gibert, K. (1994), *L'ús de la informació simbòlica en l'automatització del tractament estadístic de dominis poc estructurats*, Tesis Doctoral, Universitat Politècnica de Catalunya.
- [13] Herrera, F., Herrera-Viedma, E., Verdegay, J.L., (1995), A Sequential Selection Process in Group Decision Making with a Linguistic Assessment Approach, *Int. J. Information Science*, 80, 1-17.
- [14] Herrera, F., Herrera-Viedma, E., Verdegay, J.L., (1996), A model of consensus in group decision making under linguistic assessments, *Fuzzy Sets and Systems*, 78, 73-87.
- [15] Kelly, G.A., (1955), *The Psychology of Personal Construct*, W.W. Norton & Co.: New York.
- [16] Kohonen, T., (1988), *Self-Organization and Associative Memory*, Springer, Berlin.
- [17] López de Mántaras, R., Sanz, F., Sierra, C., Verdaguer, A., (1987), Milord + Pneumon-ia: Un outil et une application en médecine, *Colloque Intelligence Artificielle et Santé*, Toulouse, 45-54.
- [18] Martín, M., Sangüesa, R., Cortés, U., (1991), Knowledge acquisition combining analytical and empirical techniques". Proceedings of the eighth international workshop on Machine learning, Ed. Morgan-Kaufman, 657-661.
- [19] Mayor, G. Torrens, J., (1993), On a Class of Operators for Expert Systems, *Int. J. of Intelligent Systems*, 8, 771-778.
- [20] Minsky, M., (1975), A framework for representing knowledge. In P.H. Winston (ed.), *The Psychology of Computer Vision*. New York: McGraw-Hill, 211-277.
- [21] Murphy, P., M., Aha, D. W., (1994), UCI Repository machine learning databases, <http://www.ics.uci.edu/~mllearn/MLRepository.html>, Irvine, CA: University of California, Department of Information and Computer Science.
- [22] Mustagh, F., Hernandez-Pajares, M., (1995), The Kohonen Self-Organizing Map Method: An Assessment, *Journal of Classification*, 12:2, 165-189.
- [23] Plaza, E., (1987), *EAR*: Sistema d'Ajut a l'adquisició i estructuració de coneixements*, Tesis doctoral, Universitat Politècnica de Catalunya.
- [24] Plaza, E., Alsina, C., López de Mántaras, R., Aguilar, J., Agustí, J., (1986), Consensus and knowledge acquisition, In R.R. Yager, B. Bouchon (Eds), *Uncertainty Management in Knowledge-Based Systems*, LNCS, Springer-Verlag.
- [25] Quinlan, J.R. (1986), Induction of Decision Trees, *Machine Learning* 1, 81-106.
- [26] Riaño, D., (1994), Knowledge abstraction using heuristic search I: A specialization approach, Report CS-R9469 of the Centrum voor Wiskunde en Informatica (CWI), Amsterdam, Netherlands.
- [27] Torra, V., Cortés, U., (1995), Towards an Automatic Consensus Generator Tool: EGAC, *IEEE Trans. on Systems, Man and Cybernetics*, 25:5, 888-894.
- [28] Torra, V., (1996), Negation functions based semantics for ordered linguistic labels, *Int. J. of Int. Syst.*, (Accepted).
- [29] Valls, A., (1995), *Sedàs. General Classification Module for ill structured domains*, (in catalan), Master Thesis, ETSE, Universitat Rovira i Virgili.
- [30] Yuan, Y., Shaw, M. J. (1995), Induction of fuzzy decision trees, *Fuzzy Sets and Systems*, 69, 125-139.

INDICE DE AUTORES

ACEVEDO SOTOCA, M. Isabel	173	JACAS, Joan	41
AMO, Ana del	1	JESUS, M ^a José del	95
ARANGUREN, Gerardo	255	JIMENEZ, Jorge	161, 195
ARENAS, M. Mar	7	JIMENEZ, Mariano	7
		JUNCO, L.	267
BARRIGA, Angel	167		
BARRO, Senén	77	LAFARGA, A.	47, 261
BARRON, Mariano	255	LAMATA, M. Teresa	143
BENITEZ, Jose M.	13	LINARES, L.J.	101
BERTOLUZZA, Carlo	19	LOPEZ, Diego	167
BILBAO, Amelia	7	LOPEZ, Enrique	231
BLANCO, A.	23, 29, 35	LOPEZ ALIGUE, Francisco J.	173
BODINI, Antonella	19	LOPEZ MUÑOZ, Javier	225
BOIXADER, D.	41		
BOLAÑOS, M. J.	47	MARTIN, F.	277
BUGARIN, A.	77	MARTINEZ, Daniel	179
BULDAIN, G.	47, 261	MARTINEZ-BEJAR, R.	277
BULDAIN, P.	47, 261	MARTINEZ TORRE, José Ignacio	107
BURUSCO, A.	53	MAS, M.	71
BUSTINCE, Humberto	59	MAS, Orestes	179, 185
		MATELLAN, Vicente	191
CADENAS, J. M.	277	MAYOR, G.	71
CALDERON, M.	59	MEDINA, J. Miguel	219, 281
CAMPOS, Luis de	65	MEGIDO, J. Alfonso	271
CARBONELL, M.	71	MIRO, Joan M.	185
CARIÑENA, P.	77	MOHEDANO, V.	59
CASTRO, Juan L.	13, 83	MOLINA, Elisenda	197
CASTRO SANCHEZ, J.J.	83	MOLINA, José M.	191
CONEJO, R.	213	MONTERO, Javier	1
CORDON, Oscar	89, 95	MONTES, Susana	161, 201
CUBERO, J. C.	213	MONTSENY, Eduard	207
		MORALES BUENO, R.	213
DELGADO, Miguel	23, 29, 101	MORENO, Rafael	107
DESCHAMPS, Jean Pierre	107		
ESTEVA, Francesc	113	PALA, Pere	185
		PARDO, Xoan Carlos	125
FAJARDO, Waldo	23, 35	PEGALAJAR, M. C.	29, 35
FERNANDEZ, Camino	191	PEREZ, J.	271
FERNANDEZ, M. José	119	PEREZ, Raúl	131
FERREIRO, Ramón	125	PEREZ de la CRUZ, J. L.	213
FUENTES GONZALEZ, R.	53	PINA, A.	47, 261
		PONS, O.	219
GARCIA, Pere	113	PUYOSA, H. D.	287
GIL, Pedro	201		
GODO, Lluís	113	RAMOS JIMENEZ, Gonzalo	225
GOMEZ-SKARMETA, A.F.	101	RASHID, David	281
GONZALEZ, Antonio	131	RECASENS, J.	41
GRAVALOS, C.	271	RECHE, Fernando	47
GUARDIOLA, F.	287	REQUENA, Ignacio	13, 231
		RODRIGUEZ, Miguel	255
HERENCIA, José A.	137, 143	RODRIGUEZ, M. Victoria	7
HERNANDEZ MOLINERO, Luis D.	149	RODRIGUEZ, R.	113
HERRERA, Francisco	89, 95, 155	ROLANDO, Martín	179
HERRERA-VIEDMA, E.	155		
HUETE, Juan F.	65	SALMERON, Antonio	149
		SANCHEZ, Luciano	267
		SANCHEZ SOLANO, Santiago	167

SERON, F.J.	47, 261
SOBREVILLA, Pilar	207
SOBRINO, Alejandro	235
SOTO ALARCON, P.	287
SUAREZ, Emilio	241
SUAREZ, Fermín	119, 161, 241
SUÑER, J.	71
TEJEDA, Juan	197
TOMAS, L. M.	287
TORRA, Vicenç	247
TORRALBO, J. Luis	281
TORRENS, J.	71
VALLS, Aida	247
VELA, Rafael	283
VERDEGAY, J. L.	155
VILA, M. Amparo	219, 281
ZURITA, J.M.	83