Decisión y Optimización

Decisión Interactiva para un Problema General Fuzzy Multiobjetivo

Jose M. Cadenas Fernando Jiménez
Universidad de Murcia
30071-Espinardo, Murcia
e-mail: jcadenas@dif.um.es, fernan@dif.um.es

Resumen

Presentamos un sistema interactivo que ayuda a resolver un problema general fuzzy multiobjetivo, es decir, problemas de programación multiobjetivo con metas fuzzy sujetos a un conjunto de restricciones donde cada uno de los elementos que lo componen estan dados de forma imprecisa. Se propone un sistema interactivo donde despues de proponer el decisor las metas fuzzy con funciones de pertenencia y las distintas imprecisiones de los elementos de las restricciones, se obtienen una solución satisfactoria adaptando el conjunto de valores de pertenencia interactivamente. Discutimos en este trabajo el método de resolución del problema general de programación lineal fuzzy multiobjetivo y la programación fuzzy multiobjetivo interactiva.

Palabras clave: Programación Multiobjetivo, conjuntos fuzzy, sistema interactivo de ayuda a la decisión.

1 Introducción

Los modelos que consideran multiples criterios estan teniendo incrementada importancia en la decisión, [3]. Al decisor no se le fuerza a restringir sus consideraciones a un aspecto principal sino que puede tomar diferentes puntos de vista. En modelos de programación multiobjetivo, el criterio se describe por funciones objetivo y las posibles alternativas estan implícitamente determinadas por restricciones.

Esta demostrado que dada una clase de funciones de pertenencia de las metas fuzzy asignadas a las funciones objetivo y de las restricciones fuzzy en el problema, el uso de los métodos de programación lineal clásicos son posibles para resolver y analizar el problema,[12]. Como resultado de aplicación del método basado en la técnica de programación paramétrica, podemos obtener una solución fuzzy del problema. Esta solución es un subconjunto del conjunto de soluciones eficientes del problema. La solución la cual pertenece a este conjunto a grado más alto es una maximinimización (en terminología de Bellman y Zadeh). Es decir, una decisión fuzzy D es el conjunto fuzzy de todos los elementos que pertenecen a los conjuntos Z_l, l=1,..,k, que describen las metas y los conjuntos fuzzy R_i, i=1,..,m, describiendo las restricciones:

$$D = \bigcap_{l=1}^{k} Z_l \cap \bigcap_{i=1}^{m} R_i$$

Escoger un elemento optimal entonces significa seleccionar el elemento x^0, el cual tiene el grado más alto de la función de pertenencia del conjunto de decisión fuzzy:

$$\mu_D(x^0) = \max_{x \in X} \mu_D(x) =$$

$$= \max_{x \in X} \min \left\{ \min_{l=1,..,k} \mu_{Z_l}(x), \min_{i=1,..,m} \mu_{R_i}(x) \right\}$$

Consideraremos modelos lineales:

$$\begin{array}{ll} \underset{\sim}{Max} & (z_1, z_2, .., z_k) \\ s.a: & \\ & \sum_{j=1}^{n} a_{ij} x_j \underset{\sim}{\leq} b_i \quad , i = 1, .., m \\ & x_j \geq 0 \qquad\qquad , j = 1, .., n \end{array}$$

Si las metas estan descritas por funciones de pertenencia $\mu_{Z_l}(x)$ y las restricciones estas descritas por funciones de pertenencia $\mu_{R_i}(x)$ la decisión optimal es x^0 con

$$\mu_D(x^0) = \max_{x \in X} \min \{\mu_{Z_1}(x), .., \mu_{Z_k}(x), \mu_{R_1}(x), .., \mu_{R_m}(x)\}$$

Una formulación equivalente esta dada por el siguiente modelo de programación clásico.

$$\begin{array}{ll} Max & \alpha \\ s.a: & \\ & z_l(x) \geq \overline{z}_l - (1-\alpha)(\overline{z}_l - \underline{z}_l) \quad , l = 1, .., k \\ & \sum_{j=1}^{n} a_{ij} x_j \leq b_i + d_i(1-\alpha) \quad , i = 1, .., m \\ & x_j \geq 0, \quad \alpha \in [0,1] \qquad , j = 1, .., n \end{array}$$

2 Un Modelo General de Programación Lineal Fuzzy Multiobjetivo

En esta sección proponemos un modelo general de Programación Lineal Fuzzy (PLF) Multiobjetivo en el que consideramos que todos los elementos que interviene en el problema estan dados de forma imprecisa, es decir, un problema del tipo:

$$\begin{array}{ll} \underset{\sim}{Max} & (z_1, z_2, .., z_k) \\ s.a: & \\ & \sum_{j=1}^{n} \underset{\sim}{a_{ij}} x_j \lesssim \underset{\sim}{b_i} \quad , i \in M = \{1, .., m\} \\ & x_j \geq 0 \qquad\qquad , j \in N = \{1, .., n\} \end{array} \tag{1}$$

donde los elementos fuzzy se consideran dados por:

i) Para cada función objetivo, $\exists \mu_l^0 \in F(R)$ tal que $\mu_l^0 : R \to [0,1]$, l=1,..,k, las cuales definen las metas fuzzy.

$$\mu_l^0(z_l(x)) = \begin{cases} 0 & si \;\; z_l(x) \leq \underline{z}_l \\ \frac{z_l - \underline{z}_l}{\overline{z}_l - \underline{z}_l} & si \;\; \underline{z}_l \leq z_l(x) \leq \overline{z}_l \\ 1 & si \;\; z_l(x) \geq \overline{z}_l \end{cases}$$

donde los 2 valores $\underline{z}_l$, $\overline{z}_l$ denotan el mejor valor posible y el pero valor posible de la l-esima función objetivo, respectivamente.

ii) Para cada fila, $\exists \mu_i \in F(R)$ tal que $\mu_i : R \to [0,1]$, i=1,..,m, las cuales definen el número fuzzy en la parte derecha.

iii) Para cada i=1,..,m y j=1,..,n, $\exists \mu_{ij} \in F(R)$ tal que $\mu_{ij} : R \to [0,1]$, las cuales definen los números fuzzy en la matriz tecnológica.

iv) Para cada fila, $\exists \mu^i \in F(F(R))$ tal que $\mu^i : F(R) \to [0,1]$, i=1,..,m, que nos dá para cada x$\in R^n$, el grado de acoplamiento del número fuzzy

$$\underset{\sim}{a_{i1}} x_1 + \underset{\sim}{a_{i2}} x_2 + .. + \underset{\sim}{a_{in}} x_n, i = 1, .., m,$$

con respecto a la i-ésima restricción, es decir, la adecuación entre estos números fuzzy y el correspondiente $\underset{\sim}{b_i}$ con respecto a la i-ésima restricción.

Evidentemente, este problema generaliza los problemas propuestos en la literatura donde aparecen supuestos algunos de los elementos dados de forma imprecisa.

3 Método de resolución del Modelo General de PLF Multiobjetivo

Proponemos un método de resolución para el modelo general que, brevemente, consiste en la sustitución del conjunto de restricciones por un conjunto fuzzy convexo.

Consideramos $A, B \in F(R)$. Un método simple de comparación entre números fuzzy consiste en la definición de una función $g : F(R) \to R$. Si se conoce esta función $g(\cdot)$, entonces

$$\begin{array}{c} g(A) < g(B) \Rightarrow A \; es \; menor \; que \; B \\ g(A) > g(B) \Rightarrow A \; es \; mayor \; que \; B \\ g(A) = g(B) \Rightarrow A \; es \; igual \; que \; B \end{array}$$

Usualmente, g se llama función ordenadora lineal si

a) g(A+B)=g(A)+g(B), $\forall$ A, B$\in F(R)$

b) g(rA)=rg(A), r$\in R$, r$>$ 0, A$\in F(R)$

Utilizando los resultados obtenidos en, [2], obtenemos el siguiente problema auxiliar para resolver (1):

$$\begin{array}{ll} \underset{\sim}{Max} & (z_1, z_2, .., z_k) \\ s.a: & \\ & \sum_{j=1}^{n} \underset{\sim}{a_{ij}} x_j \lesssim_g \underset{\sim}{b_i} + \underset{\sim}{d_i}(1-\alpha) \\ & x_j \geq 0, \alpha \in [0,1], i \in M, j \in N \\ & g \; funcion \; ordenadora \; lineal \end{array} \tag{2}$$

Podemos utilizar para resolver dicho problema las distintas relaciones de comparación de números fuzzy lo que permitirá la obtención de una solución eficiente al problema (1).

3.1 Resolución del Modelo General

En esta sección indicaremos distintas aproximaciones para estudiar una solución del problema general de PLF multiobjetivo. Las distintas aproximaciones utilizaran las funciones ordenadoras lineales para la sustitución del conjunto de restricciones tal y como se ha comentado anteriormente.

La primera aproximación supondrá que los números fuzzy del problema han sido generados por funciones homogeneas lineales fuzzy, [7], [6], y utilizando las funciones ordenadoras lineales se podrá obtener una solución al problema fuzzy. La siguiente aproximación supondrá que los números fuzzy no han sido generados por funciones homogeneas lineales fuzzy.

3.1.1 Funciones homogeneas lineales fuzzy.

Consideramos el problema multiobjetivo de PLF (1) en el que proponiamos el problema auxiliar (2).

Este problema auxiliar (2) puede transformarse en un problema equivalente de PLF multiobjetivo, si los números fuzzy se suponen definidos por una función homogénea lineal fuzzy (f.h.l.f), es decir,

$$\begin{array}{l} \underset{\sim}{Max} \quad (z_1, z_2, .., z_k) \\ s.a: \\ \qquad \sum_{j=1}^{n} f(a_{ij})x_j \leq_g f(b_i) + f(d_i)(1-\alpha) \\ \qquad x_j \geq 0, \alpha \in [0,1], i \in M, j \in N \\ \qquad g \text{ funcion ordenadora lineal} \end{array} \tag{3}$$

donde f:$R \longrightarrow F(R)$ es una f.h.l.f con f(t)=tτ, t$\in R$ y $\tau \in F(R)$, [6], [7].

Teorema 1 *Sea f una f.h.l.f y sea g una función ordenadora lineal. Entonces la solución de (2) es*
i) la solución del problema de PLF multiobjetivo siguiente si $g(\tau) \geq 0$.

$$\begin{array}{l} \underset{\sim}{Max} \quad (z_1, z_2, .., z_k) \\ s.a: \\ \qquad \sum_{j=1}^{n} a_{ij}x_j \leq b_i + d_i(1-\alpha) \\ \qquad x_j \geq 0, \alpha \in [0,1], i \in M, j \in N \end{array} \tag{4.1}$$

ii) la solución del problema de PLF multiobjetivo siguiente si $g(\tau) \leq 0$.

$$\begin{array}{l} \underset{\sim}{Max} \quad (z_1, z_2, .., z_k) \\ s.a: \\ \qquad \sum_{j=1}^{n} a_{ij}x_j \geq b_i + d_i(1-\alpha) \\ \qquad x_j \geq 0, \alpha \in [0,1], i \in M, j \in N \end{array} \tag{4.2}$$

Esta aproximación generaliza a los problemas multiobjetivos con metas fuzzy con/sin restricciones fuzzy.

Si $f(t) = t\tau$ *con* $\tau = \underset{\sim}{1}$ $(\underset{\sim}{1} = (1,1,1,1))$, *obtenemos* :

$$\begin{array}{l} \underset{\sim}{Max} \quad (z_1, z_2, .., z_k) \\ s.a: \\ \qquad \sum_{j=1}^{n} a_{ij}x_j \underset{\sim}{\leq} b_i \\ \qquad x_j \geq 0, i \in M, j \in N \end{array}$$

ó

Si $f(t) = t\tau$ *con* $\tau = \underset{\sim}{1}$, $\alpha \in [0,1]$ $(\underset{\sim}{1} = (1,1,1,1))$, *obtenemos* :

$$\begin{array}{l} \underset{\sim}{Max} \quad (z_1, z_2, .., z_k) \\ s.a: \\ \qquad \sum_{j=1}^{n} a_{ij}x_j \leq b_i \\ \qquad x_j \geq 0, i \in M, j \in N \end{array}$$

3.1.2 Funciones Ordenadoras Lineales

Sea el problema de PLF (1) y supongamos que los números fuzzy que intervienen en él no han sido generados por f.h.l.f. Sea la función ordenadora g lineal. Entonces, para resolver (1) y supuesto el problema auxiliar (2), consideramos el siguiente problema de PL multiobjetivo asociado

$$\begin{array}{l} \underset{\sim}{Max} \quad (z_1, z_2, .., z_k) \\ s.a: \\ \qquad \sum_{j=1}^{n} g(\underset{\sim}{a_{ij}})x_j \leq g(\underset{\sim}{b_i}) + \underset{\sim}{d_i}\,(1-\alpha)) \\ \qquad x_j \geq 0, \alpha \in [0,1], i \in M, j \in N \end{array}$$

que como g es lineal, puede reescribirse de la siguiente forma:

$$\begin{array}{l} \underset{\sim}{Max} \quad (z_1, z_2, .., z_k) \\ s.a: \\ \qquad \sum_{j=1}^{n} g(\underset{\sim}{a_{ij}})x_j \leq g(\underset{\sim}{b_i}) + g(\underset{\sim}{d_i})(1-\alpha) \\ \qquad x_j \geq 0, \alpha \in [0,1], i \in M, j \in N \end{array} \tag{5}$$

Es obvio que con el uso de las distintas funciones ordenadoras que podamos considerar, [13], obtendremos diferentes modelos de programación multiobjetivo.

Esta aproximación generaliza a los problemas multiobjetivo con metas fuzzy con/sin restricciones fuzzy.

Si $g(\underset{\sim}{u}) = u$, *obtenemos el problema* :

$$\begin{array}{l} \underset{\sim}{Max} \quad (z_1, z_2, .., z_k) \\ s.a: \\ \qquad \sum_{j=1}^{n} a_{ij}x_j \underset{\sim}{\leq} b_i \\ \qquad x_j \geq 0, i \in M, j \in N \end{array}$$

ó

Si $g(\underset{\sim}{u}) = u, \alpha \in [0,1]$, *obtenemos el problema* :

$$\begin{aligned} &\underset{\sim}{Max} \quad (z_1, z_2, .., z_k) \\ &s.a: \\ &\qquad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \\ &\qquad x_j \geq 0, i \in M, j \in N \end{aligned}$$

4 Decisión Interactiva en Problemas de Programación Lineal Fuzzy Multiobjetivo

En la literatura especializada pueden encontrarse muchos modelos y métodos que se mantienen con estos problemas. Pero, aunque la PLF es uno de los tópicos más estudiados en el contexto fuzzy, contrariamente a lo que ha ocurrido con los desarrollos teóricos, son muy escasas las aportaciones prácticas orientadas a la implementación software de los resultados logrados en el dominio en el que nos movemos, [9], [5], [4], [8], [2]. Desde este punto de vista se justifica la necesidad de desarrollar más trabajo y esfuerzo en este area.

Proponemos un proceso interactivo para resolver estos problemas de PLF multiobjetivo de una manera práctica y eficiente. Una vez resuelto dicho problema se muestran los resultados al decisor. El decisor puede estar satisfecho con esta solución o puede querer modificar alguna situación y/o cambiar el modelo original. Si satisface, el problema ha sido resuelto. Si no, se procede a un proceso interactivo. Este procedimiento de resolución se continua hasta que el decisor obtenga una solución que le satisfaga.

Una vez que tengamos definidos los números fuzzy y las restricciones fuzzy, y elegida una función ordenadora, calculamos el mínimo valor individual y el máximo valor para cada función objetivo en el conjunto de restricciones, [11]. Es decir, utilizando la información de la siguinte tabla de soluciones

	c_1	c_2	c_3	...	c_k
x^{01}	c_1x^{01}	c_2x^{01}	c_3x^{01}	...	c_kx^{01}
x^{02}	c_1x^{02}	c_2x^{02}	c_3x^{02}	...	c_kx^{02}
...	...	...	...	...	...
x^{0k}	c_1x^{0k}	c_2x^{0k}	c_3x^{0k}	...	c_kx^{0k}
x^{11}	c_1x^{11}	c_2x^{11}	c_3x^{11}	...	c_kx^{11}
x^{12}	c_1x^{12}	c_2x^{12}	c_3x^{12}	...	c_kx^{12}
..	...	...	...	...	...
x^{1k}	c_1x^{1k}	c_2x^{1k}	c_3x^{1k}	...	c_kx^{1k}

el sistema indica las funciones de pertenencia de las funciones objetivo ($x^{01}, .., x^{0k}$, optimos individuales para cada función objetivo con α=0, $x^{11}, .., x^{1k}$, optimos individuales para cada función objetivo con α=1) Los óptimos individuales son la diagonal de la mitad superior. Los valores pesimistas pueden ocurrir o en la mitad superior o inferior.

Con $\overline{c}_l = \sum_j c_{lj} x_j^0$ y $\underline{c}_l = \min\left\{\min_{l'} \sum_j c_{l'j} x_j^0, \min_{l'} \sum_j c_{l'j} x_j^1\right\}$, las funciones de pertenencia estan determinadas con la información sobre la estructura del modelo, [11].

$$\mu_{Z_l}(x) = \begin{cases} 0 & , \sum_j c_{lj} x_j < \underline{c}_l \\ \frac{\sum_j c_{lj} x_j - \underline{c}_l}{\overline{c}_l - \underline{c}_l} & , \underline{c}_l \leq \sum_j c_{lj} x_j \leq \overline{c}_l \\ 1 & , \overline{c}_l \leq \sum_j c_{lj} x_j \end{cases}$$

El decisor toma la cantidad de grado de satisfación en el rango máximo y mínimo valor de cada función objetivo y asignamos la función de pertenencia $\mu_l^0(z_l(x))$. El dominio de definición de la función de pertenencia tiene un rango de entre $\underline{z}_l$ para grado 0 y $\overline{z}_l$ para grado 1 (entre $z_l^{\min}$, $z_l^{\max}$) de satisfación de cada función objetivo.

En el proceso de decisión interactivo, si el decisor no esta de acuerdo o satisfecho con el nivel alcanzado en las funciones objetivo con la solución obtenida, puede cambiar los elementos del modelo paso a paso hasta que le satisfaga la solución obtenida.

Un algoritmo interactivo para encontrar una solución satisfactoria para el decisor puede construirse como sigue:

Algoritmo interactivo para problemas de PLF multiobjetivo *(ver fig. 1)*

1 Formulación del modelo.

2 Decidir las funciones de pertenencia para cada número fuzzy del conjunto de restricciones y de las restricciones fuzzy (Si cada número fuzzy de este conjunto de restricciones ha sido generado por f.h.l.f introducir la función f).

3 Elegir una función ordenadora lineal.

4 Encontrar los valores mínimos y máximos individuales para cada función objetivo.

5 Decidir las funciones de pertenencia para cada función objetivo.

6 Resolver el problema fuzzy multiobjetivo (problemas (4) o (5)).

7 Si satisface la solución actual. Parar. En otro caso, cambiar los distintos parámetros que definen el problema. Ir a Paso 2.

Los pasos 2,3,5,7 son los interactivos donde el decisor ira probando distintos elementos hasta encontrar una solución que le satisfaga.

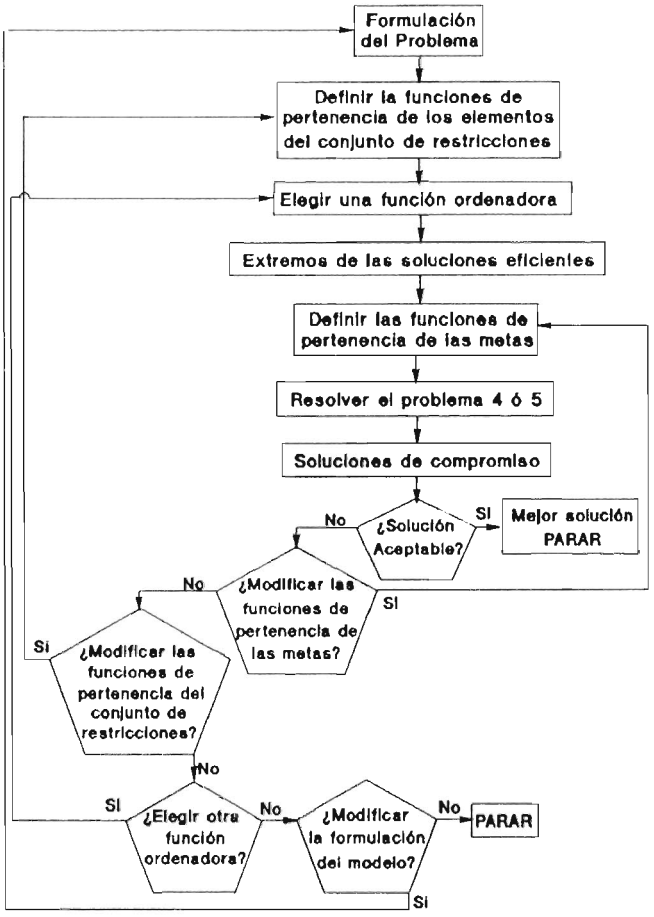


Figure 1. Organigrama del Sistema Interactivo

5 Conclusiones

Hemos dado un método de resolución para el problema general de programación lineal fuzzy Multiobjetivo, siguiendo la decisión de maximinimización de Bellman y Zadeh, ademas hemos dado una aproximación interactiva para resolver estos tipos de problemas multiobjetivo, transformandolos en problemas de programación lineal clásicos.

La interactividad nos sirve para escoger la función apropiada (según el usuario), para transformar el conjunto de restricciones en un conjunto clásico y para definir las funciones de pertenencia de cada elemento impreciso.

La satisfación de las soluciones para el decisor pasa por la adaptación interactiva de los elementos del problema. El método interactivo para resolver problemas de programación lineal fuzzy multiobjetivo es el foco de este trabajo.

Referencias

[1] R. E. Bellman y L. A. Zadeh, Decision Making in a Fuzzy Environment. Management Science 17 (B) 4 (1970) 141-164.

[2] J.M. Cadenas. Diseño e Implementación de un Sistema Interactivo para resolver Problemas de Optimización Fuzzy. Tesis Doctoral (1993). Universidad de Granada.

[3] L. Campos y J.L. Verdegay, Linear Programming Problems and Ranking of Fuzzy Numbers. Fuzzy Sets And Systems, 32 (1989), 1-11.

[4] V. Chankong, Y.Y. Haimes, J. Thadathil and S. Zionts, "Multiple criteria Optimization: A state-of-the art review" in Y.Y. Haimes, V. Chankong (eds.) Decision Making with Multiple Objectives, Proceeding, Cleveland, Ohio, 1984, Springer, Berlin-Heidelberg (1985), 36-90.

[5] S. Chanas. Fuzzy programming in multiobjective linear programming, a parametric approach. Fuzzy Sets and Systems 29 (1989), 303-313. North-Holland.

[6] S. Chanas, D. Kuchta and Z. Nowak, FPLP - A package for Fuzzy and Parametric Linear Programming Problems. In M. Fedrizzi et al. (Eds.). Interactive Fuzzy Optimization and Mathematical Programming. Springer Verlag, (1991) 188-201.

[7] P. Czyzak and R. Slowinski, FLIP: Multiobjective Fuzzy Linear Programming Software with Graphical Facilities. In M. Fedrizzi et al. (Eds.). Interactive Fuzzy Optimization and Mathematical Programming. Springer Verlag, (1991) 168-187.

[8] F. Herrera, M. Kovacs y J.L. Verdegay. Fuzzy Linear programming problems with homogeneous linear fuzzy functions. Presented to IPMU'92.

[9] Y. Lai, C. Hwang, IFLP-II: A decision Support Systems. Fuzzy Sets and Systems 54 (1993) 47-56.

[10] H. Rommelfanger. A PC-Supported Procedure for Solving Multicriteria Linear Programming Problems with Fuzzy Data. In M. Fedrizzi et al. (Eds.). Interactive Fuzzy Optimization and Mathematical Programming. Springer Verlag, (1991) 154-167.

[11] M. Sakawa, Fuzzy interactive multiobjective programming. Japanese Journal of fuzzy theory and Systems. Vol. 4. Número 1 (1992).

[12] B. Werners, Interactive multiple objective programming subject to flexible constraints. European Journal of Operational Research 31 (1987) 342-349. North-Holland.

[13] H J Zimmermann, Fuzzy Programming an Lineal Programming with several objective functions, Fuzzy Sets and Systems 1 (1978) 45-55.

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

Un modelo general de decisión*

María Teresa Lamata
Departamento de Ciencias de la Computación e I.A
Universidad de Granada
e-mail:mtl@robinson.ugr.es

Resumen

El objetivo de este trabajo es resolver el problema de seleccionar una estrategia óptima cuando la información disponible sobre los estados de la naturaleza la conocemos en términos lingüísticos y además es posible que los elementos de la matriz de recompensas sean tambien lingüísticos.

Mostramos nuevos operadores para dar solución a este problema demostrando como los operadores OWA nos proporcionan un buen marco para representar el grado de optimismo del decisor, en el caso de no disponer de información sobre el verdadero estado de la naturaleza.

palabras clave: Toma de decisión. Etiquetas lingüísticas. Incertidumbre. Operadores.

1 Introducción

Un problema de decisión lleva consigo la selección de la mejor alternativa, a la vista de unos resultados y de la información que sobre los estados de la naturaleza hayamos sido capaces de conseguir.

En muchos casos, la información disponible sobre los estados de la naturaleza se obtiene de forma auxiliar, pero normalmente, en el modelo clásico esta información está disponible en términos de una distribución de probabilidad ó bien puede suceder que nos encontremos ante la ignorancia total.

Por otra parte, los pagos que percibe el decisor como consecuencia de su elección que serán numéricos, son función del estado de la naturaleza que se presenta.

Si pensamos que en la vida cotidiana, la forma de expresar una idea es más lingüística que numérica, significará que tanto la información disponible sobre los estados de la naturaleza, como las posibles recompensas las conozcamos en términos lingüísticos.

Así, nosotros supondremos que tanto la información, en el caso de riesgo, como las recompensas puedan ser lingüísticas, con lo cual tendremos que desarrollar distintos operadores que sean capaces de agregar valores numéricos con lingüísticos.

Por ello, primero discutiremos el problema de decisión bajo riesgo, donde la información sobre los estados de la naturaleza es conocida en términos de una distribución de probabilidad pero dada en términos lingüísticos.

Por último, trataremos el caso de decisión bajo ignorancia donde no se dispone de información sobre los estados de la naturaleza y por tanto, para poder actuar es necesario conocer la actitud del decisor frente al riesgo; resolveremos el problema introduciendo los operadores OWA (The Ordered Weighted Averaging) que nos proporcionan un marco mas general que el hasta ahora conocido.

*Este trabajo ha sido soportado por la DGICYT mediante el proyecto PB92-0959

2 El problema de decisión

Consideremos un problema de decisión con los siguientes elementos:

$$\{\Omega, D, \mathbf{r}, I, \leq\}$$

donde:

1. Ω representa el conjunto de *estados de la naturaleza* que supondremos finito

2. D es el conjunto de *alternativas* disponibles para el decisor, que tambien supondremos que es finito.

3. $\mathbf{r}$:$D \times \Omega \rightarrow R$ es una función tal que a cada decisión d_i y a cada estado ω_j hace corresponder un número real

$$(d_i, \omega_j) \longrightarrow \mathbf{r}(d_i, \omega_j) \equiv r_{ij}$$

siendo r_{ij} la recompensa que percibe el decisor por haber elegido la decisión d_i cuando se ha presentado el estado ω_j. $\mathbf{r}$ se puede representar por una matriz que llamamos *Matriz de recompensas*.

$D\backslash^{\Omega}$	w_1	w_2	$\cdots$	w_j	$\cdots$	w_m
A_1	r_{11}	r_{12}	$\cdots$	r_{1j}	$\cdots$	r_{1m}
A_2	r_{21}	r_{22}	$\cdots$	r_{2j}	$\cdots$	r_{2m}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
A_i	r_{i1}	r_{i2}	$\cdots$	r_{ij}	$\cdots$	r_{im}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
A_n	r_{n1}	r_{n2}	$\cdots$	r_{nj}	$\cdots$	r_{nm}

En estas condiciones, la recompensa que podrá obtener el decisor al efectuar una decisión será uno cualquiera de los valores del vector m-dimensional asociado a la alternativa elegida

$$A_i \longrightarrow [r_{i1}, r_{i2}, ..., r_{ij}, ..., r_{im}] \equiv \vec{r_i} \in R^m$$

y el valor concreto de la recompensa dependerá del estado de la naturaleza que se presente.

4. Una relación de preferencias $\leq$ por parte del decisor. Supondremos un decisor coherente y por tanto tratará de maximizar sus ganancias ó de minimizar sus pérdidas.

5. Una cierta información I sobre el verdadero estado de la naturaleza. Dependiendo del tipo de conocimiento que el decisor tiene sobre el verdadero estado , aparecen en la literatura tres ambientes.

   a) *Ambiente de certidumbre.* El decisor conoce cual va a ser el estado en presentarse. En este caso, el decisor debe seleccionar aquella alternativa que suponga el máximo pago/mínima pérdida para ese estado de la naturaleza.

   b) *Ambiente de riesgo.* En la decisión bajo riesgo, se supone conocida una distribución de probabilidad sobre los estados. En este caso el procedimiento standard es usar el valor esperado:
   Algoritmo de decisión

      i. Para cada alternativa A_i, calculamos su esperanza

      $$E_i = r_{ij} \times p_j$$

      ii. Seleccionamos como óptima aquella decisión cuyo valor esperado sea mayor

      $$E* = Max_i E_i$$

   c) *Ambiente de incertidumbre.* El tercer ambiente corresponde al caso de ignorancia. Al no disponer de información sobre los estados de la naturaleza, los criterios de decisión pasan por tener en cuenta la actitud del decisor frente al riesgo. Entre estas actitudes, las mas representativas son las siguientes:

      - *Actitud pesimista.* Usando esta estrategia el decisor se pone en el peor de los casos; selecciona para cada alternativa el peor resultado y despues selecciona la que consigue el máximo. Tambien se conoce esta estrategia como *Maximin*.

      - *Actitud optimista.* Bajo esta estrategia, el decisor presupone el mejor resultado posible; seleccionando para cada alternativa el mejor resultado posible y eligiendo la alternativa que consiga el máximo. Esta estrategia es conocida como *Maximax*

      - *Criterio de Hurwicz.* En este enfoque el decisor selecciona un valor $\alpha \in [0, 1]$ que representa su grado de optimismo. Para cada alternativa, lo que hacemos es ponderar los valores máximo y mínimo por α y $(1 - \alpha)$ respectivamente.

        $$H_i = \alpha \times Max_i r_{ij} + (1 - \alpha) \times Min_i r_{ij}$$

        Finalmente, se elige la altermativa que consigue el máximo.

      - *Criterio Normativo.* En este enfoque , el decisor suma todos los valores relativos a cada una de las alternativas para al final elegir aquella cuya suma sea la mayor. Equivale a suponer una distribución equiprobable sobre Ω.
        En este caso el **algoritmo de decisión** sería el siguiente:

        i. Para cada alternativa calculamos una función valor $V_i = F(r_{i1}, r_{i2}, .., r_{im})$, donde F es cualquier función de agregación de las que se exponen a continuación.

        ii. Seleccionamos aquella alternativa $A*$ para la cual se consiga el máximo $A* = Max_i Vi$.

ESTRATEGIA	*FUNCION DE AGREGACION*
Pesimista	$F = \text{Min}_j r_{ij}$
Optimista	$F = \text{Max}_j r_{ij}$
Hurwicz	$F = \alpha \times \text{Max}_j r_{ij} + (1 - \alpha) \times \text{Min} r_{ij}$
Normativo	$F = \Sigma_j r_{ij}$

A partir de ahora nosotros supondremos que tanto las recompensas r_{ij} como la información I sobre los estados de la naturaleza pueden estar expresadas en términos lingüísticos. Esto llevará consigo algunos problemas como el tener que agregar etiquetas con valores numéricos. Es por ello, que tendremos que introducir nuevos operadores similares a los que aparecen en la literatura y que sean capaces de resolvernos el problema que apuntamos. Para ello empezaremos definiendo el concepto de

3 Conjunto de etiquetas generalizado

Sea $L = \{0, l_1, l_2, ..., l_n\}$ el conjunto ordenado de etiquetas lingüísticas. Es decir, $L = \{l_i / i \in I = 1, .., n\}$ y

tal que $l_i < l_j$ si $i < j$.

Consideremos otro conjunto $P = \{0 = p_0, p_1, ..., p_m\}$ con $p_j \in [0,1]$. En el caso que nos ocupa los p_j representan la distribución de probabilidad sobre Ω, y por lo tanto, el conjunto de posibles ponderaciones de las distintas etiquetas.

Con base en estos conjuntos que pueden ser lingüísticos o numéricos, definimos el conjunto de etiquetas generalizadas [4].

$$\mathcal{L} = L \times P$$

donde el par $(l_i, p_j) \in \mathcal{L}$ representa el producto de $p_j \odot l_i$.

- Si $p_j = 1 \Rightarrow (l_i, 1) \equiv l_i$
- Si $p_j = 0 \Rightarrow (l_i, 0) \equiv 0$.

Con lo cual, la etiqueta que representa el valor nulo, tiene que pertenecer al conjunto de etiquetas L.

3.1 Adición

Sean (l_{i1}, p_{j1}) y (l_{i2}, p_{j2}) dos etiquetas generalizadas; definimos su suma como:

$$\oplus : \mathcal{L}^2 \longrightarrow \mathcal{L}$$

$$(l_{i1}, p_{j1}) \oplus (l_{i2}, p_{j2}) = (l_s, p_{i1} + p_{j2}) \quad (1)$$

donde

$$s = \frac{(i1 \times p_{j1}) + (i2 \times p_{j2})}{p_{j1} + p_{j2}}$$

ahora bien, este valor s no tiene porque ser un número entero, con lo cual la etiqueta l_s no siempre será un elemento de L y por tanto, es necesario hacer una aproximación a la etiqueta mas próxima, siendo lo mas sencillo un redondeo. Si tenemos que sumar varias etiquetas generalizadas, deberemos extender (1) en la forma siguiente:

$$(l_{i1}, p_{j1}) \oplus (l_{i2}, p_{j2}) \oplus ..... \oplus (l_{1n}, p_{jn}) = (l_s, \overset{n}{\Sigma}_{k=1}\ p_{jk})$$

siendo

$$s = \text{redondeo} \frac{\overset{n}{\Sigma}_{k=1}\ (ik \times p_{jk})}{\overset{n}{\Sigma}_{k=1}\ p_{jk}} \quad (2)$$

Propiedades

1. commutativa. Al sumar dos etiquetas generalizadas el resultado no depende del orden en el que este se efectue.
2. asociativa. Por la forma en que se ha definido la operación en (2).
3. elemento neutro. Será cualquier etiqueta de la forma
$$(l_i, 0) \forall i = 0, 1, .., n$$

4 Decisión bajo riesgo

Cuando nos encontramos en un ambiente de riesgo, suponemos que sobre los estados de la naturaleza disponemos de una distribución de probabilidad que nos informa de la incertidumbre asociada a la experiencia , en este caso, el operador clásico que para cada decisión pondera las distintas recompensas es la Esperanza Matemática. Para el problema que nos ocupa, en el que ó bien las recompensas ó las probabilidades pueden ser lingüísticas proponemos distintos operadores.

4.1 Si el conjunto de probabilidades es lingüístico y los pagos numéricos

. Consideramos $P = \{p_k / k = 0, 1, .., m\}$ un conjunto finito y ordenado de etiquetas lingüísticas; definimos el orden de cada etiqueta probabilistica por $O(p_k) = k$ Entonces la esperanza matemática será la suma de m etiquetas generalizadas

$$E : (R \times P)^m \longrightarrow R$$

$$\overset{m}{\Sigma}_{j=1}\ (u_{ij} \odot p_j) \longrightarrow U_i$$

siendo

$$E_1[A_i] = U_i$$

donde

$$U_i = \frac{\Sigma_j u_{ij} \times O(p_j)}{\Sigma_j O(p_j)}$$

y donde $\Sigma_j O(p_j)$ es el factor de normalización y R el cuerpo de los números reales.

La decisión óptima $A*$ será aquella para la cual se verifique que $E[A*] = \text{Max}_i U_i$ y diremos que la acción A_i es preferida a la A_j si y solo si $U_i > U_j$.

4.2 Cuando el conjunto de probabilidades es numérico y los pagos lingüísticos

Considerando $L = \{l_i / i = 1, .., n\}$ el conjunto finito ordenado de las posibles etiquetas lingüísticas, y puesto que cada utilidad u_{ij} es una etiqueta $l_j \in L$, definimos al igual que en el apartado anterior su orden por $O(u_{ij}) = O(l_j) = j$.
Entonces

$$E_2[A_i] = l_{s_i}$$

donde

$$s_i = \text{redondeo}(U_i)$$

y

$$U_i = \Sigma_j O(l_j) \times p_j$$

Al ser el conjunto L finito y normalmente de pequeña granularidad, supone que es muy probable que varias decisiones consigan la misma esperanza, en cuyo caso, para adoptar una decisión se debería elegir aquella acción para la cual se consiga el $Max_i U_i$

4.3 Ambos conjuntos son lingüísticos

La primera opción consistiría en calcular las tablas para la adición y el producto para posteriormente operar sobre ellas; pero esto tiene muchos problemas como se expone en [8].

Por ello, una via alternativa es construir un operador relacionado con los ordenes de las respectivas etiquetas.
De esta forma

$$E_3[A_i] = l_{si}$$

donde

$$s = \text{redondeo}(u_i)$$

y

$$u_i = \frac{\Sigma_j O(l_j) \times O(p_j)}{\Sigma_j O(p_j)}$$

Al igual que en el primer caso $\Sigma_j O(p_j)$ es el factor de normalización y la decisión óptima será aquella acción que consiga el máximo sobre L y en caso de no ser este valor único, tendriamos que hallar el $Max_i u_i$.

5 Decisión bajo ignorancia

Al no disponer de información sobre los estados de la naturaleza, y supuesto que el criterio de dominancia no nos conduce a ningún resultado, es el decisor quien debe decidir teniendo en cuenta alguna regla subjetiva.

Supongamos que las utilidades son lingüísticas. En este caso, las estrategias clásicas de decisión ***Maximax*** y ***Minimax*** no son las mas apropiadas ya que, al ser el conjunto L finito y de poca granularidad, es muy probable que dos alternativas alcancen el mismo máximo y/o mínimo, por ello tambien obviamos el criterio de Hurwicz, pareciendonos mas apropiado introducir los operadores OWA [11].

5.1 Operadores OWA

Definición (Yager). *An ordered weighted averaging operator (OWA) de dimension n is una función*

$$F : R^n \longrightarrow R$$

que tiene asociada con ella un vector de pesos w

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

tal que
1) $w_i \in [0, 1]$

2)$\Sigma_i w_i = 1$

donde para cualquier conjunto de valores $a_1, .., a_n$, *se verifica:*

$$F(a_1, \cdots a_n) = \Sigma_i \omega_i \times b_i$$

donde b_i *es el valor i-ésimo valor mayor de* $a_1, .., a_n$.

Para cualquier coeficiente de optimismo, el operador F que lo representa siempre está comprendido entre los operadores extremos máximo y mínimo.

$$Min(a_1, ..., a_m) \leq F(a_1, ..., a_m) \leq Max(a_1, ..., a_m)$$

Teniendo en cuenta la actitud del decisor, reproducimos los criterios clásicos de decisión bajo ignorancia cuando las recompensas son lingüísticas y operamos con el concepto de etiqueta generalizada introducido anteriormente así como de los operadores OWA.

1) *Actitud Pesimista*: Para ello seleccionamos w_*

$$w_* = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

entonces

$$F(a_1, a_2, ..., a_m) = Min_j[a_j] = l_{*j} \in L$$

que es la regla de agregación usada en la estrategia pesimísta

2) *Actitud Optimísta*: Seleccionamos w^*

$$w^* = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

entonces

$$F(a_1, a_2, ..., a_m) = Max_j[a_j] = l_j^* \in L)$$

, con lo que obtenemos el operador propio de la estrategia optimísta.

3) *Estrategia de Hurwicz*: Seleccionamos

$$w_H = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \\ (1-\alpha) \end{bmatrix}$$

entonces

$$\begin{aligned} F(a_1, .., a_m) &= \alpha \times Max[a_j] + (1-\alpha) \times Min[a_j] \\ &= \alpha \times l_{*j} + (1-\alpha) \times l_{*j} = l_j \in L \end{aligned}$$

Siendo el operador usado en la estrategia de Hurwicz.

4) *Estrategia de Hurwicz Generalizada*. Seleccionamos

$$w_{HG} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$$

con $\Sigma\alpha = 1$ y cuyos valores α_i son el resultado del problema de programación lineal propuesto por O'Hagan

Maximizar:

$$\Sigma_j w_j ln(w_j)(entropia)$$

sujeto a:

$$\Sigma_j(h_n(j) \times w_j) = \beta$$

$$\Sigma_j w_j = 1$$

$$w_j \leq 0 \text{ para todo } j = 1, .., m$$

donde β es el grado de optimísmo del decisor y $h_n(j) = (n-j)/(n-1)$.

5) *Enfoque Normativo*: Si seleccionamos

$$w_N = \begin{bmatrix} 1/n \\ 1/n \\ \vdots \\ 1/n \end{bmatrix}$$

entonces $F(a_1, a_2, ..., a_m) = 1/n\Sigma_j a_j = l_j \in L$. Esta función es esencialmente la usada en el criterio normativo.

6 Conclusión

En este trabajo, proponemos un método para poder agregar valores numéricos ó lingüísticos, incorporando distintos operadores aplicables al problema de decisión. Se demuestra como estos operadores cumplen las propiedades que son mas deseables para la toma de decisiones.

Bibliografía

[1] R.Bellman and L.A.Zadeh, "Decision making in a fuzzy environment". *Management Science, 17, 1970, pp. 141-164.*

[2] M.J.Bolaños, M.T.Lamata and S.Moral, "Decision making problems in a general environment". *Fuzzy Sets and Systems*, 18, 1988, pp. 135-144.

[3] P.P.Bonissone, "A fuzzy sets based linguistic approach". *Approximate Reasoning in Decision Analysis*, M.M:Gupta, E.Sanchez (eds), North Holland Publishing Company, 1982, pp. 329-339.

[4] M.Delgado, J.L.Verdegay and A.Vila, "On aggregation operations of linguistic labels". *International Journal of Intelligent Systems*, 8, 1993, pp. 351-370.

[5] P.C.Fishburn, *"Utility theory for decision making"* John Wiley 1970.

[6] M.T.Lamata, "Problemas de decisión con informacion general. *Tesis doctoral.* Universidad de Granada, 1986.

[7] M.T.Lamata and S.Moral, "Decisión; Utilidades lingüísticas". III *Congreso Español sobre tecnologías y lógica fuzzy*, 1993, pp. 321-328.

[8] M.T.Lamata and S.Moral, "Calculus with linguistic probabilities and beliefs". *Advances in the Dempster-Shafer theory of evidence*, R.R.Yager, M.Fedrizzi, J,Kacprzyk (eds), John Wiley,1994, pp. 133-152.

[9] M.O'Hagan, "Aggregating template rule antecedents in real- time expert systems with fuzzy set logic". *Int. J. of Man-Machine Studies*, (to appear).

[10] J.Q.Smith, *Decision Analysis. A Bayesian Approach*,Chapman & Hall.1988.

[11] R.R.Yager, "Decision making under Dempster-Shafer uncertainties". 20, 1992, pp. 233-245.

[12] R.R.Yager, "Families of OWA operators". *Fuzzy Sets and Systems*,59, 1993, pp. 125-148.

[13] L.A.Zadeh, "Fuzzy sets and information granularity" *Advances in fuzzy sets theory and applications*, M.M.Gupta, E.Sanchez (eds), North Holland Publishing Company, New York, 1979, pp. 3-19.

Ordenación de números borrosos mediante la comparación de sus intervalos esperados

Mariano Jiménez

Departamento de Economía Aplicada I
E. U. EE. Empresariales. Universidad del País Vasco
Paseo de Vizcaya nº22.
20010 San Sebastián, España
e-mail: eupjilom@se.ehu.es

Resumen

En este trabajo se propone un método de ordenación de números borrosos, basado en la comparación de los intervalos esperados de dichos números. Esta relación de preferencia es borrosa y verifica las cualidades de distinguibilidad, racionalidad y robustez. Se extiende la noción de intervalo esperado al caso de números borrosos no normales, con lo que el método propuesto permite comparar también este tipo de conjuntos.

Palabras clave: Fuzzy number; nonnormal fuzzy set; expected interval; fuzzy preference relation; fuzzy ranking.

1 Introducción.

Cuando se trabaja en un ambiente de incertidumbre, y se utilizan números para delimitar los posibles valores que pueden tomar las variables que intervienen en un modelo de decisión, el resultado esperado de las acciones a emprender será la solución de una ecuación borrosa , la cual a su vez también es un número borroso [5]. Así pues, el agente decisor se ve obligado a elegir entre distintos números borrosos .

Diversos autores han propuesto métodos distintos para ordenar números borrosos, sin que se haya encontrado una respuesta enteramente satisfactoria, ya que en muchas ocasiones dichos métodos no proporcionan el mismo resultado. Esto hace que los diferentes autores, para justificar la bondad de su método, utilicen diversas propiedades o cualidades que se piensa debe verificar todo sistema de ordenación de números borrosos. Las cualidades más mencionadas son: expresión borrosa o lingüística de la relación de preferencia [6]; distinguibilidad [2]; racionalidad [15]; robustez [16].

En este trabajo se propone un método de ordenación de números borrosos que verifica las anteriores propiedades, y que además puede extenderse al caso de que deban ordenarse también subconjuntos convexos no normales.

2 Intervalo esperado de un número borroso.

Definamos en primer lugar algunos conceptos que utilizaremos a lo largo de este trabajo:

Definición 1. *Un número borroso es un subconjunto borroso normal y convexo de la recta real, cuya función de pertenencia es semicontinua superiormente.*

De acuerdo con esta definición podemos describir la función de pertenencia de un número borroso de la siguiente manera:

$$\mu_A(x)=\begin{cases}0 & \forall x\in(-\infty,a_1]\\ f_A(x) & \text{creciente en } [a_1,a_2]\\ 1 & \forall x\in[a_2,a_3]\\ g_A(x) & \text{decreciente en } [a_3,a_4]\\ 0 & \forall x\in[a_4,\infty)\end{cases}$$

a $f_A(x)$ y $g_A(x)$ les llamaremos parte izquierda y derecha de la función de pertenencia respectivamente. Con el fin de asegurarnos la existencia e integrabilidad de sus respectivas funciones inversas: $f_A^{-1}(x)$ y $g_A^{-1}(x)$, supondremos que $f_A(x)$ es continua y estrictamente creciente y que $g_A(x)$ es continua y estrictamente decreciente. Podemos denotar un número borroso por $[a_1, a_2, a_3, a_4; 1]$

Definición 2. *Un intervalo aleatorio S, es una aplicación medible del espacio probabilístico* (Ω, A, Pr) *en la familia de todos los intervalos cerrados de la recta real .* [9, 10]:

$$S(\omega) = [s_1(\omega), s_2(\omega)]$$

donde $\omega \in \Omega$ *y* $s_1(\omega) \le s_2(\omega) \ \forall \omega \in \Omega$.

Un intervalo aleatorio es una variable aleatoria que toma valores en R^2, es decir esta compuesto por dos variables reales aleatorias s_1, s_2, cuyas funciones de distribución denotaremos $F_1(x)$, $F_2(y)$, respectivamente.

Definición 3. *Llamaremos valor esperado E(S) de un intervalo aleatorio S, al intervalo cuyos extremos son los valores esperados de las variables aleatorias* s_1 *y* s_2 [11].

Es decir:

$$E(S) = [E_1, E_2] = \left[\int_{-\infty}^{+\infty} x\, dF_1(x), \int_{-\infty}^{+\infty} x\, dF_2(x) \right] \quad (1)$$

Un número borroso $\tilde{A}$, induce una clase $RI(\tilde{A})$ de intervalos aleatorios [8], que satisfacen la siguiente condición:

$$Pr(x \in S) = \mu_A(x), \qquad S \in RI(\tilde{A})$$

El soporte de un intervalo aleatorio S, al que representaremos por $sop(S)$, es el más pequeño conjunto que satisface la condición $Pr(S \subset sop(S)) = 1$. Es fácil comprobar que el rectángulo $[a_1, a_2] \times [a_3, a_4]$ contiene al soporte de un intervalo aleatorio S inducido por el número borroso $\tilde{A} = [a_1, a_2, a_3, a_4; 1]$, [9]. Entonces el valor esperado de S (véase (1)) puede escribirse:

$$E(S) = [E_1, E_2] = \left[\int_{a_1}^{a_2} x\, dF_1(x), \int_{a_3}^{a_4} x\, dF_2(x) \right] \quad (2)$$

Definición 4. *El intervalo esperado, de un número borroso es el valor esperado de un intervalo aleatorio* $S \in RI(\tilde{A})$ *y lo denotaremos por* $EI(\tilde{A})$, [11].

Es fácil demostrar [11] que el valor esperado de todos los miembros de $RI(\tilde{A})$ es el mismo, y que además :

$$F_1(x) = f_A(x), \quad \text{para} \quad a_1 \le x \le a_2$$
$$F_2(x) = 1 - g_A(x), \quad \text{para} \quad a_3 \le x \le a_4 \quad (3)$$

Por lo tanto, después de (2), podemos escribir:

$$EI(\tilde{A}) = [E_1^A, E_2^A] = \left[\int_{a_1}^{a_2} x\, df_A(x), -\int_{a_3}^{a_4} x\, dg_A(x) \right]$$

de donde integrando por partes y haciendo los cambios de variables $\alpha = f_A(x)$, $\alpha = g_A(x)$:

$$EI(\tilde{A}) = [E_1^A, E_2^A] = \left[\int_0^1 f_A^{-1}(\alpha)\, d\alpha, \int_0^1 g_A^{-1}(\alpha)\, d\alpha \right] \quad (4)$$

Heilpern [11], demuestra que el concepto de intervalo esperado conduce al mismo resultado que el concepto de "valor medio" de Dubois y Prade [7]. Por otro lado la expresión (4), nos dice que los extremos inferior y superior del intervalo esperado son respectívamente lo que Liou y Wang [14] llaman valor integral izquierdo y valor integral derecho de un número borroso.

De (4) se deduce de forma inmediata que si si f_A y g_A son lineales, es decir si el número borroso $\tilde{A}$ es triangular o trapezoidal, su intervalo esperado es:

$$EI(\tilde{A}) = \left[\frac{1}{2}(a_1 + a_2), \frac{1}{2}(a_3 + a_4) \right]$$

3 Un método de ordenación de números borrosos basado en la comparación de sus intervalos esperados.

Sean dos números borrosos $\tilde{A}$ y $\tilde{B}$.

Lema 1. *El intervalo esperado de* $\tilde{A} - \tilde{B}$ *es*:

$$EI(\tilde{A} - \tilde{B}) = [E_1^A - E_2^B, E_2^A - E_1^B] = EI(\tilde{A}) - EI(\tilde{B})$$

Demostración. Por ser la operación diferencia creciente respecto de A y decreciente respecto de B, los α – cortes: $(A - B)_\alpha$, de $\tilde{A} - \tilde{B}$, se calculan así [4]:

$$(A - B)_\alpha = [f_A^{-1}(a) - g_B^{-1}(a), \ g_A^{-1}(a) - f_B^{-1}(a)]$$

entonces, de acuerdo con (1), y por aritmética de intervalos, el intervalo esperado de $\tilde{A} - \tilde{B}$, es:

$$EI(\tilde{A} - \tilde{B}) =$$
$$= \left[\int_0^1 (f_A^{-1}(\alpha) - g_B^{-1}(\alpha))d\alpha, \ \int_0^1 (g_A^{-1}(\alpha) - f_B^{-1}(\alpha))d\alpha \right] =$$
$$= [E_1^A - E_2^B, E_2^A - E_1^B] = [E_1^A, E_2^A] - [E_1^B, E_2^B]$$

Definición 5. *dados cualquier par de números borrosos* $\tilde{A}$, $\tilde{B}$, *definimos la relacion de preferencia borrosa* $M(\tilde{A}, \tilde{B})$, *por medio de la siguiente función de pertenencia:*

$$\mu_M(\tilde{A},\tilde{B}) = \begin{cases} 0 & E_2^A - E_1^B < 0 \\ \dfrac{E_2^A - E_1^B}{E_2^A - E_1^B - (E_1^A - E_2^B)}; & 0 \in [E_1^A - E_2^B,\ E_2^A - E_1^B] \\ 1 & E_1^A - E_2^B > 0 \end{cases} \tag{5}$$

representando $\mu_M(\tilde{A}, \tilde{B})$ el grado de preferencia de $\tilde{A}$ sobre $\tilde{B}$. Obsérvese que $\mu_M(\tilde{A}, \tilde{B})$ es el cociente entre la parte positiva del intervalo diferencia de los intervalos esperados de los dos números borrosos que estemos comparando y la longitud total de dicho intervalo diferencia. También, después del lema 1, puede interpretarse como una comparación de $\tilde{A} - \tilde{B}$ con cero, ya que nos da la proporción en que el intervalo esperado de $\tilde{A} - \tilde{B}$ es mayor que cero.

3.1 Racionalidad.

Cuando se trabaja en lógica binaria toda relación de orden debe verificar las propiedades: reflexiva, antisimétrica y transitiva. Cuando se trabaja con lógica borrosa, ésta debe ser trasladada también a dichas propiedades [17], [13 p. 105].

Nakamura [15] , introduce los adjetivos "fuerte", "débil" y "moderado" para indicar el grado en que una relación borrosa P verifica una determinada propiedad. De entre las propiedades expresadas por Nakamura destacamos las siguientes:

a) Reflexividad débil : $\mu_P(\tilde{A}, \tilde{A}) \geq \frac{1}{2}$;

b) Antisimetría moderada : $\mu_P(\tilde{A}, \tilde{B}) + \mu_P(\tilde{B}, \tilde{A}) \leq 1$;

c)Transitividad moderada:

$$\mu_P(\tilde{A},\tilde{B}) \wedge \mu_P(\tilde{B},\tilde{C}) \geq \frac{1}{2} \Rightarrow \mu_P(\tilde{A},\tilde{C}) \geq \mu_P(\tilde{A},\tilde{B}) \wedge \mu_P(\tilde{B},\tilde{C})$$

d) Comparabilidad moderada: $\mu_P(\tilde{A}, \tilde{B}) + \mu_P(\tilde{B}, \tilde{A}) \geq 1$

e) Reciprocidad : $\mu_P(\tilde{A}, \tilde{B}) + \mu_P(\tilde{B}, \tilde{A}) = 1$.

$\mu_P(\tilde{A}, \tilde{B})$, representa el grado de preferencia de $\tilde{A}$ sobre $\tilde{B}$, entonces : "puesto que la preferencia de $\tilde{A}$ sobre $\tilde{B}$ es opuesta a la preferencia de $\tilde{B}$ sobre $\tilde{A}$, parece natural exigir que $\mu_P(\tilde{A}, \tilde{B}) = 1 - \mu_P(\tilde{B}, \tilde{A})$", [16]. Es decir que una relación de preferencia borrosa P, debe ser reciproca. Es evidente que una relación borrosa es recíproca si y solo si es moderadamente antisimétrica y moderadamente comparable. También resulta evidente que la reciprocidad implica que $\mu_P(\tilde{A}, \tilde{A}) = 1/2$, es decir que se verifica la reflexividad débil. Así pues sólo falta la transitividad para completar las propiedades que debe verificar una relación de orden borroso. Luego podemos dar la siguiente definición:

Definición 6. *Si una relación borrosa es recíproca y moderadamente transitiva, es un orden total borroso* [15].

Teorema 1. *La relación de preferencia borrosa M (véase definición 2), es un orden total borroso.*

Demostración. 1) M es reciproca: es evidente por su definición (véase expresión (5)).

2) M es moderadamente transitiva: En efecto, supongamos que:

$$\mu_M(\tilde{A},\tilde{B}) = \frac{E_2^A - E_1^B}{E_2^A - E_1^B - (E_1^A - E_2^B)} = p \geq \frac{1}{2}$$

$$\mu_M(\tilde{B},\tilde{C}) = \frac{E_2^B - E_1^C}{E_2^B - E_1^C - (E_1^B - E_2^C)} = q \geq \frac{1}{2} \tag{6}$$

supongamos también que $min\{p, q\} = q$. Veamos que ocurre con:

$$\mu_M(\tilde{A},\tilde{C}) = \frac{E_2^A - E_1^C}{E_2^A - E_1^C - (E_1^A - E_2^C)}$$

de acuerdo con (6):

$$E_2^A - E_1^B = p \cdot (E_2^A - E_1^B - (E_1^A - E_2^B))$$

$$E_2^B - E_1^C = q \cdot (E_2^B - E_1^C - (E_1^B - E_2^C))$$

luego:

$$E_2^A - E_1^B + E_2^B - E_1^C \geq$$

$$\geq q \cdot [E_2^A - E_1^B - (E_1^A - E_2^B) + E_2^B - E_1^C - (E_1^B - E_2^C)] =$$

$$= q \cdot [E_2^A - E_1^C - (E_1^A - E_2^C) + 2(E_2^B - E_1^B)]$$

de donde:

$$E_2^A - E_1^C \geq q \cdot [E_2^A - E_1^C - (E_1^A - E_2^C)] + (2q - 1)(E_2^B - E_1^B)$$

pero como $q \geq \frac{1}{2}$, se verifica que $(2q-1) \geq 0$, por lo tanto

$$E_2^A - E_1^C \geq q \cdot \left[E_2^A - E_1^C - \left(E_1^A - E_2^C\right)\right]$$

así pues, podemos concluir que

$$\mu_M(\tilde{A}, \tilde{C}) \geq q = min\left\{\mu_M(\tilde{A}, \tilde{B}), \mu_M(\tilde{B}, \tilde{C})\right\}.$$

3.2 Robustez.

Yuan[16] introduce un nuevo criterio para estudiar la validez de un método de comparación de números borrosos: su robustez. Es sabido que normalmente no se puede determinar con precisión la forma de la función de pertenencia, por ello parece conveniente que el orden que proporcione una determinada relación de preferencia, no cambie significativamente ante cambios suficientemente pequeños en la función de pertenencia.

Definición 7. *Dados dos números borrosos $\tilde{A}$ y $\tilde{A}'$, llamaremos máxima desviación entre $\tilde{A}$ y $\tilde{A}'$ a la siguiente expresión :*

$$D\left(\tilde{A}, \tilde{A}'\right) = \max_{\alpha \in (0,1]}\left\{\left|f_A^{-1}(\alpha) - f_{A'}^{-1}(\alpha)\right|, \left|g_A^{-1}(\alpha) - g_{A'}^{-1}(\alpha)\right|\right\}$$

De forma que si $\tilde{A}'$ es una aproximación de $\tilde{A}$, $D\left(\tilde{A}, \tilde{A}'\right)$ permite evaluar la validez de dicha aproximación [12, pp. 51-53].

Definición 8. *Se dice que una relación de preferencia P es robusta si y sólo si se verifica que:*

$$\forall \varepsilon > 0 \ \exists \delta / \forall \tilde{A}', D\left(\tilde{A}, \tilde{A}'\right) < \delta \Rightarrow$$
$$\Rightarrow \left|\mu_P(\tilde{A}, \tilde{B}) - \mu_P(\tilde{A}', \tilde{B})\right| < \varepsilon$$

Teorema 2. *La relación de preferencia M (véase definición 5) es robusta.*

Demostración. Supongamos que $\tilde{A}'$ es una aproximación de $\tilde{A}$. Las relaciones de preferencia de ambos números borrosos respecto de un tercer subconjunto $\tilde{B}$, son:

$$\mu_M\left(\tilde{A}, \tilde{B}\right) = \frac{E_2^A - E_1^B}{E_2^A - E_1^B - \left(E_1^A - E_2^B\right)}$$

$$\mu_M\left(\tilde{A}', \tilde{B}\right) = \frac{E_2^{A'} - E_1^B}{E_2^{A'} - E_1^B - \left(E_1^{A'} - E_2^B\right)}$$

pero, para los numeradores encontramos que:

$$\left|E_2^A - E_1^B - \left(E_2^{A'} - E_1^B\right)\right| = \left|\int_0^1 \left(g_A^{-1}(a) - g_{A'}^{-1}(a)\right) d\alpha\right| \leq$$

$$\leq \int_0^1 \left|g_A^{-1}(a) - g_{A'}^{-1}(a)\right| d\alpha \leq D$$

de igual forma para los denominadores:

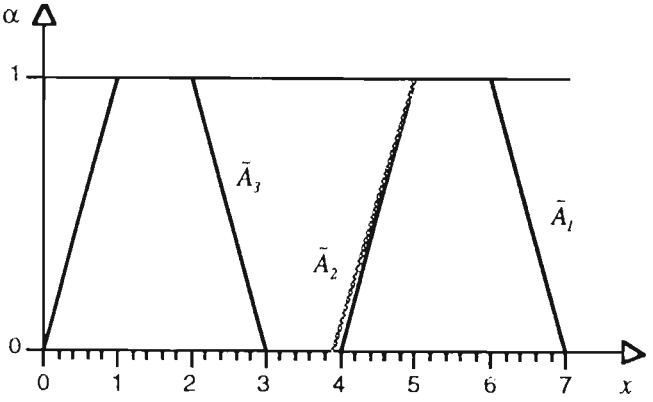
$$\left|E_2^A - E_1^B - \left(E_1^A - E_2^B\right) - \left(E_2^{A'} - E_1^B - \left(E_1^{A'} - E_2^B\right)\right)\right| \leq 2D$$

luego dado $\varepsilon > 0$, siempre será posible encontrar un $\delta > 0$ suficientemente pequeño tal que si $D < \delta$, entonces,

$$\left|\mu_M(\tilde{A}, \tilde{B}) - \mu_M(\tilde{A}', \tilde{B})\right| < \varepsilon.$$

3.3 Distinguibilidad.

La distinguibilidad de un método de ordenación de números borrosos, se refiere a la capacidad para distinguir o diferenciar números borrosos, en términos de grado de preferencia, cuando las diferencias entre esos números son interesantes para el decisor, [16]. Así si consideramos el ejemplo de la figura:



el método de Nakamura [15] no distingue entre los grados de preferencia de $\tilde{A}_1$=(4, 5, 6, 7;1) sobre $\tilde{A}_2$=(3'9, 5, 6 , 7; 1) y sobre $\tilde{A}_3$=(0, 1, 2, 3; 1), en ambos casos es 1, sin embargo lo razonable es que el grado de preferencia de $\tilde{A}_1$ sobre $\tilde{A}_2$ sea cercano a 1/2, es decir que sean casi indiferentes, esto ocurre porque el método de Nakamura no es robusto. Lo mismo pasa con el método de Baas y Kwakernak [1]. Sin embargo el método propuesto eneste trabajo proporciona los siguientes resultados:

$$\mu_M(\tilde{A}_1, \tilde{A}_2) = 0,506, \quad \mu_M(\tilde{A}_1, \tilde{A}_3) = 1$$

Al comparar dos números borroso $\tilde{A}$ y $\tilde{B}$, el método M propuesto en este trabajo, al igual que el propuesto por Yuan [16], relaciona los mejores resultados de $\tilde{A}$ con los peores de $\tilde{B}$, y los peores resultados de $\tilde{A}$ con los mejores de $\tilde{B}$. Por lo tanto parece indicado para comparar resultados de acciones en los que la obtención de los mejores resultados en una presuponga que en la otra se obtienen los peores, o que simplemente no se pueda presuponer nada en este sentido, en cuyo caso adoptamos la posición más prudente. Sin embargo el método de Nakamura relaciona los mejores resultados de $\tilde{A}$ con los mejores resultados de $\tilde{B}$ y los peores de $\tilde{A}$ con los peores de $\tilde{B}$, por ello puede ser utilizado de forma complementaria dependiendo del contexto del problema y de la actitud del decisor.

En la práctica los resultados proporcionados por el método de Yuan son iguales o prácticamente iguales a los obtenidos por el método que se propone en este trabajo, pero el cálculo de éste es mucho más sencillo y además como veremos más adelante permite introducir en la comparación números no normales. Es fácil demostrar [12 pp. 99-101] que la clasificación proporcionada por estos tres métodos (M, Yuan y Nakamura) es la misma, aunque lógicamente el grado de preferencia no tiene porqué ser igual.

3.4.- Expresión borrosa o lingüística de la relación de preferencia.

Cuando se trabaja con lógica borrosa no debemos conformarnos simplemente con concluir que $\tilde{A}$ es preferido a $\tilde{B}$, sino que es más consecuente con la vaguedad inherente a la incertidumbre, poder decir en que grado $\tilde{A}$ es preferido a $\tilde{B}$, ($\tilde{A}$ es moderadamente mejor que $\tilde{B}$, $\tilde{A}$ es bastante mejor que $\tilde{B}$,...). Conformarse con lo primero, representa perder al final buena parte de la información que cuidadosamente hemos conservado utilizando la lógica fuzzy para nuestros cálculos. Atendiendo a este criterio los diferentes métodos se pueden agrupar en dos clases: aquellos que establecen una relación de preferencia ordinaria y aquellos que establecen una relación de preferencia borrosa. Los primeros actúan asimilando cada número borroso a un número preciso, y ordenando después dichos números ordinarios. Heilpern [11] propone asignar a cada número borroso su "valor esperado" , que es un número crisp perteneciente al intervalo esperado, más próximo a su extremo inferior o a su extremo superior, dependiendo del grado de aversión al riesgo del decisor. Exactamente lo mismo proponen Liou y Wang [14], puesto que su "valor integral total" no es otra cosa que el valor esperado de Heilpern. El método propuesto en este trabajo también opera con el intervalo esperado, pero al proporcionar el grado de preferencia en que una alternativa domina a otra, permite al decisor sopesar posibles factores que no pueden ser cuantificados y que sin embargo deben considerarse antes de tomar una decisión.

4 Intervalo esperado de un número borroso no normal

En ocasiones un número borroso puede ser no normal, vamos a ver que el método de clasificación propuesto en este trabajo permite comparar números borrosos de diferentes alturas.

Definición 9. *Un número borroso no normal $\tilde{A}$, es todo aquel cuya altura es menor que 1, es decir cuya función de pertenencia es:*

$$\mu_A(x) = \begin{cases} 0 & \forall x \in (-\infty, a_1] \\ f_A(x) & \text{creciente en } [a_1, a_2] \\ \varpi & \forall x \in [a_2, a_3] \\ g_A(x) & \text{decreciente en } [a_3, a_4] \\ 0 & \forall x \in [a_4, \infty) \end{cases}$$

donde ϖ es una constante llamada altura de $\tilde{A}$ tal que $0 < \varpi < 1$. El número borroso $\tilde{A}$ se denota por $[a_1, a_2, a_3, a_4; \varpi]$. Un número borroso no normal puede ser normalizado dividiendo su función de pertenencia por su altura. De manera que si denotamos por $\overline{\tilde{A}}$ al número normalizado, las partes izquierda y derecha de su función de pertenencia pasan a ser:

$$\bar{f}_A(x) = \frac{f_A(x)}{\varpi} \; ; \; \bar{g}_A(x) = \frac{g_A(x)}{\varpi} \qquad (7)$$

entonces el intervalo esperado de un número borroso no normal $\tilde{A}$, se podrá calcular así:

$$EI(\tilde{A}) = EI(\overline{\tilde{A}}) = \left[\int_{a_1}^{a_2} x \, d\bar{f}_A(x), \; -\int_{a_3}^{a_4} x \, d\bar{g}_A(x) \right] =$$

$$= \left[\frac{1}{\varpi} \int_{a_1}^{a_2} x \, df_A(x), \; -\frac{1}{\varpi} \int_{a_3}^{a_4} x \, dg_A(x) \right] =$$

$$= \left[\frac{1}{\varpi} \int_0^{\varpi} f_A^{-1}(\alpha) \, d\alpha, \; \frac{1}{\varpi} \int_0^{\varpi} g_A^{-1}(\alpha) \, d\alpha \right]$$

es inmediato comprobar que si f_A y g_A son lineales, es decir si el numero borroso $\tilde{A}$ es triangular o trapezoidal, su intervalo esperado es el siguiente:

$$EI(\tilde{A}) = \left[\frac{1}{2}(a_1 + a_2),\ \frac{1}{2}(a_3 + a_4)\right]$$

Una vez calculado el intervalo esperado, puede utilizarse para, por medio de la relación borrosa *M* (véase definición 5), comparar este número no normal con otro u otros, normales o no.

5 Conclusiones

En este trabajo se ha propuesto un método de ordenación de números borrosos, basado en la comparación de los intervalos esperados, que verifica cuatro interesantes propiedades: expresión borrosa de la relación de preferencia, distinguibilidad, racionalidad y robustez. Este método proporciona unos resultados muy parecidos a los del método de Yuan, pero es de cálculo mucho más sencillo, y además permite introducir en la comparación números borrosos de diferentes alturas, cosa que con el de Yuan no es posible.

Desarrollo futuro: en algunos modelos de decisión los resultados obtenidos no vendrán representados por números borrosos, sino por la unión de varios subconjuntos borrosos, algunos de los cuales proceden del truncamiento de un número borroso, lo que puede dar lugar a un subconjunto borroso no convexo, [3], [12 pp. 127-131]. Sería conveniente disponer de un sistema que permitiera extender la comparación a este tipo de conjuntos. La generalización del concepto de intervalo esperado a estos subconjuntos borroso no convexos, nos permitiría utilizar el método aquí propuesto para ello.

Referencias.

[1] BAAS, S.M; AND KWAKERNAK, H. Rating and ranking of multiple-aspect alternatives using fuzzy sets. *Automatica* 13 (1977), 47-58

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

[3] BUCKLEY, J.J. The fuzzy mathematics of finance. *Fuzzy Sets and Systems 21* (1987), 257-273.

[4] BUCKLEY, J.J., AND QU, Y. On using α-cuts to evaluate fuzzy ecuations. *Fuzzy sets and Systems 38* (1990), 309-312.

[5] BUCKLEY, J.J., AND QU, Y. Solving fuzzy equations: A new solution concept. *Fuzzy setes and Systems 39* (1991), 291-301

[6] DELGADO, M., VERDEGAY, J.L. AND VILA, M.A. A procedure for ranking fuzzy relations. *Fuzzy Sets and Systems 26* (1988), 49-62.

[7] DUDOIS, D., AND PRADE, H. The mean value of fuzzy number. *Fuzzy Sets and Systems 24* (1987), 279-300.

[8] GOODMAN, I.R. Fuzzy sets as equivalence classes of random sets: R.R. Yager, Ed. *Recent Developments in Fuzzy Sets and Possibility Theorie.* Pergamon Press, New York (1982), 327-432.

[9] HEILPERN, S. Fuzzy number and interval ramdom sets. *Tamsui Oxford J. Management Sci.* (1989), 4-5, 1-14.

[10] HEILPERN, S. Interval ramdom sets and entropy. *Fuzzy Sets and Systems 35* (1990), 213-218.

[11] HEILPERN, S. The expected value of a fuzzy number. *Fuzzy Sets and Systems 47* (1992), 81-86.

[12] JIMENEZ, M *Modelos matemáticos aplicados a la toma de decisiones financieras en condiciones de incertidumbre.* Universidad del País Vasco, Bilbao 1.994. (Tesis Doctoral).

[13] KAUFMANN, A. *Introduction a la théorie des sous-ensembles flous.* Tome I. Paris. Masson, 1.973.

[14] LIOU, T-S. AND WANG, M-J. Ranking fuzzy numbers with integral value. *Fuzzy Sets and Systems 50* (1992), 247-255.

[15] NAKAMURA, K. Preference relations on a set of fuzzy utilities as a basis for decision making. *Fuzzy Sets and Systems 20* (1986), 147-162.

[16] YUAN, Y. Criteria for evaluating fuzzy ranking methods. *Fuzzy Sets and Systems 44* (1991), 139-157.

[17] ZADEH, L.A. Similarity relations and fuzzy orderings. *Inform. Sci. 3* (1971), 177-200.

El Problema del Transporte Sólido Multiobjetivo No Lineal con Metas Fuzzy: Un Método de Solución basado en Algoritmos Genéticos

Jose M. Cadenas Fernando Jiménez
Universidad de Murcia
30071-Espinardo, Murcia, España
e-mail: jcadenas@dif.um.es, fernan@dif.um.es

Resumen

En el Problema del Transporte, normalmente se supone que el decisor conoce con certeza todos los datos que se manejan, pero a menudo esto no es así. En estos casos, los Conjuntos Fuzzy son herramientas muy apropiadas para modelizar este tipo de procesos. En este trabajo intentamos realzar la interface entre los Algoritmos Genéticos y los Conjuntos Fuzzy ([10]) proponiendo un Método de Solución basado en Algoritmos Genéticos para el caso en el cual se asumen Metas Fuzzy en el Problema del Transporte Sólido Multiobjetivo No Lineal. El trabajo es una extensión a nuestras investigaciones anteriores para el Problema del Transporte Sólido Multiobjetivo Lineal ([5]).

Palabras clave: Transporte, Sólido, Multiobjetivo, Metas Fuzzy, Programación No Lineal, Algoritmo Genético, Función de Evaluación, Operadores Genéticos.

1 Introducción

El Problema del Transporte es uno más comunes problemas combinatorios con restricciones que han sido estudiados. Su clara relevancia e interpretación económica, así como la existencia de fáciles y robustos algoritmos que lo resuelven, justifica la atención prestada a dicho problema. Como ocurre en el caso clásico, existe una extrecha relación entre el Problema del Transporte en un ambiente fuzzy ([6]) y la Programación Fuzzy ([16]), pero aparecen casos para los cuales los algoritmos existentes no trabajan bien, por ejemplo, el Problema del Transporte Sólido Multiobjetivo (PTSM) No Lineal. Actualmente, los Algoritmos Genéticos (AG) están siendo muy tenidos en cuenta sobre todo en aquellos problemas con espacios de solución complejos para los cuales no se dispone de buenos algoritmos que los resuelvan. En muchos de estos casos sería deseable disponer de métodos de resolución que nos proporcionen "buenas" soluciones si estas son obtenidas con razonable rapidez. Los AG son procedimientos adaptativos de optimización y búsqueda inspirados en los mecanismos de la selección natural y en la genética. En un AG, a una solución de nuestro problema se le denomina *individuo*. En esencia, se trata de mantener una *población* de un determinado número de individuos, cada uno de ellos caracterizado por un código genético (*genotipo*) que lo identifica unívocamente; se simula entonces una evolución de dicha población a lo largo del tiempo, basada en la aparición de nuevos individuos resultado de *cruces*, *mutaciones* y *reproducciones directas* de los antiguos individuos. Una función objetivo o de evaluación juega el papel de *entorno* para decidir en cada *generación* qué soluciones relativamente buenas se reproducen, y qué soluciones relativamente malas mueren, para ser sustituidas por los descencientes de las buenas.

Estos algoritmos cuentan con la ventaja de ser una técnica extraordinariamente general, lo que aumenta el número de posibles áreas de aplicación. Básicamente, podemos decir que un AG se basa en los siguientes componentes, para cualquier tipo de problema:

- Una *representación genética* de las soluciones del problema.
- Una forma de crear una *población inicial* de soluciones.
- Una *función de evaluación* capaz de medir la bondad de cualquier solución, y que hará el papel de "entorno", en el cual las mejores soluciones (esto es, aquellas que tengan un mejor valor objetivo) tengan mayor probabilidad de supervivencia.
- Una serie de *operadores genéticos* para combinar estas soluciones entre sí con el propósito de obtener otras mejores.
- El valor de unos *parámetros* de entrada que el AG usa para guiar su evolución: tamaño de la población, número de generaciones, probabilidades de cruce y mutación, etc.

Como puede observarse, estos algoritmos son muy sencillos y, sobre todo, muy generales. Es esta misma sencillez y generalidad la que origina su gran potencia, y la gran diversidad de áreas en la que pueden

funcionar. Una extensa documentación sobre AG la podemos encontrar en [7], [9], [11], [13]

2 El Problema del Transporte Sólido Multiobjetivo (PTSM)

El Problema del Transporte, introducido originalmente por F.L. Hitchcock, es uno de los más comunes problemas combinatorios con restricciones que han sido estudiados. En la mayoría de los casos, se requiere resolver el problema teniendo en cuenta más de un criterio de decisión, dando así lugar al Problema del Transporte Multiobjetivo. Muchos autores han desarrollado una gran variedad de enfoques de resolución para el Problema del Transporte Multiobjetivo (ver [1], [2], [3], [12], [15]), y en este trabajo se propone un enfoque de resolución con AG. El planteamiento del PTSM es el siguiente: Un producto homogéneo ha de ser transportado desde m orígenes hasta n destinos. Los orígenes son instalaciones de producción, almacenes o puntos de suministro, y están caracterizados por unas capacidades disponibles a_i, para $i = 1, 2, \ldots, m$. Los destinos son instalaciones de consumo, almacenes o puntos de demanda, y se caracterizan por requerir niveles de demanda b_j, para $j = 1, 2, \ldots, n$. Para llevar a cabo el transporte, disponemos de K modos de transporte (*conveyances*), y sean e_k $(k = 1, 2, \ldots, K)$ las cantidades del producto que pueden ser transportadas por los K diferentes modos de transporte. Asociado con el transporte unidad de dicho producto desde el origen i hasta el destino j a través del k-ésimo modo de transporte, existe un *costo* c^p_{ijk} correspondiente al p-ésimo *criterio de decisión*. El criterio de decisión puede ser costo del transporte, tiempo de reparto, cantidad de mercancía repartida, demanda incumplida, capacidad infrautilizada, fiabilidad de reparto, seguridad de reparto (desperdicio), y muchos otros.

El problema consiste en determinar las cantidades x_{ijk} que deberán de ser transportadas desde cada origen i a cada destino j a través de cada modo de transporte k de forma que todos los P criterios de decisión hayan sido tenidos en cuenta de manera satisfactoria para el decisor.

Cuando los valores c^p_{ijk} son directamente proporcionales a la cantidad transportada x_{ij}, el Problema del Transporte es *Lineal*, y en caso contrario es *No Lineal*.

El PTSM No Lineal puede ser representado como:

$$Minimizar\ Z_p = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{K} f^p_{ijk}(x_{ijk}),\ p = 1, 2, \ldots, P$$

sujeto a las siguientes restricciones:

$$\sum_{j=1}^{n}\sum_{k=1}^{K} x_{ijk} = a_i,\quad i = 1, 2, \ldots, m, \qquad (1)$$

$$\sum_{i=1}^{m}\sum_{k=1}^{K} x_{ijk} = b_j,\quad j = 1, 2, \ldots, n, \qquad (2)$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ijk} = e_k,\quad k = 1, 2, \ldots, K, \qquad (3)$$

$$x_{ijk} \geq 0,\quad para\ todo\ i, j, k, \qquad (4)$$

donde el subíndice en Z_p y el superíndice en c^p_{ijk} denota el p-ésimo criterio de decisión; $a_i > 0$ para todo i, $b_j > 0$ para todo j, $e_k > 0$ para todo k, $c^p_{ijk} \geq 0$ para todo i, j, k, p; $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = \sum_{k=1}^{K} e_k$ (condición de equilibrado), y las funciones f^p_{ijk}, para todo i, j, k, p, son funciones no lineales. Obsérvese que cuando $f^p_{ijk} = c^p_{ijk} x_{ijk}$, para todo i, j, k, p, entonces el problema es Lineal. No obstante, aunque este trabajo generaliza el problema para cuando éste es lineal, las soluciones serán diferentes a las obtenidas en [5] para el tratamiento exclusivo de problemas lineales, ya que en [5] se 'fuerza' a que las soluciones obtenidas sean enteras, mientras que aquí se obtendrán soluciones reales.

3 El PTSM con Metas Fuzzy

Un enfoque de Programación Fuzzy para el PTSM ha sido estudiado recientemente por A.K. Bit, M.P. Biswal y S.S. Alam (ver [2], [3]). En este modelo, las funciones objetivo están representadas por *conjuntos fuzzy* y el *conjunto de decisión* se define como la intersección de todos los conjuntos fuzzy y las restricciones. La *regla de decisión* seleccionaría la solución con valor de satisfacción más alto en el conjunto de decisión.

Sean L_p y U_p los límites inferior y superior de la p-ésima función objetivo, respectivamente. L_p representa el nivel de aspiración de realización, y U_p el nivel más alto aceptable de realización. El modelo fuzzy inicial puede venir dado entonces de la siguiente forma: Encontrar $\{x_{ijk},\ i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n;\ k = 1, 2, \ldots, K\}$ de forma que satisfaga $Z_p \lesssim L_p$, para $p = 1, 2, \ldots, P$, y las restricciones (1)-(4).

De cara a suprimir los cálculos de los valores L_p y U_p, para $p = 1, 2, \ldots, P$, proponemos lo siguiente (ver [8]): suponemos que el decisor puede fijar, para cada objetivo p, dos valores g_p y G_p, con $g_p < G_p$, donde g_p es la 'meta' en el objetivo p-ésimo, y G_p es el límite superior de los valores deseables para el decisor en el objetivo p-ésimo. Ahora podemos medir el grado de satisfacción de una solución $X = \{x_{ijk}\}$ para cada objetivo p con las funciones $\mu_p : R^n \to R$:

$$\mu_p(X) = \frac{G_p - Z_p(X)}{G_p - g_p},\ p = 1, 2, \ldots, P$$

donde $Z_p(X) = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{K} f^p_{ijk}(x_{ijk}),\ p = 1, 2, \ldots, P$

El siguiente paso es formular el problema de Programación No Lineal equivalente. Aquí proponemos el siguiente: $Maximizar\ \min_p \{\mu_p(X)\}$, sujeto a las restricciones (1)-(4).

Por último, aplicamos un AG para resolver este problema. Una vez ejecutado el método, las funciones μ_p así definidas nos permiten obtener las siguientes conclusiones para la solución $X^* = \{x^*_{ijk}\}$ encontrada:

- Si $\exists p/\mu_p(X^*) < 0 \Rightarrow X^*$ no es satisfactoria para el decisor (X^* viola G_p).
- Si $\exists p/\mu_p(X^*) > 1 \Rightarrow X^*$ satisface en exceso las necesidades del decisor (la meta g_p puede ser más ambiciosa).
- Si $0 \leq \mu_p(X^*) \leq 1, \forall p \Rightarrow X^*$ es satisfactoria para el decisor.

Para ilustar gráficamente estas conclusiones construimos las funciones $h_p : R \rightarrow R$ como sigue:

$$h_p(Z_p) = \frac{G_p - Z_p}{G_p - g_p}, \ p = 1, 2, \ldots, P.$$

Los resultados anteriores equivalen a los siguientes:

- Si $\exists p/h_p(Z^*_p) < 0 \Rightarrow X^*$ no es satisfactoria para el decisor (ver figura 1).
- Si $\exists p/h_p(Z^*_p) > 1 \Rightarrow X^*$ satisface en exceso las necesidades del decisor (ver figura 2).
- Si $0 \leq h_p(Z^*_p) \leq 1, \forall p \Rightarrow X^*$ es satisfactoria para el decisor (ver figura 3).

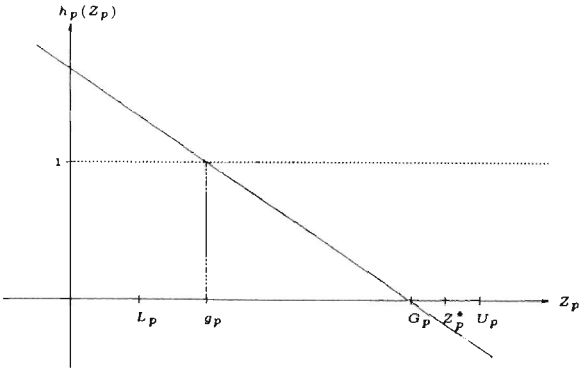


Figura 1

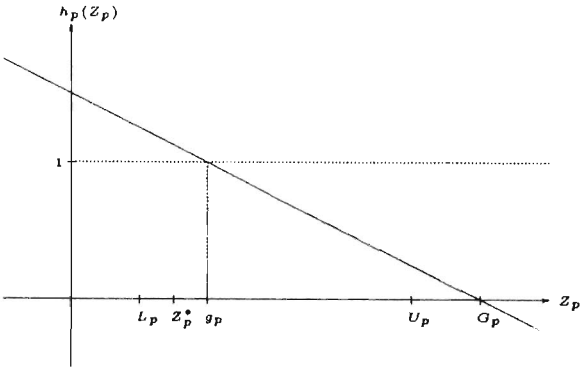


Figura 2

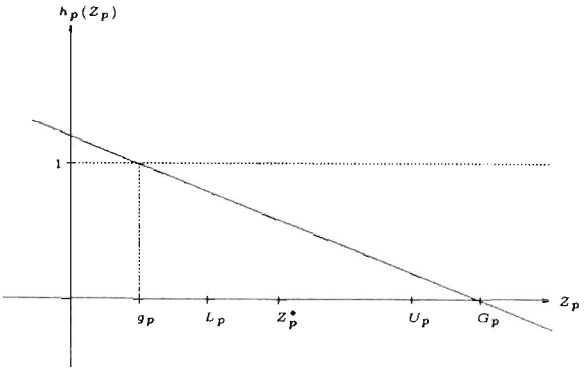


Figura 3

4 Un Método de Solución basado en AG

G.A. Vignaux y Z. Michalewicz proponen en [14], [17] un AG para el Problema del Transporte Lineal y No Lineal. Para el Problema del Transporte Sólido, la manera más natural de representar una solución es con un array tridimensional de rango $[m, n, K]$. Veamos las implicaciones de esta representación en satisfacción de restricciones, función de evaluación y operadores genéticos. Primero describiremos una forma de crear una solución que satisfaga todas las restricciones, con a lo sumo $m + n + K - 2$ valores no cero:

entrada: vectores a, b y c, de rangos m, n y K, respectivamente.
salida: un array $V = (v_{ijk})$ que representa una solución al Problema del Transporte Sólido, i.e., satisface las restricciones (1)-(4).
procedimiento inicializar;
begin
 inicializar todos los números del 1 al $m \cdot n \cdot K$ como 'no visitados';
 repeat
  seleccionar aleatoriamente un número 'no visitado' q y ponerlo como 'visitado';
  $i = \lfloor (q-1)/n \cdot K + 1 \rfloor$;
  $j = \lfloor ((q-1) \ mod \ (n \cdot K))/K + 1 \rfloor$;
  $k = ((q-1) \ mod \ (n \cdot K)) \ mod \ K + 1$;
  $val = min(a[i], b[j], c[k])$;
  $v_{ijk} = val$;

$a[i] = a[i] - val$;
$b[j] = b[j] - val$;
$e[k] = e[k] - val$;
until todos los números estén 'visitados';
end;

Satisfacción de restricciones: El procedimiento inicializar garantiza que cualquier permutación de $m \cdot n \cdot K$ números distintos desde el 1 hasta el $m \cdot n \cdot K$ produce una única solución que satisface las restricciones. Los operadores de mutación y cruce garantizan también que las soluciones que producen satisfacen todas las restricciones.

Función de evaluación: La función de evaluación es la función objetivo del Problema de Programación No Lineal:

$$eval(v_{ijk}) = \underset{p}{Min} \{\mu_p(Z_p)\}.$$

Operadores genéticos: Se han estudiado cuatro operadores genéticos válidos para el PTSM No Lineal, dos de mutación y dos de cruce.

Mutación1

Sea $V = (v_{ijk})$ un array $(m \times n \times K)$ seleccionado como padre para la mutación, y denotemos un descendiente por el array $V' = (v_{ijk})$ de igual dimensión que el padre. Usamos el siguiente algoritmo para obtener V' como resultado de la mutación de V:

1. Hacer $V' = V$;
2. Seleccionar aleatoriamente el conjunto de índices $I = \{i_1, i_2, \ldots, i_p\} \subset \{1, 2, \ldots, m\}$ tal que $2 \leq p \leq m$;
3. Seleccionar aleatoriamente el conjunto de índices $J = \{j_1, j_2, \ldots, j_q\} \subset \{1, 2, \ldots, n\}$ tal que $2 \leq q \leq n$;
4. Seleccionar aleatoriamente el conjunto de índices $L = \{l_1, l_2, \ldots, l_s\} \subset \{1, 2, \ldots, K\}$ tal que $2 \leq s \leq K$;
5. Crear un subarray W de dimensión $(p \times q \times s)$ a partir de los elementos del array V' de la siguiente forma: un elemento $v_{ijk} \in V'$ pertenece a W *sii* $i \in I$, $j \in J$ y $k \in L$;
6. Asignar nuevos valores a a_i^W, b_j^W y e_k^W con $1 \leq i \leq p$, $1 \leq j \leq q$ y $1 \leq k \leq s$ para el array W:

$$a_i^W = \sum_{j \in J} \sum_{k \in L} v_{ijk}, \quad 1 \leq i \leq p,$$

$$b_j^W = \sum_{i \in I} \sum_{k \in L} v_{ijk}, \quad 1 \leq j \leq q,$$

$$e_k^W = \sum_{i \in I} \sum_{j \in J} v_{ijk}, \quad 1 \leq k \leq s.$$

7. Usar el procedimiento inicializar para asignar nuevos valores al array W de forma que todas las restricciones a_i^W, b_j^W y e_k^W se satisfagan;
8. Sustituir los correspondientes elementos en el array V' por los nuevos elementos del array W De esta forma, se satisfacen todas las restricciones globales a_i, b_j y c_k

Mutation2

Este operador de mutación es una modificación del operador *Mutación1* para no elegir entradas cero seleccionando para ello valores aleatorios de un rango. El operador *Mutación2* es idéntico a *Mutación1* excepto que para recalcular el subarray W seleccionado usamos una versión modificada del procedimiento inicializar. Los cambios requeridos en el procedimiento inicializar son los siguientes:

El paso:
$val = min(a[i], b[j], e[k])$
es sustituido por:
$val_1 = min(a[i], b[j], e[k])$;
If (i ó j ó k son la última vez que se 'visitan') then
$val = val_1$
else
val = número aleatorio (real) entre 0 y val_1;

Este procedimiento debe ser finalmente modificado ya que normalmente produce un array-solución que viola las restricciones. Cuando esto ocurre, los vectores a_i, b_j y e_k contienen valores distintos de cero para algún o algunos i, j, k. Si aplicamos el procedimiento inicializar con estos valores de a_i, b_j y e_k, el resultado es un array-deudas que sumado al array-solución dará como resultado una solución factible a nuestro problema.

Cruce Geométrico

Asumimos que dos arrays $V_1 = (v_{ijk}^1)$ y $V_2 = (v_{ijk}^2)$ son seleccionados como padres para la operación de cruce. El cruce de ambos produce dos descendientes $V_3 = (v_{ijk}^3)$ y $V_4 = (v_{ijk}^4)$ como sigue:

1. $V_3 = V_1$; $V_4 = V_2$;
2. Seleccionar aleatoriamente el conjunto de índices $I = \{i_1, i_2, \ldots, i_p\} \subset \{1, 2, \ldots, m\}$ tal que $2 \leq p \leq m$;
3. Seleccionar aleatoriamente el conjunto de índices $J = \{j_1, j_2, \ldots, j_q\} \subset \{1, 2, \ldots, n\}$ tal que $2 \leq q \leq n$;
4. Seleccionar aleatoriamente el conjunto de índices $L = \{l_1, l_2, \ldots, l_s\} \subset \{1, 2, \ldots, K\}$ tal que $2 \leq s \leq K$;
5. Intercambiar $v_{ijk}^3 \leftrightarrow v_{ijk}^1$, para todo $i \in I$, $j \in J$ y $k \in L$.
6. Hacer los cambios mínimos en los elementos v_{ijk}^1 y v_{ijk}^2 para todo $i \in I$, $j \in J$ y $k \in L$ de forma que:

   1) Todas las restricciones se satisfagan;

   2) Ambos descendientes V_3 y V_4 conserven la máxima información posible de V_2 and V_1 respectivamente.

Cruce Aritmético

Al igual que en el *cruce geométrico*, denotamos como padres para el cruce los arrays $V_1 = (v^1_{ijk})$ y $V_2 = (v^2_{ijk})$, que darán lugar a dos descendientes $V_3 = (v^3_{ijk})$ y $V_4 = (v^4_{ijk})$ de la siguiente forma:

$$V_3 = c_1 \ V_1 + c_2 \cdot V_2$$

$$V_4 = c_1 \cdot V_2 + c_2 \cdot V_1$$

donde $c_1, c_2 \geq 0$ y $c_1 + c_2 = 1$, tomados aleatoriamente en cada operación de cruce.

5 Experimentos y Resultados

Un Sistema Genético Interactivo, **GA_FTP2**, ha sido construido con estas características. Más de 100 ejecuciones han sido llevadas a cabo durante el proceso de prueba con diferentes problemas. Atendiendo a los parámetros propios del AG, los mejores resultados se han obtenido para: generaciones $\geq$ 100; tamaño de la población $\geq$ 50; pocentaje de cruces (geométricos mas aritméticos) entre 5% y 20%; porcentaje de mutaciones (tipo 1 mas tipo 2) entre 10% y 40%; parámetro de distribución de probabilidad *sprob* (que controla la distribución geométrica usada para seleccionar los padres y los individuos que serán sustituidos) entre 0.7 y 0.9.

Para ilustrar el AG para el PTSM con Metas Fuzzy, hemos considerado el siguiente problema, con $f^p_{ijk} = c^p_{ijk}\sqrt{x_{ijk}}$, para todo i, j, k, p:

Ofertas : $a_1 = 24$; $a_2 = 8$; $a_3 = 18$; $a_4 = 10$;

Demandas : $b_1 = 16$; $b_2 = 19$; $b_3 = 25$;

Capacidades med. de transp. : $e_1 = 17$; $e_2 = 31$; $e_3 = 12$;

Metas : $g_1 = 100$; $G_1 = 350$; $g_2 = 50$; $G_2 = 150$;

Costes c^1_{ijk}

	D_1			D_2			D_3		
	C_1	C_2	C_3	C_1	C_2	C_3	C_1	C_2	C_3
O_1	15	18	17	12	22	13	10	4	12
O_2	17	20	19	21	21	22	21	19	18
O_3	14	11	12	25	34	33	20	16	15
O_4	22	18	13	24	35	32	18	21	14

Costes c^2_{ijk}

	D_1			D_2			D_3		
	C_1	C_2	C_3	C_1	C_2	C_3	C_1	C_2	C_3
O_1	6	7	8	10	6	5	11	3	7
O_2	13	8	11	12	2	9	20	15	13
O_3	5	6	7	11	9	7	10	5	2
O_4	13	6	6	17	11	18	12	16	12

Los valores de los parámetros del AG sobre los que se ha llevado a cabo el problema anterior son los siguientes: generaciones = 100; tamaño de la población = 100; número de cruces geométricos= 5; número de cruces aritméticos = 5; número de mutaciones tipo 1 =10; número de mutaciones tipo 2 =10; sprob = 0.9.

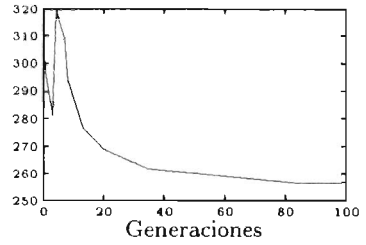
Los resultados obtenidos fueron los siguientes:

$$x_{121} = 11.0 \quad x_{132} = 13.0 \quad x_{222} = 8.0$$
$$x_{311} = 6.0 \quad x_{333} = 12.0 \quad x_{412} = 10.0$$

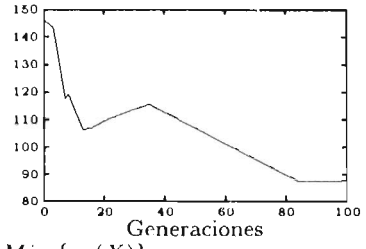
$$Z_1 = 256.79 \quad Z_2 = 87.79$$

Los siguientes gráficos muestran la evolución del AG ilustrando por separado las funciones objetivo Z_1 y Z_2, así como la evolución global del AG siguiendo el criterio max-min.

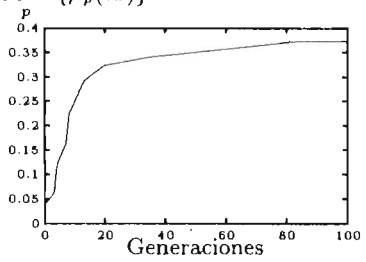
Objetivo 1



Objetivo 2



$Min_p \{\mu_p(X)\}$



6 Conclusiones

El estudio aquí realizado muestra un triple interés. Por un lado, supone una expansión a los estudios realizados por G.A. Vignaux y Z. Michalewicz en [14], [17]. Estos desarrollan un AG para el Problema del Transporte Lineal y No Lineal, y aquí se muestra como generalizar el AG cuando el Problema del Transporte es Sólido y Multiobjetivo. Por otro lado, y como ya se dijo en [14], [17], esta técnica de optimización supone una alternativa seria a los métodos clásicos cuando las funciones objetivo son no lineales. En [2], [3] se propone un enfoque de Programación Difusa al PTSM, desembocando en un modelo que resuelven con el Método del Simplex, para el caso lineal, y con alguna otra técnica clásica de optimización para el caso no lineal. Un sistema comercial, GAMS (ver [4]), basado en técnicas de gradiente controlado, ha sido comparado por otros autores ([14], [17]) con un Sistema Genético para el Problema del Transporte. Las conclusiones son que para funciones convexas ambos sistemas evolucionan bastante bien, pero cuando las funciones son concavas el sistema genético gana a GAMS, encontrando mejores soluciones. La causa fundamental es la tendencia por parte de los métodos clásicos de optimización no lineal a "caer" en óptimos locales, mientras que los AG realizan una búsqueda

global en todo el espacio de las soluciones. Finalmente, el trabajo aporta un paso más hacia la obtención de la interface entre los Algortimos Genéticos y los Conjuntos Fuzzy.

Referencias

[1] V.P Aneja and K.P.K. Nair, *Bicriteria transportation problem*, Management Sci., vol. 25., pp. 73-78, 1979.

[2] A.K. Bit, M.P Biswal and S.S. Alam, *Fuzzy programming approach to multicriteria decision making transportation problem*, Fuzzy Sets and Systems, vol. 50, pp. 135-141, 1992.

[3] A.K. Bit, M.P. Biswal, S.S. Alam, *Fuzzy programming approach to multiobjective solid transportation problem*. Fuzzy Sets and Systems, vol. 57, pp. 183-194, 1993.

[4] A. Brooke, D. Kendrick, A. Meeraus, *GAMS: A user's guide*. The Scientific Press, Redwood City, CA, 1988.

[5] J.M. Cadenas, F. Jiménez, *A genetic algorithm for the multiobjective solid transportation problem: a fuzzy approach*. Technical Report DIS TR 3-94, Dpto. de Informática y Sistemas, Facultad de Informática, Universidad de Murcia, 1994.

[6] S. Chanas, M. Delgado, J.L. Verdegay and M.A. Vila, *Interval and fuzzy extensions of classical transportation problems*, Transportation Planning and Tecnology, vol. 17, no. 2, pp. 203-218, 1993. Special Issue: Application of Fuzzy Set Theory to Transportation.

[7] L. Davis, Ed., *Genetic algorithms and simulated annealing*. London: Pitman, 1987.

[8] M. Delgado, *A resolution method for multiobjective problems*, European Journal of Operations Research, vol 13, pp. 165-172, 1983.

[9] D.E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989.

[10] F. Herrera and J.L. Verdegay, *Genetic algorithms with fuzzy performance function and applications*, Actas del II Congreso Español sobre Tecnologías y Lógica Fuzzy. Boadilla del Monte, Madrid, pp. 147-159, 1992 (en español).

[11] J.H. Holland, *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, 1975.

[12] S.M. Lee and L.J. Moore, *Optimizing transportation problem with multiple objectives*, AIIE Transactions, vol. 5, pp. 333-338, 1973.

[13] Z. Michalewicz, *Genetic algorithms + data structures = evolution programs*. Springer Verlag, 1992.

[14] Z. Michalewicz, A. Vignaux, and M. Hoobs, *A non-standard genetic algorithm for the nonlinear transportation problem*. Submitted to the ORSA Journal on Computing, vol. 3, no. 4, pp. 307-316, 1991

[15] J.L. Ringuest and D.B. Rinks, *Interactive solutions for the Linear multiobjective transportation problem*, Eur. J. Oper Res., vol. 32, pp. 96-106, 1987.

[16] J.L. Verdegay, *Fuzzy mathematical programming*, in "Fuzzy Information and Decision Processes". M.M. Gupta and E. Sánchez (Eds). North Holland, pp. 231-237, 1982.

[17] G.A. Vignaux and Z. Michalewicz, *A Genetic algorithm for the linear transportation problem*. IEEE Transactions on Systems, Man, and Cibernetics, vol. 21, no. 2, pp. 445-452, March/April, 1991.

Un Proceso de Decisión en Grupo con Preferencias Lingüísticas y Selección Secuencial

F. Herrera, E. Herrera-Viedma, J.Luis Verdegay
Dpto. de Ciencias de la Computacion e I.A
Universidad de Granada,
18071 - Granada, España
e-mail: herrera,viedma,verdegay@robinson.ugr.es

Resumen

En este trabajo se presenta un método para resolver problemas de decisión en grupo con preferencias lingüísticas. Este método se basa en los conceptos de mayoría fuzzy, alternativas no-dominadas, grado de no-dominancia, de dominancia lingüística, y de dominancia estricta definidos sobre preferencias lingüísticas. Se utiliza el operador lingüístico LOWA (linguistic ordered weighted averaging), cuyos pesos son elegidos de acuerdo al concepto de mayoría fuzzy especificado por un cuantificador lingüístico. Con este operador a partir del conjunto de relaciones lingüísticas, que representan las preferencias de los individuos, se obtiene una relación de preferencia lingüística colectiva. A partir del grado de no-dominancia se deriva el conjunto maximal de alternativas no-dominadas. Finalmente se definen los grados de dominancia y de dominancia estricta para obtener la solución al problema de decisión.

Palabras clave: Decisión en grupo, etiquetas lingüísticas, preferencias lingüísticas, mayoría fuzzy, cuantificador lingüístico fuzzy, grado de no-dominancia lingüística, grado de dominancia lingüística, grado de dominancia estricta.

1 Introducción

Un proceso de decisión en grupo se lleva a cabo en situaciones de decisión que se caracterizan: (i) por la presencia de dos o más individuos, cada uno determinado por sus aptitudes, motivaciones y personalidad; (ii) por la existencia de un problema común y de un conjunto de posibles soluciones; (iii) y por la intención de alcanzar una decisión colectiva sobre la solución que resuelve el problema [2].

Suponiendo un conjunto de alternativas o decisiones posibles, la cuestión básica es cómo relacionar los diferentes esquemas de decisión. De acuerdo a [3] hay al menos dos posibilidades: (i) establecer un consenso algebraico o proceso de selección o decisión en grupo; (ii) establecer un consenso topológico o proceso de consenso. El primero calcula un esquema de decisión de valor significativo dentro de un conjunto de decisiones, mientras el segundo calcula la medida de distancia entre los diferentes esquemas de decisión. En este trabajo exponemos un método para alcanzar un consenso algebraico.

En la teoría de decisión en grupo es usual expresar las valoraciones u opiniones personales mediante relaciones de preferencia. En el contexto de la lógica difusa podemos manejar las preferencias de los expertos, que a menudo son vagas o poco concretas. En este entorno de imprecisión se supone la existencia de un conjunto finito de alternativas $X = \{x_1, \ldots, x_n\}$, así como de un conjunto finito de individuos $N = \{1, \ldots, m\}$, de forma que cada individuo k de N expresa su relación de preferencias sobre X, $P_k \subset X \text{x} X$, donde $\mu_{P_k}(x_i, x_j) \in [0, 1]$, nota el grado de preferencia de la alternativa x_i sobre la x_j [6].

Sin embargo, no siempre el experto puede dar con exactitud el grado de preferencia de las alternativas x_i sobre x_j. Bajo estas circunstancias parece más adecuado el uso de valoraciones lingüísticas en lugar de numéricas. Así el experto podría expresar su preferencia utilizando una etiqueta lingüística extraída de un conjunto de términos lingüísticos que representan una escala de expresiones de certeza lingüísticamente valorados [5].

El proceso de decisión en grupo que se presenta en este trabajo, representado gráficamente en la figura 1, se realiza siguiendo los siguientes pasos:

1. A partir de un conjunto de relaciones de preferencia lingüísticas se obtiene una relación de preferencia colectiva haciendo uso del concepto de mayoría fuzzy representado por un cuantificador lingüístico y un operador de agregación OWA lingüístico [5].

2. Se aplica un proceso de selección secuencial que actúa sobre la relación de preferencia colectiva en dos fases:

   2.1. Utilizando el concepto de alternativas no-dominadas [7] se calcula el grado de no-dominancia lingüístico de cada alternativa, y seleccionamos el conjunto de alternativas no-dominadas maximales.

   2.2. Definiendo los conceptos de grado de dominancia lingüístico y grado de dominancia estricta sobre el conjunto de alternativas no-dominadas maximales obtenemos la solución al proceso de decisión.

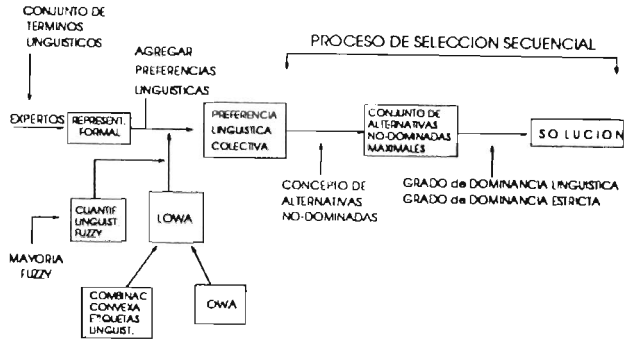


Fig. 1. Proceso de Decisión en Grupo

El trabajo se encuentra estructurado de la siguiente forma: la sección 2 presenta el operador LOWA; la sección 3 muestra como obtener la relación de preferencia lingüística colectiva; la sección 4 presenta el proceso de selección secuencial; la sección 5 un ejemplo de aplicación del proceso de decisión expuesto; y finalmente se presentan algunas conclusiones y posibles líneas de actuación futura.

2 Operador LOWA

Sea $S = \{s_i\}, i \in H = \{0, \ldots, T\}$ un conjunto finito y totalmente ordenado de términos lingüísticos que representan un valor posible de una variable real lingüística en el sentido usual [1, 10], siendo propiedades o restricciones vagas sobre [0,1]. La semántica de cada etiqueta es dada por números difusos definidos sobre el intervalo [0,1], descritos por funciones de pertenencia trapezoidales lineales. La etiqueta del centro representa una incertidumbre de "aproximadamente 0.5" y el resto de etiquetas está distribuído simétricamente a ambos lados de la misma [1]. El conjunto de etiquetas debe cumplir las siguientes propiedades:

1) $s_i \geq s_j$ si $i \geq j$.

2) $\mathrm{Neg}(s_i) = s_j$ tal que $j = T - i$.

3) $\mathrm{Max}(s_i, s_j) = s_i$ if $s_i \geq s_j$.

4) $\mathrm{Min}(s_i, s_j) = s_i$ if $s_i \leq s_j$.

Hay dos formas de agregar información lingüística incierta: la primera actúa por computación directa sobre las etiquetas [4]; y la segunda segunda usa las funciones de pertenencia asociadas [9, 1]. Esta segunda plantea el problema de que muchas veces los conjuntos fuzzy resultantes no se corresponden con ninguna etiqueta del conjunto de términos y entonces hay que hacer una aproximación lingüística [9, 10, 1], muchas veces inadecuada. En este trabajo usamos la primera aproximación.

Definición 1. [8] *Una función F definida sobre*

$$I^n \rightarrow I (\text{donde } I = [0, 1])$$

es un operador OWA de dimensión n si tiene asociado un vector de pesos W,

$$\begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_n \end{bmatrix}$$

tal que, i) $w_i \in [0, 1]$, *ii)* $\Sigma_i w_i = 1$, *y donde*

$$F(a_1, \ldots, a_m) = W_1 b_1 + W_2 b_2 + \ldots + W_n b_n$$

donde b_i es el i-ésimo mayor elemento de la colección $(a_1, \ldots, a_m)$

Este es un operador de agregación que cumple la propiedad siguiente:

$$Min\{a_1, \ldots, a_n\} \leq F(a_1, \ldots, a_m) \leq Max\{a_1, \ldots, a_n\}.$$

Este operador puede extenderse a argumentos lingüísticos mediante la combinación convexa de etiquetas lingüísticas definida en [4]. Allí se define la agregación de etiquetas por adición, la diferencia de etiquetas generalizada, y el producto por un número positivo real sobre un *espacio generalizado de etiquetas $\mathcal{S}$*, $\mathcal{S} = SxZ^+$, basado en S, conjunto de etiquetas base $S = \{(s_i, 1), i \in H\}$. De hecho, sea $\{p_1, \ldots, p_m\}$ un conjunto de etiquetas para ser agregadas, y asumiendo sin pérdida de generalidad que $p_m \leq p_{m-1} \leq \ldots \leq p_1$, para algún conjunto de coeficientes $\{\lambda_k \in [0, 1], k = 1, 2, .., m, \Sigma\lambda_k = 1\}$, la combinación convexa de estas m etiquetas genéricas viene dada por la expresión siguiente:

$$C\{\lambda_k, p_k, k = 1, \ldots, m\} =$$

$$= \lambda_1 \odot p_1 \oplus (1 - \lambda_1) \odot C\{\beta_h, p_h, h = 2, \ldots, m\}$$

con $\beta_h = \lambda_h / \sum_2^n \lambda_k; h = 2, \ldots, m$. Para el caso en que $\lambda_j = 1$ y $\lambda_i = 0$ con $i \neq j \; \forall i$, se define la combinación convexa como:

$$C\{\lambda_i, p_i, i = 1, \ldots, m\} = p_j.$$

En nuestro contexto todas las operaciones son realizadas sobre el conjunto de etiquetas $S.C\{\ldots\}$ es el operador de combinación convexa de etiquetas lingüísticas, que para el caso de la expresión $\lambda \odot s_j \oplus$

$(1-\lambda) \odot s_i, j \geq i$, resulta la etiqueta s_k tal que $k = \min\{T, i + \text{round}(\lambda \cdot (j-i))\}$, donde *round* es la función matemática de redondeo usual. Un ejemplo de la aplicación de este operador sería el siguiente. Supongamos el conjunto de términos:

$$S = \{s_8 = C, s_7 = EP, s_6 = MP, s_5 = JP,$$

$$s_4 = P, s_3 = PP, s_2 = MPP, s_1 = EI, s_0 = I\}$$

y $\lambda = 0.4$,

		$1-\lambda = 0.6$			
		PP	**MPP**	**C**	**EP**
$\lambda = 0.4$	**MP**	P	P	EP	EP
	I	MPP	EI	JP	P
	P	PP	PP	MP	MP
	MPP	PP	MPP	MP	JP

donde por ejemplo:

$$k_{11} = min\{8, 3 + round(0.4 * (6-3))\} = 4 \text{ (P)}$$

$$k_{21} = min\{8, 0 + round(0.6 * (3-0))\} = 2 \text{ (MPP)}$$

La generalización del operador OWA para etiquetas linguísticas (LOWA) se puede definir como:

$$F(a_1, \ldots, a_m) = W \cdot B^T = C\{w_k, b_k, k = 1, \ldots, m\} =$$

$$= w_1 \odot b_1 \oplus (1-w_1) \odot C\{\beta_h, b_h, h = 2, \ldots, m\}$$

donde $\beta_h = w_h / \Sigma_2^m w_k, h = 2, \ldots, m$, y B es el vector de etiquetas ordenado asociado. Cada elemento $b_i \in B$ es la i-ésima mayor etiqueta de la colección $a_1, \ldots, a_m$.

3 Relación de Preferencia Lingüística Colectiva

Supongamos que tenemos un conjunto de n alternativas $X = \{x_1, \ldots, x_n\}$ y un conjunto de individuos $N = \{1, \ldots, m\}$. Cada individuo $k \in N$ expresa su relación de preferencia valorada lingüísticamente en el conjunto de términos S,

$$\phi_{P_k} : X x X \rightarrow S,$$

donde $\phi_{P_k}(x_i, x_j) = p_{ij}^k \in S$ representa el grado lingüístico de preferencia de la alternativa x_i sobre la x_j. Asumimos que P_k es recíproca en el sentido de que $p_{ij}^k = \text{Neg}(p_{ji}^k)$, y por definición $p_{ii}^k = Imposible$ (la etiqueta mínima de S).

En primer lugar introducimos el concepto de cuantificador lingüístico fuzzy usado para especificar el concepto de mayoría fuzzy. Los cuantificadores lingüísticos fuzzy fueron presentados por Zadeh en 1983 [11]. Un cuantificador lingüístico es un conjunto fuzzy definido sobre $[0,1]$ que representa una propiedad linguísticamente cuantificada. Zadeh distinguió dos tipos de cuantificadores, los absolutos como *"alrededor de 7"* y los proporcionales o relativos como *"muchos"*. Utilizamos los cuantificadores proporcionales porque reflejan la esencia de la mayoría, como por ejemplo *muchos, al menos la mitad, todos, tantos como sea posible.* Si Q es un cuantificador relativo, entonces Q puede representarse como un subconjunto fuzzy de $[0,1]$ tal que para cada $r \in [0,1]$, $Q(r)$ indique el grado en el que la proporción r de objetos satisface el concepto referido por Q.

La función de pertenencia de un cuantificador relativo puede expresarse como

$$Q(r) = \begin{cases} 0 & \text{if } r < a \\ \frac{r-a}{b-a} & \text{if } a \leq r \leq b \\ 1 & \text{if } r > b \end{cases}$$

con $a, b, r \in [0,1]$. La gráfica 2 muestra el cuantificador *Al menos la mitad* con el par (0,0.5).

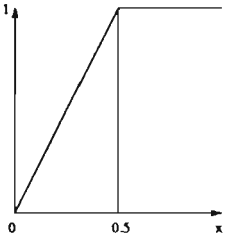


Fig. 2. Cuantificador "Al menos la mitad"

Usando el concepto de *mayoría fuzzy* especificado por un cuantificador lingüístico fuzzy no-decreciente podemos calcular un vector de pesos W, y usando el operador LOWA, la relación de preferencia colectiva, P, se obtiene como

$$P = F(P_1, \ldots, P_m)$$

con $p_{ij} = F(p_{ij}^1, \ldots, p_{ij}^m)$, y $p_{ij} \in S$.

Los pesos w_i para la agregación se pueden calcular a partir de la función que describe al cuantificador Q [8]. En el caso de un cuantificador relativo se obtienen mediante la expresión:

$$w_i = Q(i/m) - Q((i-1)/m), i = 1, \ldots, m.$$

4 Proceso de Selección Secuencial

El proceso de selección secuencial actúa sobre la relación de preferencia colectiva en dos pasos:

1. Utilizar el concepto de alternativas no-dominadas [7], para definir el grado de no-dominancia lingüística y obtener el conjunto maximal de alternativas no-dominadas a partir de la relación de preferencia lingüística colectiva [5]. Este concepto detecta contradicciones e incongruencias

entre las preferencias expresadas por los expertos. Esto es patente sobre todo cuando tenemos un conjunto maximal de alternativas no-dominadas que engloba al total de las alternativas.

2. Aplicar los conceptos de dominancia y dominancia estricta con etiquetas lingüísticas al conjunto de alternativas no-dominadas maximal para hallar las mejores alternativas. Sirven para discernir entre las alternativas no-dominadas aportando más consideraciones sobre ellas para poder seleccionar la mejor.

4.1 Grado de No-Dominancia Lingüística

Partiendo de la preferencia colectiva lingüística $P = (p_{ij}), i, j = 1, \ldots, n$, sea P^s una *relación de preferencia estricta lingüística* con $\mu_{P^s}(x_i, x_j) = p^s_{ij}$ tal que,

$$p^s_{ij} = Imposible \text{ si } p_{ij} < p_{ji},$$

$$\text{ó } p^s_{ij} = s_k \in S \text{ si } p_{ij} \geq p_{ji}$$

$$\text{con } p_{ij} = s_l, p_{ji} = s_t \text{ y } l = t + k.$$

El *grado de no-dominancia lingüística* de x_i se define como

$$\mu_{ND}(x_i) = \text{Min}_{x_j \in X}[\text{Neg}(\mu_{P^s}(x_j, x_i))]$$

donde el valor $\mu_{ND}(x_i)$ es un grado lingüístico que muestra el grado en el que la alternativa x_i es no-dominada por las demás

El conjunto de *alternativas no-dominadas maximales*, $X^{ND} \subset X$, se halla como

$$X^{ND} = \{x \in X \setminus \mu_{ND}(x) = \text{Max}_{y \in X}[\mu_{ND}(y)]\}.$$

4.2 Grado de Dominancia Lingüística

Definimos un grado lingüístico que sirve para seleccionar las mejores alternativas de X^{ND},

$$LDD(x_i) = F_{Q, i \neq j}(p_{ij}),$$

donde F_Q es el operador LOWA cuyos pesos son calculados usando el cuantificador Q, y cuyos componentes son los elementos de la correspondiente fila de P, tal que para x_i tenemos el conjunto de $n-1$ etiquetas $\{p_{ij} \setminus j = 1, \ldots, n \text{ and } i \neq j\}$.

Esta medida nos permite hallar el conjunto de alternativas no-dominadas con máximo grado de dominancia lingüística:

$$X^{LDD} = \{x \in X^{ND} \setminus LDD(x) =$$

$$= \text{Max}_{y \in X^{ND}}[LDD(y)]\}.$$

4.3 Grado de Dominancia Estricta

Este es un grado que sirve para discernir las mejores alternativas de entre las de X^{LDD},

$$SDD(x_i) = Q(\frac{r_i}{n-1}),$$

donde

$$r_i = card\{s_q \in S \setminus p_{iq} > p_{qi}\}.$$

Finalmente se obtiene una solución al proceso de decisión hallando el siguiente conjunto de alternativas

$$X^{SDD} = \{x \in X^{LDD} \setminus SDD(x) =$$

$$\text{Max}_{y \in X^{LLD}}[SDD(y)]\}.$$

Es importante destacar que los dos últimos grados usan un cuantificador lingüístico fuzzy en su definición, y por tanto son grados de dominancia basados en el concepto de mayoría fuzzy.

5 Ejemplo

Consideremos el siguiente conjunto de 9 etiquetas, S,

s_8	C	$Cierto$
s_7	EP	$Extremadamente\_Probable$
s_6	MP	$Muy\_Probable$
s_5	JP	$Justificadamente\_Probable$
s_4	P	$Posible$
s_3	PP	$Poco\_Probable$
s_2	MPP	$Muy\_Poco\_Probable$
s_1	EI	$Extremadamente\_Improbable$
s_0	I	$Imposible$

cuya semántica asociada es, [1]:

C	$(1, 1, 0, 0)$
EP	$(.98, .99, .05, .01)$
MP	$(.78, .92, .06, .05)$
JP	$(.63, .80, .05, .06)$
P	$(.41, .58, .09, .07)$
PP	$(.22, .36, .05, .06)$
MPP	$(.1, .18, .06, .05)$
EI	$(.01, .02, .01, .05)$
I	$(0, 0, 0, 0)$

donde los primeros dos parámetros indican el intervalo en el que la función de pertenencia vale 1.0; y el tercero y cuarto indican las holguras izquierda y derecha de la distribución.

Supongamos que tenemos cuatro alternativas y cuatro expertos cuyas preferencias lingüísticas usando el

anterior conjunto de 9 etiquetas son.

$$P_1 = \begin{bmatrix} - & PP & JP & MPP \\ JP & - & P & P \\ PP & P & - & MPP \\ MP & P & MP & - \end{bmatrix}$$

$$P_2 = \begin{bmatrix} - & P & P & MPP \\ P & - & JP & P \\ P & PP & - & MPP \\ MP & P & MP & - \end{bmatrix}$$

$$P_3 = \begin{bmatrix} - & P & JP & I \\ P & - & MP & P \\ PP & MPP & - & MPP \\ C & P & MP & - \end{bmatrix}$$

$$P_4 = \begin{bmatrix} - & P & JP & PP \\ P & - & P & PP \\ PP & P & - & MPP \\ JP & JP & MP & - \end{bmatrix}$$

Usando el cuantificador lingüístico *"Al menos la mitad"* con parametros $(0.0, 0.5)$, la relación de preferencias lingüísticas colectiva es:

$$P = \begin{bmatrix} - & P & JP & MPP \\ P & - & JP & P \\ PP & P & - & MPP \\ EP & P & MP & - \end{bmatrix}$$

donde por ejemplo el elemento p_{12} es obtenido como:

$$p_{12} = F(PP, P, P, P,) =$$

$$= (0.5, 0.5, 0, 0) \begin{pmatrix} P \\ P \\ P \\ PP \end{pmatrix} =$$

$$= C_0\{(0.5, P), (0.5, P), (0, P), (0, PP)\} =$$
$$= 0.5 \odot P \oplus (1 - 0.5) \odot C_1\{(1, P),$$
$$(0, P), (0, PP)\} = s_4 = P$$

pues

$$C_1\{(1, P), (0, P), (0, PP)\} = P,$$

como ya se indicó en la definición.

En el primer paso del proceso secuencial obtenemos la relación lingüística de preferencia estricta

$$P^s = \begin{bmatrix} - & I & MPP & I \\ I & - & EI & I \\ I & I & - & I \\ JP & I & P & - \end{bmatrix}$$

donde por ejemplo los elementos p^s_{13} y p^s_{31} se obtienen como:

$$p^s_{13} = p_{13} - p_{31} = JP - PP = MPP$$

$$p^s_{31} = I \text{ ya que } p_{13} \succ p_{31}$$

En base a la anterior relación hallamos el grado de no-dominancia lingüístico para cada alternativa

$$\mu_{ND}(x_1, x_2, x_3, x_4) = [PP, C, P, C]$$

Con ésto el conjunto de alternativas no-dominadas maximales, X^{ND}, es:

$$X^{ND} = \{x_2, x_4\}.$$

Como obtenemos un conjunto de alternativas no-dominadas con más de un elemento, entonces aplicamos el grado de dominancia lingüística sobre el conjunto X^{ND}. Para calcular dicho grado usamos el mismo cuantificador lingüístico, obteniendo:

$$(LDD(x_2), LDD(x_4)) = [JP, EP],$$

y por tanto,

$$X^{LDD} = \{x_4\}.$$

Como X^{LDD} tiene sólo una alternativa, ésta es la alternativa solución al proceso de decisión en grupo. Obviamente si hubiese tenido más de una alternativa tendríamos que haber aplicado en un siguiente paso el grado de dominancia estricta para discernir más entre las mejores alternativas.

6 Conclusiones

En este trabajo hemos presentado un proceso de decisión en grupo con selección secuencial y donde las valoraciones de los expertos son representadas mediante relaciones de preferencia lingüística. El proceso de decisión se ha basado en los conceptos de *mayoría fuzzy, cuantificadores lingüísticos fuzzy, el operador de agregación OWA, concepto de alternativas no-dominadas, y grados de no-dominancia, dominancia y dominancia estricta.*

Aquí hemos expuesto una de las posibilidades, el consenso algebraico o proceso de decisión en grupo con preferencias lingüísticas. La segunda posibilidad, un consenso topológico o proceso de consenso, queda pendiente para trabajos futuros en orden a determinar una medida de consenso lingüístico que muestre el grado de acuerdo global.

Referencias

1 BONISSONE, P.P., DECKER K.S., Selecting Uncertainty Calculi and Granularity. An Experiment in Trading-off Precision and Complexity. En: L.H Kanal and J.F. Lemmer (Eds.), *Uncertainty in Artificial Intelligence*, (North-Holland, 1986) 217-247

2. BUI, T.X., *Co-op, A Group Decision Support System for Cooperative Multiple Criteria Group Decision Making* (Springer-Verlag, Berlin, 1987).

3. CARLSSON, C., EHRENBERG, D., EKLUND, P., FEDRIZZI, M., GUSTAFSSON, P., LINDHOLM, P., MERKURYEVA G., RIISSNEN, T., VENTRE, A.G.S., Consensus in Distributed Soft Environments. *European Journal of Operational Research* 61 (1992) 165-185.

4. DELGADO, M., VERDEGAY, J.L., VILA, M.A., On Aggregation Operations of Linguistic Labels. *International Journal of Intelligent Systems* 8 (1993) 351-370.

5. HERRERA, F., VERDEGAY, J.L., Linguistic assessments in group decision. *Proc. of First European Congress on Fuzzy and Intelligent Technologies*, Aachen, 1993, 941-948.

6. KACPRYCK, J.,FEDRIZZI, M., NURMI, H., *Fuzzy Logic with Linguistic Quantifiers in Group Decision Making.* En: R.R. Yager, L.A. Zadeh (Eds)., An Introduction to Fuzzy Logic Applications in Intelligent Systems. (Kluwer Academic, Boston, 1992) 263-280.

7. ORLOVSKY, S.A., Decision-Making with a Fuzzy Preference Relation. *Fuzzy Sets and Systems* 1 (1978) 155-167.

8. YAGER, R.R., On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics* 18 (1988) 183-190.

9. ZADEH, L., The Concept of a Linguistic Variable and its Applications to Approximate Reasoning. *Part I, Information Sciences* 8 (1975) 199-249, *Part II, Information Sciences* 8 (1975) 301-357, Part III, *Information Sciences* 9 (1975) 43-80.

10. ZADEH, L., *Fuzzy Sets and Information Granularity.* En: M.M. Gupta et al. Eds., Advances in Fuzzy Sets Theory and Applications (North Holland, 1979) 3-18.

11. ZADEH, L. A Computational Approach to Fuzzy Quantifiers in Natural Language. *Comps & Maths. with Appls* 9 (1983) 149-184.

UN MODELO DE GOAL PROGRAMMING LINEAL DIFUSO CON NUMEROS DIFUSOS TRIANGULARES EN COEFICIENTES Y NIVELES DE ASPIRACIÓN.

ARENAS PARRA, M.MAR
Dpto. Matemáticas.
Universidad de Oviedo

RODRIGUEZ URIA, M.VICTORIA
Dpto. Matemáticas.
Universidad de Oviedo.

1 INTRODUCCIÓN.

La Programación Multiobjetivo puede definirse como una parte de la Investigación Operativa que trata de proporcionar métodos eficientes para la toma de decisiones sobre problemas que incluyen diversidad de objetivos, a veces contradictorios, que son evaluados de acuerdo a múltiples criterios y donde no es evidente la alternativa óptima. La expresión matemática del problema sería:

$$\text{Optimizar} \quad f_r(x) \quad r = 1, 2, \ldots, m$$

$$\text{sujeto a} \quad x \in X$$

En el mundo real, no obstante, el concepto de meta o goal puede ser más útil que el concepto de óptimo de una función objetivo, ya que se tienen en cuenta las preferencias del decisor reflejadas, de un lado en la fijación de niveles de aspiración aceptables para el mismo y de otro en la ordenación de las funciones objetivo, por su importancia relativa para el decisor, estableciendo entre las mismas un sistema de prioridades de antemano o excluyentes (pre-emptive priorities), e incluso ponderaciones dentro de cada nivel de prioridad, si hubiere lugar. La metodología Goal Programming [G.P.] o Programación por Metas, en sí misma, está alejada de toda pretensión optimizadora, basándose en una filosofía satisfaciente en la línea que propone Simon [1979] y por ello se muestra como un modelo especialmente adecuado para ser expresado cuantitativamente en el marco de la teoría de subconjuntos difusos, que proporciona herramientas de lógica flexible, muy bien adaptadas al pensamiento humano.

En general, el decisor empresarial expresa sus metas y preferencias, así como algunos de sus datos, en términos vagos, inciertos o imprecisos; a pesar de lo cual, en los modelos habituales del Goal Programming los datos, sean coeficientes tecnológicos del problema o sean niveles de aspiración del decisor empresarial, son representados por números exactos que no siempre responden a un conocimiento preciso de los mismos, sino al intento de manipulación cuantitativa de los problemas. Proponemos en este trabajo tratar los datos imprecisos como números difusos de tipo triangular $\tilde{a} = (\alpha, \sigma)$.

2 FORMULACIÓN DE UN MODELO G.P. DIFUSO.

La programación por metas –GOALS– es un modelo de optimización con objetivos múltiples basado en las preferencias del decisor –de los llamados con información a priori– reflejadas de un lado en la fijación de niveles de aspiración aceptables para el mismo y de otro en la ordenación de las funciones objetivo por su importancia relativa para el decisor, estableciéndose entre las mismas un sistema de prioridades de antemano (preemptive priority), e incluso ponderaciones dentro de cada nivel de prioridad, si hubiere lugar.

El problema tendrá habitualmente la forma:

Encontrar un $X^* = (x_0^*, x_1^*, \ldots, x_n^*)^t$, tal que:

$$\left.\begin{array}{l} \text{GOALS} \left\{ \begin{array}{l} G_r \equiv c_r^1 x_1 + c_r^2 x_2 + \ldots + c_r^n x_n \geq g_r \\ r = 1, 2, \ldots, m \end{array} \right. \\ \text{RESTRICCIONES} \left\{ \begin{array}{l} AX \leq b, \\ X \geq 0 \end{array} \right. \end{array} \right\} \text{(I)}$$

donde g_r, $r = 1, 2, \ldots, m$ representa el nivel de aspiración del decisor para el objetivo r-ésimo.

En el caso en que tanto los niveles de aspiración, g_r, como los coeficientes, c_r^i, son conocidos de modo impreciso, podrían ser cuantificados mediante números difusos triangulares simétricos, obteniendo la expresión del problema **(II)**:

Encontrar un $X^* = (x_0^*, x_1^*, \ldots, x_n^*)^t$ tal que:

$$\tilde{G}_1 \equiv \tilde{g}_1 x_0 + \tilde{c}_1^1 x_1 + \tilde{c}_1^2 x_2 + \cdots + \tilde{c}_1^n x_n \gtrsim 0$$
$$\tilde{G}_2 \equiv \tilde{g}_2 x_0 + \tilde{c}_2^1 x_1 + \tilde{c}_2^2 x_2 + \cdots + \tilde{c}_2^n x_n \gtrsim 0$$
$$\vdots$$
$$\tilde{G}_m \equiv \tilde{g}_m x_0 + \tilde{c}_m^1 x_1 + \tilde{c}_m^2 x_2 + \ldots + \tilde{c}_m^n x_n \gtrsim 0$$

$$AX \leq b$$

$$X \geq 0$$

donde

$$\tilde{g}_r = (\alpha_r, \sigma_r)$$

representa el nivel de aspiración difuso asociado al r-ésimo objetivo transformándolo en un goal difuso,

$$\tilde{c}_r^i = (\alpha_r^i, \sigma_r^i)$$

representa un coeficiente (coste o beneficio) difuso y la desigualdad difusa

$$\tilde{G}_r \gtrsim 0$$

significa *casi positivo*[1]. A y b representan una matriz y un vector convenientes, respectivamente,

$$X = (x_1, x_2, \ldots, x_n)^t, \; x_0 = 1$$

. No haremos distinción entre restricciones y goals difusos.

Conocidos los niveles de aspiración del centro decisor, g_r, para cada objetivo, G_r, clásicamente, el primer paso operativo en la formulación del problema por metas (goal programming) reside en conectar cada objetivo con el nivel de aspiración respecto del mismo por medio de la introducción de las variables de desviación negativa, n_r y positiva, p_r, variables lógicas que representan las diferencias entre el valor que alcanza cada objetivo para cierto $\mathbf{x}$ y el nivel de aspiración que sobre tal objetivo tiene el decisor.

Así para el atributo r-ésimo del problema **(I)**, descrito mediante G_r, tendremos la meta:

$$G_r(\mathbf{x}) + n_r - p_r = g_r, \quad r = 1, 2, \ldots, m$$

[1]La relación $\tilde{G} \gtrsim 0$ -la función $\tilde{G}$ es "casi positiva"- equivale a decir que:

$$\mu_{\tilde{G}}(0) \leq 1 - \mathbf{h}$$

donde $\mathbf{h}$, $(0 \leq \mathbf{h} \leq 1)$, es un parámetro que mide el grado de *casi positividad* del goal difuso $\tilde{G}$: a mayor valor de $\mathbf{h}$ más fuerte es el significado de *casi positivo*.

Los deseos del centro decisor respecto de cada meta se cuantifican ahora mediante la expresión de la "desviación no deseada" en cada caso. El problema reside ahora en minimizar tales desviaciones, que son recogidas en la llamada "función de realización o de logro".

Supuesto que el decisor empresarial asocia prioridades excluyentes (preemptive priorities) a las diferentes metas, podemos transformar el programa en un modelo de programación por metas lexicográficas. Se trata de minimizar lexicográficamente la función vectorial de logro cuyas componentes son función, $a_k(n,p)$, de las desviaciones no deseadas de las metas, es decir, se trata de resolver el problema:

Encontrar un $X^* = (x_0^*, x_1^*, \ldots, x_n^*)^t$, tal que:

$$\text{Lexmin } \mathbf{a} = \{a_1 = a_1(n,p), \ldots, a_K = a_K(n,p)\}$$

$$G_r \equiv c_r^1 x_1 + c_r^2 x_2 + \ldots + c_r^n x_n + n_r - p_r = g_r$$
$$r = 1, 2, \ldots, m$$

$$AX \leq b$$
$$X \geq 0$$

Proposición 1 *La función lineal difusa*

$$\tilde{G} = \tilde{C}_0 x_0 + \tilde{C}_1 x_1 + \ldots + \tilde{C}_n x_n = \tilde{C}\, X$$

$$\tilde{C} = \left(\tilde{C}_0, \tilde{C}_1, \ldots, \tilde{C}_n\right) = (\alpha, \sigma)$$

donde $\tilde{C}_r$, $r = 0, 1, \ldots, n$ *son números difusos de tipo triangular simétricos caracterizados por* (α_r, σ_r) *y que constituyen el parámetro difuso* $\tilde{C}$.

$$\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_n)^t, \qquad \sigma = (\sigma_0, \sigma_1, \ldots, \sigma_n)^t$$

$$X = (x_0, x_1, \ldots, x_n)^t \geq 0 \qquad \text{con} \qquad x_0 = 1.$$

está definida por la siguiente función pertenencia[2]*:*

$$\mu_{\tilde{G}}(y) = \begin{cases} 1 - \dfrac{|y - (\alpha_0 + \sum_{r=1}^n \alpha_r x_r)|}{\sigma_0 + \sum_{r=1}^n \sigma_r x_r} & X > 0 \\ 1 & X = 0, y = 0 \\ 0 & X = 0, y \neq 0 \end{cases}$$

Teniendo en cuenta las consideraciones anteriores obtenemos el siguiente resultado:

[2]Arenas, M.M.; Rodríguez, M.V.: Fuzzy linear Goal Programming with fuzzy numbers in coefficients and target values. XIth International Conference on M.C.D.M., Coimbra 1994

Proposición 2 *El problema* **(II)** *tendrá la siguiente forma:*

Encontrar un $X^* = (x_0^*, x_1^*, \ldots, x_n^*)^t$ *tal que minimice lexicográficamente la función* **a**:

$$Lexmin\ \mathbf{a} = \{a_1 = a_1(n,p), \ldots, a_K = a_K(n,p)\}$$

sujeto a

$$\mathbf{P_1}\begin{cases} \mu_{\tilde{G}_1}(0) + n_1 - p_1 = 1 - h \\ \vdots \\ \mu_{\tilde{G}_{l_1}}(0) + n_{l_1} - p_{l_1} = 1 - h \end{cases}$$

$$\mathbf{P_2}\begin{cases} \mu_{\tilde{G}_{l_1+1}}(0) + n_{l_1+1} - p_{l_1+1} = 1 - h \\ \vdots \\ \mu_{\tilde{G}_{l_2}}(0) + n_{l_2} - p_{l_2} = 1 - h \end{cases}$$

$$\vdots$$

$$\mathbf{P_m}\begin{cases} \mu_{\tilde{G}_{l_{K-1}+1}}(0) + n_{l_{K-1}+1} - p_{l_{K-1}+1} = 1 - h \\ \vdots \\ \mu_{\tilde{G}_m}(0) + n_m - p_m = 1 - h \end{cases}$$

$$AX \leq b$$

$$X \geq 0$$

$$0 \leq h \leq 1$$

$$n = (n_1, n_2, \ldots, n_m) \geq 0, p = (p_1, p_2, \ldots, p_m) \geq 0$$

Utilizamos para su resolución el algoritmo secuencial del goal programming no lineal. La solución $x^*(h)$ asociada al último paso del algoritmo se propone como solución del problema inicial. Considerando h como un parámetro, tendríamos un programa de goal programming paramétrico cuya solución $x^*(h)$ dependerá de h, presentando al decisor las soluciones para cada valor de h este elegiría aquella que mejor se adaptase a sus deseos.

Formalmente el problema se reduce a encontrar el mayor h compatible con el problema anterior. Obsérvese que sería posible determinar el conjunto de soluciones eficientes y en tal conjunto resolver el problema:

$$\text{máx} \quad h$$

$$\text{s.t.} \quad x \in eff[FGPP]$$

EJEMPLO.–

Una empresa produce los outputs X_1, X_2, X_3, X_4. La aportación de cada producto a la facturación y a los beneficios de la empresa se recogen en la tabla siguiente:

PROD.	PRECIO/UNID.	BENEF.
X_1	0.40	0.12
X_2	0.045	0.018
X_3	0.06	0.03
X_4	0.06	0.03

Utilizamos como u.m. 100.000 pesetas.

En la producción de cada output intervienen dos profesionales P_1 y P_2, así como dos máquinas, M_1 y M_2; detallamos en la siguiente tabla su contribución en tiempo a cada producto:

PRODUCTO	P_1	P_2	M_1	M_2
X_1	60	–	15	30
X_2	20	–	–	10
X_3	–	20	–	10
X_4	10	–	40	10

Utilizamos como u.t. minutos.

Suponemos que las restricciones del total de horas de trabajo de los profesionales y de las máquinas son:

$$1620 \leq minutos\ P_1 \leq 2700$$

$$540 \leq minutos\ P_2 \leq 900$$

$$540 \leq minutos\ M_1 \leq 900$$

$$1260 \leq minutos\ M_2 \leq 2160$$

La pretención de la empresa es aumentar su facturación y beneficios anuales, junto con la redistribución de la contibución del cuarto producto a la facturación total. Este problema es de modo evidente un problema de programación multiobjetivo.

El decisor empresarial establece sus niveles de aspiración de modo difuso para cada uno de los objetivos o goals; representándolos mediante números difusos el problema tendrá la siguiente forma:

Encontrar un $\mathbf{x}^t = (x1, x2, x3, x4)$, tal que

Prioridad 1

$$G_1 \equiv 0.06x4 \gtrsim (2.8, 0.3)$$

$$G_2 \equiv 0.12x1 + 0.014x2 + 0.018x3 + 0.01x4 \lesssim (4.69, 0.3)$$

Prioridad 2

$$G_3 \equiv 0.4x1 + 0.045x2 + 0.06x3 + 0.06x4 \gtrsim (12.5, 6)$$

Restricciones

$1620 \leq 60x1 + 20x2 + 10x4 \leq 2700$

$540 \leq 20x3 \leq 900$

$540 \leq 15x1 + 40x2 \leq 900$

$1260 \leq 30x1 + 10x2 + 10x3 + 10x4 \leq 2160$

$xi \geq 0,\ i = 1,2,3,4$

donde xi es la variable unidades de producto X_i , $i = 1,2,3,4$. o bien

Encontrar un $\mathbf{x^t} = (x1, x2, x3, x4)$, tal que

$$\mathbf{P_1}\begin{cases} 1 - \dfrac{-2.8 + 0.06x_4}{0.3} \leq 1 - h \\ 1 - \dfrac{4.69 - 0.12x_1 - 0.018x_2 - 0.03x_3 - 0.03x_4}{0.3} \leq 1 - h \end{cases}$$

$$\mathbf{P_2}\left\{ 1 - \frac{-12.5 + 0.4x_1 + 0.045x_2 + 0.06x_3 + 0.06x_4}{6} \leq 1 - h \right.$$

$1620 \leq 60x_1 + 20x2 + 10x_4 \leq 2700$
$540 \leq 20x_3 \leq 900$
$540 \leq 15x_1 + 40x_2 \leq 900$
$1260 \leq 30x_1 + 10x_2 + 10x_3 + 10x_4 \leq 2160$

$\mathbf{x} \geq 0$

Usando la definición de casi positividad de las funciones difusas e introduciendo las desviaciones negativas y positivas del modelo goal programming transformamos el problema para aplicarle el algoritmo secuencial anteriormente descrito obteniendose los siguientes resultados en el último paso:

Encontrar un $\mathbf{x^t} = (x1, x2, x3, x4)$, tal que

Lexmin $a = (n1 + p2, n3)$

sujeto a:

$-2.8 - 0.3h + 0.06x4 + n1 - p1 = 0$

$-4.69 + 0.3h + 0.12x1 + 0.014x2 + 0.018x3 + 0.01x4 + n2 - p2 = 0$

$-12.5 - 6h + 0.4x1 + 0.045x2 + 0.06x3 + 0.06x4 + n3 - p3 = 0$

$1620 \leq 60x_1 + 20x2 + 10x_4 \leq 2700$

$540 \leq 20x_3 \leq 900$

$540 \leq 15x_1 + 40x_2 \leq 900$

$1260 \leq 30x_1 + 10x_2 + 10x_3 + 10x_4 \leq 2160$

$0 \leq h \leq 1$

$x = (x1, x2, x3, x4) \geq 0$

$n = (n1, n2, n3) \geq 0$

$p = (p1, p2, p3) \geq 0$

```
MAX h
SUBJECT TO
0.06x4 - 0.3h - p1 = 2.8
0.12x1 + 0.014x2 + 0.018x3 + 0.01x4 + 0.3h + n2 = 4.69
0.4x1 + 0.045x2 + 0.06x3 + 0.06x4 - 6h - p3 = 12.5
60x1 + 20x2 + 10x4 <= 2700
60x1 + 20x2 + 10x4 >= 1620
20x3 <= 900
20x3 >= 540
15x1 + 40x2 <= 900
15x1 + 40x2 >= 540
30x1 + 10x2 + 10x3 + 10x4 <= 2160
30x1 + 10x2 + 10x3 + 10x4 >= 1260
h <= 1
h >= 0
-0.4x1 - 0.045x2 - 0.06x3 - 0.06x4 + G1 = 0
-0.06x4 + G2 = 0
-0.12x1 - 0.014x2 - 0.018x3 - 0.01x4 + G3 = 0
-60x1 - 20x2 - 10x4 + P1 = 0
-20x3 + P2 = 0
-15x1 - 40x2 + M1 = 0
-30x1 - 10x2 - 10x3 - 10x4 + M2 = 0
END
GIN x1
GIN x2
GIN x3
GIN x4

LP OPTIMUM FOUND AT STEP 50
OBJECTIVE VALUE =.868333300
```

VARIABLE	VALUE	REDUCED COST
$x1$	22.000000	−.036667
$x2$	6.000000	.002500
$x3$	33.000000	.000000
$x4$	111.000000	.000000
h	.868333	.000000
$p1$	3.599500	.000000
$n2$	.001500	.000000
$p3$	.000000	.166667
$G1$	17.710000	.000000
$G2$	6.660000	.000000
$G3$	4.428000	.000000
$P1$	2550.000000	.000000
$P2$	660.000000	.000000
$M1$	570.000000	.000000
$M2$	2160.000000	.000000

Bibliografía

[1] Bellman, R. E.;Zadeh, L.A. (1970): "Decision-making in a fuzzy environment", *Management Science 17*, B141-B164.

[2] Carlsson, C.; Korhonen, P. (1986): "A parametric approach to fuzzy linear programming", *Fuzzy sets and Systems 20*, 17-30.

[3] Chanas, S. (1989): "Fuzzy programming in multiobjective linear programming - a parametric approach", *Fuzzy sets and Systems 29*, 303-313.

[4] Delgado, M.; Verdegay, J L., Vila, M.A. (1989): "A general models for fuzzy linear programming", *Fuzzy sets and Systems 29*, 21-29.

[5] Dubois,D.;Prade,H., (1980): "Systems of linear fuzzy constraints", *Fuzzy Sets and Systems 3*, 37-48.

[6] Fedrizzi,M., (1987): "Introduction to Fuzzy Sets and Possibility Theory", *Optimization Models using fuzzy Sets and Possibility Theory*, 13-26.

[7] Hannan, E. L. (1980): "Nondominance in goal programming". *INFOR, Canadian Journal of O.R. and Inf. Processing, 18*, 300-309.

[8] Hannan, E. L. (1981): "Linear programming with multiple fuzzy goals". *Fuzzy Sets and Systems, 6*, .

[9] Ignizio, J.P. (1976): *Goal Programming and Extensions*. Lexington Books. Lexington.

[10] Lai, Y.; Hwang, C. (1992): *Fuzzy Mathematical Programming. Methods and Aplications*, Springer-Verlag, Berlin Heidelberg.

[11] Narasimhan (1980): "Goal Programming in a fuzzy environment". *Decision Sciences, 99*. 325-336.

[12] Negoitia, C.V.; Ralescu, D.A. (1977): "On fuzzy optimization". *Kybernetes, 6*, 193-196.

[13] Rodríguez , M. V.; Arenas, M. M. (1992): *Programación Multiobjetivo: Goal Programming*, Servicio de publicaciones del Dpto. Matemáticas, Universidad de Oviedo.

[14] Romero, C. (1991): *Handbook of critical issues in goal programming*, Pergamon Press. Oxford.

[15] Simon, S. A. (1979): "La toma racional de decisiones en la empresa", *Económicas y Empresariales 10*, 94-112.

[16] Tanaka, H.; Okuda, T.; Asai, K. (1974): "On fuzzy-mathematical programming", *Journal of Cibernetics 3*, 37-46.

[17] Tanaka, H.; Asai, K., (1984): "Fuzzy linear Programming Problems with fuzzy numbers", *Fuzzy sets and Systems 13*, 1-10.

[18] Verdegay, J. L. (1982): "Fuzzy mathematical programming", *Approximate Reasoning in Decision Analysis*, eds. M. M. Gupta and E. Sanchez, North-Holland, Amsterdam, 231-236.

[19] Verdegay, J. L. (1984): "A dual approach to solve the fuzzy linear programming problems", *Fuzzy sets and Systems 14*, 131-141.

[20] Zadeh, L. A. (1965): "Fuzzy sets". *Information and Control, 8*, 338-353.

[21] Zimmermann, H. J., (1976): "Description and optimization of fuzzy systems", *International Journal General Systems 2*, 209-215.

Control Fuzzy

Un Algoritmo Iterativo para la Generación de Reglas Difusas *

Antonio GONZALEZ, Raúl PEREZ
Departamento de Ciencias de la Computación e Inteligencia Artificial
Universidad de Granada
ETS de Ingeniería Informática
Avda. Andalucía, 38
18071-Granada
e-mail: fgr@robinson.ugr.es

Resumen

En este trabajo proponemos un algoritmo iterativo para la generación de reglas difusas. La base del proceso es un algoritmo genético que nos devuelve la mejor regla para identificar a un valor de la variable consecuente o variable de clasificación. Puesto que pueden existir diversas reglas para caracterizar a un único consecuente, repetiremos la ejecución del algoritmo genético varias veces, hasta que tengamos la certeza suficiente de que el sistema no requiere más reglas. Este algoritmo se aplicará en diversos problemas de clasificación utilizando distintas bases de datos.

Palabras Clave: aprendizaje automático, clasificación, reglas difusas, métodos de optimización, algoritmos genéticos.

1 Introducción

La adquisición de conocimiento es uno de los mayores problemas que se han encontrado en el desarrollo de diversos sistemas basados en el conocimiento. La elicitación de este conocimiento es una tarea compleja que consume mucho tiempo. Una alternativa a los métodos tradicionales que ha tenido recientemente un amplio desarrollo consiste en la utilización de técnicas de aprendizaje automático [11, 12].

Uno de los modelos más estudiados es el aprendizaje inductivo que trata de obtener conocimiento de un sistema utilizando un conjunto de ejemplos. A menudo este conocimiento se representa en forma de reglas que expresan las relaciones existentes entre las distintas variables del sistema. Si consideramos que algunas o todas estas variables son linguísticas [14] entonces estaremos manejando reglas difusas y tratando de aprender el comportamiento de un sistema mediante modelos difusos. Podemos encontrar en la literatura diversos trabajos que han planteado el problema de describir el comportamiento de un sistema mediante la generación de reglas difusas, ver por ejemplo [3, 9, 13].

En este trabajo describimos un algoritmo de aprendizaje sobre reglas difusas que permite obtener la estructura adecuada para cada regla. El tipo de regla que utilizaremos en el algoritmo fijará la variable consecuente o variable de clasificación y buscará las mejores reglas que describan a dicho consecuente. La parte antecedente de la regla consistirá en una conjunción de una o más variables, mientras que la parte consecuente indicará el valor que se asignará a los ejemplos que emparejen con la parte antecedente. Además permitiremos que en cada miembro de la conjunción haya una disyunción interna, es decir, permitiremos que el valor que se pueda asignar a una variable sea un conjunto de valores. Por ejemplo una regla podría tomar la siguiente forma:

Si (X=Muy_Grande) y (Y=Muy_Pequeño o Pequeño o Mediano)
Entonces (Z=Aproximadamente 0)

Este tipo de regla es básico en el modelo que estamos desarrollando para aprender la estructura de la misma, ya que si el algoritmo decide que la mejor regla para un consecuente es la que toma en una variable concreta todo el dominio de variación de la variable podemos concluir que la variable es irrelevante para

*Este trabajo ha sido financiado por el CICYT dentro del proyecto TIC92-0665

describir a dicho consecuente y puede ser eliminada de la descripción de la regla, ver [2, 8] para una discusión sobre el tema. Este tipo de regla fue utilizado en [4, 7] y coincide con una modificación de la llamada *forma normal disyuntiva* (DNF) [11].

Una vez que hemos descrito el modelo de regla que vamos a utilizar el problema que nos planteamos puede formularse de la siguiente manera: encontrar para cada valor de la variable consecuente la mejor combinación de valores de la variable antecedente. Para conseguir este objetivo es necesario dar un criterio que nos permita calcular la bondad de una regla dado un conjunto de ejemplos. En [8] propusimos un criterio basado en la utilización del número de ejemplos positivos y negativos cubiertos por una regla. En concreto el criterio consiste en encontrar, para cada consecuente, el antecedente que permita obtener reglas que cubran el máximo número de ejemplos positivos y el mínimo número de ejemplos negativos.

Este criterio implica resolver dos problemas parciales:

a) Encontrar definiciones apropiadas para los conceptos de ejemplo positivo y ejemplo negativo a una regla difusa.

b) Resolver el problema de optimización propuesto para cada consecuente.

En [7, 8] se propusieron distintas soluciones para estos problemas. En este trabajo extendemos los conceptos de ejemplo positivo y negativo a una regla para poder trabajar con ejemplos que a su vez sean conjuntos difusos.

En [7] el problema de optimización fue resuelto mediante un algoritmo genético que manejaba una codificación particular de las reglas. El problema fundamental es que usualmente cada ejecución del algoritmo genético devuelve una única regla y que las soluciones devueltas por el genético estan expuestas a cambios probabilísticos que pueden conducir al mismo a explorar zonas alejadas del óptimo. Por todo ello, en este trabajo volveremos a utilizar el algoritmo genético para resolver el problema de optimización asociado, pero proponemos un algoritmo de aprendizaje más complejo que resuelve al menos parcialmente los problemas anteriores.

Este algoritmo se probará sobre distintas bases de datos en diversos problemas de clasificación.

2 Algoritmo de Aprendizaje

Este algoritmo trata de corregir los problemas presentados anteriormente por el método de optimización. Para ello, introduciremos dos validaciones locales a distintos niveles en el proceso de aprendizaje.

- **Validación de una solución simple o** *Test de Bondad.* Este "Test de Bondad" corrige el comportamiento del algoritmo de optimización que puede tender a evolucionar hacia soluciones no óptimas en ejecuciones aisladas debido a sus componentes probabilísticas. Una implementación simple de este test es repetir un número de veces el algoritmo de optimización y quedarnos con la mejor solución de entre las obtenidas.

- **Validación del conjunto de soluciones para una clase o** *Test de Completitud.* Este "Test de Completitud" trata de determinar cuando el número de reglas obtenidas para una clase es suficiente para cubrir todos los ejemplos de dicha clase en el conjunto de entrenamiento. Este criterio de completitud admite muchas posibles implementaciones, entre ellas, nosotros hemos optado por una muy simple que consiste, en suponer cierto el test cuando se cumple alguna de las dos siguientes condiciones:

  a) que el valor asignado a la mejor regla devuelta por el Test de Bondad no supere un umbral δ.

  b) que la regla obtenida no se adecúe a más de un cierto nivel λ para ninguno de los ejemplos del conjunto de entrenamiento.

El algoritmo de aprendizaje consta de los siguientes pasos:

1. Sea REGLAS un conjunto de reglas, inicialmente vacío. Sea E un conjunto de ejemplos de entrenamiento. Sea B un caso de la variable consecuente.

2. Ejecútese el **algoritmo de optimización** para encontrar una regla R que tenga como consecuente B y que represente al conjunto de ejemplos E.

3. **Mientras** R no supere el "Test de Bondad" **ir al paso 2**. En caso contrario añadir R al conjunto de reglas REGLAS y eliminar los ejemplos de E que estén cubiertos a un cierto nivel por la regla R. Este proceso es necesario para permitir al algoritmo de optimización la localización de nuevas soluciones sobre el resto de ejemplos existentes en el conjunto de entrenamiento que tengan algún grado de pertenencia a la clase sobre la que se esta buscando.

4. **Mientras** REGLAS no supere el "Test de Completitud" para la clase B de la variable consecuente **ir al paso 2**. En caso contrario añadir todos los ejemplos eliminados en algún proceso anterior al conjunto de entrenamiento y tomar un nuevo caso B de la variable consecuente. Si se han recorrido todos los casos simples de la variable consecuente, **ir al siguiente paso** en caso contrario **ir al paso 2** a buscar las reglas que describen un nuevo consecuente B.

5. Ordenar las reglas obtenidas en REGLAS de más especificas a más generales.

3 Módulo de Optimización: Algoritmos Genéticos

El módulo central del algoritmo descrito en la sección anterior es un algoritmo de optimización que devuelve la mejor regla en función de la información disponible. Utilizaremos un algoritmo genético [10] similar al propuesto en [7] pero modificando el concepto de compatibilidad entre ejemplos y antecedentes y consecuentes de las reglas allí descritas.

3.1 Modelo general

Consideremos la siguiente regla:

$$Si\ X_1\ es\ A_1\ y \ldots y\ X_p\ es\ A_p\ entonces\ Y\ es\ B$$

donde X_i es una variable sobre un referencial U_i, el valor de A_i es un subconjunto de D_i, siendo D_i un dominio difuso compuesto por subconjuntos difusos sobre un referencial U_i y asociado a la variable antecedente X_i y B $\in$ D', siendo D' el dominio difuso de la variable consecuente Y con referencial V. Notaremos por $R_B(A_1, \ldots, A_p)$ a la regla anteriormente escrita.

Sea el conjunto de ejemplos $E = \{e_1, \ldots, e_k\}$ en donde permitimos que cada e_i pueda ser un conjunto difuso sobre el referencial U_i, es decir, $e_i \in \widetilde{\mathcal{P}}(U_1) \times \widetilde{\mathcal{P}}(U_2) \times \ldots \times \widetilde{\mathcal{P}}(U_p) \times \widetilde{\mathcal{P}}(V)$. Las componentes del vector e_i son

$$(ex_{i1}, ex_{i2}, \ldots, ex_{i_p}, ey_i)$$

donde $ex_{ij} \in \widetilde{\mathcal{P}}(U_j)$ y $ey_i \in \widetilde{\mathcal{P}}(V)$.

El problema planteado consiste en: Para cada valor B de la variable consecuente Y, encontrar una combinación de valores X_1 es A_1 y ... y X_p es A_p de las variables antecedente que mejor describan, a través de una regla y el conjunto de ejemplos E, a dicha variable consecuente.

Se hace preciso establecer un criterio que nos determine la bondad de una regla. El criterio que usaremos será el siguiente: Una regla representa a un conjunto de ejemplos si cubre muchos ejemplos positivos y pocos ejemplos negativos.

Para aplicar este criterio tenemos que definir una medida que determine si un ejemplo es positivo o negativo a una regla dada, y además una forma de calcular el número de ejemplos positivos y negativos a una regla. Estos conceptos se basan en la siguiente definición de compatibilidad entre conjuntos difusos.

Definición 1: *Compatibilidad entre conjuntos difusos*

*Sean a y b dos conjuntos difusos sobre un referencial U y * una t-norma, definimos la siguiente función de compatibilidad:*

$$\sigma(a, b) = \sup_x \{\mu_a(x) * \mu_b(x)\}$$

Para poder aplicar esta definición a nuestro problema, necesitamos extender el concepto anterior para trabajar con la compatibilidad entre dos conjuntos de etiquetas difusas.

Definición 2: *Compatibilidad entre conjuntos de etiquetas difusas*

Sean Dom_1 y Dom_2 dos dominios compuestos por conjuntos difusos sobre un referencial U, y sean dos conjuntos de etiquetas $F_1 \subseteq Dom_1$ y $F_2 \subseteq Dom_2$, definimos la siguiente función de compatibilidad:

$$\sigma(F_1, F_2) = \sup_{a \in F_1} \sup_{b \in F_2} \sigma(a, b)$$

Esta definición asigna a dos conjuntos de etiquetas difusas la mejor compatibilidad parcial entre sus elementos.

Para aplicar el concepto de compatibilidad a nuestro problema, extendemos esta definición al producto cartesiano. Para ello, consideremos un ejemplo $e_i = (ex_{i1}, ex_{i2}, \ldots, ex_{ip}, ey_i)$ donde $ex_i \in \widetilde{\mathcal{P}}(U_i)$ y una regla $R_B(A)$ donde $A = (A_1, \ldots, A_p)$, $A_i \subseteq D_i$ y $B \subseteq D'$, la compatibilidad entre ejemplo y antecedente se construye utilizando la definición anterior de la siguiente forma

$$\beta(e, A) = *_j\left(1 - \frac{\sigma(ex_j, \overline{A_j})}{\sigma(ex_j, D_j)}\right)$$

y la compatibilidad entre ejemplo y consecuente se define como:

$$\beta(e, B) = 1 - \frac{\sigma(e_y, \overline{B})}{\sigma(e_y, D')}$$

Puesto que la normalización de la compatibilidad parcial anteriormente definidas es interpretable como una distribución de necesidad, podemos dar una interpretación global la función de la siguiente manera: la función $\beta(e, A)$ se obtiene a través de una combinación mediante una t-norma de la necesidad de cada componente del antecedente condicionada a su respectiva componente del ejemplo.

$$\beta(e, A) = *_j Nec(A_j | ex_j)$$

3.2 Ejemplos Positivos y Negativos

Las definiciones que utilizaremos en este trabajo serán las ya descritas en [7] pero modificando el operador de compatibilidad, de foma que dada una regla $R_B(A)$ y un conjunto de ejemplos $E = \{e_1, e_2, \ldots, e_k\}$, podemos definir:

- Conjunto de ejemplos positivos a la regla $R_B(A)$

$$E^+(R_B(A)) = \{(e_i, \beta(e_i, A)^*\beta(e_i, B))/e_i \in E\}$$

- Conjunto de ejemplos negativos a la regla $R_B(A)$

$$E^-(R_B(A)) = \{(e_i, \beta(e_i, A)^*\beta(e_i, \bar{B}))/e_i \in E\}$$

Definimos ahora la estimación del número de ejemplos positivos a una regla $n^+(R_B(A))$ como el cardinal del conjunto $E^+(R_B(A))$, tomando el cardinal como

$$\mid E^+(R_B(A)) \mid = \sum_{e \in E} \mu_{E^+(R_B(A))}(e)$$

De forma análoga podemos definir la estimación del número de ejemplos negativos a una regla $n^-(R_B(A))$ como

$$\mid E^-(R_B(A)) \mid = \sum_{e \in E} \mu_{E^-(R_B(A))}(e)$$

3.3 Función a optimizar por el Genético

El criterio que consideraremos para determinar la bondad de una regla puede ser expresado como maximizar el conjunto de ejemplos positivos y minimizar el conjunto de ejemplos negativos. Como ya se vio en [7] este problema de optimización biobjetivo no es fácil de resolver. Por tanto, definimos un nuevo criterio que es una aproximación al definido anteriormente pero de más fácil resolución. Este nuevo criterio puede ser expresado como maximizar el número de ejemplos positivos pero no permitiendo que el número de ejemplos negativos pueda superar un umbral determinado. Una función que recoge la idea anterior y que notaremos por FITNESS($R_B(A)$) es

$$f = \begin{cases} n^+ & \text{si } n^- \leq k_1 n^+ \\ \frac{k_2 n^+ - n^-}{k_2 - k_1} & \text{si } k_1.n^+ \leq n^- \leq k_2.n^+ \\ 0 & \text{en otro caso} \end{cases}$$

donde f es FITNESS($R_B(A)$), n^+ es $n^+(R_B(A))$, n^- es $n^-(R_B(A))$ y se ha de cumplir que $k_1, k_2 \in [0,1]$ y $k_2 \leq k_1$

El parámetro k_1 indica el umbral de ruido permitido o tolerable, y el parámetro k_2 indica el umbral de ruido máximo admisible.

Esta función actua considerando que pequeños porcentajes del número de ejemplos negativos con respecto al número de ejemplos positivos para una regla dada son provocados por el propio ruido existente en el conjunto de ejemplos de entrenamiento. Sin embargo, cuando el número de ejemplos negativos es excesivo (es decir, los ejemplos negativos dada una regla superan el umbral "duro"), la función no lo considera como ruido, sino como que la regla no es buena para representar al conjunto de ejemplos. Entre ambos umbrales la función decrementa linealmente el valor dado por número de ejemplos positivos en función del número de ejemplos negativos.

Finalmente, como proceso de optimización del algoritmo de aprendizaje utilizaremos el algoritmo genético descrito en [7]

4 Resultados Experimentales

Hemos probado el comportamiento del algoritmo propuesto sobre tres bases de datos, dos de las cuales han sido generadas de forma artificial y la otra es la famosa base de datos *Iris Plants Database* que notaremos por IRIS creada por *R.A. Fisher* y utilizada en multitud de publicaciones tales como [6, 5, 1]

Una de las bases de datos artificiales, que notaremos por PRUB1, ha sido generada por simulación a partir de un conjunto de reglas difusas elegidas a priori. La otra base de datos, que notaremos por PRUB2, ha sido generada a partir de las mismas reglas difusas que definían PRUB1 pero añadiendo un porcentaje de ruido a los datos. También, hemos usado algoritmos clásicos de aprendizaje tales como CART y C4 para poder comparar la bondad del sistema desarrollado.

La discretización en término de etiquetas difusas de los dominios continuos de los atributos para cada una de las bases de datos se ha realizado sin información específica del problema, en el que se ha recurrido a un algoritmo simple de reparto uniforme del espacio de cada atributo. El número de etiquetas linguíticas varía entre cinco y siete.

Los parámetros utilizados para la obtención de resultados son.

Parámetros	Valor
k_1	0.01
k_2	0.1
λ	0.8
δ	0.1
Tamaño Población	30
Número Iteraciones	1000
t-norma	min

En la siguiente tabla se muestran los porcentajes de aprendizaje para cada una de las bases de datos y cada uno de los algoritmos probados. El algoritmo descrito en esta comunicación es el notado FL.

Método Aprendizaje	PRUB1	PRUB2	IRIS
CART	97,45	97,1	92,2
C4	97,69	97,85	92,2
FL	100	100	94,74

En la tabla se desprende el buen comportamiento de algoritmo aquí descrito ya que el porcentaje de apredizaje de FL es superior en todas las bases de datos a los obtenidos por los otros métodos clásicos.

Los algoritmos de aprendizaje detectan las regularidades más importantes existentes en una base de

datos para describir su comportamiento. Supongamos dos bases de datos de la cuales una ha sido generada a partir de la otra introduciéndole ruido. Las regularidades existentes en la primera base de datos deben aparecer en la segunda aunque más difuminadas por el efecto del ruido. Parece una cualidad muy importante que un método que detecte las reglas que describen a la segunda base de datos, estas mismas reglas sean aplicables con buenos resultados a la primera. Justo en esta situación se encuentran las bases de datos PRUB1 y PRUB2, por ello, realizamos la prueba e inferimos con los resultados obtenidos sobre PRUB2 en los conjuntos de entrenamiento y test de la base de datos PRUB1. Los resultados obtenidos son los siguientes:

Método	Training	Test
CART	98,35	99,48
C4	97,69	97,45
FL	100	100

El cumplimiento de esta propiedad asegura un correcto comportamiento del método de aprendizaje ante el ruido, ya que aprende las reglas más relevantes y sólo genera nuevas reglas para cubrir los casos especiales provocados por el ruido cuando no conlleva fuertes contradicciones.

5 Conclusiones

En esta comunicación hemos propuesto un algoritmo de aprendizaje de reglas difusas en entornos inciertos. Aplicando este algoritmo a un conjunto de bases de datos, dos artificiales y una real, se han obtenidos buenos resultados comparándolos con otros métodos clásicos de aprendizaje. Se ha observado que la bondad de los resultados depende de los valores elegidos para los parámetros k_1 y k_2. Esto nos lleva a estudiar el comportamiento de la función FITNESS y buscar métodos simples de bajo costo computacional que nos permitan estimar de forma correcta los parámetros mencionados.

Podemos deducir del comportamiento observado al algoritmo de aprendizaje que existen algunos tipos de problemas para los cuales es recomendable el uso de este tipo de métodos de aprendizaje que utilizan reglas difusas para la descripción del sistema. Su utilización puede ser provechosa, ya sea por la obtención de buenos resultados medidos en términos de porcentaje de aprendizaje, o bien, porque aún teniendo un porcentaje de aprendizaje menor que otros métodos se ha de valorar la interpretación semántica de las reglas obtenidas por estos métodos frente a los obtenidos por otros. Esta capacidad de "fácil interpretación" de las soluciones lo hacen aplicable a ciertos campos donde tan importante es el porcentaje de aprendizaje, como la comprensión de los resultados obtenidos.

Bibliografía

[1] Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System Structure and Classification Rule for Recognition in Partially Exposed Environments". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 1, 67-71

[2] De Jong, K., Spears, W.M., Gordon, D.F., Using genetic algorithms for concept learning, Machine learning, 13, pp. 161-188, (1993).

[3] De Mori, R., Saitta, L., Automatic learning of fuzzy naming relations over finite languages, Information Sciences 21, pp.93-139, (1980).

[4] Delgado, M., González, A., An inductive learning procedure to identify fuzzy systems, Inter. Journal for Fuzzy Sets and Systems 55 121-132 (1993).

[5] Duda,R.O., and Hart,P.E. (1973) Pattern Classification and Scene Analysis. (Q327.D83) John Wiley and Sons. ISBN 0-471-22361-1. See page 218.

[6] Fisher,R.A. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to Mathematical Statistics" (John Wiley, NY, 1950).

[7] González, A., Pérez, R., Un modelo experimental para determinar la estructura de una regla, Comunicaciones del Tercer Congreso Español sobre Tecnología y Lógica Fuzzy, pp. 17-24, (1993).

[8] González, A., Pérez, R., Verdegay, J.L., Learning thes structure of a fuzzy rule: a genetic approach, Proc. First European Congress on Fuzzy and Intelligent Technologies, Vol. 2, pp. 814-819, (1993)

[9] González, A., A learning methodology in uncertain and imprecise environments, aceptado en el International Journal of Intelligent Systems.

[10] Goldberg, D.E., Genetic algorithms in search, optimization and machine learning, Addison-Wesley, New York, (1989)

[11] Michalski, R.S., A theory and methodology of inductive reasoning, in R.S. Michalski, J. Carbonell and T. Mitchel (Eds.), Machine Learning: An artificial intelligence approach (Vol. 1). San Mateo, CA: Morgan Kaufmann.

[12] Quinlan, J.R., Induction of decision trees, Machine learning, 1 (1), pp. 81-106, (1986).

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

[14] Zadeh, L.A., The concept of a linguistic variable and its applications to approximate reasoning, Part I Information Sciences vol. 8, pp. 199-249, Part II Information Sciences vol. 8, pp. 301-357, Part III Information Sciences vol. 9, pp. 43-80, (1975)

SUPERVISION DE REGULADORES ADAPTATIVOS CLASICOS MEDIANTE TECNICAS DIFUSAS

Ramón Ferreiro García y Benigno Rodriguez Gómez
Universidad de La Coruña, Dep. Electrónica y Sistemas
E.S.Marina Civil, Paseo de Ronda, 51- 15011
Tel: 34. 81. 25 67 00. Fax: 34 81 25 15 68

Resumen

Este trabajo describe la tarea de supervisión de plantas dotadas de lazos de control basado en reguladores adaptativos clásicos. La tarea de supervisión incluye la detección del comportamiento dinámico, y en función del mismo se toman decisiones sobre la conveniencia de realizar el autoajuste, como resultado de la defuzificación de una base de reglas propuesta a tal efecto. El resultado de la base difusa de reglas es aplicado en modo secuencial a una base de reglas determinista booleana, cuya conclusión sirve para indicar el estado del regulador en la planta.

Palabras clave: Supervisión, lógica difusa, autoajuste, diagnosis, memoria asociativa difusa.

1 Introducción

Este trabajo describe el problema de supervisión el comportamiento de los reguladores convencionales adaptativos usando técnicas de supervisión experta y difusa.

Las perturbaciones en lazos de realimentación son una buena prueba de desajuste de los parámetros del regulador. Tales perturbaciones pueden entrar al lazo de control desde una fuente externa, pero también pueden ser generadas dentro del lazo de control debido a fricción de los elementos finales de control como actuadores etc. Por tanto el objetivo principal es el establecimiento de los algoritmos de detección de tales perturbaciones así como aquellas acciones necesarias para eliminarlas. La deteccion de perturbaciones está encaminada a la toma de decisiones en la tarea de ajuste de parámetros [2].

La asumción de perturbaciones en el bucle de realimentación concierne a todas las señales introducidas en el lazo de realimentación, tales como cambios del valor de consigna, ruido de alta frecuencia añadido a señal de medida, cambios de carga o variación de parámetros de la planta.

Dependiendo del origen de las perturbaciones, resulta de gran interés el criterio de eliminarlas de ser posible. Esto puede ser llevado a cabo de diferentes formas dependiendo de su origen [1]. Para perturbaciones de origen externo, es

Este trabajo está financiado por el Ministerio español de Educación y Ciencia bajo el código de proyecto número TIC92-0267-C02-02 of the C.I.C.Y.T department.

necesario eliminar las causas. Si ello no es posible se aplicará compensación por prealimentación. Si las perturbaciones son generadas dentro del lazo de realimentación por fricción del actuador, este debe ser corregido convenientemente.

El proceso, siendo una aplicación industrial clásica está constituido normalmente como una característica de filtraje de paso bajo, lo que significa baja ganancia a altas frecuencias.

Las perturbaciones de carga son de gran importancia cuando su frecuencia esta en la vecindad de la frecuencia ultimada debido a su bajo comportamiento en control adaptativo. El regulador adaptativo interpreta la frecuencia de tales perturbaciones como una alta ganancia de lazo [1]. Por tanto, es de gran importancia la detección de las oscilaciones de esta frecuencia y supervisar el regulador en el sentido de evitar el ajuste de parámetros debido a esta causa. Un problema añadido aparece cuando la frecuencia ultimada del sistema cambia debido a la variación de parámetros del sistema controlado.

Por tanto es necesario investigar como las perturbaciones a la carga V(s) que entran al lazo de control son transferidas a la señal de medida Y(s). La función de transferencia entre la salida Y(s) y la entrada r(s) y entre la salida y la perturbación de origen externo V(s) están dadas en (1) y (2) en forma de diagrama de bloques y están ilustradas en la figura 1

$$G_{yr}(s) = C(s)P(s)/[1+C(s)P(s)] \qquad (1)$$

$$G_{yv}(s) = P(s)/[1+C(s)P(s)] \qquad (2)$$

La salida del sistema es debida a las dos entradas. Existe otra causa de mal comportamiento debido a la variación de parámetros en la función de transferencia del proceso.

$$Y(s)=[V(s)+r(s).C(s)]P(s)/[1+C(s)P(s)] \qquad (3)$$

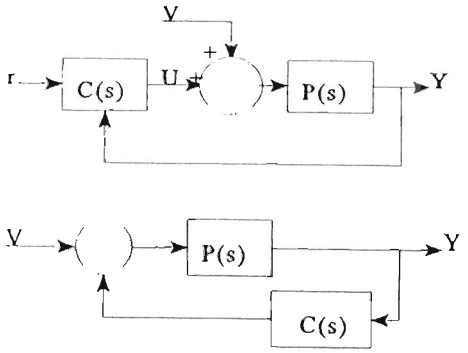


Fig. 1 Proceso perturbado.

La causa de una respuesta del estado estacionario variable debida a cambios en la referencia de entrada, variación de los parámetros del proceso y variaciones de carga. Excepto los cambios de consigna en la señal de referencia de entrada las cuales pueden ser rechazadas mediante compensación por prealimenmtación, los demás cambios tienen que ser detectados mediante el análisis de la respuesta del sistema por identificación del exceso de carga y la frecuencia de las oscilaciones.

El procedimiento de detección de las perturbaciones debidas a carga puede ser realizado mediante el análisis de la magnitud y el valor absoluto del error (IAE) entre pasos sucesivos por error cero de la variable de control. Para los períodos de buen control la magnitud del error de control es pequeño. Se asume que el controlador emite acción de control, por tanto el error promedio es nulo.
Para mejorar el procedimiento de detección es necesario establecer un valor límite adecuado para el IAE. La elección del valor límite está sujeto a un compromiso entre la demanda entre una alta probabilidad de detección y la demanda de una baja probabilidad de obtener falsas detecciones. Asumiendo conocido el valor del

tiempo integral T_i de regulador, el límite para el IAE ha sido establecido en [1] como

$$IAE_{lim} = 1/w_i, \text{ donde } w_i = 2*Pi/T_i \qquad (4)$$

Tal método requiere que el error de control sea una senoide pura, lo cual hace el método impracticable en la industria, debido a causas obvias tales como la presencia de perturbaciones a la carga entra las causas mas importantes.

2 Detección de oscilaciones y carga

En este trabajo se adopta un criterio práctico con el objeto de evitar la manipulación de señales de características standard, lo cual en tiempo real son imposibles de generar y aplicar en simultaneidad con el proceso de control porque varia los objetivos de control. Por tanto es mejor seleccionar un valor límite para el IAE para cada proceso de acuerdo a su máximo soportado sin deteriorar el control debido a perturbaciones de carga.

La detección de oscilaciones en el lazo de realimentación está basada en la conclusión de que existen oscilaciones si la frecuencia de las perturbaciones a la carga se hace alta. Esto es, el comportamiento de la dinámica de control es supervisada durante el tiempo T_{sup}. Si la cantidad de oscilaciones N_{osc} detectadas debidas a carga excede el valor límite denotado como N_{lim} durante ese tiempo, entonces, la conclusión es que hay presencia de oscilaciones. Como consecuencia de este razonamiento se establecen las reglas siguientes:

If IAE > IAE_{lim} then existen pertuebaciones a la carga

If $N_{osc} > N_{lim}$ then las perturbaciones de carga son oscilación.

3 Criterios de decisión sobre el ajuste del regulador

El criterio para decidir sobre la conveniencia de realizar el autoajuste del regulador depende de la capacidad deseada del mismo para seguir una señal de referencia de entrada o la capacidad de rechazar perturbaciones de carga. Debe existir cierto compromiso entro los dos conceptos los cuales están en oposición entre sí. La detección de variación en la ganancia de lazo u otro parámetro significativo es generalmente de mucha ayuda. Para la adquisición de tal información se necesita el modelo matemático nominal del proceso a controlar, con el cual es posible la evaluación de las variaciones de ganancia de lazo a partir del análisis de la respuesta temporal de ambos, el modelo y el proceso en tiempo real, sin la necesidad de aplicación de ningún método clásico de identificación. Se intenta simplificar las estrategias de supervisión con objeto de hacer aplicable y fiable al algoritmo de ajuste. Por esta razón, resulta de gran importancia la mejora de la tarea de supervisión sobre la base del modelo de comportamiento del ser humano.

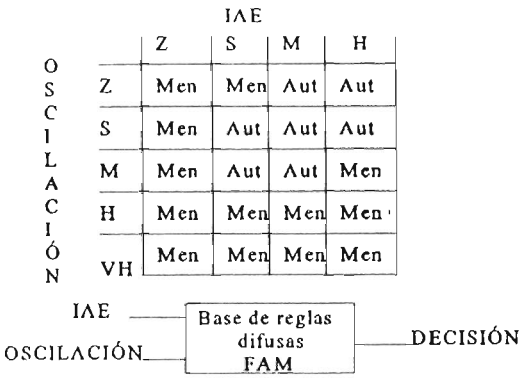


Fig. 2 Base de reglas difusas.

Para ello va a ser aplicado una memoria asociativa difusa [3], FAM (Fuzzy asociative

memory), consecuencia de una base de reglas para decidir cuando debe ser ajustado el controlador. El hipercubo resultante tiene dos dimensiones en la familia de variables de entrada:

-- presencia de carga bajo el criterio IAE
-- presencia de oscilaciones

Por tanto, el universo de discurso de ambas variables expresado en funciones de pertenencia es cero Z, pequeño S, mediano M y alto H para la carga, mientras que para las oscilaciones son cero Z, pequeño S, mediano M, alto H y muy alto VH. El universo de discurso de la variable de salida de la base difusa de reglas contiene dos funciones de pertenencia, a saber, un emisor de mensajes Men, y orden de autoajuste (Aut) [4].

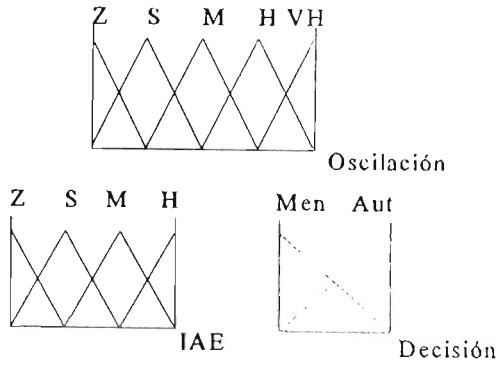


Fig. 3 Funciones de pertenencia

La función de pertenencia mensaje (Men), contiene tres salidas deterministas: el indicador de dinámica del proceso normal, presencia de ruido en la medida (rui) y indicación de puesta en modo manual (MM). El procedimiento para la toma de decisiones consiste en la aplicación de la base de reglas de acuerdo a la figura 3 en donde aut es la salida defuziticada que ordena la operación de autoajuste, mientras que A es la conclusión optativa de la base de reglas que aplicada como entrada a una base de reglas determinista, en unión de las demás variables, conforman el motor de inferencia para la toma de decisiones en modo determinista, la cual permite decidir el mensaje como consecuencia del valor lógico de las entradas bajo una memoria determinista asociativa (DAM). La toma de decisiones bajo la DAM establecida se muestra en la figura 4, en la cual, existen tres salidas como consecuencia de las variables de entrada.

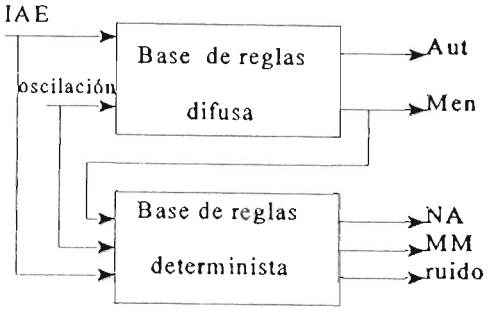


Fig 4. Generador de decisiones.

Finalmente, el algoritmo de supervisión se muestra en la figura 5 en donde la tarea global consiste en supervisar secuencialmente una serie de reguladores de distintos procesos de la planta, los cuales serán barridos o rastreados bajo un procedimiento secuencial. El conocimiento histórico acerca de cada proceso tiene que estar disponible con objeto de comparar los parámetros de un regulador ajustado con los correspondientes parámetros del proceso nominal para validar el ajuste. Con esta información es posible la discriminación de autoajustes de dudosa precisión.

Puesto que se desconoce el origen del exceso de carga y de presencia de oscilaciones debido a que puede ser por grandes variaciones de parámetros o exceso de carga así como excitaciones en el límite del rango de la frecuencia ultimada, se hace necesario después

de una sesión de autoajuste chequear si la nueva configuración paramétrica del PI(D) está dentro de límites admisibles. Tales valores tienen que haber sido determinados bajo la asumción de un entrenamiento en tiempo real del proceso baja la máxima variación de parámetros y cambios de carga. Por ejemplo, la masa de un buque solo puede variar para un buque en particular desde 3/4 hasta 4 veces su valor nominal y la carga debido al viento oleaje y corrientes en la misma dirección solo va desde 0 a 1/3 de su máxima fuerza de control. Para cada variación de carga o de parámetros del proceso dentro de los márgenes de variación definidos tiene que existir un juego de posibles valores para el regulador. Si después de una sesión de autoajuste y consecuente chequeo de los valores del regulador, estos están fuera de los límites admisibles, concluimos que algo anormal ha ocurrido y se interpreta como la necesidad de una puesta en manual para corregir el defecto.

La tarea de supervisión consiste en un rastreo secuencial de cada lazo controlado de la planta mediante reguladores adaptativos clásicos para detectar cualquier anomalía dinámica de la información adquirida por el sistema supervisor. A tal efecto, el proceso completo de supervisión para cada lazo consiste en los siguientes pasos:

-- entrar al proceso i y detectar estados de IAE (carga) y presencia de oscilaciones por los algoritmos dados en [1] y [2].
-- procesar la base de reglas difusas con los datos adquiridos, obteniéndose una conclusión parcial del problema:se ordena el autoajuste o se habilita la base de reglas booleanas.
-- actuar según las conclusiones de la base de reglas booleanas de acuerdo a las siguientes reglas:
(a)if Men and oscilación son cero (zero output) then la tarea de este ciclo de supervisión está terminada.

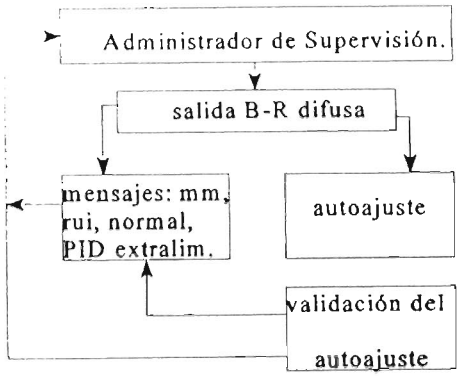


Fig. 5 Tarea de Supervisión.

(b)if Men and IAE=0 and osc=umbral ruido, then hay ruido en el lazo de realimentación.
(c)if Men and IAE $>>0$ then modo manual.
Una vez finalizado el autoajuste, el algoritmo de supervisión comprueba los parámetros (PI)D comparando los nuevos valores con los nominales y deducir si están dentro de los límites admisibles. Si es así, la tarea de autoajuste de este lazo se da por finalizada, saltando al siguiente lazo de control. Por el contrario, si los nuevos parámetros están fuera de los límites admisibles se concluye que los parámetros del proceso o la carga han cambiado demasiado y se pasa al proceso de reglas booleanas según muestra la figura 4.

4 Implementación

Se realiza la tarea de supervisión sobre un sistema controlado, el cual consiste en un proceso de nivel controlado por realimenmtación mediante un controlador adaptativo clásico PI(D) Shimaden del tipo SR25 dotado de medios de comunicación via bus serie y un protocolo de comunicación binario de 8 bits que permiten fijar ordenar el autoajuste y hacer cumplir así las conclusiones de las bases de reglas.

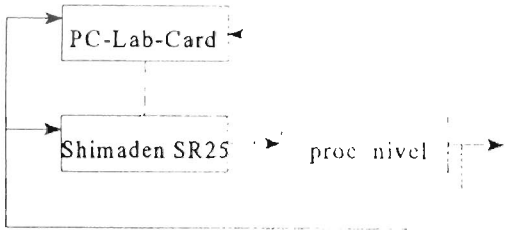


Fig. 6 Supervisión del proceso de nivel controlad por un regulador Shimaden SR25

5 Conclusión

La generación de información adecuada para introducir en la bese difusa de reglas es un problema resuelto en [1] y [2] entre otros. La aplicación eficiente de tal información es afrontada en este trabajo como una aplicación de la lógica difusa a la tarea de diagnosis de plantas controladas mediante reguladores autoajustables clásicos. Se desprende de la aplicación que las reglas de supervisión son híbridas, esto es, no todas reglas son aplicadas en modo difuso ni todas son aplicadas en modo booleano, y que debe ser establecida una estrtategia que resuelve el conflicto entre las reglas difusas y las booleanas en función de las necesidades del problema, para obtener una solución aceptable.

References

[1] Hagglund, (1993)
Distrubance supervision in feedback loops
International conference on Fault Diagnisis, TOOLDIAG'93
Toulouse, France. pp 132-139

[2] Astrom K.J. and T.Hagglund (1984)
Automatic tuning of simple regulators with specifications on phase and amplitude margins
Automatica, Vol. 20, No.5, pp. 645-651.

[3] Bart Kosko, (1992
Neural networks and fuzzy systems
Prentice Hall International, Inc.
Chp. 8, pp 299-339

[4] Ferreiro G. R, & Rodriguez G. B. (1993)
Process Control Failure Diagnostic Fuzzy Expert System. Prepirnts of the International Conference on Fault Diagnosis.TOOLDIAG'93. Labege Congress Center. Toulouse, France
Vol I, pp. 197-206

Aprendizaje mediante la aplicación de técnicas fuzzy a espacios discretos

J. Riquelme, M. Toro
Facultad de Informática y Estadística. Universidad de Sevilla.
Av. Reina Mercedes sn. 41012 Sevilla.
Tfno. (95) 4552775. Fax (95) 4557139. E-Mail mtoro@sevaxu.cica.es

Resumen

La necesidad de aprender, *es decir, de extraer información útil de un conjunto de datos muy grande o de un sistema no completamente conocido, puede venir dada, entre otras motivaciones, como base para construir un modelo cualitativo del sistema. Un modelo que nos permita conocer de manera automática el comportamiento del sistema ante una entrada determinada. En esta comunicación se pretende mostrar una variante del algoritmo Fuzzy C-Means adaptado a la característica de que la variable de salida es discreta. A partir de éste se encontrará un modelo cualitativo que describirá el sistema mediante variables lingüísticas. Esta metodología se ha aplicado al estudio del comportamiento que, en régimen estacionario, tienen diversos sistemas dinámicos en función de sus parámetros.*

Palabras clave: Aprendizaje, modelo cualitativo, técnicas difusas, sistemas dinámicos.

1 Introducción

El análisis cualitativo de un sistema dinámico $\dot{x} = f(x,p)$ con $x \in R^n$, $p \in R^m$, es una herramienta conceptual adecuada para comprender los modos de comportamiento que pueda presentar dicho sistema dinámico en régimen estacionario. Cuando el sistema tiene alguna complejidad o el número de parámetros es grande, no existe la posibilidad de encontrar soluciones matemáticas, y debemos de recurrir a otros métodos de aprendizaje.

El razonamiento cualitativo empieza desde el momento en el que, aunque pueda ser obtenido un modelo cuantitativo completo, su complejidad hace que no proporcione información intuitiva sobre la estructura general del sistema. En [1] se expresa esta idea con el nombre de *principio de incompatibilidad*, es decir, cuando la complejidad de un sistema se incrementa, nuestra habilidad para precisar resultados significativos sobre su comportamiento disminuye. Además, otras ventajas de la obtención de un modelo cualitativo, radica en casos donde no es posible construir un modelo matemático por integración de otros subsistemas, o bien, sistemas donde sólo podemos observar las entradas y salidas.

Es, entonces, cuando intentamos automatizar la obtención de un modelo cualitativo, para el estudio de los comportamientos que en regimen estacionario puede presentar un sistema dinámico. Este proceso conlleva dos fases: en la primera se intentan encontrar resultados empíricos (en esta comunicación a partir del modelo cuantitativo) sobre el sistema creando una base de información, que será explotada en la segunda fase, consistente en aprender de esa información hasta conseguir un conjunto de reglas, que nos haga comprender mejor la evolución del sistema en función de sus parámetros.

Para lograr este propósito, en primer lugar, se hallarán cluster difusos sobre el espacio de parámetros donde podamos afirmar con cierta seguridad que el sistema tendrá un atractor puntual, un ciclo límite, o un *atractor extraño*. Con posterioridad, se hará una aproximación de este modelo difuso a un modelo cualitativo mediante una aproximación lingüística.

La metodología empleada consiste básicamente, en sortear puntos en el espacio de parámetros, y simular el sistema en esos puntos, asociando a cada simulación un código que nos represente el tipo de atractor resultante. Con estos puntos y el atractor resultante se forma una base de datos, que explotada mediante la lógica difusa nos lleve al objetivo propuesto.

La explotación de la base de datos se realiza mediante la aplicación del algoritmo Fuzzy C-Means [2] adaptado a la característica de que la variable de salida es discreta, que conlleva una serie de modificaciones sobre el desarrollo tradicional de los modelos de lógica difusa.

Los resultados obtenidos, sobre todo en el modelo difuso, con diversos sistemas dinámicos, son muy esperanzadores de encontrar una herramienta que permita prever el comportamiento en el régimen estacionario de un sistema dinámico.

2 Técnicas de aprendizaje.

El problema planteado de forma general es el de extraer información de una base de datos, formada por

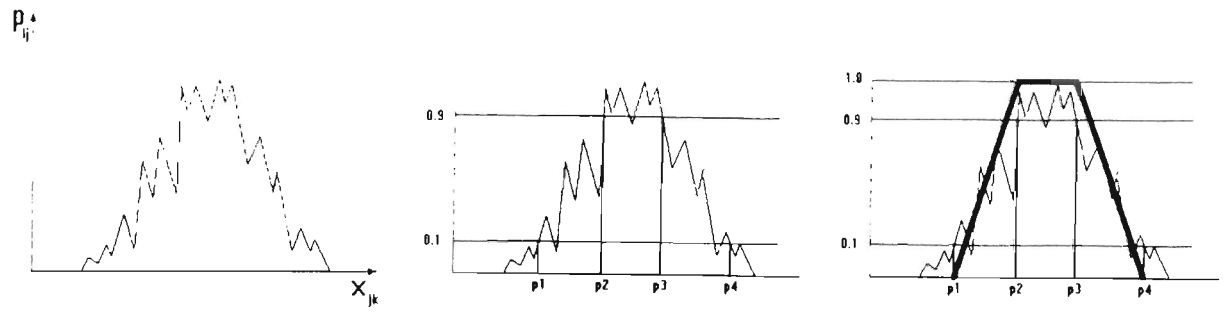


Figure 1 Construcción del conjunto difuso

```
        finpara
      sino
        para cada variable k del registro j
          si (x < p^1)
            p^1 = x
          si (x > p^4
            p^4 = x
          si (x < p^2) y (x ≥ p^1)
            p^2_ki = x_jk
          si (x > p^3) y (x ≤ p^4)
            p^3 = x
          si (p^3 > p^4)
            p^4 = p^3
      finsi
    finsi
  finpara
finpara
```

Al aplicar este algoritmo a la variable discreta que representa el atractor del modelo, puede darse el caso de que el trapecio quede reducido a un sólo punto, el valor de la variable para un determinado tipo de atractor. Hay que tener en cuenta que, en principio, no tiene sentido hablar de conjunto difuso para representar los valores de una variable discreta, sin embargo, mantendremos esta notación hasta el final del proceso, pues es uno de los factores que influyen en la optimización del modelo difuso.

4.3 Optimización de la solución obtenida.

Del procedimiento anterior obtenemos una aproximación al modelo difuso buscado que nos sirve de punto de partida para una optimización del mismo. Para calcular la "bondad" del modelo encontrado podemos estimar el resultado que se producirá para la variable de salida, "defuzificando" el modelo mediante la ecuación:

$$\hat{y} = \Sigma w_i b_i / \Sigma w_i$$

donde w_i es el grado de pertenencia de los valores de las variables de entrada de cada registro al cluster i. y b_i el centro de gravedad del cluster i para la variable de salida.

Una manera obvia de medir el error cometido con el modelo es calcular la media de los cuadrados de los errores para todos los registros de la base de datos. Sin embargo, en el caso que nos ocupa comparar directamente el valor de $\hat{y}$ con el valor real, no parece adecuado, por la naturaleza discreta del espacio de salida. Si los valores que representan los distintos tipos de atractores se encuentran "suficientemente" distanciados, podemos otorgar un intervalo de confianza alrededor de éstos para considerar la previsión como acertada. Es decir, lo que realmente nos interesa no es minimizar el error medio, sino el número total de errores cometidos en la previsión con el modelo difuso.

Pretendemos, por tanto, minimizar el número de errores cometidos con el modelo difuso, al modificar los valores que determinan los trapecios que definen los clusters iniciales. Después de probar con diversos algoritmos de optimización, es la búsqueda aleatoria el que mejores resultados nos a proporcionado.

Algoritmo de optimización:

Sea p^k_{ij} el valor del parámetro k en el trapecio que define el cluster difuso j para la variable i, con $1 \leq k \leq 4$, $1 \leq j \leq NC$, $1 \leq i \leq NV$, donde NC es el número de clusters y NV el número de variables.

```
Obténgase p^k_ij del modelo difuso inicial
Evaluar el modelo inicial
Para un numero grande de iteraciones
    Escójase de forma aleatoria unos valores α, β y ω
      con 1 ≤ α ≤ 4, 1 ≤ β ≤ NC, 1 ≤ ω ≤ NV
    Escójase de forma aleatoria un valor de ajuste va
      tal que -20 % rango ω ≤ va ≤ 20 % rango ω
    Modificar p^ω_αβ = p^ω_αβ + va
    Evaluar el nuevo modelo
    Si no es mejor deshacer el cambio
```

Es necesario hacer algunas puntualizaciones sobre el algoritmo anterior, la primera es $p^{\omega}_{\alpha\beta}$ no puede ser

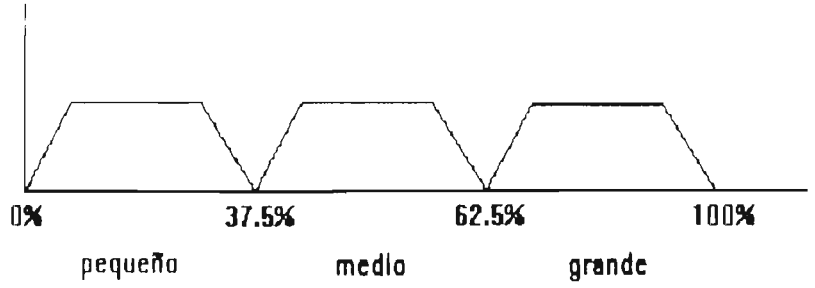


Figure 2: División del rango de valores

mayor que $p^{\omega}_{\alpha+1\beta}$, para $\alpha < 4$ o igualmente $p^{\omega}_{\alpha\beta}$no puede ser menor que $p^{\omega}_{\alpha-1\beta}$ para $\alpha > 1$, en estos casos deben hacerse iguales. La segunda es sobre la función objetivo, se ha discutido anteriormente la necesidad de minimizar el número de errores, lo que daría lugar a una función objetivo discreta, sin embargo para una mejor convergencia del algoritmo, se puede añadir una parte decimal al número de errores, consistente en la suma de los errores cometidos al estimar el valor de la variable de salida, corregida para que tome valores menores que uno. Así la función objetivo quedaría:

$$cardinal\{k/|\hat{y}_k - y_k| > \delta\} + 10^{-n} * \Sigma(\hat{y}_k - y_k)^2$$

donde δ (intervalo de confianza) y n dependen de los valores que pueda tomar la variable de salida.

Otra observación que ha de hacerse es que la optimización también se debe llevar a cabo con los clusters de la variable de salida, pues aunque estrictamente hablando no tiene sentido un conjunto difuso para definir un valor discreto, el ajuste de la variable de salida ayuda a ajustar mejor las variables de entrada. Con posterioridad en el modelo cualitativo, los intervalos difusos para la variable discreta se ignorarán, teniéndose en cuenta sólo el valor discreto que indique el cluster inicial que proporciona el algoritmo FCM

5 Aplicación.

5.1 Modelo cualitativo.

Una vez conseguido un modelo difuso óptimo, nos planteamos obtener un modelo cualitativo mediante una aproximación lingüística a los clusters difusos obtenidos. La técnica desarrollada consiste en medir el grado de equiparación entre los conjuntos conseguidos y otros previamente establecidos según las características de las variables del modelo.

Entenderemos por modelo cualitativo un conjunto de reglas definidas por la gramática siguiente:

```
<modelo>          ::= <lista_reglas>
<lista_reglas>    ::= <lista_reglas>
                    | <regla>
<regla>           ::= <lista_premisas>
                      ENTONCES <conclusion>
<lista_premisas>  ::= <lista_premisas>
                      Y <premisa>
                    | <premisa>
<premisa>         ::= SI PARAMETRO ES
                      <modificador> <termino>
<conclusion>      ::= <comportamiento>
```

Donde *comportamiento* es alguno de los atractores posibles en un sistema dinámico y cada *termino* viene dado por un conjunto difuso definido sobre el intervalo de variación de cada variable, asignándole a cada uno un nombre. Ver Fig. 2. Estos serían los términos lingüísticos básicos, que podrían venir acompañados por modificadores, para un mejor ajuste del modelo cualitativo. Definimos la unión entre modificador y termino lingüístico básico como una operación que transforma el conjunto difuso básico en otro (Fig. 3), obteniendo así una colección de conjuntos difusos preestablecidos con la que comparar los clusters del punto anterior (Fig. 4).

La medida de similaridad entre dos conjuntos difusos usada es la propuesta en [11]: dados C_1 y C_2 su grado de similitud es

$$S(C_1, C_2) = \| C_1 \cap C_2 \| / \| C_1 \cup C_2 \|$$

La metodología a seguir es comparar los conjuntos difusos obtenidos para las variables de entrada, con la colección de conjuntos difusos resultado de aplicar todos los modificadores posibles a todos los términos lingüísticos básicos. Como resultado de esta comparación obtendremos un grado de similitud máxima entre cada cluster del modelo difuso y un término lingüístico, probablemente con algún modificador, consiguiendo las premisas de las reglas del modelo. De la variable de salida obtenemos la consecuencia de cada regla, basándonos, como ya se ha señalado, en el valor discreto alrededor del que se situaban los cluster resultado del algoritmo FCM para la variable de salida.

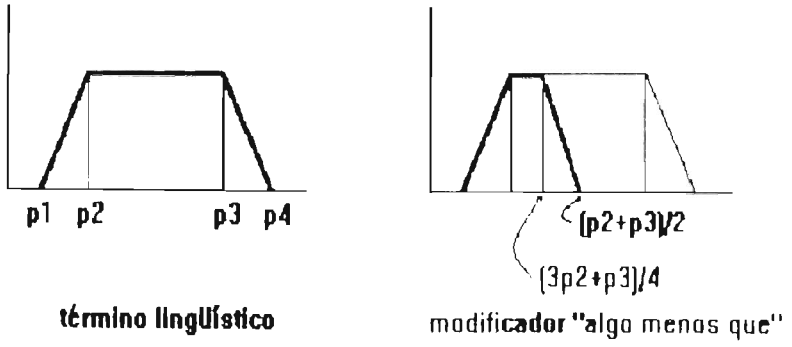


Figure 3: Obtención del conjunto fuzzy de un termino modificado.

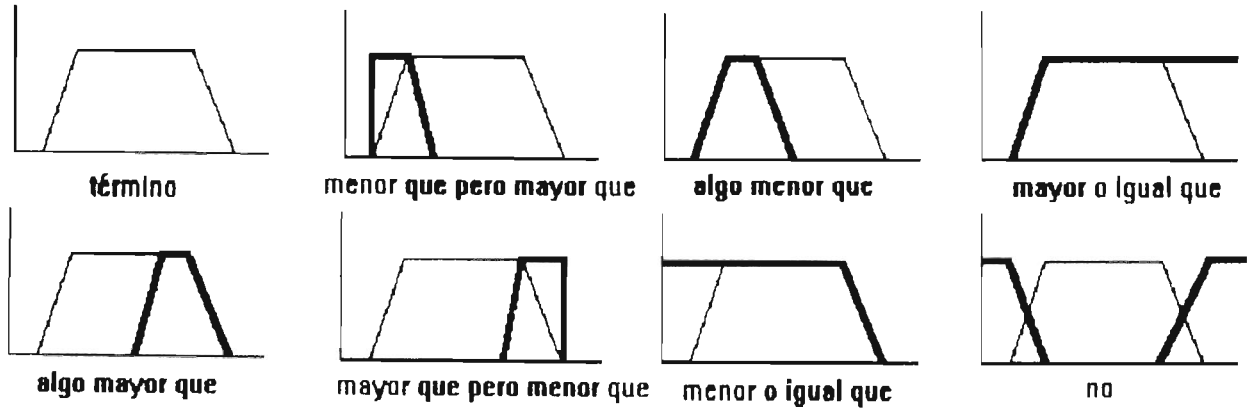


Figure 4: Modificadores propuestos.

Hemos aplicado esta metodología a dos sistemas dinámicos que representan modelos económicos.

5.2 Modelo simple.

El primero es un modelo económico cíclico simple propuesto en [13] y formulado en [14], es un modelo con dos variables de nivel y dos parámetros. Disponemos de una base de datos con 100 registros, cada uno con dos variables de entrada y una variable discreta de salida con dos valores posibles según el modelo tuviera un estacionario con punto de equilibrio o ciclo límite.

Los resultados del algoritmo de optimización del modelo difuso fueron inmejorables, al obtenerse un modelo con seis clusters, donde el número de errores en la estimación del estacionario fue cero, y donde el modelo podía incluso, diferenciar los puntos de equilibrio que tenían el transitorio oscilante de los que tenían un transitorio asintótico. El error medio de los errores al estimar el valor de la variable discreta fue de 0.0294

Al comparar los clusters difusos para obtener el modelo cualitativo los resultados son.

Variable no: 1

cluster	similitud máxima	término lingüístico
1	0.7403	menor o igual que grande
2	0.6815	menor que pequeño
3	0.7661	menor o igual que medio
4	0.7157	no grande
5	0.7607	no pequeño
6	0.5990	algo menor que grande

Variable no: 2

cluster	similitud máxima	término lingüístico
1	0.8485	menor o igual que pequeño
2	0.5177	menor que medio pero mayor que pequeño
3	0.8630	grande
4	0.6441	grande
5	0.7869	medio
6	0.6689	menor que grande pero mayor que medio

Lo que da lugar a un modelo cualitativo como el

que sigue:

R1: Si p2 es menor o igual que pequeño entonces ciclo límite.

R2: Si p1 es muy pequeño y p2 es menor que medio pero mayor que pequeño entonces punto equilibrio.

R3: Si p1 es menor o igual que medio y p2 es grande entonces punto de equilibrio.

R4: Si p1 es no grande y p2 es grande entonces punto equilibrio.

R5: Si p1 es no pequeño y p2 es medio entonces ciclo límite.

R6: Si p1 es algo menor que grande y p2 es menor que grande pero mayor que medio entonces ciclo límite.

5.3 Modelo complejo.

El segundo modelo es propuesto en [15], compuesto de tres variables de nivel y siete parámetros, donde además existen relaciones de dependencia entre los parámetros. Este sistema fue estudiado con métodos estadísticos en [16], y su nivel de complejidad era acentuado por el hecho de que, todos los parámetros influían bastante en el comportamiento del sistema. Para no obtener un modelo lingüístico demasiado confuso limitamos el número de reglas a ocho y utilizando sólo los cuatro parámetros que sabíamos más influían en el comportamiento del modelo. Partimos, entonces, de una base de datos con 200 registros, (parece lógico aumentar la muestra ante la complejidad del modelo), con cuatro variables de salida y una variable de salida con dos valores.

Dos son los principales problemas en la aplicación del método propuesto a este modelo. Uno radica en la autolimitación impuesta en el número de reglas que conlleva la posibilidad de que haya puntos en la base de datos que no queden recogidos por ninguno de los cluster resultado del algoritmo FCM. Esto se puede resolver en la optimización del modelo, añadiendo a la función objetivo un sumando que pondere de manera adecuada el cardinal del conjunto de puntos que no están en ningún cluster.

El segundo problema, es que dada la limitación del número de variables de entrada a cuatro, en un sistema que tiene siete, es que el error cometido en la previsión puede variar bastante de un modelo difuso a otro. Los resultados obtenidos para este sistema, después de aplicarle el algoritmo de optimización varían entre un 2% y un 4% de tasa de error. Si además, intentamos discriminar el transitorio de los puntos de equilibrio el número de errores se incrementa algo, pero siempre por debajo del 8%. Otro indicador que nos puede medir el grado de bondad del modelo en caso de igualdad en la tasa de error, es el grado de "solapamiento", es decir, cuantos puntos de la base de datos tienen un valor de pertenencia alto a más de un cluster

6 Conclusiones.

Se han adaptado conocidas técnicas difusas a sistemas con variable de salida discreta. Asimismo se presentan modificaciones de algoritmos de optimización de modelos difusos en este mismo caso. Estas técnicas se han aplicado, con resultados bastante aceptables a la creación de modelos cualitativos que expliquen el comportamiento que en régimen estacionario puede presentar un sistema dinámico, en función de sus parámetros.

7 Bibliografía.

[1] L. A Zadeh, "Outline of a new approach to tge analysis of complex systems and decision processes", IEEE Trans. Syst., Man, Cybern., vol. 3, pp. 28-44, 1973.

[2] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithm", New York· Plenum Press, 1981

[3] M. R. Anderberg, "Cluster Analysis for Applications", Academic Press, 1973.

[4] R. C. Tryon y D. E. Bailey, "Cluster Analisys", McGraw-Hill, 1970.

[5] B. V Dasarathy (Ed.), "Nearest Neighbor Pattern classification techniques", IEEE Computer Society Press, 1991.

[6] Y-H. Pao, "Adaptive Pattern Recognition and Neural Networks", Addison-Wesley, 1989.

[7] T. Kohonen, "Self-Organization and Associative Memory", Springer-Verlag, 1987.

[8] T. Khanna, "Foundations of neural Networks", Addison-Wesley, 1990.

[9] I. Bratko, S. Muggleton y A. Varsek, "Learning qualitative models of dynamic systems, En "Inductive Logic Programming", Academic Press, 1992.

[10] J R. Quinlan, "Learning logical definitions from relations". Machine Learning Journal, Vol. 5. pp. 239-266, 1990.

[11] M. Sugeno y T Yasukawa, "A fuzzy-logic-based approach to qualitative modeling", IEEE Transactions on fuzzy systems, vol. 1., n. 1, pp 7-31. 1993.

[12] D. Dubois y H. Prade, Fuzzy Sets and Systems. Theory and Aplications. New York. Academic Press. 1980.

[13] N. D. Kondratieff, "The long wave in economic life", Review of Economic Statistics n. 17 pp 105-115, 1935.

[14] S. Rassmussen, E. Mosekilde y J.D. Stearman, "Bifurcations and Chaotic Behavior in a simple model of the economic long wave", System Dynamics Review 1 (1985) pp. 92-110.

[15] J. D. Stearman, "A Behavioral model of the economic long wave", Journal of Economic Behavior and Organization, n. 6 pp 17-53, 1985.

[16] M Toro. J. Riquelme y J Aracil, "Classifying system behavior modes by statistics search in the parameter space", European Simulation Multiconference, York, Inglaterra, pp 181-185, 1991

Aplicación de algoritmos borrosos al guiado de un robot móvil con visión artificial

Enrique Santiso Gómez
Dpto. de Electrónica
Universidad de Alcalá
28871 Alcalá de Henares - Madrid
e-mail: elesg@alcala.es

Manuel Mazo, Carmen Serra,
J.Jesús García, F. Javier Rodríguez.
Dpto. de Electrónica
Universidad de Alcalá
e-mail: eldep@alcala.es

Resumen

En esta comunicación se describe la estrategia de guiado de un robot móvil, a través de trayectorias definidas en el suelo por medio de dos trazas paralelas de pintura, utilizando algoritmos de control borrosos y donde las variables a controlar se obtienen a partir de técnicas de procesamiento de imágenes.

Palabras clave: Control Fuzzy, Robot Autónomo, Visión Artificial.

1. Introducción.

De un tiempo a esta parte se ha tratado de dotar de autonomía a los robots móviles industriales. Esto se había conseguido hasta ahora principalmente mediante guiado a través de campos magnéticos generados por hilos enterrados en el suelo, los cuales indicaban la trayectoria a seguir (carros filoguiados). Con objeto de dotar a los robots de una mayor autonomía, y poder modificar más fácilmente la trayectoria a seguir, los últimos estudios se han enfocado al guiado mediante visión artificial.

Debido a los retardos introducidos por el procesamiento de imágenes, a la falta de linealidad del robot y por tanto a su difícil modelación, se hace harto complicado realizar un control clásico y obtener unos resultados aceptables. Es por esto que con este estudio hemos tratado de mejorar el guiado utilizando para ello un control borroso.

2. Configuración del sistema.

El sistema utilizado por nosotros responde a la configuración de la fig.1, y está constituido básicamente por:

- Robot móvil con una estructura rectangular (1025 mm de largo, 680 mm de ancho y 440 mm de alto) sobre cuatro ruedas: dos locas y dos motrices con tracción independiente; con un peso de 150 Kg, cargas de hasta 120 Kg y velocidad máxima de 2 m/s.
- Sistema de visión artificial con cámara a bordo del robot y control y tratamiento de imágenes sobre un ordenador personal 486, siendo la comunicación entre y PC vía radio.

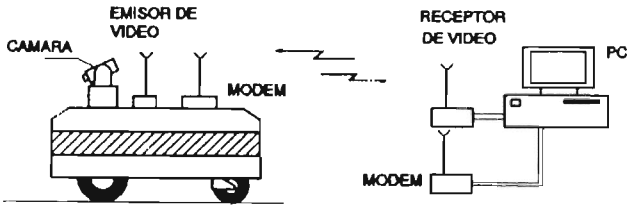


Fig.1. Configuración del sistema.

3. Estructura de control

La estructura de control que se ha utilizado es la que se muestra en la fig.2. En ella se puede observar que existen dos lazos principales de control denominados como:

- Control de bajo nivel (control clasico).
- Control de alto nivel (control borroso).

El primero de ellos está constituido por dos lazos de control tipo PID, uno de las velocidades angulares (w_r, w_l) de cada una de

las ruedas motrices y otro de las velocidades de traslación (V) y rotación (Ω) del robot.

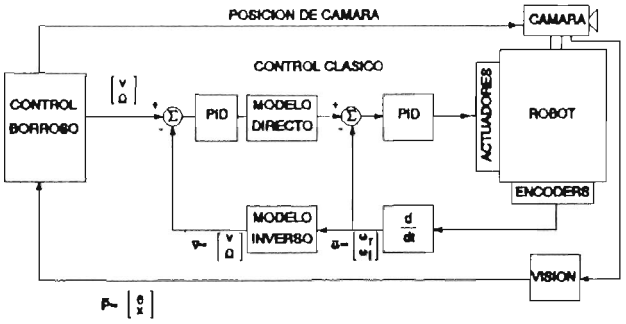


Fig.2. Estructura de control

El control borroso tiene como variables de entrada (fig.3) el ángulo (θ) que forma el eje longitudinal del robot con el eje central de la trayectoria a seguir, la distancia (X) entre el eje longitudinal del robot y la tangente a la trayectoria y la derivada de X (ΔX) que es calculada internamente.

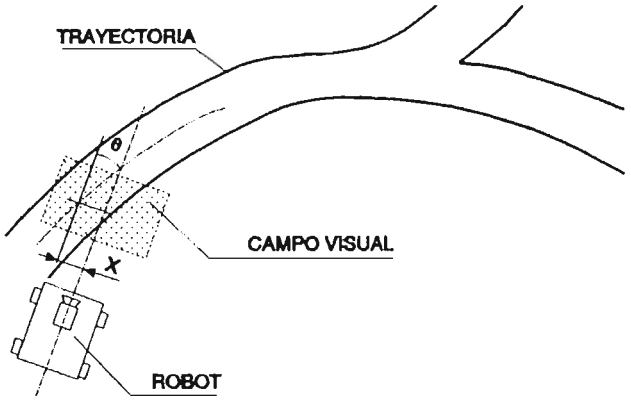


Fig.3. Variables de entrada del control borroso.

4. Obtención de las variables de control.

La obtención de las variables de entrada al control borroso se obtienen analizando los sucesivos cuadros de imagen. Sobre las imágenes captadas se aplican diferentes algoritmos de filtrado y segmentación que permiten obtener una imagen de salida donde las líneas que definen los bordes de la trayectoria a seguir quedan perfectamente definidas, aun en condiciones de baja iluminación o presencia de ruidos (puntos del suelo con características similares a las trazas que definen la trayectoria). Sobre esta imagen de salida se analizan uno o varios cuadros (fig.4) localizando los puntos de corte de éste con las trazas que definen la trayectoria y el ancho de las trazas. Esto nos permite detectar la presencia de cruces y, previa calibración de la cámara, calcular las variables: "X" y "θ" en un punto cualquiera dentro del campo visual.

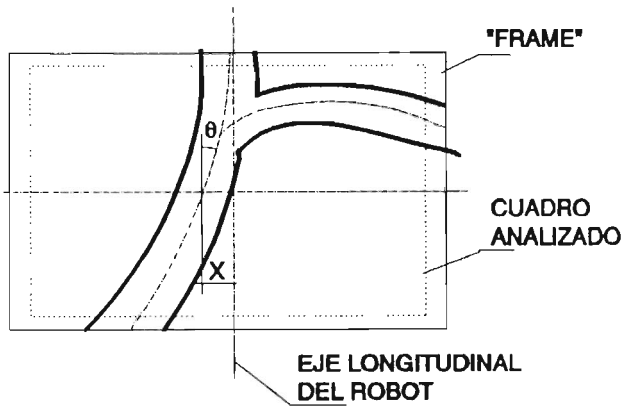


Fig.4. Obtención de las variables de control

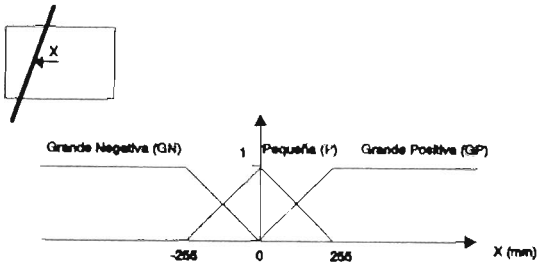
5. Control borroso.

El control borroso tomará como entradas la variables antes mencionadas y proporcionará como salida las velocidades lineales V_L y V_R, a partir de las cuales se obtienen las velocidades de traslación (V) y de giro (Ω) del robot:

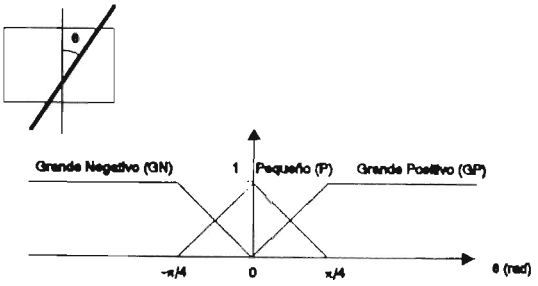
$$V=\frac{V_L+V_R}{2}$$

$$\Omega=\frac{V_L-V_R}{D}$$

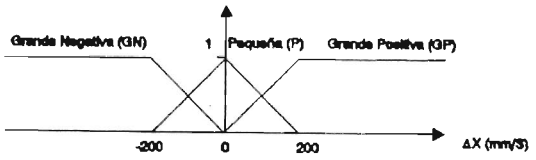
Para cada una de las tres variables de entrada ("X", "θ" y "ΔX") se han definido tres funciones de pertenencia tal como se indica en las fig.5.



X_{GN}= grado de pertenencia a distancia negativa.
X_P= grado de pertenencia a distancia pequeña.
X_{GP}= grado de pertenencia a distancia positiva.



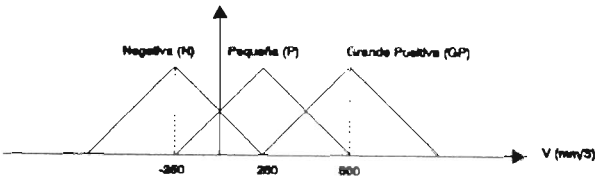
θ_{GN}= grado de pertenencia a ángulo negativo.
θ_P= grado de pertenencia a ángulo pequeño.
θ_{GP}= grado de pertenencia a ángulo positivo.



ΔX_{GN}= grado de pertenencia a derivada negativa.
ΔX_P= grado de pertenencia a derivada pequeña.
ΔX_{GP}= grado de pertenencia a derivada positiva.

Fig.5 Funciones de pertenencia de las variables "X", "θ" Y "ΔX".

Las clasificaciones del grado de actuación de las variables VL y VR de salida son idénticas y toman los valores mostrados en la figura 6.



V_N = velocidad negativa.
V_P = velocidad pequeña positiva.
V_{GP} = velocidad grande positiva.

Fig.6 Grado de actuación de las variables de salida VR y VL.

Se ha comprobado que con solamente tres subconjuntos para cada variable se obtiene una respuesta buena, por lo que se consideró que era innecesario aumentar este número, lo que implicaría un mayor número de reglas y por tanto un mayor programa.

Los valores de las funciones de pertenencia se obtienen experimentalmente tras múltiples pruebas.

Las reglas que se enuncian a continuación se obtuvieron a partir de un control manual, que tomaba como referencia para la conducción del móvil, los datos proporcionados por el sistema de visión artificial.

La variable X se considera positiva hacia la derecha y θ se considera positiva en el sentido de giro de las agujas del reloj.

1- Si X es GP , θ es GP y ΔX es GP
ENTONCES VL es N y VR es G
2- Si X es GP , θ es GP y ΔX es P
ENTONCES VL es N y VR es G
3- Si X es GP , θ es GP y ΔX es GP
ENTONCES VL es P y VR es G
4- Si X es GP , θ es P y ΔX es GP
ENTONCES VL es P y VR es G
5- Si X es GP , θ es P y ΔX es P
ENTONCES VL es G y VR es G
6- Si X es GP , θ es P y ΔX es GP
ENTONCES VL es G y VR es G
7- Si X es GP , θ es GP y ΔX es GP
ENTONCES VL es G y VR es G

8- Si X es GP , θ es GP y ΔX es P
ENTONCES VL es G y VR es G
9- Si X es GP , θ es GP y ΔX es GP
ENTONCES VL es G y VR es P
10- Si X es P y , θ es GP
ENTONCES VL es P y VR es G
11- Si X es P , θ es P y ΔX es GP
ENTONCES VL es P y VR es G
12- Si X es P , θ es P y ΔX es P
ENTONCES VL es G y VR es G
13- Si X es P , θ es P y ΔX es GP
ENTONCES VL es G y VR es P
14- Si X es P y , θ es GP
ENTONCES VL es G y VR es P
15- Si X es GP , θ es GP y ΔX es GP
ENTONCES VL es P y VR es G
16- Si X es GP , θ es GP y ΔX es P
ENTONCES VL es G y VR es G
17- Si X es GP , θ es GP y ΔX es GP
ENTONCES VL es G y VR es G
18- Si X es GP , θ es P y ΔX es GP
ENTONCES VL es G y VR es G
19- Si X es GP , θ es P y ΔX es P
ENTONCES VL es G y VR es G
20- Si X es GP , θ es P y ΔX es GP
ENTONCES VL es G y VR es P
21- Si X es GP , θ es GP y ΔX es GP
ENTONCES VL es G y VR es P
22- Si X es GP , θ es GP y ΔX es P
ENTONCES VL es G y VR es N
23- Si X es GP , θ es GP y ΔX es GP
ENTONCES VL es G y VR es N

Originalmente no se había incluido como señal de entrada la derivada de X. Al introducirla, y actuando tal como indican las reglas, la distancia del centro del robot al centro de la trayectoria aumentó ligeramente, sin embargo, se redujeron considerablemente las oscilaciones del robot y se aumento la velocidad media.

6. Resultados.

Con este control se ha mejorado la respuesta obtenida hasta el momento con controles clásicos, de manera que se llegan a alcanzar velocidades de hasta 1 m/s. en entornos complejos, velocidad que viene limitada principalmente por el período de muestreo, fijado actualmente en 300 ms.

Actualmente se está trabajando con una tarjeta digitalizadora más rápida, con la consiguiente reducción del período de muestreo, y se va a realizar la implantación de un nuevo control basado en la variación de la elevación de la cámara para conseguir un campo de visión mayor.

Esta nueva opción ya está simulada y se han obtenido unos resultados realmente buenos. En este momento se está probando el nuevo programa en el robot con la nueva tarjeta, y se prevé obtener buenos resultados muy próximamente.

En la figura 7 se muestra un ejemplo de los resultados obtenidos, donde los trazos gruesos y finos representan las trayectorias a seguir y la realmente seguida, respectivamente.

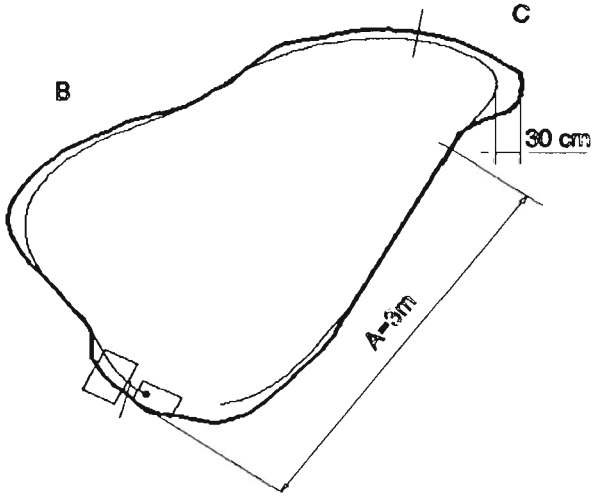


Fig.7 Ejemplo de trayectorias.

En un experimento realizado sobre un trayecto cuadrado de 3 m de lado, se ha obtenido con un control clásico una velocidad media de 0,04 m/s, mientras que con el control borroso esta media se vio incrementada hasta 0,3 m/s.

La figura 8 muestra la variación de la velocidad de traslación del robot para los diferentes tramos de la trayectoria. El máximo valor de la velocidad de traslación está limitado por el tiempo de procesado de imágenes.

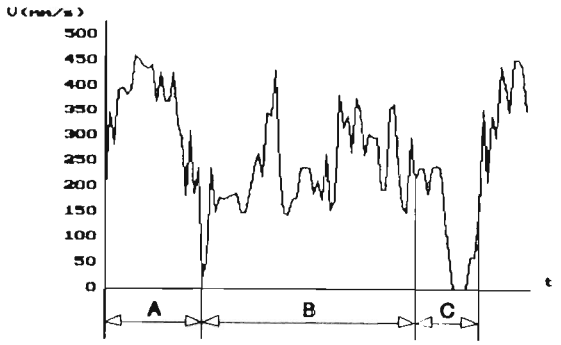


Fig.8 Variación de la velocidad V para la trayectoria de la figura 7.

7. Conclusiones.

Se ha comprobado que en sistemas difíciles de modelar, o con modelos no lineales y variables, y que además introducen grandes retardos, el control borroso introduce grandes mejoras respecto al control clásico. También se ha comprobado que aunque el control borroso nunca realiza un mejor control que el experto del cual se obtienen las reglas, si podrá realizar dicho control a una velocidad mayor. Debido a estas variaciones en la velocidad de proceso en algunas ocasiones habrá que modificar las funciones de pertenencia con respecto a las definidas inicialmente por el experto.

8. Bibliografía.

Zadeh, L.A. (1973). "Outline of approach to the analysis of complex systems and decision processes". Transactions on Systems, Man and Cybernetics. IEEE

The Institute of Electrical and Electronics Engineers, Inc.(1991) "Fuzzy Logic: Applications and Perspectives". Video Conferences. IEEE

Toshiro Terano, Kiyoji Asai, Michio Sugeno, "Fuzzy Systems, Theory and Its Applications". Academic Press, Inc. (1992)

M. Sugeno, "Industrial Aplications of Fuzzy control". (1985)

L.A. Zadeh, "Fuzzy logic and its aplications to approximate reasoning". (1973)

Depuración y generación de reglas de forma automática. Aplicación sobre el control de una planta piloto utilizando autómata programable con módulo fuzzy

Juan Carlos Hernández, Francisco Jiménez, Joseba Quevedo, Josep Aguilar

Universidad Politécnica de Catalunya
Campus de Terrassa
Colón 11, 08222 Terrassa
España
Teléfono: 739 82 27 - 739 82 35
Fax: 739 82 25

Resumen

En este artículo, se recogen las experiencias surgidas al utilizar un autómata programable comercial (C200H - Omron) como controlador fuzzy de una planta piloto de laboratorio (viga-bola).

También se recogen en este trabajo unas ideas de tipo práctico a las que hemos recurrido con objeto de crear o depurar la base de reglas necesaria para llevar a cabo los objetivos de control perseguidos.

Palabras clave: Lógica Fuzzy, Control Fuzzy, Autómata Programable, Planta Piloto, Generación de reglas desde ejemplos, Depuración de bases de reglas.

1 Introducción

En este artículo, se intentará resumir las experiencias que han ido apareciendo al intentar realizar un control fuzzy sobre una planta piloto de laboratorio (viga-bola), la cual presenta una gran inestabilidad. La estructura de control utilizada es intrínsecamente fuzzy.

El controlador fuzzy adoptado es el módulo C200H-FZ001 de la firma Omron, el cual es insertado dentro de la estructura del autómata programable C200H de la misma firma.

El módulo fuzzy FZ001 dispone de 8 entradas y de 4 salidas y la base de reglas que maneja puede estar compuesta por 128 reglas que utilicen 8 antecedentes y 2 consecuentes.Cada una de las variables de entrada al módulo fuzzy puede ser dividida en un máximo de 7 conjuntos (labels).

El método de inferencia utilizado por este módulo es el MIN MAX, mientras que el método de defusificación que utiliza puede ser elegido por el usuario entre el de máxima pertenencia y el de centro de gravedad.

Una vez examinadas las características más relevantes del módulo fuzzy, la figura 1 nos da una idea general del sistema de control utilizado. En ella puede observarse al autómata programable utilizado, junto con dos módulos de conversión Analógico Digital (ADC) y Digital Analógico (DAC) y el módulo fuzzy FZ001 [7].

La conexión de las variables de entrada y de salida se lleva a cabo utilizando los módulos ADC y DAC, respectivamente, mientras que, internamente, la conexión entre el módulo fuzzy y los módulos de conversión se lleva a cabo por transferencia entre posiciones de memoria compartidas.

La programación del autómata programable se realiza a través de dos paquetes de software

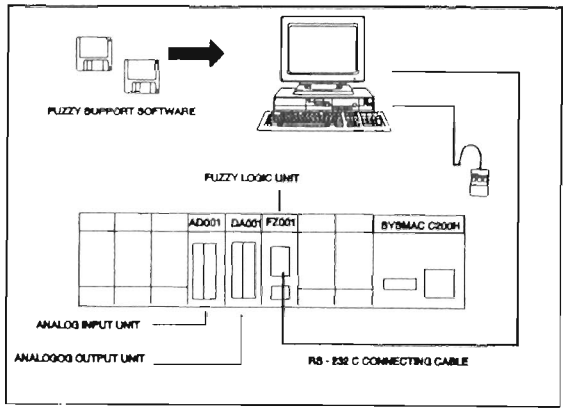


Figura 1. Esquema general de control.

distintos (LSS y FSS) con objeto de favorecer la edición y depuración de los programas de control.

Mediante el programa de programación LSS (Ladder Support Software) realizamos el programa de control, el cual utilizará como variable de salida aquella que le suministre el módulo fuzzy después de realizar la inferencia de la base de reglas que previamente se ha almacenado utilizando el paquete de software FSS (Fuzzy Support Software).

2 Descripción de la planta piloto

La planta piloto que vamos a intentar controlar consiste en un sistema denominado viga-bola (Ball and Beam), fabricado por la empresa TecQuipment [9].

Esta planta consiste en un sistema que para pequeños ángulos de inclinación puede aproximarse a un doble integrador, donde la ganancia que presenta puede ser determinada de forma experimental ajustando una función de transferencia de segundo orden al sistema. Así pues, se trata de un sistema inestable en lazo abierto.

Las variables que el sistema proporciona son el ángulo que presenta la viga y la posición de la bola.

El control consiste en proporcionar al sistema una acción de control que haga pivotar la viga sobre un eje, con objeto de posicionar la bola en el lugar deseado. La figura 2 trata de mostrarnos a grandes trazos el sistema que tratamos de controlar.

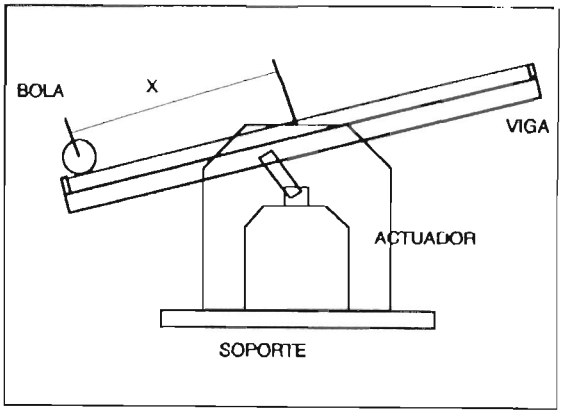


Figura 2. Planta piloto.

Finalmente, hay que indicar que este sistema no presenta ninguna dificultad de control si este es realizado en forma manual por un operador, que por otra parte no es necesario que realice un entrenamiento prolongado para situar la bola en la posición que se desea.

Es precisamente este hecho, comparado con las dificultades que han aparecido al intentar realizar un control clásico, lo que nos anima a intentar plasmar en una base de reglas el grado de conocimiento de un operador experto cuando este intenta controlar manualmente el sistema.

3 Descripción del esquema de control utilizado

El objetivo que nos hemos fijado conseguir al realizar el control fuzzy de la planta piloto comentada en el apartado anterior es colocar la bola en la posición central del viga-bola. La figura 3 trata de profundizar un poco en el esquema de control que ha sido adoptado para llevar a cabo el objetivo propuesto.

La variable de salida U representa la tensión analógica que debe proporcionarse a la bobina

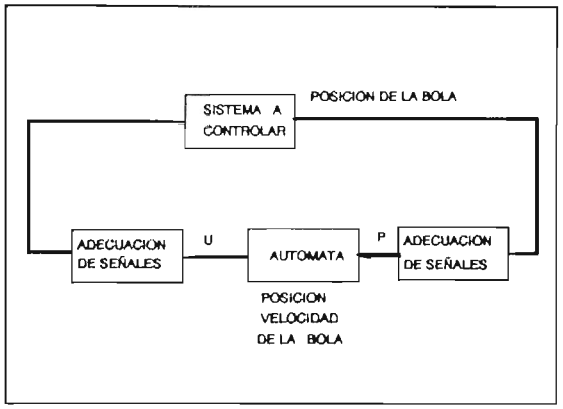


Figura 3. Esquema de control.

de control del viga-bola para que esta pivote, tratando de llevar a la bola a la posición central de la viga. Su elección como variable de salida es obvia ya que la planta piloto solo dispone de esta variable como entrada. Las variables de entrada que utiliza el autómata programable son la posición de la bola y la velocidad de la misma.

El programa principal de control implementado en el autómata [7] [8], está de acuerdo con el hardware que ha sido necesario utilizar para crear el controlador fuzzy (ver figura 2).

El proceso de programación finaliza utilizando el paquete de software FSS, el cual nos permitirá crear la base de reglas y almacenarlas en el módulo fuzzy.

4 Experimentación y conclusiones

Admitiendo que las variables de control óptimas son la posición y la velocidad de la bola, centramos nuestra atención en intentar realizar un control lo más preciso posible de la planta comentada anteriormente.

La opción de control elegida para cumplir los objetivos de precisión marcados, ha hecho que dividamos las dos señales de entrada y la de salida en siete labels cada una (precisión máxima que puede proporcionar el módulo fuzzy FZ001).

El método de defusificación elegido ha sido el de máxima pertenencia, ya que el ajuste de la matriz de inferencia final ha sido más fácil de realizar que cuando utilizamos como método de inferencia el de centro de gravedad. No obstante, es necesario advertir que utilizando como método de defusificación el de centro de gravedad se obtiene un control menos brusco del sistema que cuando se utiliza el de máxima pertenencia.

Tras un largo proceso de experimentación y de ajuste para controlar el sistema, obtenemos la matriz de inferencia representada en la figura 4.

POSICION ⇨ / VELOCIDAD ⇩	NL	NM	NS	ZR	PS	PM	PL
NL	1 NL	2 NL	3 NM	4 NM	5 NS	6 NS	7 ZR
NM	8 NL	9 NM	10 NM	11 NS	12 NS	13 ZR	14 PS
NS	15 NL	16 NM	17 NS	18 NS	19 ZR	20 NS	21 PS
ZR	22 NL	23 NS	24 ZR	25 ZR	26 ZR	27 PM	28 PL
PS	29 NL	30 ZR	31 ZR	32 ZR	33 PS	34 PM	35 PL
PM	36 NS	37 ZR	38 PS	39 PS	40 PS	41 PM	42 PL
PL	43 ZR	44 PS	45 PS	46 PM	47 PM	48 PL	49 PL

Figura 4. Matriz de inferencia.

Tal como se puede apreciar en dicha figura, la base de reglas creada dispone de 49 reglas que hacen que la planta sitúe la bola en la posición central de la viga, con independencia de donde parta la bola inicialmente.

El siguiente punto que se trató de estudiar, fue el de intentar obtener la base de reglas de una forma menos laboriosa que la experimentada en primer lugar.

Para ello, se sometió a la planta a un control manual, mientras que mediante una placa de adquisición de datos PC Labcard 718 de la firma Advantech,se iban capturando los valores de posición y velocidad de la bola, así como la acción de salida que debía de transmitirse a la bobina que controlaba el proceso, con objeto de situar la bola en la posición deseada.

Los archivos de información fueron tratados

mediante la hoja de cálculo QPRO de la firma Borland, la cuál determinaba automáticamente la base de reglas del módulo fuzzy, así como, permitia depurar la base de reglas eliminando aquellas reglas que por la naturaleza del control manual realizado no intervenían en el proceso.

La generación de los antecedentes de las reglas, se realiza de forma automática, teniendo en cuenta la definición de labels, realizada para las variables de entrada en el programa FSS, mientras que la generación de los consecuentes de las reglas, fue un poco más laboriosa, debido al hecho, de que la división de labels que presenta el módulo fuzzy estudiado es la representada en la figura 5.

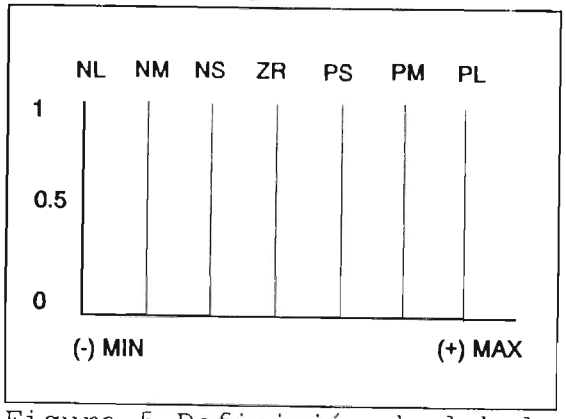


Figura 5.Definición de labels de la variable de salida.

En ella se puede observar que cuando una salida adquirida se sitúa entre dos de los labels en los que se ha dividido la señal de salida, hace que aparezca un problema de asignación de un conjunto de salida a la a variable adquirida. Hemos intentado resolver este problema introduciendo en la hoja de cálculo una definición de labels tal como la representada en la figura 6, en la cual puede observarse que se define una zona muerta alrededor de la distancia media entre labels de salida que hace que la hoja de calculo desprecie la regla que inferiría cuando el valor de la acción de salida perteneciera a ese intervalo.

La anchura de esta zona muerta se determina en forma heurística o bien intentando recoger la experiencia que se tenga sobre el proceso [12].

La matriz de inferencia representada en la figura 7 muestra las reglas generadas por este procedimiento.

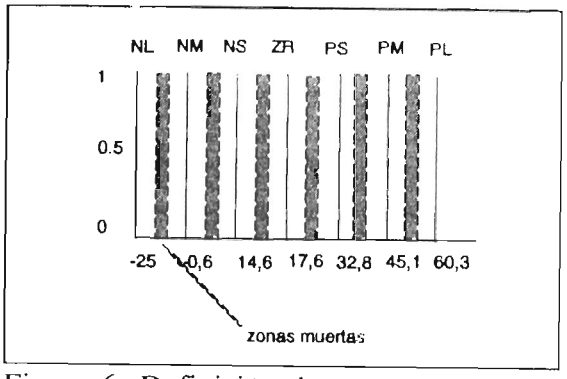


Figura 6. Definición de zonas muertas en los labels de la variable de salida.

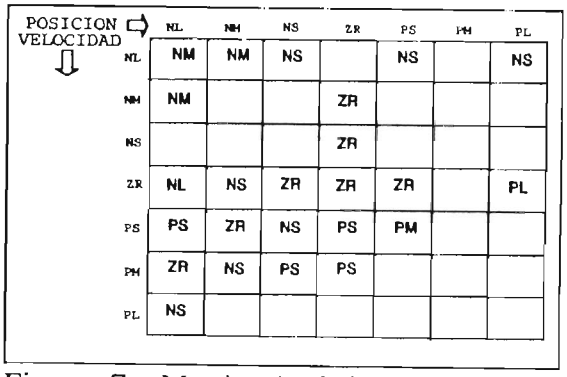


Figura 7. Matriz de inferencia ensayo manual.

Tras experimentar con la matriz de inferencia representada en la figura 7, observamos que pese a intentar estabilizar la bola en la posición central de la viga ésta no permanecia estable, teniendo que retocar de forma manual la matriz con objeto de conseguir la estabilidad buscada.

Examinando el problema, pensamos que este puede ser resuelto actuando en dos direcciones distintas. Por un lado, realizar un número elevado de ensayos en forma manual que nos permita obtener una cantidad de información lo suficientemente importante como para poder realizar un estudio de coherencia de las reglas

que se obtengan al ir realizando distintos ensayos.

Por otro lado, disponer de un módulo fuzzy que, al igual que ocurre con otras tarjetas controladoras, tenga la posibilidad de obtener un número más amplio de labels de salida. Este hecho permitirá no tener que concentrar acciones distintas sobre un número limitado de labels, no perdiendo de esta forma la riqueza del control manual.

Finalmente la figura 8 representa la misma variable controlando el sistema con la base de

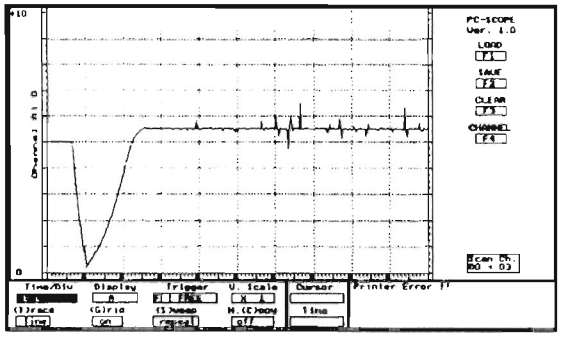


Figura 8. Variable posición de la bola.

reglas deducida mediante el procedimiento expuesto.

Agradecimientos

Este trabajo ha sido realizado gracias a la colaboración de la Comisión Interministerial de Ciencia y Tecnología (CICYT) bajo el contrato TAP93-0596-C04 . El artículo ha sido parcialmente patrocinado por el Collectiu d'Estudis i Recerca en Control i Automàtica (CERCA).

Referencias

[1] MANDAMI, E.H. Aplications of Fuzzy Algoritms for simple Dynamic Plants. Proc. IEEE, Vol 121, nº 12, pp.1585-1588, 1974.

[2] CHUEN CHIEN LEE. Fuzzy Logic in Control Systems: Fuzzy Logic Controller, part I y II. IEEE Translations on Systems, Vol 20, nº 2, pp 404-435, March/April 1990.

[3] HAO YING, SILER, W., BUCKLEY, J. Fuzzy Control Theory: A Nonlinear Case. Automática, Vol 20, nº 3, pp 513-520. Pergamon Press, 1990.

[4] TEIGHBIT, S., MALVACHE, N., WILLAEYS, D. Proposition of a modelisation of subjective process deduction of the input-output relation. European Workshop on Industrial Fuzzy Control and Applications (IFCA), ETSEI Terrassa, April 1993.

[5] YAMAMOTO ELECTRIC INDUSTRIAL Co. Ltd. Fuzzy Controller Board Technical Manual, 7-6 Shinkitano 2-chome. Yodogawa KU Osaka, 533, Japan.

[6] OMRON. Fuzzy Logic Module. European Workshop on Industrial Fuzzy Control and Applications (IFCA). ETSEI Terrassa, April 1993.

[7] OMRON. C200H-FZ001 Operation Manual. Omron Electronics S.A. C/ Arturo Soria, 95. Madrid.

[8] OMRON. C200H Operation Manual. Omron Electronics S.A. C/ Arturo Soria, 95. Madrid.

[9] WELLSTEAD, P. E. TecQuipment. Bonsall Street, Long Eaton, Nottingham, NG 10, England.

[10] ADVANTECH. Manual de Operación de la placa de Adquisición PCL-718. Advantech Co. Ltd.

[11] ARMENGOL, J. Avaluació de les prestacions de controladors digitals avançats. ETSEI Terrassa, 1992. (Proyecto Final de Carrera).

[12] LI XIN WANG AND MENDEL, J. M. Generating Fuzzy Rules by Learning from examples. IEEE Transactions on systems, man and cybernetics, vol. 22, nº 6. November/december 1992.

[13] MACVICAR-WHELAN, P. J. Fuzzy Sets for Man Machine interaction. Int. J. Man Machine Studies, 1976, 8, 687-697.

Influencia de las T-normas en Controladores Difusos

E. Cárdenas, J.C. Castillo, O. Cordón, A. Peregrín
Escuela Técnica Superior de Ingeniería Informática
Universidad de Granada
18071 - Granada

Resumen

El objetivo del presente trabajo es efectuar un estudio de la influencia de las t-normas en controladores difusos. Para ello, presentaremos la expresión genérica de la regla composicional de inferencia empleada en control difuso y enunciaremos las distintas vías mediante las cuales pueden influir las t-normas en la misma. Efectuaremos un estudio teórico del comportamiento de las t-normas al ser empleadas como implicaciones para realizar la inferencia, mostrando gráficamente sus efectos, y un estudio comparativo del rendimiento de los controladores difusos que emplean t-normas en control difuso. Como resultado principal, destacaremos el hecho de que las t-normas utilizadas para realizar la inferencia muestran un buen comportamiento y, por el contrario, las funciones de implicación no son adecuadas para su empleo en controladores difusos puesto que el hecho de verificar el Principio de Falsedad provoca un mal comportamiento en control difuso.

Palabras clave: control difuso, t-norma, t-conorma, función de implicación, método de defuzzificación.

1 Introducción al control difuso

Un componente importante de un controlador difuso es el Sistema de Inferencia (SI), encargado de recibir las entradas de las variables de estado del sistema y decidir la acción de control a aplicar sobre el mismo dando valor a las variables de control de este. El SI actúa sobre las entradas $A = (A_1 , ... , A_n)$, proporcionadas por el Interfaz de Fuzzificación, aplicándoles la regla composicional de inferencia obteniendo como resultado el conjunto difuso B'. La estructura genérica de la misma es la siguiente:

$$B'(y) = \sup_x \{ T' (A'(x), I (A(x), B(y))), x \in R^n \}$$

donde:
- $A'(x) = T (A_1'(x_1), A_2'(x_2), ..., A_n'(x_n))$
- $A(x) = T (A_1 (x_1), A_2 (x_2), ... , A_n (x_n))$
- I es una función de implicación
- T' es un conectivo u operador de conjunción

Puesto que en control difuso la entrada x correspondiente a las variables de estado del sistema a controlar es un valor puntual, $x = x_0$, la función A'(x) toma la forma:

$$A'(x) = \begin{cases} 1, & \text{si } x = x_0 \\ 0, & \text{en otro caso} \end{cases}$$

con lo que la expresión de la regla composicional de inferencia queda reducida a:

$$B'(y) = T' (1, I (A(x_0), B(y))) = I (A(x_0), B(y))$$

De este modo se observa como cada Sistema de Inferencia concreto depende de la función de implicación I y la t-norma T (operador de conjunción empleado para obtener $A(x_0)$).

Cada regla R_i incluida en la Base de Conocimiento da lugar a una acción de control representada por el conjunto difuso B'_i (y) obtenido de aplicar la regla composicional de inferencia sobre la misma. El **Interfaz de Defuzzificación** se encarga de obtener la acción de control resultante agregando estos conjuntos difusos en uno solo y aplica un método de defuzzificación D que convierte ese conjunto difuso en una acción de control en forma de valores reales asociados a las variables de control del sistema. Así, denominando S al controlador difuso, x_0 al valor de las entradas e y_0 al valor real obtenido de la defuzzificación, tenemos:

$$y_0 = S (x_0) = D (B'(y))$$

2 T-normas en control difuso

Como indicábamos anteriormente, el Sistema de Inferencia de un controlador difuso depende directamente de dos factores: el operador de conjunción y la función de implicación utilizados para el diseño de la regla composicional de inferencia. Las t-normas influyen en ambos del siguiente modo ([3,4,5,6]):

1. Puesto que el operador de conjunción que permite integrar los valores correspondientes a cada una de las

entradas al sistema viene representado por una t-norma, la elección de ésta determinará directamente la salida proporcionada por el controlador difuso.

2. Además, las t-normas influyen sobre las funciones de implicación desde una doble vertiente:
   2.1. Por un lado, existe la posibilidad de emplear una t-norma en lugar de una función de implicación para realizar la inferencia ([3,4]).
   2.2. Por otro, las t-normas son parte importante en la construcción de funciones de implicación clásicas ([2,4,5]), como a continuación se indica:
   - **Implicaciones Residuales o R-implicaciones**: Se construyen a partir de una t-norma T de la forma $I(x,y) = Sup\ \{c: c \in [0,1] / T(c,x) \leq y\}$
   - **Implicaciones de la Mecánica Cuántica o QM-implicaciones**: Familia constituida por funciones de implicación que presentan la estructura $I(x,y) = S(N(x), T(x,y))$, donde S es una t-conorma y T una t-norma.

Para la elaboración de este trabajo hemos seleccionado una serie de t-normas que emplearemos como operadores de conjunción y como sustitutas de las funciones de implicación en el proceso de inferencia. Además, añadiremos a estas funciones de implicación las tres R-implicaciones más conocidas en la literatura especializada, construidas a partir de tres de las t-normas seleccionadas. En [1] y [2] se recoge un estudio comparativo de las restantes familias de implicaciones.

3 T-normas y R-implicaciones seleccionadas

Hemos escogido seis t-normas que emplearemos como operadores de conjunción y como sustitutas de la función de implicación en la regla composicional de inferencia. Estas t-normas son las siguientes:

T1-I1.-Producto Lógico: $T(x,y) = Min(x, y)$

T2-I2.-Producto de Hamacher: $T(x,y) = \dfrac{x \cdot y}{x + y - x \cdot y}$

T3-I3.- Producto Algebraico: $T(x,y) = x \cdot y$

T4-I4.- Producto de Einstein: $T(x,y) = \dfrac{x \cdot y}{1 + (1-x) + (1-y)}$

T5-I5.-Producto Acotado: $T(x,y) = Max(0, x+y-1)$

T6-I6.- Producto Drástico:
$$T(x,y) = \begin{cases} x & \text{si } y = 1 \\ y & \text{si } x = 1 \\ 0 & \text{en otro caso} \end{cases}$$

A partir de la definición de las R-implicaciones y las t-normas T1, T3 y T6 respectivamente se obtienen las siguientes funciones de implicación pertenecientes a dicha familia:

I7.- La Implicación de Gödel:
$$I(x,y) = \begin{cases} 1 & \text{si } x \leq y \\ y & \text{si } x > 0 \end{cases}$$

I8.- La Implicación de Göguen:
$$I(x,y) = \begin{cases} Max(1, y/x) & \text{si } x \neq 0 \\ 1 & \text{si } x = 0 \end{cases}$$

I9.- La Implicación de Gaines:
$$I(x,y) = \begin{cases} 1 & \text{si } x \leq y \\ 0 & \text{en otro caso} \end{cases}$$

A continuación mostramos unos gráficos representativos del comportamiento de las tres funciones de implicación y las seis t-normas seleccionadas al ser empleadas para realizar la inferencia. Supongamos que el conjunto difuso B, asociado al consecuente de la regla de la Base de Conocimiento sobre la que se infiere, es un número difuso trapezoidal representado por la tupla (x_0, x_1, x_2, x_3). En la figura 1 se muestra su representación gráfica.

Sea $h = A(x_0) = T(A_1(x_1), A_2(x_2), \ldots, A_n(x_n))$, el valor obtenido tras aplicar el operador de conjunción T sobre las entradas proporcionadas por el sistema a controlar. La aplicación de las nueve implicaciones citadas provoca que el conjunto difuso B' resultante de la inferencia tome la forma representada en las siguientes figuras (en línea discontinua se muestra el conjunto difuso B inicial y en continua gruesa el conjunto difuso B' resultante de aplicación de la implicación):

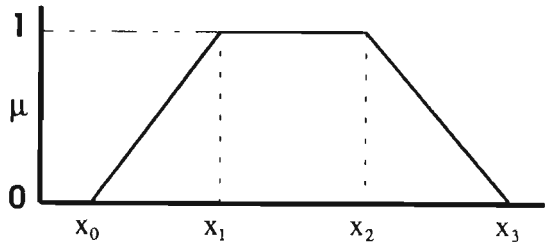


Figura 1: Representación del conjunto difuso B

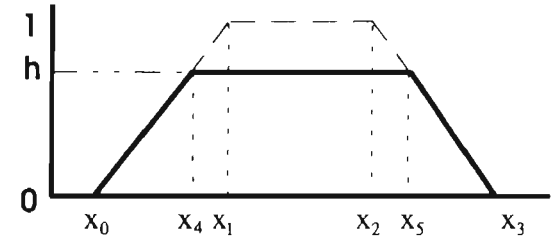


Figura 2: Conjunto difuso B' resultante de inferir mediante I1

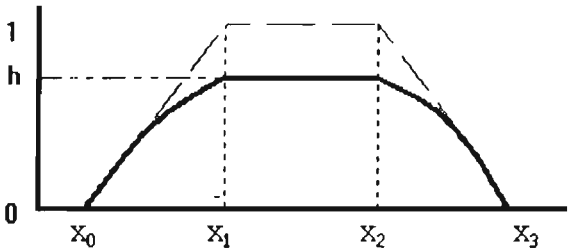


Figura 3: Conjunto difuso B' resultante de inferir mediante I2

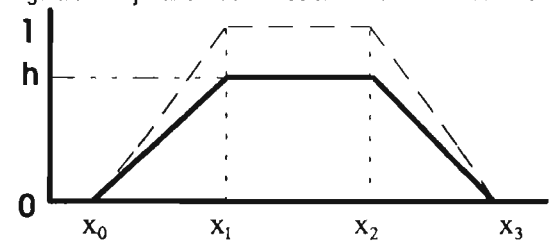


Figura 4: Conjunto difuso B' resultante de inferir mediante I3

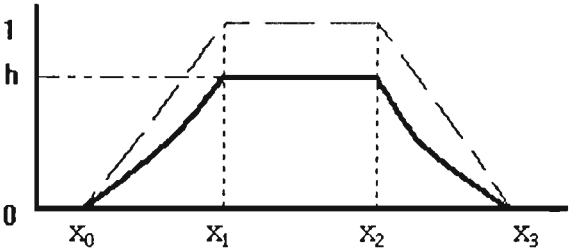


Figura 5: Conjunto difuso B' resultante de inferir mediante I4

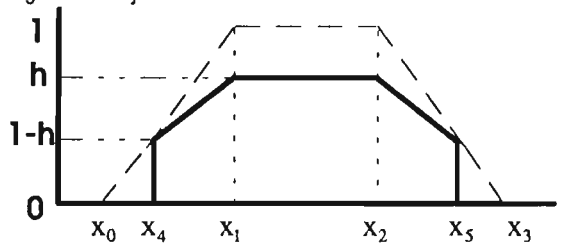


Figura 6: Conjunto difuso B' resultante de inferir mediante I5

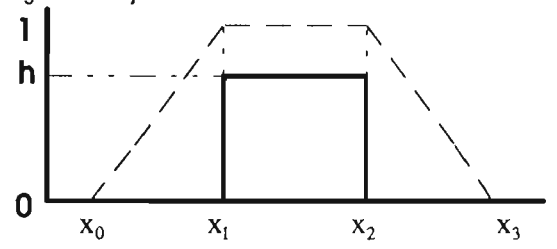


Figura 7: Conjunto difuso B' resultante de inferir mediante I6

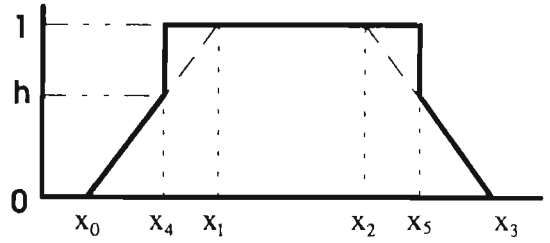


Figura 8: Conjunto difuso resultante de inferir mediante I7

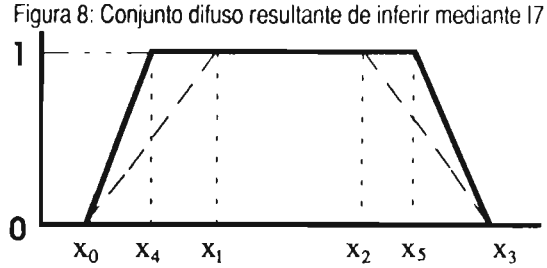


Figura 9: Conjunto difuso B' resultante de inferir mediante I8

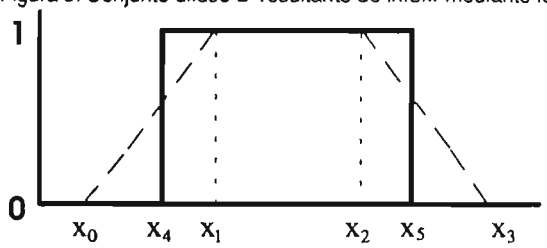


Figura 10: Conjunto difuso B' resultante de inferir mediante I9

Los valores x_4 y x_5 poseen las expresiones siguientes: $x_4 = x_0 + (x_1 - x_0)\cdot h$ y $x_5 = x_3 - (x_3 - x_2)\cdot h$.

4 Métodos de defuzzificación

Los estudios realizados en [1] y [2] perfilaban el método de defuzzificación **Punto de Máximo Criterio ponderado por el Grado de Emparejamiento** como el mejor de los allí citados y mostraban cómo los defuzzificadores basados en ponderaciones superaban ampliamente a los basados en el mayor Grado de Importancia. Para efectuar este análisis trabajaremos con los defuzzificadores basados en la ponderación del Valor Característico Punto de Máximo Criterio (PMC), G, citados en dichas referencias, los cuales responden a la expresión:

$$y_0 = \frac{\sum_i T_i \cdot G_i}{\sum_i T_i}$$

donde T_i equivale a S_i, Y_i ó H_i según el Grado de Importancia usado para ponderar sea el Area (**D1**), la Altura (**D2**) o el Grado de Emparejamiento (**D3**) repectivamente.

La forma de los conjuntos difusos resultantes de la aplicación de las distintas implicaciones seleccionadas, mostrada gráficamente en la sección anterior, y la formu-

lación de los métodos de defuzzificación enunciada arriba, da lugar a las siguientes observaciones:

1. El empleo de t-normas como implicaciones provoca que la Altura del conjunto difuso resultante coincida con el Grado de Emparejamiento de los conjuntos difusos antecedentes de la regla de control disparada. De este modo, los defuzzificadores basados en estos dos Grados de Importancia (en nuestro caso, D2 y D3) coincidirán cuando se empleen t-normas para inferir.
2. Puesto que el Punto de Máximo Criterio depende de los dos valores centrales del conjunto difuso resultante de la inferencia, este valor será idéntico cuando se empleen las R-implicaciones y la t-norma del Mínimo por un lado, y al emplear las cinco t-normas restantes por otro. Así, puesto que el Grado de Emparejamiento no depende de la función de implicación utilizada, D3 presentará el mismo valor para las cuatro primeras implicaciones y para las cinco restantes (valores distintos entre si a menos que se dé la simetría de los conjuntos difusos que constituyen las reglas de control de la Base de Conocimiento).

5 Medidas de comparación

En [2] se muestran dos grupos principales de medidas de rendimiento, Medidas de Convergencia y Medidas de Error, basadas respectivamente en las oscilaciones del sistema a controlar con respecto a su punto de equilibrio y en el estudio de la actuación del controlador difuso con respecto a una tabla de datos de evaluación del sistema. Para efectuar nuestro estudio hemos seleccionado una medida de cada grupo, una **Medida de Convergencia (MC)** y la **Medida de Error Cuadrático Medio (EC)**. Considerando que S [i,j,k] es el controlador difuso construido empleando la t-norma i como operador de conjunción, la implicación j y el método de defuzzificación k, dichas medidas responden a las siguientes expresiones:

$$MC\ (S[i,j,k]) = \frac{\sum_{ti=m}^{n} |e(ti)|}{(n-m)/\Delta t}$$

$$EC\ (S[i,j,k]) = \frac{\frac{1}{2}\sum_{k=1}^{N} (y_k - S[i,j,k](x_k))^2}{N}$$

donde e (ti) es el estado del sistema en un determinado instante de tiempo ti y n,m son los extremos del intervalo temporal en el que se realiza la toma de datos en la Medida de Convergencia; mientras que (x_k,y_k) es el par de valores (variables de estado,variables de control) contenidos en la tabla de datos del comportamiento del sistema en la Medida de Error Cuadrático Medio.

6 Experimento utilizado: Péndulo Invertido

La aplicación escogida para efectuar el estudio del rendimiento de los distintos controladores difusos ha sido ampliamente estudiada en Teoría de Control clásica, el problema del "Péndulo Invertido".

Hemos trabajado con el modelo de simulacion del sistema ya empleado para desarrollar los trabajos [1] y [2]. De nuevo, los parámetros utilizados han sido Masa = 5 Kilogramos y Longitud = 5 metros. El conjunto de reglas de control que constituye la Base de Conocimiento es el clásico de Yamakawa.

7 Resultados obtenidos

En las tablas que mostramos a continuación se recogen los valores obtenidos por los distintos controladores difusos en las medidas de rendimiento seleccionadas. El superior de cada casilla es el perteneciente a la Medida de Convergencia y el inferior a la de Error Cuadratico:

T1.- Producto Lógico

	I1	I2	I3	I4	I5
D1	9521 **34.518**	11350 56.110	**8941** 117.201	11335 ∞	**8941** 117.201
D2	**8941** 117.187	**8941** 117.199	**8941** 117.199	**8941** 117.199	**8941** 117.199
D3	**8941** 117.187	**8941** 117.199	**8941** 117.199	**8941** 117.199	**8941** 117.199

	I6	I7	I8	I9
D1	**8941** 117.199	40431 ∞	161153 ∞	176240 ∞
D2	**8941** 117.199	29700 66.887	148367 ∞	148367 ∞
D3	**8941** 117.199	**8941** 117.187	**8941** 117.187	**8941** 117.187

T2.- Producto de Hamacher

	I1	I2	I3	I4	I5
D1	**8936** 57.437	11360 92.916	9308 47.469	11507 80.393	9308 47.469
D2	9308 **47.460**	9308 47.468	9308 47.468	9308 47.468	9308 47.468
D3	9308 **47.460**	9308 47.468	9308 47.468	9308 47.468	9308 47.468

	I6	I7	I8	I9
D1	9308 47.468	41972 89.548	159817 ∞	173153 ∞
D2	9308 47.468	29700 65.836	148367 ∞	148367 ∞
D3	9308 47.468	9308 **47.460**	9308 **47.460**	9308 **47.460**

T3.- Producto Algebraico

	I1	I2	I3	I4	I5
D1	**9871** 81.327	11561 69.474	10588 74.630	11881 72.119	10588 74.630
D2	10588 74.588	10588 74.627	10588 74.627	10588 74.627	10588 74.627
D3	10588 74.588	10588 74.627	10588 74.627	10588 74.627	10588 74.627

	I6	I7	I8	I9
D1	10588 74.627	43306 ∞	159315 ∞	172014 ∞
D2	10588 74.627	29700 **65.669**	148367 ∞	148367 ∞
D3	10588 74.627	10588 74.588	10588 74.588	10588 74.588

T4.- Producto de Einstein

	I1	I2	I3	I4	I5
D1	**10555** 73.692	11799 78.573	11272 77.085	12214 **58.892**	11272 77.085
D2	11272 77.295	11272 77.096	11272 77.096	11272 77.096	11272 77.096
D3	11272 77.295	11272 77.096	11272 77.096	11272 77.096	11272 77.096

	I6	I7	I8	I9
D1	11272 77.112	43646 ∞	158999 ∞	171314 ∞
D2	11272 77.096	29700 65.669	148367 ∞	148367 ∞
D3	11272 77.096	11272 77.295	11272 77.295	11272 77.295

T5.- Producto Acotado

	I1	I2	I3	I4	I5
D1	14998 81.249	14998 80.546	15022 81.335	15030 80.261	15022 81.335
D2	15022 **79.178**	15022 81.335	15022 81.335	15022 81.335	15022 81.335
D3	15022 **79.178**	15022 81.335	15022 81.335	15022 81.335	15022 81.335

	I6	I7	I8	I9
D1	15022 81.337	14985 81.453	158380 ∞	169976 ∞
D2	15022 81.335	**14964** 83.661	148367 ∞	148367 ∞
D3	15022 81.335	15022 **79.178**	15022 **79.178**	15022 **79.178**

T6.- Producto Drástico

	I1	I2	I3	I4	I5
D1	**106159** ∞	**106159** ∞	**106159** ∞	**106159** ∞	**106159** ∞
D2	**106159** ∞	**106159** ∞	**106159** ∞	**106159** ∞	**106159** ∞
D3	**106159** ∞	**106159** ∞	**106159** ∞	**106159** ∞	**106159** ∞

	I6	I7	I8	I9
D1	**106159** ∞	**106159** ∞	153401 ∞	159376 ∞
D2	**106159** ∞	**106159** ∞	148367 ∞	148367 ∞
D3	**106159** ∞	**106159** ∞	**106159** ∞	**106159** ∞

Las dos tablas siguientes contienen los valores medios obtenidos por los distintos operadores de conjunción y funciones de implicación en ambas medidas (Entre paréntesis en la Medida de Convergencia, el número de combinaciones que pierden el control del sistema):

	T1	T2	T3
EC	**33237**	33360	34258
MC	107.954 (6)	**53.935 (4)**	74.167 (5)

	T4	T5	T6
EC	34755	35492	113006
MC	75.7065 (5)	80.887 (4)	∞ (27)

	I1	I2	I3	I4	I5
EC	**26812**	27211	26881	27261	26881
MC	**74.6 (3)**	78.2 (3)	79.5 (3)	77.6 (4)	79.5 (3)

	I6	I7	I8	I9
EC	26881	38428	111253	115198
MC	84.9 (3)	76.2 (6)	79.1 (13)	79.1 (13)

Para finalizar, presentamos una tabla que permite comparar los valores obtenidos por los seis controladores difusos que emplean la misma t-norma como operador de conjunción e implicación con aquellos que obtienen mejores valores en ambas medidas empleando la misma implicación con cualquier otro operador de conjunción. Entre paréntesis mostramos el método de defuzzificación y el operador de conjunción (en el segundo caso) con el que consiguen dichos valores:

	I1	I2	I3
EC	8941 (D2, D3)	9308 (D2, D3)	10588 (D1, D2, D3)
MC	34.518 (D1)	47.468 (D2, D3)	74.627 (D2, D3)
EC	8936 (T2-D1)	8941 (T1-D2,3)	8941 (T1-D1,2,3)
MC	34.518 (T1-D1)	47.468 (T2-D2,3)	47.468 (T2-D2,3)

	I4	I5	I6
EC	11272 (D2, D3)	15022 (D1, D2, D3)	106159 (D1, D2, D3)
MC	58.892 (D1)	81.335 (D1, D2, D3)	∞ (D1, D2, D3)
EC	8941 (T1-D2,3)	8941 (T1-D1,2,3)	8941 (T1-D1,2,3)
MC	47.468 (T2-D2,3)	47.468 (T2-D2,3)	47.468 (T2-D1,2,3)

8 Conclusiones

Teniendo en cuenta los resultados obtenidos en la Medida de Error Cuadrático, el mejor operador de conjunción en media es **T1**, la t-norma del **Mínimo**. De igual modo, esta t-norma empleada como función de implicación es la que mejor resultado presenta en valores medios. En cambio, estudiando los resultados de la Medida de Convergencia observamos como el mejor conectivo en media es la t-norma del **Producto de Hamacher (T2)** y la mejor función de implicación vuelve a ser la t-norma del **Mínimo (I1)** cuando es empleada como tal.

Por contra, los resultados obtenidos por la t-norma del **Producto Drástico (T6)** al ser empleada como operador de conjunción han demostrado la poca utilidad que puede proporcionar este conectivo en control difuso. En cuanto a las funciones de implicación, los resultados pueden parecer engañosos en cuanto a que los valores medios obtenidos por las R-implicaciones de **Göguen** y **Gaines (I8 e I9** respectivamente) en la Medida de Convergencia no parecen mucho peores que los restantes. Lo que ocurre es que trece de las dieciocho combinaciones que las emplean como función de implicación pierden el control del sistema durante la fase de captura de datos. Este hecho, junto con los pésimos resultados que presentan en la Medida de Error Cuadrático Medio hacen que sea aconsejable desestimar su empleo en controladores difusos. Unicamente la R-implicación de Gödel presenta resultados similares a los obtenidos por las t-normas, aunque también encontramos seis controladores difusos que pierden el control empleándola como implicación.

Estudiando los resultados destacados en los trabajos [1] y [2] que mostraban las R-implicaciones como la familia de funciones de implicación clásica mejor adaptada en control difuso, podemos concluir que las t-normas, al trabajar como funciones de implicación, son la mejor familia que se puede emplear para este cometido en control difuso. Analizando más detenidamente este hecho, observamos que las t-normas no son realmente funciones de implicación ya que no satisfacen algunas de las propiedades de las mismas ([5]). Las diferencias se hallan en las dos propiedades siguientes, verificadas por las funciones de implicación pero no por las t-normas:

- Si $x' \leq x$ entonces $I(x,y) \leq I(x',y)$
- $I(0,x) = 1$ (Principio de Falsedad)

Observamos como la segunda propiedad no parece adecuada para control difuso ya que provoca la aplicación de una acción de control concreta cuando las entradas procedentes del sistema y las reglas de la Base de Conocimiento no se emparejan de forma alguna. En realidad no debería de realizarse ninguna acción, hecho que verifican las t-normas con la propiedad $T(0,x) = 0$.

Bibliografía

[1] CÁRDENAS, E., CASTILLO, J.C., CORDÓN, O. y PEREGRÍN, A., *Estudio Comparativo de Sistemas de Inferencia y Métodos de Defuzzificación aplicados al control difuso del Péndulo Invertido*, Comunicaciones del III Congreso Español sobre Tecnologías y Lógica Fuzzy, Santiago, 1993, pp. 259-266.

[2] CÁRDENAS, E., CASTILLO, J.C., CORDÓN, O., HERRERA, F. y PEREGRÍN, A., *Influence of Fuzzy Implication Functions and Defuzzification Methods in Fuzzy Control*, Busefal 57, 1994, pp.69-79.

[3] GUPTA, M.M. y QI, J., *Theory of T-norms and Fuzzy Inference Methods*, Fuzzy Sets and Systems 40, 1991, pp. 431-450.

[4] GUPTA, M.M. y QI, J., *Design of Fuzzy Logic Controllers Based on Generalized T-operators*, Fuzzy Sets and Systems 40, 1991, pp. 473-489.

[5] TRILLAS, E. y VALVERDE, L., *On Implication and Indistiguishability in the Setting of Fuzzy Logic*, J. Kacpryzk, R. R. Yager, Eds., Management Decision Support Systems Using Fuzzy Sets and Possibility Theory (Verlag TÜV Rheinland, Köln, 1985), pp. 198-212.

[6] WEBER, S., *A General Concept of Fuzzy Connectives, Negations and Implications Based on T-norms and T-conorms*, Fuzzy Sets and Systems 11, 1983, pp. 115-134.

Aplicaciones

ANALISIS DE LAS RELACIONES ENTRE LOS ATRIBUTOS DE UNA BASE DE CONOCIMIENTO USANDO MEDIDAS DE INFORMACION

M.D. Villatoro Mármol
E.U.P.
Menéndez Pidal s/n
14004 Córdoba
i02vimad@lucano.uco.es

C. Hervás Martínez
E.U.P.
Menéndez Pidal s/n
14004 Córdoba
malhemac@lucano.uco.es

J.A. Gavilán Guirao
E.U.P.
Menéndez Pidal s/n
14004 Córdoba
i02gaguj@lucano.uco.es

Resumen

El objetivo de este artículo es llevar a cabo un estudio de las relaciones (implicaciones o equivalencias) entre los atributos que forman una base de conocimiento. Presentaremos un método general en el que se calcula la reducción de incertidumbre que produce en el consecuente de una relación el conocimiento de la misma utilizando medidas de vaguedad e información actuales y posteriormente nos centraremos en su aplicación con distintas medidas de información. La obtención de dichas relaciones tienen dos usos principales: la reducción de la dimensionalidad de los datos (número de características de los elementos) y su utilización como reglas de un Sistema Experto, presentando brevemente una aplicación del método.

1 Introducción

El trabajo que se ha realizado pretende introducirse en el problema de la reducción de dimensionalidad y en la obtención de las relaciones que más información posean en ese sistema a fin de sistematizarlas en un Sistema Experto basado en reglas[4]. Para ello nos basaremos fundamentalmente en dos tipos de medidas:

* Medidas de la vaguedad[7][2] de las características en el sentido de que una afirmación no puede ser juzgada como absoluta en un sentido.

* Medidas de información[1,5,8,10] asociadas con las medidas de vaguedad enunciadas en el punto anterior.

En definitiva, se trata de la aplicación del principio fundamental de simplificación expresado en términos de minimización de la pérdida de información con respecto a la reducción de complejidad.

Presentaremos en primer lugar el método general en que nos basaremos y posteriormente la aplicación de dicho método mediante el uso de cada una de las medidas de información usadas. En una primera fase de este trabajo[4] indicábamos la aplicación del método usando la entropía de Shannon y la entropía de Kosko, por lo que nos centraremos en su aplicación con dos nuevas medidas: la entropía de Yager y la U-incertidumbre. Se han realizado, así mismo, dos aplicaciones prácticas, una en un entorno educativo y otra en un entorno de visión. En este trabajo sólo presentaremos a modo de ejemplo la primera aplicación.

Todos los métodos presentados han sido codificados en C bajo Unix, en estaciones de trabajo Sun y se hallan operativos.

2 Procedimiento general

Para el procedimiento general de actuación nos hemos basado en un método

propuesto por Gaines y Shaw[3] para la inducción de reglas de inferencia basado en distribuciones de probabilidad y en la entropía de Shannon[1]. Supondremos una base de datos formada por un conjunto de m elementos, cada uno de los cuales consta a su vez de los valores para n atributos o características.

Consideraremos que cada una de las características que se va a medir queda definida por dos conjuntos difusos, uno de ellos asociado al extremo inferior de los valores del atributo y el otro asociado al extremo superior. Llamaremos a los conjuntos asociados al extremo izquierdo y derecho del atributo A, A(I) y A(D) respectivamente. Así, generaremos un número de conjuntos difusos igual a 2*n, dos para cada característica. Cada uno de estos conjuntos estará formado por un número de elementos igual a m y definido por una función de pertenencia de la forma:

$$\mu_C(e) : X \rightarrow [0,1]$$

donde C es un conjunto difuso asociado a uno de los dos extremos de uno de los atributos, e corresponde a uno de los elementos de la base de datos y X representa al conjunto de las n-tuplas, elementos o expertos que proporcionan las medidas. Para la formación de los conjuntos difusos se ha escogido una función de pertenencia lineal, de forma que:

$$\mu_{A(I)}(e) = \frac{r_s - r}{r_s - r_i}$$

$$\mu_{A(D)}(e) = \frac{r - r_i}{r_s - r_i}$$

$$\mu_{A(I)}(e) = 1 - \mu_{A(D)}(e)$$

donde r es el valor dado por el elemento e para el atributo A y r_s y r_i representan respectivamente el extremo superior e inferior de los valores de A. Con el objetivo de simplificar, representaremos como C(e) el grado de pertenencia del elemento e al conjunto difuso C. Así, por ejemplo, suponiendo una base de datos con 10 elementos con los siguientes valores para el atributo A, cuyo rango varía de 1 a 5:

$$A = \{5,2,2,4,3,2,1,2,3,4\}$$

obtendríamos los conjuntos difusos:

$$A(I)=\{0.00,0.75,0.75,0.25,0.50,0.75,1.00,0.75,0.50,0.25\}$$
$$A(D)=\{1.00,0.25,0.25,0.75,0.50,0.25,0.00,0.25,0.50,0.75\}$$

Una forma de medir el grado de certeza de la relación (implicación o equivalencia) entre dos conjuntos será mediante la lógica de Lukasiewicz en su forma L1, donde basándose en el principio de máxima especificidad, que considera el conjunto de los grados de pertenencia como una distribución de posibilidad sobre el conjunto, se define el grado de verdad de una relación de la siguiente forma:

$$\Box(C \rightarrow D) = MIN_e \min\{1 - C(e) + D(e), 1 - D(e) + C(e)\}$$
$$\Box(C \rightarrow D) = MIN_e \min\{1, 1 - C(e) + D(e)\}$$

Sin embargo, los datos así obtenidos, presentan dos problemas:

1.- No se da un grado de significación de estas relaciones, es decir del grado en que los datos las confirman.

2.- Pueden existir relaciones que son totalmente ciertas para la mayoría de los elementos y parcialmente ciertas sólo para unos cuantos elementos, mientras que otras relaciones son parcialmente ciertas para todos los elementos. Usando los grados de verdad de la relación no hay forma de distinguirlo.

Esto lleva al siguiente paso del análisis consistente en la determinación de la

significación de los datos. Esta significación está asociada a la medida de transferencia de información. El hecho de que algunas de las medidas de información estén definidas en base a distribuciones de probabilidad nos conduce a la definición de las mismas sobre los conjuntos difusos formados.

Tal y como hemos dicho en el párrafo anterior, definiremos, para cada conjunto difuso formado, una distribución de probabilidad de los grados de pertenencia de la siguiente forma:

$$P(C) = \{f|p(f): \exists \alpha, f = C(\alpha)\}$$

donde P(C) representa la distribución de probabilidad de los grados de pertenencia al conjunto difuso C y p(f) es la probabilidad (estimada mediante la frecuencia relativa) del grado de pertenencia f al conjunto C. Así, y siguiendo con el ejemplo anterior obtendríamos las siguientes distribuciones:

$P(A(I)) = \{1.00|0.1, 0.75|0.4, 0.50|0.2, 0.25|0.2, 0.00|0.1\}$

$P(A(D)) = \{1.00|0.1, 0.75|0.2, 0.50|0.2, 0.25|0.4, 0.00|0.1\}$

De esta forma generaremos un número de distribuciones de probabilidad igual al número de conjuntos difusos formados, es decir, 2*n. El método a seguir a continuación, una vez formados los conjuntos difusos y las distribuciones de probabilidad correspondientes, es el siguiente, para cada posible relación de implicación o equivalencia entre dos conjuntos:

1.- Calculamos el grado de verdad de la relación mediante la lógica L1 de Lukasiewicz.

2.- Calculamos el valor de la información del consecuente de la regla.

3.- Calculamos el valor de la información del consecuente de la regla, teniendo en cuenta el grado de verdad de la relación y el valor del antecedente de la misma.

4.- Obtenemos la diferencia entre los valores calculados en los puntos 2 y 3. Es esta diferencia la que mide la información que introduce la relación en la base de entrenamiento.

Así, la información introducida por la relación C -> D es:

$$\rho(C \rightarrow D) = I(D) - I(D|C, C \rightarrow D)$$

Una vez obtenidos los valores de la reducción de incertidumbre para todas las posibles relaciones, las podremos jerarquizar en función de la información que introducen, para su posterior uso en la reducción de la dimensionalidad de los datos, como reglas de inferencia para un Sistema Experto, etc.

3 Aplicación del método usando distintos tipos de entropía

Además de la visión tradicional de los conjuntos fuzzy existe otra caracterización de los mismos como puntos dentro de un hipercubo unidad[6], $[0,1]^m$ donde m es la cardinalidad del conjunto. Asociada a esta visión se hallan medidas de entropía basadas en distancia: la entropía de un conjunto se define como la distancia desde el punto que éste representa dentro del hipercubo al punto de entropía máxima.

Una de tales medidas de entropía, es la enunciada por Yager[9] en la que se realizan las siguientes consideraciones:

1.- Se define una distribución de probabilidad P como un vector o punto en el hipercubo.

2.- La entropía o incertidumbre de una distribución de probabilidad P se define como la distancia desde el punto que ésta define en el hipercubo hasta el punto de máxima entropía, el centro del hipercubo que es el punto definido por la distribución de probabilidad $p_i=1/m$ que llamaremos [1/m]. En nuestro caso, en lugar de trabajar con la distribución de probabilidad, lo

haremos con el conjunto formado al sustituir en los conjuntos difusos los grados de pertenencia por su probabilidad correspondiente, que para un conjunto C llamaremos C'.

Consideremos una función Q(C) que mide la distancia en el espacio m-dimensional de un conjunto de probabilidades C' al vector [1/m]. En cierta forma este valor se puede considerar como una medida de la no-incertidumbre de C. Con la distancia euclídea:

$$Q(C) = \left(\sum_e (p(C(e)) - \frac{1}{m})^2 \right)^{1/2}$$

si, por simplicidad, prescindimos de la raiz:

$$R(C) = \sum_e \left(p(C(e)) - \frac{1}{m} \right)^2$$

Llamaremos [I] a aquella distribución de probabilidad tal que uno de sus elementos toma valor uno y el resto valor cero, de forma que [I'] estará formado por valores todos iguales a no y R(C) toma valor máximo para C=[I]. Puesto que, como ya hemos dicho, R mide la no-incertidumbre de C, si definimos:

$$H(C) = R([I]) - R(C) = m * \left(1 - \frac{1}{m}\right)^2 - \sum_e \left(p(C(e)) - \frac{1}{m} \right)^2$$

obtenemos una medida de la entropía de C. Así:

$$I(D) = m * \left(1 - \frac{1}{m}\right)^2 - \sum_e \left(p(D(e)) - \frac{1}{m} \right)^2$$

$$I(D|C \rightarrow D) = m * \left(1 - \frac{1}{m}\right)^2 - \sum_e \left(p(D(e)|C(e), C \rightarrow D) - \frac{1}{m} \right)^2$$

$$\rho(C \rightarrow D) = I(D) - I(D|C \rightarrow D)$$

Así, siguiendo con el ejemplo presentado en la sección anterior y aplicando la expresión de la entropía, obtenemos los siguientes valores:

$$I(D) = 7.70$$

$$I(D|C \rightarrow D) = 7.53$$

$$\rho(C \rightarrow D) = 0.17$$

Por otra parte, otra entropía basada en la visión de los conjuntos fuzzy como puntos de un hipercubo[6] es la entropía de Kosko. Se trata de una entropía basada en la distancia al punto de máxima incertidumbre que es el centro del hipercubo (aquel cuyos grados de pertenencia toman todos el valor 0.5). Por otra parte, se define la inclusión de un conjunto difuso en otro como una relación a la que los conjuntos pueden tener diferentes grados de pertenencia:

$$S(C,D) = 1 - \frac{\sum_e \max\{0, C(e) - D(e)\}}{M(C)}$$

donde

$$M(C) = \sum_e C(e)$$

y de esta forma, con los conjuntos de la sec. 3:

$$S(C,D) = 1 - \frac{1.25}{5} = 0.75$$

El cálculo del grado de contenido del antecedente en el consecuente de la relación nos ha proporcionado una medida bastante precisa de la certeza de las relaciones. Ello es debido a que esta medida es indicativa tanto del número de infracciones que se cometen en la inclusión como de la gravedad de las infracciones cometidas.

4 U-Incertidumbre

La U-incertidumbre[3] es una medida de información, concretamente de no-especificidad, definida como:

$$U(r) = \sum_{i=1}^{m} (p_i - p_{i+1}) \log_2 |A_i|$$

donde p_i representa la distribución de posibilidad y p_{m+1} es igual a 0 por acuerdo. Es necesario aclarar que la distribución de posibilidad ha de estar ordenada de forma que p_{i+1} es menor o igual que p_i, y A_i es:

$$A_i = \{e \in X / r(e) \geq p_i\}$$

Una definición similar de la U-incertidumbre es:

$$U(r) = \sum_{i=1}^{n} (p_i - p_{i+1}) \log_2 i$$

y, como en nuestro caso tendremos una distribución de posibilidad por cada conjunto difuso formado, calcularemos su U-incertidumbre de la siguiente forma:

$$U(C) = \sum_{i=1}^{m} (C(e_i) - C(e_{i+1})) \log_2 i$$

donde e_i representa el elemento i-ésimo.

Para el caso de la U-incertidumbre condicionada utilizamos la restricción producida en el consecuente de la relación por el valor del antecedente y el grado de verdad de dicha relación según la lógica L1 de Lukasiewicz. Así el proceso que seguimos es distinto según que el consecuente de la regla sea un conjunto asociado al extremo izquierdo o al extremo derecho:

1.- Consecuente asociado al extremo izquierdo: Se considera cómo se modifica el intervalo por el extremo derecho según el valor del antecedente (siempre considerándolo como asociado al extremo izquierdo) y el valor de verdad de la relación. Así, por ejemplo, si el antecedente tiene un grado de pertenencia de 0.5 y el valor de verdad de la relación es 1, entonces el consecuente sólo puede tomar valores de pertenencia superiores a 0.5, lo que, considerando un rango entre 1 y 5, modifica el intervalo para el consecuente a un rango entre 1 y 3. El grado de pertenencia del consecuente a este nuevo intervalo será el que usemos para el cálculo de la U-incertidumbre condicionada.

2.- Consecuente asociado al extremo derecho: Se considera cómo se modifica el intervalo por el extremo izquierdo según el valor del antecedente (siempre considerándolo como asociado al extremo derecho) y el valor de verdad de la relación.

Una vez obtenida la distribución de posibilidad condicionada, calculamos su U-incertidumbre usando la misma expresión que para las distribuciones de posibilidad no condicionadas. De forma que la reducción de incertidumbre se calcula como:

$$p(C \rightarrow D) = U(D) - U(D|C \rightarrow D)$$

y, siguiendo con el ejemplo presentado en la sección 3 obtenemos los siguientes valores:

$U(D) = 1.7692039$

$U(D|C \rightarrow D) = 1.3474937$

$p(C \rightarrow D) = 0.4217102$

5 Resultados y conclusiones

En este trabajo se ha presentado un método para el cálculo de las relaciones entre los atributos que forman una base de entrenamiento, con todos los posibles usos de dichas relaciones. Para la aplicación de dicho

método se usan medidas de información tanto para distribuciones de probabilidad como de posibilidad.

Hemos realizado dos aplicaciones prácticas del método de las cuales nos centraremos en la de educación. Partimos, para ello, de un conjunto de notas y de respuestas a dos encuestas realizadas a los alumnos, una a principio y otra a final de curso. Un ejemplo de los resultados obtenidos son:

Relación	Shannon	Kosko	U-inc.	Yager
FINAL(I) -> E1(I)	8.55	0.89	0.46	595.27
E1(D) -> FINAL(D)	8.26	0.92	0.50	259.43
FINAL(I) -> E2(I)	7.84	0.94	0.39	486.51
E2(D) -> FINAL(D)	7.79	0.95	0.49	206.44
E2(I) -> FINAL(I)	4.80	0.86	0.43	218.79
E2(I) -> E1(I)	4.69	0.84	0.34	272.28
FINAL(D) -> E2(D)	4.55	0.89	0.36	78.271
E1(D) -> E2(D)	4.50	0.88	0.31	115.01
E2(D) <-> FINAL(D)	3.88	0.95	-	106.78

Como conclusiones generales de los resultados podemos señalar:

1.- Son resultados bastante similares usando los distintos métodos, ya que aunque no sigan exactamente el mismo orden, se encuentran dentro del mismo rango.

2.- Para el entorno de educación las variables que más información introducen son las de las notas y las de la encuesta posterior.

3.- Dentro de las respuestas de la encuesta posterior las más importantes son las del bloque relacionado con la calidad de la educación en función del profesor que imparte la asignatura y con la consideración por parte del profesor del trabajo, interés y esfuerzo de los alumnos.

BIBLIOGRAFIA

[1] Aczél J. and Daróczy Z., "On measures of information and their characterizations", Academic Press, 1975

[2] Alsina C., Trillas E. and Valverde L., "On some logical connectives for fuzzy sets theory", Journal of Mathematical Analysis and Applications, 93, pp 15-26, 1983.

[3] Gaines B.R. and Shaw M.L.G., "Induction of inference rules for expert systems", Fuzzy sets and systems, North Holland, pp 313-328, 1986.

[4] Hervás C., Gavilán J., Villatoro M. y Mohedano J., "Análisis de características y de sus implicaciones en un entorno educativo", III Congreso Español sobre Tecnologías y Lógica Fuzzy, Santiago de Compostela, 1993.

[5] Klir G.J. and Folger T.A., "Fuzzy sets, uncertainty and information", Prentice-Hall Int. Editions, 1988.

[6] Kosko B., "Neural networks and fuzzy systems. A dinamical systems approach", Prentice-Hall, 1992.

[7] Kruse T., Schwecke E. and Heinschn J., "Uncertainty and Vageness in Knowledge based systems", Springer-Verlog, 1991.

[8] Shannon C. and Warren W., "The mathematical theory of comunication", University of Illinois Press, 1971.

[9] Yager R., "On some measures of entropy related to agregation operators", Proc. de estudios de lógica borrosa y sus aplicaciones, pp 365-377, Universidad de Santiago de Compostela, 1993.

[10] Zadeh L.A., "Fuzzy sets as a basis for a theory of possibility", Fuzzy sets and systems, 1, pp 3-18, 1978.

Aplicación de la Lógica Fuzzy al Control de Inventarios.

Luis Miguel Marín Trechera
Escuela Universitaria Politécnica
Universidad de Cádiz
C. Chile s/n
11010 Cádiz, España
e-mail: Quico@czv1.uca.es

Francisco Mesa Varela
Escuela Universitaria Politécnica
Universidad de Cádiz
C. Chile s/n
11010 Cádiz, España
e-mail: paco@sunca1.uca.es

Resumen

Se aborda el problema del Control de Inventarios, con especial incidencia en la determinación de los diversos costes intervinientes en un modelo. Como alternativa, se propone el uso de técnicas de Lógica Fuzzy, de modo que a partir de un conjunto de reglas lingüísticas se determinen, mediante los procedimientos de fuzzificación y defuzzificación, los distintos costes. Se muestran ejemplos concretos y se exponen listados de programas que realicen los cálculos de los costes y la simulación de un Modelo de Inventarios.

Palabras clave: Inventario, lógica fuzzy, control.

1.- Descripción del problema de Control de Inventarios.

El Control de Inventarios es una disciplina incluida dentro de la Investigación Operativa cuyo objetivo es determinar la política de pedidos más adecuada a fin de optimizar los costes derivados del mantenimiento de materiales en un almacén. Como definición de Inventario adoptaremos la siguiente: [2]

> Una cantidad de mercancías o materiales en el control de una empresa y conservada durante un tiempo en un estado relativamente ocioso o improductivo esperando su uso posterior o su venta.

Son muchos los factores que intervienen en un Sistema de Inventarios. Además, siempre es posible añadir nuevas restricciones o condiciones que hacen que la complejidad del sistema vaya aumentado. Es por esto que, aunque se han estudiado gran cantidad de modelos, no existe una solución para la formulación general del problema. Es en esta circunstancia en la que la simulación se convierte en una eficaz herramienta de resolución, ya que permite la comparación de la efectividad de distintas alternativas. Un estudio más detallado de las características que debe tener un modelo de simulación y del análisis estadístico de los resultados puede encontrarse en [1].

En un modelo de Control de Inventarios podemos distinguir los siguientes elementos característicos:

- En primer lugar debemos de tener identificados cuáles son los ítems sobre los que se va a realizar el Control de Inventarios. Los sistemas más estudiados, por ser los más simples, son los de un único ítem. Diversos trabajos sobre modelos multi-nivel pueden encontrarse en [4].
- Se producen unas demandas del producto. Estas demandas pueden realizarse bien de forma determinista o bien de forma aleatoria. Se necesita, por tanto, para tener

identificado el modelo, conocer la distribución de las demandas.

- Con el fin de mantener el nivel del inventario, se realizan una serie de pedidos. La política óptima nos determinará cuándo realizar los pedidos y qué cantidad de producto pedir
- Al realizar un pedido, existe un tiempo de demora hasta su recepción. Como en el caso de las demandas, debemos de tener identificado la distribución de los tiempos de demora.

Además de los elementos anteriores, también tenemos una serie de costes asociados. La determinación de dichos costes es fundamental, ya que el objetivo del Control de Inventarios es precisamente el encontrar la política que optimice los costes. De hecho, la justificación para llevar un control de inventarios es que los costes evitados sobrepasen los costes de mantenimiento. Una pequeña clasificación de los costes intervinientes puede ser la siguiente:

- En primer lugar tenemos que considerar el coste de mantener el inventario. El hecho de tener los artículos almacenados supone un gasto en sí, ya que están ocupando un espacio, necesitan determinadas condiciones para evitar su deterioro, etc.
- También debemos considerar el coste de realizar un pedido, que normalmente constará de un coste fijo más el coste unitario por elemento pedido.
- Costes por demanda no satisfecha. Cuando nos quedamos sin existencias, no podemos satisfacer la demanda que se produzca, lo que nos acarrea una serie de costes como pueden ser penalizaciones recogidas en contratos, desaparición de la venta, pérdida del cliente, etc.

Llegamos, por tanto, a la conclusión de que para tener determinado un modelo de Control de Inventarios necesitamos conocer las distintas distribuciones estadísticas implicadas así como los distintos costes. Para la determinación de las distribuciones, podemos aplicar técnicas estadísticas de bondad de ajuste para comprobar si los datos se ajustan a alguna de las distribuciones más usuales. En el caso de que las demandas sean variables a través del tiempo se pueden aplicar estudios de series temporales. Todo esto puede hacerse de modo objetivo, realizando un estudio de los datos.

Sin embargo, la determinación de los costes no siempre resulta fácil. Así, podemos poner como ejemplo que mientras que determinar el coste unitario por cada ítem pedido o el coste global de realizar un pedido no suele ser difícil al tratarse de cantidades bien conocidas, no ocurre lo mismo con el coste por demanda insatisfecha. Este hecho es recogido en [2] y en [3].

En las experiencias acumuladas por los autores al intentar implementar un sistema de Control de Inventarios hemos podido constatar que al preguntar al gestor del almacén por dichos costes no se ha podido tener una respuesta concreta, ni tan siquiera al requerirle una estimación. En algunos casos se consideraba la demanda insatisfecha como algo no deseable bajo ningún concepto, por lo que implicaba de pérdida de negocio y de ganancia para la competencia. Sin embargo esta actitud cambiaba al preguntar hasta cuánto estaba dispuesto a gastar para que no se produjera esta situación era apreciable un cambio en la percepción del problema. Algunos gestores señalaban que su dificultad en dar una estimación estribaba en los diversos factores que intervenían en dicho coste, que además no eran considerados constantes a lo largo del tiempo, y que dependían fundamentalmente de las circunstancias del mercado.

Entre los factores más citados como intervinientes en la determinación de los costes por demanda insatisfecha podemos citar los siguientes: número de competidores, inversión publicitaria de la competencia, ofertas realizadas por la

competencia, cuota del mercado ocupada por la empresa, grado de fidelización de la clientela, etc.

También la determinación de las distribuciones de las demandas y del tiempo de demora entre la realización y la recepción de un pedido resulta difícil en la práctica. Usualmente se constata el hecho de que dichas distribuciones no son constantes a lo largo del tiempo, sino que sufren fluctuaciones que no son exclusivamente temporales, sino que dependen de las campañas publicitarias realizadas tanto por la propia empresa como por la competencia. Además, en la mayoría de los casos no se disponen de suficientes datos fiables como para poder estudiar la serie temporal, y no se está dispuesto a esperar a que se recopilen dichos datos antes de realizar el Control de Inventarios.

Llegamos pues a una situación en la que resulta muy difícil, cuando no imposible, tener definidos con precisión todos los elementos intervinientes en un Sistema de Inventarios ya resuelto analíticamente. Incluso para los casos en los que la solución analítica no es posible y se recurre a la simulación también es necesario tener determinados los costes y las distribuciones. Nos encontramos pues en una situación práctica de difícil resolución, aunque en la teoría se hayan avanzado soluciones para un nutrido conjunto de situaciones.

2.- Aportaciones de Lógica Fuzzy.

A pesar de todo lo anterior, sí es frecuente que los gestores de los almacenes sean capaces de expresar en términos lingüísticos en qué condiciones tendremos unos costes "grandes" y en qué condiciones dichos costes serán "pequeños". También es posible que sean capaces de explicitar aproximadamente entre qué límites se encuentran lo que entienden por "grande" o "pequeño". En estas circunstancias podemos expresar el modelo en términos de reglas que afectan a conjuntos difusos.

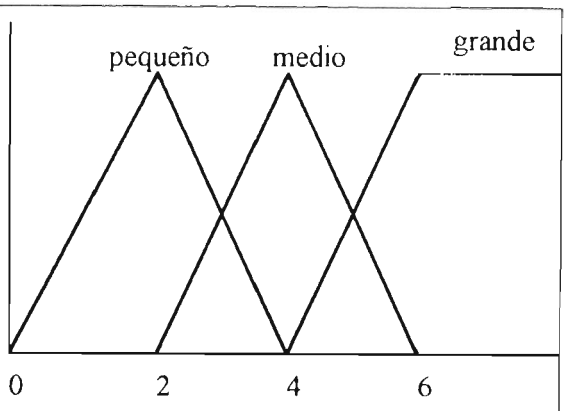


Figura 1.- Función de pertenencia para el número de competidores

Supongamos un modelo muy simplificado, en el que para determinar los costes por demanda insatisfecha consideraremos que estos sólo dependen del número de empresas que pueden ofertar artículos similares y de las inversiones publicitarias que realice la competencia. En estas circunstancias se nos puede plantear el siguiente conjunto de reglas:

COMPETENCIA	INVERSION	COSTE
pequeña	pequeña	pequeño
pequeña	media	media
pequeña	grande	media
media	pequeña	pequeño
media	media	media
media	grande	grande
grande	pequeña	medio
grande	media	grande
grande	grande	grande

A continuación, a partir de las funciones de pertenencia a los distintos conjuntos, podemos hacer corresponder a cada conjunto de entradas una salida. La técnica a emplear sería la siguiente:

1.- Dado un conjunto de entradas, x_1 y x_2, calcular el grado de cumplimiento w_i de cada una de las reglas. Dicho grado de cumplimiento se calculará como el producto de las funciones de pertenencia de los valores x_1 y x_2 a cada una de las premisas.

2.- Defuzzificar las salidas, hallando el centro de gravedad b_i de cada uno de los conjuntos que aparecen en las conclusiones.

3.- Calcular el valor estimado del coste mediante la ponderación $y = \Sigma w_i b_i / \Sigma w_i$

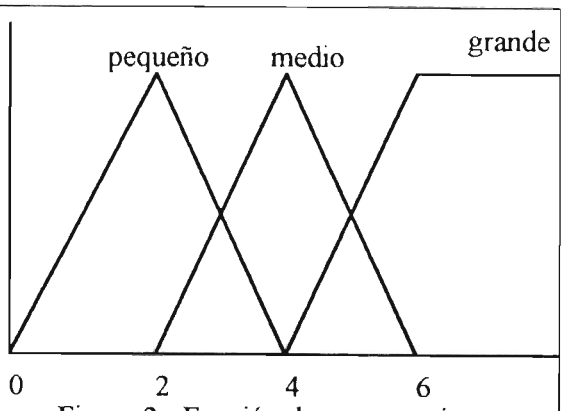


Figura 2.- Función de pertenencia para Inversión de la competencia (en millones).

Este procedimiento está detalladamente descrito y ampliado en numerosos trabajos, entre los que podemos citar [5] y [6] que incluyen diversos ejemplos y programas de ordenador.

Así, en nuestro ejemplo podríamos definir las funciones de pertenencia de las figuras 1, 2 y 3 .

Si con las funciones de pertenencia de los gráficos anteriores tenemos que el número de competidores es 3 y la inversión publicitaria es de 5 millones, procederíamos de la siguiente manera:

1.- Calcularíamos los grados de cumplimiento de cada una de las reglas. En nuestro caso todas las reglas tienen grado 0 excepto las números 5, 6, 8 y 9 que tienen de grado 1/4.

2.- Calcular los centros de gravedad de los conjuntos que definen la variable de salida. En nuestro ejemplo estos son 5 para poca, 10 para media y 15 para alta. Con lo que la salida de cada una de las reglas implicadas es de 10 para la regla 5 y 20 para las reglas 6, 8 y 9.

3.- Calcular el valor estimado, en nuestro caso 17.5.

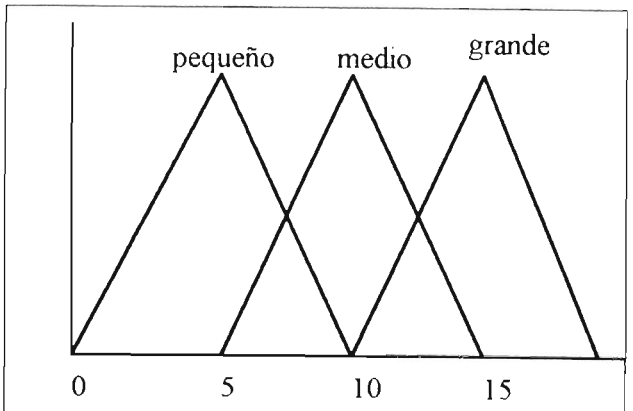


Figura 3.- Función de pertenencia para coste por demanda insatisfecha

Idéntico procedimiento puede utilizarse para la estimación de los parámetros de las distribuciones de las demandas y los tiempos de demora al verse afectado por factores externos (por ejemplo, una campaña publicitaria de la competencia) que no son cíclicos ni estacionales, con la cual no pueden ser previstos al analizar la serie temporal.

A continuación, una vez calculados estos datos, se podría proceder a realizar una evaluación de un determinado conjunto de políticas. Esto se podría realizar mediante un programa de simulación como el que se adjunta, escrito en el lenguaje de propósito específico SIMSCRIPT II.5. No se incluyen las rutinas de entrada y salida de datos, tan sólo los módulos de carácter más específico. Un listado completo de un programa similar puede encontrarse en [1].

```
PREAMBLE
THE SYSTEM HAS A PROB RANDOM STEP VARIABLE
AND OWNS A ALMACEN
DEFINE ALMACEN AS A FIFO SET
TEMPORARY ENTITIES
EVERY ITEM HAS A TPO.LLEG AND FCH.CAD AND
MAY BELONG TO THE ALMACEN
DEFINE TPO.LLEG AND FCH.CAD AS REAL
VARIABLES
PROCESSES INCLUDE ITEM.PROCESS AND PEDIDO
EVERY LLEGADA HAS A COMPRA
EVERY DEMANDA HAS A VOLUMEN
DEFINE VOLUMEN,COSTE_COMPRA AS AN INTEGER
VARIABLE
DEFINE INVENT .N ,I_MAS ,I_MENOS ,SP ,C, SG ,T
,C.FIJO ,C.UNI ,I ,J ,K,
INVENT_INICIAL,NUMERO_POLITICAS,NUM_MESES
AS INTEGER VARIABLES
DEFINE PROB AS A REAL,STREAM 6 VARIABLE
DEFINE DISTRIBUCION AS A TEXT,1-DIMENSIONAL
ARRAY
DEFINE
ME,MP,PB,KTER,MN,SD,QBET,PBET,PGAM,AGAM,PC,Q
C
            AS A REAL,1-DIMENSIONAL ARRAY
DEFINE PD,NB,QD,KTE,KERL, AS A
INTEGER,1-DIMENSIONAL ARRAY
DEFINE GENERA AS A REAL FUNCTION
DEFINE H,TPO.LIMITE AS REAL VARIABLES
DEFINE A1,A2 AS INTEGER VARIABLES
DEFINE CUM ,CUDI AS REAL VARIABLES
DEFINE MESES TO MEAN UNITS
ACCUMULATE COSTE_MANT AS THE AVERAGE OF
N.ALMACEN
ACCUMULATE COSTE_DEM AS THE AVERAGE OF
I_MENOS
TALLY MED_COSTE_COMPRA AS THE AVERAGE OF
COSTE_COMPRA
END

MAIN
OPEN UNIT 1 FOR INPUT.NAME ="P.DAT",
DEFAULT = "SYS$USUARIO:[CDCUS.TRE.SIMPRU]"
OPEN UNIT 2 FOR OUTPUT.NAME = "SALIDA.DAT",
DEFAULT = "SYS$USUARIO:[CDCUS.TRE.SIMPRU]"
USE 1 FOR INPUT
USE 2 FOR OUTPUT
CALL LECTURA.DATOS
FOR I=1 TO NUMERO_POLITICAS
DO
CALL INICIALIZA
START SIMULATION
LOOP
END

FUNCTION GENERA GIVEN DISTRIB,J AND SEMILLA
DEFINE DISTRIB AS TEXT VARIABLE
DEFINE SEMILLA AS INTEGER VARIABLE
DEFINE I,J AS A INTEGER VARIABLE
IF DISTRIB EQ "EXPONENCIAL"
 LET H=EXPONENTIAL.F(ME(J),SEMILLA)
ELSE
IF DISTRIB EQ "CONSTANTE"
 LET H=KTE(J)
ELSE
IF DISTRIB EQ "UNIFORME"
 LET H=UNIFORM.(PC(J),QC(J),SEMILLA)
ELSE
IF DISTRIB EQ "NORMAL"
 LET H=NORMAL.F(MN(J),SD(J),SEMILLA)
ELSE
IF DISTRIB EQ "BETA"
 LET H=BETA.F(PBET(J),QBET(J),SEMILLA)
ELSE
IF DISTRIB EQ "GAMMA"
 LET H=GAMMA.F(PGAM(J),AGAM(J),SEMILLA)
ELSE
IF DISTRIB EQ "CONSTANTE REAL"
 LET H=KTER(J)
ELSE
IF DISTRIB EQ "POISSON"
 LET H=POISSON.F(MP(J),SEMILLA)
ELSE
IF DISTRIB EQ "ERLANG"
 LETH=ERLANG.F(MERL(J),KERL(J),SEMILLA)
ELSE
IF DISTRIB EQ "UNIF_DISC"
 LET H=RANDI.F(PD(J),QD(J),SEMILLA)
ELSE
IF DISTRIB EQ "BINOMIAL"
 LET H =BINOMIAL.F(NB(J),PB(J),SEMILLA)
ELSE
IF DISTRIB EQ "PROBABILISTICA"
 LET H =PROB
ALWAYS ALWAYS ALWAYS ALWAYS ALWAYS
ALWAYS ALWAYS ALWAYS
                ALWAYS ALWAYS ALWAYS
ALWAYS
 RETURN WITH H
 END
```

Ejemplos de rutinas de un programa de simulación de inventarios

3.- Conclusiones.

Se ha expuesto una visión general de las problemas de Control de Inventarios así como de las dificultades que se plantean a la hora de determinar los costes y las distribuciones que aparecen en el modelo. Gracias a las técnicas de lógica Fuzzy se puede obtener una aproximación de dichos valores a partir de las reglas lingüísticas dadas por el gestor del almacén. Realizando posteriormente una simulación del proceso se puede elegir de entre las distintas políticas la que optimice los costes.

Referencias.

[1] Marín Trechera, L. M. Modelización y *Simulación de Sistemas. Aplicación a Sistemas de Control de Inventarios.* Tesina de Licenciatura. 1986.

[2] Love, S. F *Inventory Control.* McGraw-Hill. 1979

[3] Elsayed y Boucher. *Analysis and Control of Production Systems.* Prentice-Hall. 1985.

[4] Schwarz. *Multi-level Production/ Inventory Control Systems: Theory and Practice.* 1981

[5] Sugeno y Yasukawa. *A Fuzzy-Logic-Based Aproach to Qualitative Modeling.* IEEE Transactions on fuzzy systems. Feb 93

[6] Viot. *Fuzzy Logic in C* Dr Dobb's Journal. Feb 93

ESTUDIO DE FUNCIONES DE COMPOSICION DE EVIDENCIAS EN UN PROBLEMA DE RECONOCIMIENTO DE TEXTOS

Javier Etxanobe[1], José R. González de Mendivil[1], José R. Garitagoitia[2].

[1] Dpto. Electricidad y Electrónica
Facultad de Ciencias de Leioa.
Universidad del País Vasco.
Apartado 644.
48080 Bilbao España
email: webecarj@lg.ehu.es

[2] Dpto. Matemática e Informática
Universidad Pública de Navarra
Campus de Arrosadía s/n
Pamplona. España
email: wepgapaj@lg.ehu.es

Resumen:

En este trabajo se presenta un sistema de reconocimiento que utiliza en su etapa de procesamiento contextual un sistema deformado. El sistema deformado realiza transiciones entre conjuntos fuzzy siguiendo diferentes leyes de composición de evidencias. El trabajo muestra los resultados en un problema de reconocimiento de textos para cada una de las leyes de composición propuestas.

Palabras clave: Reconocimiento de Texto, Sistemas Deformados, Composición de Evidencias, Procesamiento Contextual.

1 Introducción

Los reconocedores de caracteres aislados (ICC), son capaces de reconocer hasta cierto límite debido a la presencia de distintas fuentes de error en los documentos de texto de entrada. En la literatura podemos encontrar diferentes métodos para mejorar las tasas de reconocimiento de estos sistemas, mediante la incorporación del contexto lexicográfico. Estos métodos se pueden clasificar en [1]:(i) Métodos estadísticos [2][3][4]; (ii) Métodos basados en diccionario [5][6][7]; y (iii) Métodos híbridos [8][9][10].

Todos estos métodos utilizan las decisiones previas obtenidas del reconocedor de caracteres aislados. Sin embargo, es conocido el hecho de que los reconocedores aislados obtienen una clasificación de la letra de entrada en función de las proximidades (en términos de semejanza) entre un grupo de caracteres patrón y la letra de entrada. Es posible recoger toda la información que proporcionan esos valores de proximidad y construir para cada letra de entrada un carácter fuzzy cuyas componentes representen estas proximidades, en lugar de un único carácter aislado de salida.

Para poder utilizar toda esa información conjuntamente con un diccionario que representa el contexto lexicográfico, es necesario definir un nuevo sistema de reconocimiento basado en el concepto de los Sistemas Deformados [12]. Dada la flexibilidad de este tipo de sistemas, es posible emplear las distintas funciones de composición de evidencias con objeto de comprobar cuál ofrece mejores resultados en un problema concreto de reconocimiento. Las funciones de composición de evidencias son la particularización, en la mayor parte de los casos, de la composición "max-estrella" definida entre conjuntos fuzzy.

En este artículo se hace la revisión de los sistemas deformados utilizando diferentes funciones de composición y se presentan los resultados obtenidos en un experimento de reconocimiento de textos manuscritos.

El resto del artículo se organiza de la siguiente forma: En la sección 2, se presenta formalmente el sistema deformado para un

autómata de estados finitos, y se describe el método para utilizar este sistema como herramienta que introduce el contexto lexicográfico en un sistema de reconocimiento. En la sección 3 se presentan las ecuaciones del sistema deformado utilizando las diferentes funciones de composición de evidencias. La sección 4 describe los resultados de un experimento de reconocimiento de textos manuscritos, en el que se utiliza un sistema deformado con las diferentes funciones de composición. Finalmente, la sección 5 muestra las conclusiones más importantes de este trabajo.

2 Sistemas Deformados

Una forma tradicional de representar el contexto lexicográfico definido por un diccionario es utilizando un autómata de estados. Este autómata recibe como entrada los caracteres que le proporciona un ICC y determina, después de hacer todas las transiciones, si la palabra está o no en el diccionario (comprobando si se ha alcanzado algún estado final).

Sin embargo, tal y como hemos dicho más arriba, nuestro propósito es utilizar toda la información que puede proporcionar el ICC para cada letra, y no únicamente el carácter de salida. Puesto que, toda esa información, la vamos a expresar en forma de conjuntos fuzzy, necesitamos un sistema que introduzca el contexto y que pueda manejar conjuntos fuzzy. Estos sistemas son los Sistemas Deformados [12][13]. A continuación presentamos la descripción formal de un sistema deformado para un autómata de estados, finito y determinista

Sea $A \equiv (\Sigma, Q, q_0, \delta, \lambda, \Delta)$ un autómata Moore, finito y determinista, que acepta el lenguaje definido por un diccionario D. Los elementos del autómata son: (i) Σ es el alfabeto, $\Sigma=\{\sigma_1, \sigma_2, ..., \sigma_r, ...\}$, y es tal que $D \subseteq \Sigma^*$; (ii) Q es el conjunto de estados; (iii) $q_0 \in Q$ es el estado inicial; (iv) δ es la función de transición definida por un grafo δ: $Q \times \Sigma \times Q \rightarrow \{0, 1\}$ siendo $\delta(q, \sigma, q') = 1$ si existe la transición de q a q' con el símbolo σ, y $\delta(q, \sigma, q') = 0$ si no existe tal transición; (iv) Δ es el alfabeto de salida y está definido como $\Delta = D \cup \{\varepsilon\}$ (ε denota la cadena vacía); (v) λ es la función de salida definida como λ: $Q \rightarrow \Delta$.

Dado un diccionario D, los elementos anteriores pueden ser definidos como sigue: Sea Q un conjunto finito de estados; inicialmente $\forall q, q' \in Q$: $\delta(q, \sigma, q') = 0$ y $\lambda(q) = \varepsilon$. Dada una secuencia $w=\sigma_1\sigma_2...\sigma_m$ de caracteres en Σ, y tales que $w \in D$, definimos por construcción una secuencia de estados $q_0q_1...q_m$ siendo q_0 el estado inicial y $q_t \in Q$ con $1 \leq t \leq m$, haciendo además que $\delta(q_{t-1}, \sigma_t, q_t) = 1$ con $1 \leq t \leq m$. Finalmente, $\lambda(q_m) = w$. El método se lleva a cabo de tal forma que el autómata obtenido sea determinista.

El Sistema Deformado [12] para el autómata anterior se define como una 6-tupla $AD \equiv ((\Sigma, \tilde{\sigma}), (Q, S), q_0, \delta, \lambda, (\Delta, T))$ donde Σ, Q, Λ son los conjuntos de entradas, estados y salidas definidos anteriormente, los cuales están limitados de forma fuzzy por los conjuntos fuzzy $\tilde{\sigma}$, S y T respectivamente; δ es la función de transición definida antes cuyo dominio esta limitado de forma fuzzy por $S \times \tilde{\sigma} \times S$; λ es la función de salida cuyo dominio está limitado de forma fuzzy por S y su rango por T. Consideramos los conjuntos fuzzy $\tilde{\sigma}$, S y T definidos como $\tilde{\sigma} = \{(\mu_{\tilde{\sigma}}(\sigma), \sigma), \sigma \in \Sigma\}$; $S = \{(\mu_S(q), q), q \in Q\}$; $T = \{(\mu_T(w), w), w \in \Delta\}$ siendo los respectivos universos Σ, Q y T.

Las ecuaciones de estado del sistema deformado son (t denota el paso t-ésimo):

(1) Dado $\delta(q_t, \sigma_t, q_{t+1}) \Longrightarrow \mu_S(q_{t+1}) \geq \min(\mu_S(q_t), \mu_{\tilde{\sigma}}(\sigma_t)\, \delta(q_t, \sigma_t, q_{t+1}))$.

(2) Dado $w = \lambda(q_t) \Longrightarrow \mu_T(w) \geq \mu_S(q_t)$.

De este modo, si tenemos una palabra fuzzy $\tilde{\sigma}_1 \tilde{\sigma}_2 \ldots \tilde{\sigma}_m$ el sistema deformado, realiza las transiciones con cada carácter fuzzy $\tilde{\sigma}_t$ ($1 \leq t \leq m$) de la palabra fuzzy de entrada: $s_0\sigma_1 s_1 \ldots \sigma_t s_t \ldots \sigma_m s_m$. A partir de s_m mediante la ecuación (2) se construye el conjunto T. La palabra final reconocida es aquella palabra del diccionario w para la cual $\mu_T(w)$ tiene el máximo valor en el conjunto T.

3 Composición de evidencias

Tal y como hemos visto en la definición que se da del sistema deformado, se utiliza como ley de composición de evidencias, la relación max-min. Sin embargo, es posible definir la ecuación (1) del sistema deformado de tal forma que admita diferentes leyes de composición. La ecuación (2) se mantiene para todos los casos

En efecto, podemos introducir las siguientes leyes de composición [11]:

I) Max-min.

$$\mu_S(q_{t+1}) \geq \min(\mu_S(q_t), \mu_{\tilde{\sigma}}(\sigma_t), \delta(q_t, \sigma_t, q_{t+1}))$$

Esta composición, presentada inicialmente, resulta muy drástica en muchos problemas de reconocimiento; así por ejemplo, si una palabra tiene una letra poco reconocible (borrosa, distorsionada, etc.), la función de pertenencia para dicha palabra, tendrá un valor muy bajo a pesar de que el resto de las letras estén reconocibles. Debido a este motivo, es preferible utilizar composiciones más suavizadas.

II) Max-Prod

$$\mu_S(q_{t+1}) \geq \mu_S(q_t) \cdot \mu_{\tilde{\sigma}}(\sigma_t) \cdot \delta(q_t, \sigma_t, q_{t+1})$$

La composición max-prod no tiene el inconveniente anterior puesto que al calcular el producto de las evidencias se están teniendo en cuenta todas ellas en conjunto. Sin embargo es dependiente de la longitud de la palabra

III) Promedio deslizante global

$$\mu_S(q_{t+1}) \geq \frac{\mu_S(q_t)t + \mu_{\tilde{\sigma}}(\sigma_t)}{t+1} \delta(q_t, \sigma_t, q_{t+1})$$

En este caso se efectúa una media aritmética entre todos los valores de la función de pertenencia. De este modo, a pesar de que haya en una palabra alguna letra poco legible, la palabra puede ser bien reconocida si el resto de las letras de la palabra están claras y reconocibles. Además, esta composición es independiente de la longitud de la palabra.

IV) Media cuadrática

$$\mu_S(q_{t+1}) \geq \frac{\mu_S(q_t)t + (\mu_{\tilde{\sigma}}(\sigma_t))^2}{t+1} \delta(q_t, \sigma_t, q_{t+1})$$

Al utilizar el cuadrado de la función de pertenencia conseguimos que la diferencia sea más acusada entre los valores bajos y los valores altos de la función de pertenencia.

V) Promedio local de relajación

$$\mu_S(q_{t+1}) \geq \alpha(\mu_S(q_t) + \gamma\, \mu_{\tilde{\sigma}}(\sigma_t)) \cdot \delta(q_t, \sigma_t, q_{t+1})$$

donde $0 \leq \alpha \leq 1$ y $\gamma = \dfrac{1}{\sum_{n=0}^{\infty} \alpha^n}$

Esta función tiene la propiedad de crecer y decrecer siguiendo localmente las evidencias de las transiciones con una "constante de tiempo" dependiente de α. También esta composición es independiente de la longitud de la palabra.

4 Resultados

Con el fin de estudiar el sistema deformado con cada una de las funciones de composición de evidencias, hemos desarrollado un sistema experimental de reconocimiento de textos. Este sistema trata de reducir los errores de clasificación que produce un ICC, mediante la incorporación del contexto lexicográfico a través del sistema deformado. Se ha construído la siguiente simulación:

En primer lugar creamos una colección de letras mayúsculas escritas a mano; en concreto 100 muestras para cada letra (100 para la A, 100 para la B, ...). A continuación cada una de estas letras es procesada por un ICC, que en nuestro caso es una red neuronal, obteniendo así, un conjunto de caracteres fuzzy (100 caracteres fuzzy para la A, 100 para la B, ...). En cada una de estas colecciones de caracteres fuzzy tendremos algunos caracteres fuzzy erróneos, es decir, caracteres fuzzy para los que el máximo de la función de pertenencia no se corresponde con la letra escrita debido a un error de clasificación de la red neuronal.

Una vez que disponemos de esta base de datos de caracteres fuzzy, tomamos un texto lo suficientemente extenso (6392 palabras; 36645 letras) y sustituimos cada letra por uno de sus respectivos caracteres fuzzy. Además, introducimos de forma controlada un cierto grado de error; es decir sustituimos algunas letras por caracteres fuzzy erróneos. De este modo, estamos simulando el comportamiento de un ICC cuando está reconociendo las letras de un texto manuscrito e introduce un cierto grado de error de clasificación.

A continuación el sistema deformado recibe como entrada el texto obtenido y procesa los caracteres fuzzy de cada una de las palabras dando como resultado el texto final. Este texto se compara con el texto de salida del ICC y calculamos así, en cuánto se ha reducido el error introducido por el clasificador aislado al haber tenido en cuenta el contexto lexicográfico.

Todo este proceso (sustituir en el texto de entrada cada letra por un carácter fuzzy introduciendo un cierto grado de error, procesar el resultado por el sistema deformado y calcular la tasa de reducción del error) se lleva a cabo repetidas veces para obtener resultados estadísticos. Además, en el sistema deformado se van utilizando las diferentes funciones de composición que hemos presentado en la sección anterior.

Las gráficas (I) a (V) muestran los resultados experimentales para cada función de composición. En concreto presentan, para cada porcentaje de (A) error introducido, (B) el porcentaje medio de reconocimiento absoluto y (C) el porcentaje medio de reducción del error. Mas exactamente:

$$\%\ \text{error introd.} = \frac{\text{nº caracteres fuzzy erron.}}{\text{nº letras en el texto}} \times 100$$

$$\%\ \text{reconocimiento} = \frac{\text{nº caracteres reconocidos.}}{\text{nº letras en el texto}} \times 100$$

$$\%\ \text{error reduc.} = 100 - \left(\frac{\text{nº caracteres erron. finales}}{\text{nº caracteres fuzzy erron.}} \times 100\right)$$

4.89	9.75	16.22	19.50	24.44	32.46	48.77	(A)
96.25	93.80	91.30	89.50	87.49	84.12	77.45	(B)
23.21	36.30	46.39	46.17	48.60	51.08	53.77	(C)
Max-min							

Fig. I
Resultados obtenidos con la ley de composición "**max-min**": A) % error introducido. B) % letras reconocidas. C) % reducción del error

4.89	9.75	16.22	20.50	24.44	32.46	48.77	(A)
99.46	99.27	99.01	98.93	98.70	98.19	96.66	(B)
88.98	92.47	93.91	94.49	94.66	94.41	93.16	(C)
max-prod							

Fig. II
Resultados obtenidos con la ley de composición "**max-prod**": A) % error introducido. B) % letras reconocidas. C) % reducción del error

4.89	9.75	16.22	19.50	24.44	32.46	48.77	(A)
99.27	98.87	98.24	98.05	97.30	96.04	91.22	(B)
85.06	88.46	89.16	90.00	88.90	87.81	82.00	(C)
promedio deslizante global							

Fig. III
Resultados obtenidos con la ley de composición "**promedio deslizante global**": A) % error introducido. B) % letras reconocidas. C) % reducción del error

4.89	9.75	16.22	19.50	24.44	32.46	48.77	(A)
98.82	98.00	96.67	96.23	94.72	92.18	83.52	(B)
75.80	79.55	79.46	80.68	78.31	75.91	66.22	(C)
Media cuadrática							

Fig. IV
Resultados obtenidos con la ley de composición "**media cuadrática**": A) % error introducido. B) % letras reconocidas. C) % reducción del error

4.89	9.75	16.22	19.50	24.44	32.46	48.77	(A)
99.28	98.88	98.25	98.07	97.37	96.05	90.81	(B)
85.31	88.47	89.19	90.10	89.20	87.84	81.16	(C)
promedio local de relajación							

Fig. V
Resultados obtenidos con la ley de composición "**promedio local de relajación (α=0.944)**": A) % error introducido. B) % letras reconocidas. C) % reducción del error

En estas tablas podemos ver cómo los valores más bajos corresponden a la función de composición Max-min, que como ya habíamos comentado es demasiado restrictiva.

Valores más altos se obtienen con el promedio deslizante global y con el promedio local de relajación, ya que en ambos casos se efectúa una evaluación de manera conjunta entre todas las evidencias de cada palabra.

Sin embargo, las tasas más altas se alcanzan con la composición max-prod que también supone una evaluación conjunta pero con la propiedad de que las evidencias más bajas hacen caer considerablemente el valor total

5 Conclusiones

En primer lugar, se ha hecho una evaluación de diversas leyes de composición de evidencias a través de un problema concreto de reconocimiento de textos manuscritos. En esta evaluación se ha podido determinar cuál o cuáles de esas funciones resulta más adecuada para el problema que nos ocupa. En concreto los mejores resultados se han obtenido con la ley "max-prod" y con los dos promedios. Por el contrario los resultados más bajos se obtienen con la ley "max-min" .

Por otra parte, los resultados obtenidos, nos permiten concluir que un sistema deformado con una ley de composición adecuada, constituye una herramienta muy útil para introducir el contexto lexicográfico en un sistema de reconocimiento de caracteres aislados.

Finalmente, y habida cuenta que el sistema deformado se construye a partir de un autómata de estados finitos, es posible aplicar el método a un problema de reconocimiento en el que el lenguaje a reconocer venga definido por un autómata tal.

Agradecimientos

El autor desea agradecer al Gobierno Vasco la financiación de este trabajo.

Referencias

[1] D.G. Elliman and I.T. Lancaster, A Review of Segmentation and Contextual Analysis Techniques for Text Recognition, *Pattern Recognition*, **23** (3/4), 337-346 (1990).

[2] W. Dost and J. Schurman, An Application of The Modified Viterbi Algorithm in Text Recognition, *Proc. 5th Int. Conf. Pattern Recognition*, Miami Beach, pp. 855-863 (1980).

[3] J. J. Hull and S.N. Srihari, Experiments in Text Recognition with Binary n-Grams and Viterbi Algorithm, *IEEE Trans. Pattern Analysis Mach. Intell.*, **PAMI-4**(5), 520-530 (1982).

[4] A. Goshtasby and R.W. Ehrich, Contextual Word Recognition Using Probabilistic Relaxation Labeling, *Pattern Recognition*, **21** (5), 455-462 (1988).

[5] R.L. Kashyap and B. J. Oommen, An effective algorithm for string correction using generalized edit distances, *Inform. Sci.*, **23**, 123-142 (1981).

[6] G.M. Landau, Fast string mathing with k differences, *J. Comput. Syst. Sci.*, **37**, 63-78 (1988).

[7] R.L. Kashyap and B. J. Oommen, Spelling Correction Using Probabilistic Methods, *Pattern Recognition letters*, **2** (4), 147-154 (1984).

[8] J.J. Hull, S.N. Srihari and R. Choudhari, An Integrated Algorithm for Text Recognition: Comparison with a Cascaded Algorithm, *IEEE Trans. Pattern Analysis Mach. Intell.*, **PAMI-5**(4), 384-395 (1983).

[9] R. Bozinovic and S Srihari, Off-Line Cursive Script Word Recognition, *IEEE Trans. Pattern Analysis Mach. Intell.*, **PAMI-11**(1), 68-83 (1989).

[10] R.M.K. Sinha, B. Prasada, G.F. Houle and M. Sabourin, Hybrid Contextual Text recognition with String Matching, *IEEE Trans. Pattern Analysis Mach. Intell.*, **PAMI-15**(9), 915-925 (1993).

[11] F. Casacuberta, E. Vidal. "Reconocimiento automático del habla". Marcombo, Boixerau Editores, Barcelona 1987.

[12] C.V. Negoita, D.A. Ralescu, Application of Fuzzy Sets to System Analysis, *Birkaeuser, Basilea 1975.*

[13] J. Echanove, R. Reina, J.R. Garitagoitia and J.R. González de Mendívil, Deformed Systems in Text Recognition, *International Conference On Artificial Neural Networks*., 26-29 Mayo 1994 pp. 977-980. Sorrento, Italia.

Un Procesador Elemental de Fuzzy SQL*

J. M. Medina Rodríguez, O. Pons Capote, A. Vila Miranda
Dpto. Ciencias de la Computación e Inteligencia Artificial
E.T.S. de Ingeniería Informática.
Universidad de Granada 18071. Granada (Spain)
email: medina@robinson.ugr.es
Tlf. 958-244079
Fax. 958-243317

Resumen

En esta comunicación se propone una alternativa para implementar una extensión difusa de SQL en un Sistema de Bases de Datos Relacionales Difusas. La idea central que subyace a esta propuesta es la de construir un procesador de FSQL empleando los mecanismos de representación y de manipulación brindados por el SBDR que opere como anfitrión. Para llevar a cabo este propósito, nos basamos en la formulación de un modelo teórico de Bases de Datos Relacionales Difusas [11], en la adopción de un esquema para la representación e implementación de información difusa en SBDR convencionales [10] y en las herramientas para el desarrollo de aplicaciones proporcionadas por el SBDR anfitrión. Como resultado de todo ello, el procesador de FSQL es capaz de traducir sentencias formuladas en FSQL a sentencias SQL clásicas directamente ejecutables por el SBDR adoptado como anfitrión.

1 Introducción

Partiendo del modelo de Bases de Datos Relacionales, propuesto en [5], han surgido diversas aproximaciones, que tratan de proporcionar un marco teórico para la representación y tratamiento de información de naturaleza difusa. Los enfoques aparecidos, que se basan en el empleo de la teoría de subconjuntos difusos [23] como herramienta de representación y manipulación, se agrupan principalmente en torno a dos tendencias: *Modelos de Unificación Mediante Relaciones de Similitud* y *Modelos Relacionales sobre Distribuciones de Posibilidad.*

La primera línea se encuentra recogida en los trabajos [2, 3, 4, 1], con aportaciones adicionales en [17], [18] y [16]. Dentro del segundo enfoque existen varias aproximaciones segun las propuestas de Umano, [20], la de Prade y Testemale, [14] y la contribución de Zemankova en [24].

En [11], (tambien en [9] y [12]), presentamos un modelo, denominado GEFRED, que pretende aglutinar los aspectos más relevantes de los anteriores enfoques en un marco teórico común.

Una vez establecidas las bases teóricas para la extensión difusa de las bases de datos relacionales, nos enfrentamos con el problema de encontrar los mecanismos para construir sistemas relacionales que operen de acuerdo con estos principios. Estos sistemas denominados genericamente Sistemas de Bases de Datos Relacionales Difusas, (SBDRD en lo sucesivo), precisan de un modelo teórico y de unas directrices de construcción con las que se pueda abordar una implementación operativa. Existen varias propuestas en este sentido, en su mayoria defendidas por los propursores de los diferentes modelos teóricos [19, 6, 14, 24, 7].

El problema de construir SBDRD debe abordarse siguiendo un esquema basado en dos requisitos básicos:

- Utilizar como soporte teórico un modelo de bases de datos relacionales difusas apropiado.
- Adoptar los criterios de representación y manipulación de la información difusa que favorezcan la eficiencia del sistema.

Para satisfacer el primer requisito es preciso que el modelo adoptado suponga una extensión del modelo relacional clásico, (lo cual implica que contemple a éste como un caso particular del caso difuso en el que los datos a tratar son precisos), y, que proporcione tratamiento para una amplia gama de información de naturaleza difusa.

El segundo requisito concierne a determinados criterios que deben conducir la especificación del SBDRD, en el sentido de que adoptemos aquellas alternativas que nos lleven a realizaciones operativas.

En [10] y [12] se muestran algunas ideas para abordar la construcción de SBDRD,s conforme a ambos requisitos. En el apartado segundo se resume el esquema general propuesto en dichos trabajos.

Es especialmente importante a la hora de diseñar un SBDRD el contemplar un lenguaje que recoja todos los aspectos relativos al tratamiento de la información

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

difusa preservando, además, las operaciones originales del álgebra relacional clásica. Dada la difusión de SQL como lenguaje de manipulación de SBDR.s. parece adecuado abordar una extensión del mismo que contemple las nuevas posibilidades.

Esta comunicación contempla una propuesta de procesador de FSQL construida sobre un SBDR clásico. Dicho procesador, que se integra perfectamente en el esquema recogido en el apartado segundo, permite traducir sentencias SQL difusas a sentencias SQL normales directamente ejecutables por el SBDR clásico. Las sentencias que soporta dicho procesador proporcionan los mecanismos adecuados para expresar el álgebra propuesta en GEFRED [11].

El próximo apartado proporcionará el marco estructural necesario para la formulación del procesador propuesto. Seguidamente, se abordará la descripción del procesador de FSQL y de su forma de operar en el ámbito del SBDRD mostrado en apartado segundo, para ello introduciremos una sintaxis básica compuesta de cláusulas y funciones "difusas". Completaremos el estudio mostrando un ejemplo de como el PFSQL transforma un sentencia difusa en una sentencia SQL clásica.

2 Esquema General de un SBDRD

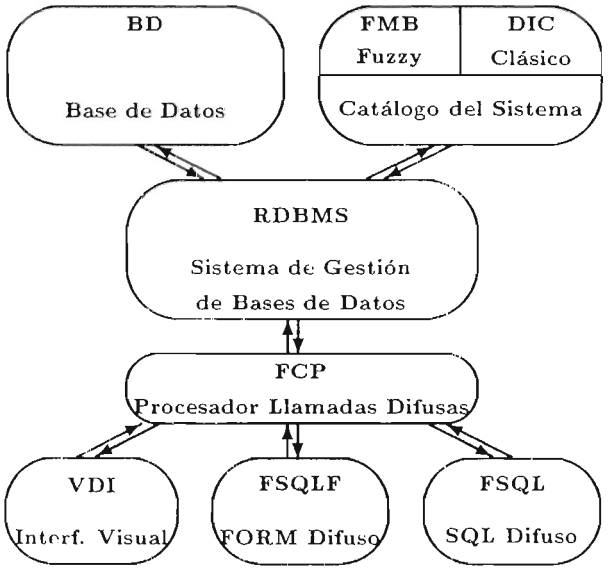


Figura 1: Esquema General de un SBDRD

La figura 1 muestra el esquema general del SBDRD sobre el que está desarrollado el procesador de FSQL. Como la idea de partida es la de construirlo sobre un gestor de bases de datos convencional, todos los desarrollos a realizar toman a dicho gestor como el elemento principal al que van dirigidas todas las peticiones.

A continuación daremos un breve recorrido por los diferentes módulos que constituyen el esquema propuesto:

- **RDBMS**. Como hemos dicho, todas las operaciones que hayamos concebido para la extensión difusa que representa nuestra implementación, se traducirán a peticiones al RDBMS anfitrión. Estas peticiones se realizarán empleando el leguaje de soporte del mismo. Generalmente, y este es nuestro caso, todas las peticiones al sistema se resuelven mediante sentencias SQL clásicas. Al final, todo el procesamiento, "difuso" o no, se traducirá a una serie de peticiones al sistema en lenguaje SQL.

- **BD**. La base de datos aglutina toda la información permanente, "difusa" o no, mediante las tablas de la base de datos. La forma en que se representan los datos en dichas tablas dependerá de la naturaleza y del tipo de los mismos. Nosotros usaremos los recursos disponibles en el sistema anfitrión para representar la información "difusa" que precisemos.

- **FMB**. El diccionario o catálogo de un SBDR representa aquella parte del sistema que almacena información sobre los datos recogidos en la Base de Datos, así como otro tipo de informaciones: usuarios, permisos, accesos, datos de control, etc. En nuestro caso extenderemos esta parte del sistema a fin de que pueda recoger aquella información necesaria relacionada con la naturaleza "imprecisa" de la nueva colección de datos a procesar. Denominaremos esta extensión como Base de Metaconocimiento Difuso, (FMB), y se organizará siguiendo la filosofía imperante en el catálogo del RDBMS anfitrión.

- **FCP**. Pretendemos con nuestra implementación, transformar el conjunto de operaciones "difusas" a soportar, en un conjunto de peticiones al SBDR anfitrión incluido en el conjunto de instrucciones que éste recoge. La misión del FCP consiste en traducir las operaciones "difusas" formuladas sobre el SBDRD en peticiones ordinarias al sistema anfitrión. Esta tradución tiene en cuenta la operación "difusa" requerida, los item difusos implicados en dicha operación, la representación de los mismos en la base de datos y la información que sobre estos posea la FMB. Por tanto, el FCP lleva implementada la forma de proceder para resolver en términos clásicos las peticiones difusas.

- **FSQL** Este modulo contempla la extensión "difusa" de SQL objeto de este trabajo. Se encargará de traducir esa sintáxis a operaciones sobre el SBDR por medio del procesador de FSQL y hará uso para ello de las rutinas del FCP.

Hasta aquí lo que podríamos considerar como módulos básicos del SBDRD propuesto.

Con la introducción de un lenguaje de consulta y manipulación tal como FSQL, tenemos resuelto en primera instancia el problema de la interfaz de un

SBDRD con el usuario. A partir de aquí, se puede pensar en la realización de otros tipos de interfaces o de lenguajes de consulta basados en entornos gráficos, lenguajes de 4^a Generación, etc

3 Descripción del Procesador de FSQL

Como hemos comentado previamente, si combinamos una representación apropiada para la información difusa, con una adecuada implementación de la misma a través la estructura de datos proporcionada por el SBDR anfitrión, es posible elaborar procedimientos que trasladen operaciones de naturaleza difusa a un conjunto de sentencias directamente ejecutables por un SBDR clásico. En este apartado vamos a ilustrar algunos aspectos de este proceso apoyándonos en una versión reducida de un procesador de FSQL (PFSQL en lo sucesivo) construido sobre el SBDR Oracle [13]. La filosofía que sustenta dicho procesador es la de integrar en una sintaxis SQL determinada un conjunto de cláusulas y funciones especiales que permitan expresar peticiones "difusas" al sistema. Según el esquema propuesto será posible expresar sentencias que involucren cláusulas "difusas" y cláusulas clásicas.

A continuacón mostramos una descripción BNF de las nuevas cláusulas "difusas" que incorpora este procesador:

```
fcond_simp    :   fcond_sin umbral
              |   fcond_sin
              ;

fcond_sin     :   column_item fcomp fexpr
              ;

umbral        :   THOLD NUMERO
              ;

fexpr         :   '$' ID
              |   NUMERO
              |   '#' NUMERO
              |   '[' NUMERO ',' NUMERO ']'
              |   '$' '[' NUMERO ',' NUMERO
                  ',' NUMERO ',' NUMERO ']'
              ;

fcomp         :   FEQ
              |   FGT
              |   FLT
              |   FGEQ
              |   FLEQ
              ;

grado_comp    :   CDEG '(' column_item ')'
              ;

column_item   :   id_usuario '.' id_tabla
                  '.' id_atributo
              |   id_tabla '.' id_atributo
              |   tabla_alias '.' id_atributo
              |   id_atributo
              ;
```

donde los símbolos no terminales `id_usuario`, `id_tabla`, `id_atributo` y `tabla_alias` son identificadores de los objetos: usuario, tabla, atributo y alias de tabla respectivamente, y siguen las reglas de formación establecidas en la sintaxis de Oracle. El símbolo terminal `NUMERO` identifica en la sintaxis a un dato con formato numérico, el analizador semántico se encargará de comprobar el rango y tipo para cada ocurrencia en la entrada. Los símbolos entre comillas simples son literales que pertenecen a la regla de producción. Los símbolos terminales restantes son palabras reservadas de la nueva sintaxis que permiten expresar condiciones, modificadores y funciones "difusas". Las siguientes frases serían sentencias válidas de la sintaxis de FSQL adoptada, (suponiendo la existencia de las tablas y etiquetas referenciadas):

```
SELECT edad, CDEG(edad) FROM personal
WHERE edad FEQ $joven THOLD 0.8

SELECT a.emp#,nombre,estudios,CDEG(estudios)
FROM personal a, aptitud b
WHERE a.emp#=b.emp# AND
estudios FEQ $licenciado THOLD 0.6

SELECT emp#,dpto#,puesto#,sueldo,
CDEG(sueldo),comision,CDEG(comision)
FROM empleos
WHERE sueldo FEQ $alto
AND comision FEQ $baja THOLD 0.8
```

El símbolo no terminal `fcomp` representa la batería de comparadores difusos comptemplados en la sintaxis de FSQL. La sintaxis completa de una comparación difusa atómica tiene la forma:

```
<atrib> fcomp <cte_difusa> [THOLD <umbral>]
```

donde los corchetes indican que el umbral puede aparecer o no en la condición, en este último caso, se toma un valor umbral por defecto que, en el caso que nos ocupa, vale 0.5.

Las condiciones compuestas se obtienen mediante el empleo de los conectivos NOT, AND y OR, pudiéndose conectar condiciones atómicas "difusas" con otras que no lo sean.

Para visualizar los grados de compatibilidad que presenta un atributo determinado se emplea la sintaxis:

```
CDEG(<atributo>)
```

donde, `<atributo>` identifica al atributo del que queremos presentar los grados de compatibilidad.

3.1 Funcionamiento del Procesador de FSQL

La figura 2 muestra el esquema del funcionamiento del procesador de FSQL. En dicha figura se muestran los módulos implicados en el procesado de una sentencia expresada en términos de FSQL, desde la entrada, hasta la obtención de los resultados, si los hubiera, en la salida.

Ayudados por la figura 2 vamos recorrer el procesado de una sentencia FSQL a través del Procesador de FSQL.

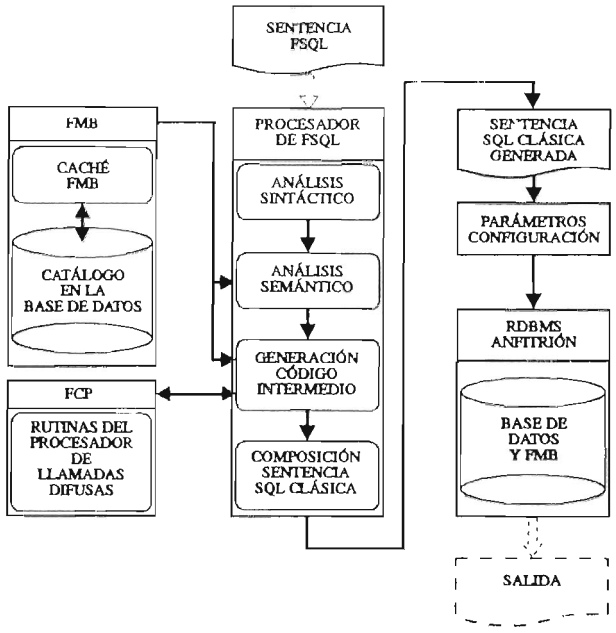


Figura 2: Procesado de una Sentencia FSQL

1. Dada una sentencia supuestamente expresada en FSQL, de acuerdo con la sintaxis introducida en este apartado, lo primero que hace el PFSQL es determinar si ésta adopta la sintaxis adecuada. Esto se realiza mediante el analizador sintáctico incorporado.

2. Simultaneamente a este proceso, el PFSQL , mediante el analizador semántico, se encarga de comprobar que todos los elementos "difusos" implicados se encuentran correctamente empleados. Para ello recurre a la información presente en la FMB ("Fuzzy Metaknowledge Base") acerca de dichos elementos.

3. Conforme se cumplimentan los procesos anteriores, el PFSQL vá generando una estructura de datos que organiza la información de partida en dos grupos de proceso:

   - El grupo de cláusulas presentes en la sentencia original que no precisan tratamiento "difuso". Sobre este grupo no se realizará ninguna operación especial.
   - Y, el grupo relativo a las cláusulas "difusas" inmersas en la sentencia de entrada.

   A partir de este último grupo, el PFSQL extrae de la FMB y de las propias cláusulas "difusas", los parámetros que se encuentren definidos al respecto, y los envía al FCP, ("Fuzzy Call Procesor"), para que genere los fragmentos de sentencias SQL correspondientes a la función requerida por dichas cláusulas. Toda la información generada queda convenientemente organizada mediante una estructura de datos adoptada al efecto.

4. Una vez realizados los pasos anteriores, el PFSQL, a partir de la información de que dispone, está en condiciones de reconstruir la sentencia SQL final

5. La sentencia generada es directamente ejecutable por el RDBMS anfitrión por lo que puede ser enviada al mismo para su inmediato procesado. No obstante, su ejecución podría verse modificada mediante la adición de una serie de parámetros de configuración que influyan sobre el formato de salida. Dichos parámetros pueden tener una fuerte dependencia del sistema empleado para ejecutarla, por lo que no entraremos en ellos.

   Dependiendo del tipo de sentencia procesada, DDL, DCL o DML, la ejecución de la sentencia generada puede, o no, generar una salida, por este motivo dicha acción se encuentra representada mediante linea discontinua en la figura 2. Cuando se trata de sentencias DDL y DCL, es posible que la FMB sufra alteraciones en su contenido de acuerdo con las modificaciones introducidas.

En los pasos anteriormente descritos están contemplados mecanismos de detección de errores que funcionan a varios niveles. A nivel del analizador sintáctico, el PFSQL detecta errores sintácticos en la entrada. Los errores semánticos son localizados, en primera instancia, cuando tienen su origen en el incorrecto empleo de items "difusos", posteriormente, en el proceso de ejecución por parte del RDBMS anfitrión, serán detectados el resto de los errores semánticos. También el gestor anfitrión será el encargado de detectar posibles errores de ejecución, por lo que habrá que remitirse a los códigos de error generados por el mismo.

4 Ejemplo de Procesamiento de una Consulta por el PFSQL

NOMBRE	DIRECCIÓN	EDAD	REND	SAL
Luis	Recogidas	31	Bueno	Alto
Antonio	Reyes Católicos	Maduro	Regular	100000
Juan Carlos	Camino Ronda	Joven	Malo	90000
Francisco	P. A. Alarcón	Mayor	Excelente	Bajo
Julia	Puerta Real	Joven	Bueno	Medio
Inés	Manuel de Falla	#28	Bueno	125000
Javier	Gran Vía	*30,35	Regular	105000

El símbolo # significa "aproximadamente". El símbolo * representa un valor intervalar.

Tabla 1: EMPLEADOS

Para ilustrar el funcionamiento del PFSQL descrito, vamos a emplear el ejemplo que se muestra en la tabla 1 que consiste en una relación que recoge datos, algunos de ellos de naturaleza difusa, sobre un conjunto de empleados.

Los atributos NOMBRE y DIRECCIÓN contienen información "crisp", siendo el primero, la clave primaria de la relación La información que contienen se implementa en la Base de Datos de la misma forma que lo hace el RDBMS anfitrión puesto que no recibirá tratamiento "difuso". Los atributos EDAD

$s_e(d,d')$	Malo	Regular	Bueno	Excelente
Malo	1	0.8	0.5	0.1
Regular	0.8	1	0.7	0.5
Bueno	0.5	0.7	1	0.8
Excelente	0.1	0.5	0.8	1

Tabla 2: Relación de Semejanza sobre RENDIMIENTO.

y SALARIO almacenan información "difusa". Sobre el atributo EDAD se encuentran definidas las siguientes etiquetas lingüísticas: *"Joven"* dado por la distribución trapezoidal [0,16,30,40], *"Maduro"* dado por [25,35,45,55] y *"Mayor"* dado por [40,50,65,80], además, el concepto *"aproximadamente n"* se representa mediante la distribución [n-5,n,n,n+5]. Sobre el atributo SALARIO las etiquetas son: *"Bajo"* representado como [0,65000,85000,95000], *"Medio"* por [85000,95000,110000,130000] y *"Alto"* dado por [110000,130000,180000, 1000000].

El atributo RENDIMIENTO permite también información "difusa", pero sobre un domininio escalar, por tanto, para poder realizar una consulta que contenga dicho atributo, necesitarémos tener definida previamente una "relación de semejanza" sobre su domino. Dicha relación se puede ver en la tabla 2

A continuación veremos como se resuelve la siguiente consulta:

"Dame el NOMBRE, la DIRECCIÓN, la EDAD, así como el grado en que se satisface la condición para el atributo EDAD, de aquellos empleados con una EDAD "madura" (con un grado $\geq$ 0.8)"

En términos de la sintaxis de FSQL, la anterior consulta queda formulada como sigue:

```
SELECT nombre, direccion, edad, CDEG(edad)
FROM empleados
WHERE edad FEQ $madura THOLD 0.8
```

De acuerdo con los criterios de representación adoptados, el PFSQL compone la siguiente sentencia SQL clásica:

```
select nombre,direccion,DECODE(EDADT,0,
'UNKNOWN',1,'UNDEFINED',2,'NULL',3,TO_
CHAR(EDAD1),4,DECODE(EDAD1,0,'JOVEN',1,
'MADURO',2,'MAYOR'),5,'['||TO_CHAR(
EDAD1)||',''||TO_CHAR(EDAD4)||']',6,'['
||TO_CHAR(EDAD1)||',''||TO_CHAR(EDAD1+
EDAD2)||',''||TO_CHAR(EDAD4)||']',7,'['
||TO_CHAR(EDAD1)||',''||TO_CHAR(EDAD2+
EDAD1)||',''||TO_CHAR(EDAD3+EDAD4)||','
||TO_CHAR(EDAD4)||']') "EDAD",ROUND(
DECODE(EDADT,0,1,2,1,3,DECODE(SIGN((EDAD1
-33)*(47-EDAD1)),-1,0,DECODE(SIGN(35-
EDAD1),1,(EDAD1-25)/10,DECODE(SIGN(EDAD1
-45),1,(EDAD1-55)/-10,1))),4,DECODE(EDAD1,
1,1.00,0),5,DECODE(SIGN((EDAD4-33)*(47-
EDAD1)),-1,0,DECODE(SIGN(35-EDAD4),1,
(EDAD4-25)/10,DECODE(SIGN(EDAD1-45),1,
(EDAD1-55)/-10,1))),6,DECODE(SIGN((EDAD4
-37)*(53-EDAD4)),-1,0,DECODE(SIGN(35-
EDAD4+5),1,(EDAD4-25)/(10+5),DECODE(SIGN(
45-EDAD4+5),1,1,(EDAD1-55)/(-10-5)))),7,
DECODE(SIGN((0.80*EDAD3+EDAD4-33)*(47
-0.80*EDAD2+EDAD1)),-1,0,DECODE(SIGN(
(EDAD3+EDAD4-35)*(45-EDAD2+EDAD1)),-1,
DECODE(SIGN((0.80*EDAD3+EDAD4-33)*(35-
EDAD3+EDAD4)),-1,DECODE(SIGN((47-0.80*
EDAD2+EDAD1)*(EDAD2+EDAD1-45)),1,(55
-EDAD1)/(EDAD2--10)),(EDAD4-25)/(10-
EDAD3)),1)),0),2) "CDEG(EDAD)" from
empleados where (EDADT=0 OR EDADT=2 OR
EDADT=3 AND EDAD1 BETWEEN 33 AND 47 OR
EDADT=4 AND EDAD1 IN (1) OR EDADT=5 AND
 EDAD1<=47 AND EDAD4>=33 OR EDADT=6 AND
EDAD4 BETWEEN 37 AND 53 OR EDADT=7 AND
EDAD4>=33 AND EDAD1<=47 AND 0.80*EDAD3+
EDAD4>=33 AND 0.80*EDAD2+EDAD1<=47)
```

El resultado[1] de ejecutar esta sentencia sobre el SBDR anfitrión es el que se muestra en la tabla 3.

NOMBRE	DIRECCIÓN	EDAD	CDEG(EDAD)
Antonio	Reyes Catolicos	Maduro	1
Javier	Gran Via	[30,35]	1

Tabla 3. Resultado de la consulta ejemplo

5 Conclusiones

En este trabajo se ha presentado el esquema de un módulo que permite traducir sentencias expresadas en FSQL a expresiones SQL estándar. El mencionado módulo se encuentra integrado en un esquema de Gestor de Bases de Datos Relacionales Difusas [10] construido sobre los conceptos teóricos formulados en GEFRED [11]. A traves de dicho Gestor es posible desarrollar extesiones "difusas" de SBDR convencionales disponibles en la actualidad, ello permite preservar las inversiones realizadas sobre dichos sistemas a la par que potenciar sus características de representación y manipulación de información.

Las líneas futuras de trabajo irán encaminadas hacia la especificación y desarrollo de un prototipo completo de procesador de FSQL siguiendo el esquema esbozado en este trabajo, así como el desarrollo de los módulos descritos en la arquitectura de SBDRD que lo sustenta. También está previsto acometer el estudio teórico del tratamiento de información "difusa" bajo otros modelos de Bases de Datos, como el modelo de Bases de Datos Orientadas a Objeto.

Bibliografía

[1] ANVARI M., ROSE G.F. "Fuzzy Relational Databases". Analysis of Fuzzy Information Bezdek ed. Vol II CRC Press. (1987)

[1] Esta sentencia y su ejecución han sido elaborados empleando la implementación de un SBDRD desarrollado sobre la plataforma Oracle

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

[3] BUCKLES B.P., PETRY F.E. "Extending the Fuzzy Database with Fuzzy Numbers". Information Sciences, 34 145-155. (1984)

[4] BUCKLES B.P., PETRY F.E., SACHAR H.S. "A Domain Calculus for Fuzzy Relational Databases". Fuzzy Sets and Systems, 29. 327-340. (1989)

[5] CODD E.F. "A Relational Model of Data for Large Shared Data Banks". Commun. ACM, 13. (6) pp. 377-387 (1970)

[6] HAMON GUY. "Extension d'un langage d'interrogation de Base de Donnees en vue vue de l'utilisation de questions imprecises", Tesis Doctoral, (1986).

[7] LI D., LIU "A Fuzzy Prolog Database System". ed John Wiley & Sons. New York. (1990)

[8] MEDINA J M., VILA M.A., "Un Modelo de Bases de Datos Difusa Aplicado a Información Médica". 1er Congreso Español sobre Tecnologías y Lógica Fuzzy. Granada. (1991)

[9] MEDINA J. M., PONS O., VILA M.A., "GEFRED. Un Modelo Generalizado de Bases de Datos Relacionales Difusas". 2º Congreso Español sobre Tecnologías y Lógica Fuzzy. Madrid.(1991)

[10] MEDINA J. M., VILA M.A., CUBERO J.C., PONS O. "Towards the Implementation of a Generalized Fuzzy Relational Database Model". Sometido a Fuzzy Sets & Systems. (1993)

[11] MEDINA J. M., PONS O., VILA M.A. "GEFRED. A Generalized Model of Fuzzy Relational Data Bases". Information Sciences,76,1-2, pp 87-109. (1994)

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

[13] ORACLE RDBMS. "SQL Language Reference Manual vs. 6.0". (1990).

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

[15] PRADE H., TESTEMALE C. "Representation of Soft Constraints and Fuzzy Attribute Values by means of Possibility Distributions in Databases". Bezdek ed. Analysis of Fuzzy Information. Vol II CRC Press. (1987)

[16] RUNDENSTEINER E.A. HAWKES L. W., BANDLER W. "On Nearness Measures in Fuzzy Relational Data Models" International Journal of Approximate Reasoning, 3.*pp*267 – 298.(1989)

[17] SHENOI S., MELTON A. "Proximity Relations in the Fuzzy Relational Database Model" Fuzzy Sets and System. 31 285 296. (1989)

[18] SHENOI S., MELTON A. "An Extended Version of the Fuzzy Relational Database Model" Information Sciences, 52. 35-52. (1990)

[19] UMANO M., MIZUMOTO M. AND TANAKA "FSTDS System. A Fuzzy-Set Manipulation System", Information Sciences. Vol. 14, pp. 115-159 (1978).

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

[21] VILA M.A., CUBERO J.C., MEDINA J. M., PONS O. "A Logic Approach to Fuzzy Relational Databases", IPMU'92 (1992)

[22] ZADEH L.A "Similarity Relations and Fuzzy Orderings". Information Sciences. vol. 3 177 200. (1971)

[23] ZADEH L.A "Fuzzy Sets as a Basis for a Theory of Possibility" Fuzzy Sets and Systems. 1, 3-28. (1978)

[24] ZEMANKOVA M., KANDEL A "Fuzzy Relational Data Bases - A Key to Expert Systems". Verlag TUV Rheinland. (1984)

DETECCIÓN DE CONTORNOS ESCALÓN MEDIANTE ALGORITMOS DIFUSOS

E. Montseny**

**Univ. Rovira i Virgili, Dep. Enginyeria Informàtica
Crta. Salou s/n 43006 Tarragona

P. Sobrevilla* y C. Garcia-Barroso**

*Univ. Politècnica de Catalunya, Dep. Matemàtica Aplicada II
Pau Gargallo, 5 08028 Barcelona

Resumen

La mayoría de los algoritmos de extracción (detección y localización) de contronos en una imagen se basan en el análisis del módulo del vector gradiente de la función iluminación de la escena.

El algoritmo de detección y localización de contornos de los objetos de una imagen, que se presenta en este trabajo está basado en el análisis simbólico-difuso del módulo y del argumento del vector gradiente, lo cual permite evitar la contradicción definida y demostrada por Marr-Hildreth [1] y Canny [2].

Las características más importantes del algoritmo són: a) Definición de medidas difusas relacionadas con los argumentos de los vectores gradiente, que permiten obtener el grado de pertenencia de cada pixel a un conjunto difuso que denominaremos lugar de paso del contorno (LPC). b) Definición de medidas difusas dadas en función de los valores de los módulos de los vectores gradiente, que permiten obtener el grado de pertenencia de cada pixel a un conjunto difuso que denominaremos punto del contorno (PC). c) Los módulos de detección y localización són robustos a la presencia de ruido en la imagen. d) El algoritmo permite realizar la extracción del contorno localizándolo con gran precisión.

1 Introducción

Los primeros trabajos de extracción de contornos como problema de visión por computador aparecieron en la década de los 60 como un intento de reducir el volumen de datos de una imagen conservando la información más relevante de la misma. En este sentido se debe tener en cuenta que, segun Rosenfeld [3], el sistema de visión de las personas, se basa en la detección de los puntos, líneas y en general, contornos de la escena, para así obtener una descripción que permita realizar una interpretación de la misma.

Los algoritmos de extracción de contornos se pueden agrupar en cuatro grandes líneas: 1) Análisis del módulo del vector gradiente [1], [2]. 2) Análisis de la respuesta de la correlación de la imagen con unos patrones (Template matching) [4], [5], [3]. 3) Detección del paso por cero de la segunda derivada de la función iluminación, en la dirección indicada por el vector gradiente de dicha función. 4) Análisis de la aproximación polinómica de la función iluminación [6].

Ultimamente han aparecido algoritmos basados en nuevas técnicas, entre las que se pueden destacar: a) Morfología matemática [7], b) Conjuntos difusos [8] y c) Redes neuronales [9].

Actualmente, la mayoría de investigadores que precisan del contorno de la imagen,

utilizan el algoritmo de Canny [2] para su extracción. Este algoritmo, incluido en la primera línea, se fundamenta en tres criterios: buena detecció, buena localización y una sola respuesta en la direccion transversal al contorno. Con estos criterios Canny obtiene los operadores óptimos para la aproximación del módulo del vector gradiente, y al mismo tiempo demuestra que los dos primeros criterios son contrapuestos, ya que para detectar un contorno hay que aproximar el módulo del vector gradiente con ventanas de convolución grandes, y para localizarlo con precisión hay que aproximarlo mediante ventanas pequeñas. Larré [10] demuestra que se puede evitar esta contradicción si para la detección del contorno se realiza un análisis del argumento del vector gradiente y para la localización se analiza el módulo.

2 Solución propuesta

En los distintos procesos para realizar la extracción de un contorno existe una vaguedad implícita, por lo que, para hacer que el sistema sea mas robusto, es necesario incorporar mecanismos que permitan representar esta vaguedad sin pérdida de información. En muchos casos el origen de la vaguedad no es únicamente aleatoria (ruido), sino que depende de otros factores que se pueden interpretar de manera más adecuada sobre la base de la teoría de conjuntos difusos. Entre estos procesos se pueden destacar: a) La perdida de resolución asociada tanto al proceso de digitalización de coordenadas espaciales, como a los niveles de intensidad de luz de la imagen. b) La vaguedad inherente a definiciones y conceptos de visión por computador, como són aristas y vértices de un contorno.

Estas razones han llevado a muchos investigadores a sustituir las técnicas y algoritmos deterministas, por nuevos métodos de naturaleza cualitativa, que hacen al sistema más robusto [11].

El algoritmo aquí propuestose divide en cuatro módulos: 1) Obtención de las componentes de los vectores gradiente de la función iluminación de la imagen. 2) Obtención del conjunto difuso LPC. 3) Obtención del conjunto difuso PC. 4) Extracción del contorno.

2.1 Obtención del vector gradiente

En [10] se introduce la obtención de un mapa denso de vectores gradiente, lo cual se realiza aproximando las componentes de dichos vectores mediante ventanas de convolución 3x3 y 4x4. El análisis del módulo del vector gradiente así obtenido permite localizar el contorno con gran precisión.

Teniendo en cuenta que cada valor muestreado de la función iluminación en el mundo continuo que representa la escena, tiene una probabilidad uniforme de pertenecer a cualquier punto del área que ocupa un pixel [12], se asigna al centro de cada pixel el valor muestreado (nivel de gris de cada pixel) a fin de minimizar el error.

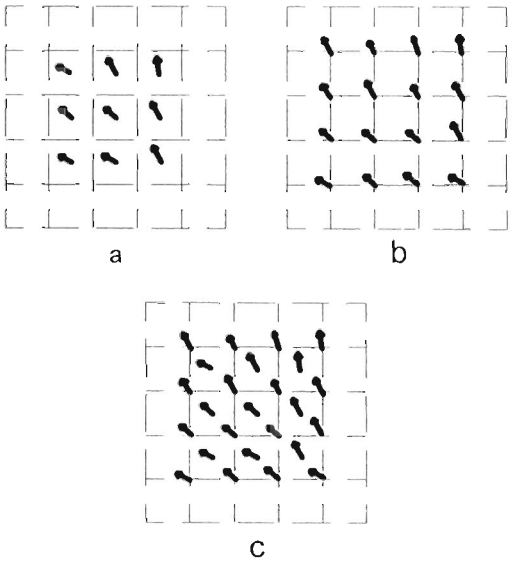


figura 1

Las ventanas de convolución 3x3 proporcionan un mapa de vectores gradiente, en el que cada vector está asignado al centro de un pixel (figura 1a). Del mismo modo, las ventanas de convolución 4x4 proporcionan un mapa de vectores gradiente,en el que cada vector se asasigna al punto de intersección definido por cuatro pixels (figura 1b). La representación en un solo plano de todos los vectores gradiente da lugar a un mapa denso de dichos vectores (figura 1c).

2.2 Obtención del conjunto difuso LPC

El contorno 2D de un objeto representado en una imagen, está definido por una línea de puntos cuya dirección varía de forma continua a lo largo del mismo, excepto en los puntos que definen vertices. Por lo tanto, puntos consecutivos pertenecientes a un contorno, tendran asociados vectores gradiente cuyos argumentos variaran de forma continua a lo largo del mismo, excepto en los vértices.

Dado que se desconoce la forma que el contorno puede tener en un pixel, se han considerado los siguientes cuatro casos: a) *Contorno recto*, en el cual los argumentos de los vectores gradiente se mantiene constantes. b) *Contorno curvo*, en el cual los argumentos de los vectores gradiente varían de forma continua a lo largo del mismo. c) *Vértice*, en él los argumentos de los vectores gradiente de los pixels a un lado y otro del mismo presentan una discontinuidad. d) *No existencia de contorno*, cuando los argumentos de los vectores gradiente toman valores aleatorios entre 0 y 360.

Para determinar el grado de pertenencia de un pixel (i, j) al conjunto difuso LPC se calculan dos medidas difusas μ_1 y μ_2, las cuales se obtienen del análisis de un conjunto de vectores gradiente cuyo origen se encuentra en un entorno del pixel en estudio. Así, para cada uno de los pixels de la imagen, en primer lugar se tomará el argumento del vector gradiente asociado al centro de dicho pixel, y a continuación se determinará el conjunto de vectores gradiente situados en el interior de una elipse centrada en el pixel (i, j), y con el eje menor orientado en la dirección del vector gradiente (figura 2).

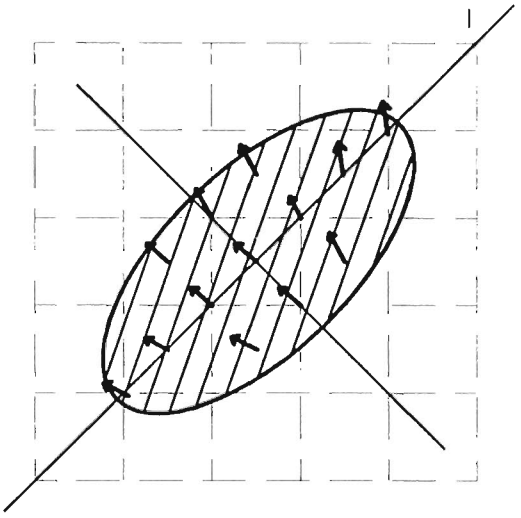


figura 2

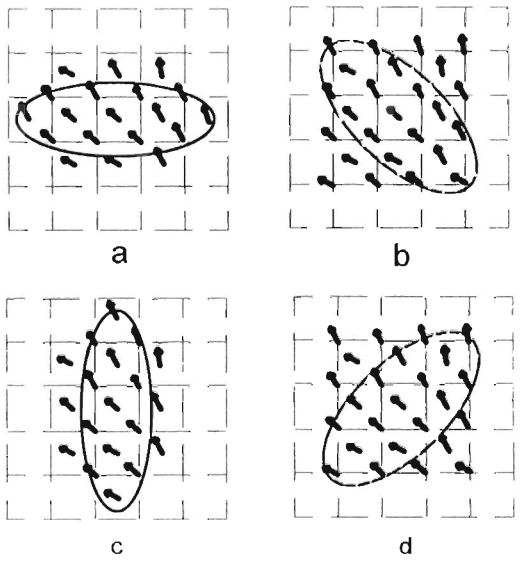


figura 3

A fin de simplificar los cálculos, se consideran únicamente 4 elipses, cuyas orientaciones del eje menor son: 0^o, 45^o, 90^o y 135^o (figura 3), de las cuales en cada caso se tomará la elipse cuya orientación sea la más cercana a la indicada por el argumento del vector gradiente del pixel en estudio.

Una vez determinado el conjunto de vectores gradiente, se representan en los planos coordenados de la figura 4, para lo cual en los ejes de abscisas se marca la distancia (positiva ó negativa) del origen del vector al eje menor de la elipse considerada (figura 2) y en los ejes de ordenadas se representa la orientación de dicho vector gradiente.

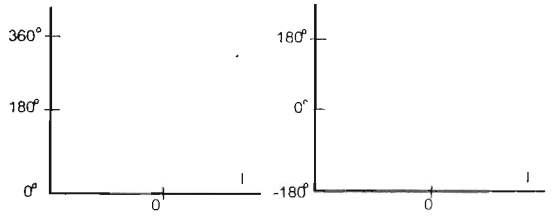


figura 4

En estas representaciones planas, los vectores gradiente correspondientes a un contorno recto estarán representados por unconjunto de puntos que definirán una recta horizontal. Los vectores gradiente correspondientes a un contorno curvo, que analizado localmente se puede tratar como un segmento rectilíneo, se representarán por un conjunto de puntos que definirán una recta horizontal. Un vértice dará lugar a la representación de dos conjuntos de puntos, cada uno de los cuales se podrá aproximar por una recta horizontal. La ausencia de contorno dará lugar a un conjunto de puntos aleatoriamente distribuidos en el plano.

Sobre estas representaciones se calculan las medidas μ_1 y μ_2 mediante las siguientes expresiones:

$$\alpha = \frac{\sum_{i=0}^{i=8} \arg(\bar{\nabla} I_i}{9}$$

$$\mu_1 = 1 - \frac{\sum_{i=0}^{i=8} \left| \arg(\bar{\nabla} I_i) - \alpha_1 \right|_{0,360}}{Error \quad Máximo}$$

$$\mu_2 = 1 - \frac{\sum_{i=0}^{i=8} \left| \arg(\bar{\nabla} I_i) - \alpha_2 \right|_{-180,180}}{Error \quad Máximo}$$

siendo $\bar{\nabla} I$ el vector gradiente en un punto de la imagen.

La medida difusa asignada a cada uno de los pixels (i, j), cuyo centro se encuentra dentro de la elipse es:

$$\mu^k_{(i,j)} = \max \mu_1, \mu_2$$

De esta forma, una vez cálculadas todas las medidas difusas, para cada pixel (i, j) de la imagen se abran obtenidos varias medidas difusas. Se definirá el grado de pertenencia de un pixel al conjunto difuso LPC:

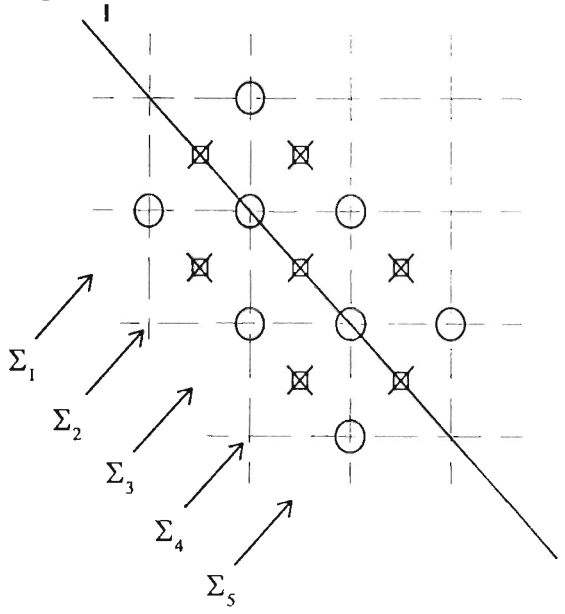
$$\mu^k_{(i,j)} = max \left\{ \mu^1_{(i,j)}, \ldots \mu^k_{(i,j)} \right.$$

2.3 Obtención del conjunto difuso PC

Se asume que un contorno se encuentra localizado en el pixel donde se produce la máxima variación de la función iluminación en la escena, en la dirección indicada por el gradiente de dicha función. Así, el grado de pertenencia de un pixel (i, j) al conjunto difuso PC, estará relacionado con dicho concepto de forma que un grado de pertenencia alto indicará que la variación máxima se produce en un punto cercano al centro de dicho pixel.

Para calcular el grado de pertenencia de cada uno de los pixels (i, j) de la imagen al conjunto difuso PC se define una medida difusa μ_1 , para ello se analiza un conjunto de vectores gradiente cuyo origen se encuentra en

un entorno del pixel en estudio. Así, en primer lugar se tomará el argumento del vector gradiente asociado al centro de dicho pixel, y a continuación se determinará el conjunto de vectores gradiente situados en el interior de un rectángulo centrado en el pixel (i, j), y orientado de manera que los lados mayores estén en la dirección del vector gradiente (figura 5).



⊠ **Módulo del vector gradiente obtenido con una ventana**

○ **Módulo del vector gradiente obtenido con una ventana**

figura 5

A fin de simplificar los cálculos, se han definido cuatro orientaciones del lado mayor del rectángulo: 0^o, 45^o, 90^o y 135^o (figura 6).

A continuación se calculan $\Sigma_1,\ldots,\Sigma_5$ como suma de los módulos de los vectores gradiente, tal como se indica en la figura 6. Una vez obtenidos estos valores se aproxima un valor proporcional al de la segunda derivada en la dirección indicada por el gradiente mediante:

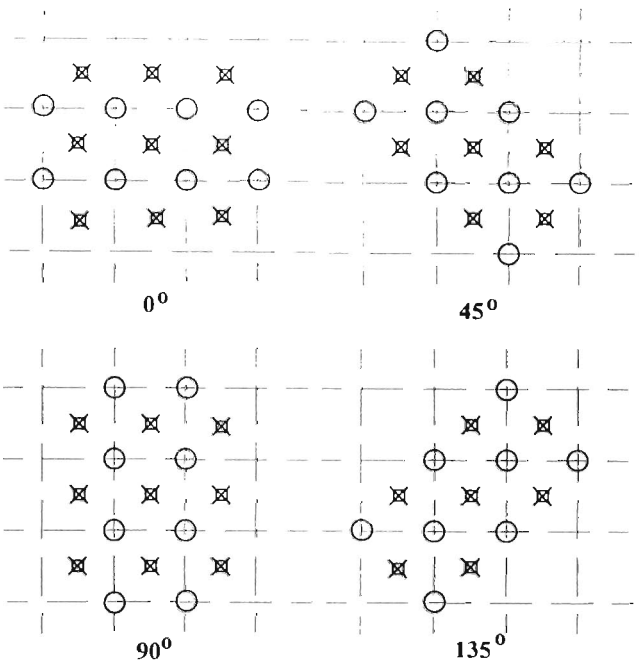


figura 6

$$r_1 = \Sigma_1 - \Sigma_3$$
$$r_2 = \Sigma_2 - \Sigma_4$$
$$r_3 = \Sigma_3 - \Sigma_5$$

y se representan estos valores en el plano coordenado de la figura 7.

Para la representación de r_1, r_2 y r_3 en este plano se considerará que estos valores corresponden a las segundas derivadas en los puntos p_1, p_2 y p_3 respectivamente (figura 6).

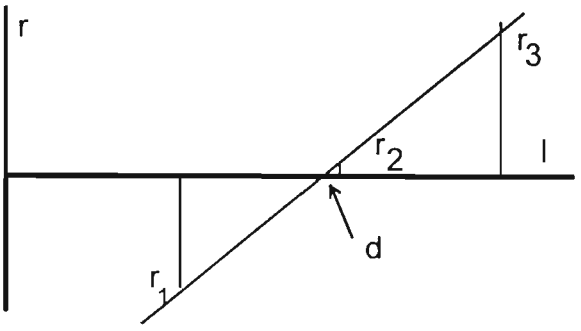


figura 7

El valor de μ_1 viene dado por:

$r_1 > 0$ y $r_3 > 0 \Rightarrow \mu_1 = 0$

$r_1 < 0$ y $r_3 < 0 \Rightarrow \mu_1 = 0$

$$\left.\begin{array}{l} r_1 > 0 \text{ y } r_3 < 0 \\ \text{o} \\ r_1 < 0 \text{ y } r_3 > 0 \end{array}\right\} \Rightarrow \mu_1 = \begin{cases} 1 - \frac{1}{2} d \text{ si } d < 2 \\ 0 \qquad \text{si } d \geq 2 \end{cases}$$

siendo d la distancia de r_2 al punto de corte del eje de abcisas y la recta definida por r_1, r_2 y r_3 por el método de los mínimos cuadrados.

2.4 Obtención del contorno

Para que un pixel (i, j) pertenezca al contorno se deben cuplir dos condiciones:

a) Que el pixel se encuentre situado en un entorno por donde pasa un contorno.

b) Que el pixel se encuentre siuado en un punto en el que en un entorno de radio < 1 se produce una variación máxima de la función iluminación en la dirección del gradiente.

De esta forma, para que un pixel (i, j) pertenezca al contorno de una imagen deberá pertenecer a los dos conjuntos difusos LPC y PC, definidos anteriormente, con un grado de pertenencia alto

3 Resultados

Este algoritmo se ha probado en distintos tipos de imagenes a fin de conocer su comportamiento, comparando los resultados con los que se obtienen aplicando los algoritmos de Canny, Marr-Hildreth y Larré. En la figura 8a se puede ver la imagen original utilizada en una de las pruebas, en 8b, c y d los contornos obtenidos de dicha imagen con los algoritmos de Canny, Marr-Hildreth y Larré respectivamente y en e los resultados obtenidos con el algoritmo que aquí se presenta. Se puede observar que en e se han conseguido extraer contornos muy poco contrastados, los cuales no aparecen en b y c.

a

b

c

Figura 8

d

e

figura 8 (cont.)

4 Conclusiones

El sistema que aquí se presenta permite realizar la detección y localización de los contornos de los objetos de la escena, incluso en imagenes afectadas de mucho ruido. Las principales características que se introducen són:

a) Para realizar la detección solo se utiliza información estructural (no se utilizan umbrales).

b) Se introducen medidas difusas para la detección y localización del contorno.

c) El algoritmo permite la localización del contorno con gran precisión incluso en imágenes con mucho ruido.

d) Se ha conseguido representar la vaguedad de la información de forma que un análisis más preciso de ésta permitirá la detección de algunos contornos que no son extraidos en estos momentos y que dificilmente lo podrán ser con algoritmos clásicos.

References

[1] D. C. Marr, E. Hildreth, "Theory of Edge Detection," Proceedings of the Royal Society of London, Series B, Vol. 207 (187-217) 1980.

[2] John F. Canny, "A Computational Approach to Edge Detection," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.PAMI-8 No.6(679-698) November 1986.

[3] Azriel Rosenfeld, M. Thurston, Y. Lee, "Edge and Curve Detection: Further Experiments," IEEE Transaction on Computers, Vol.C-21 No.7 (677-710) July 1972.

[4] G. S. Robinson, "Edge Detection by Compass Gradient Masks," Computer Graphics and Image Processing, 6 (492-501) 1977.

[5] W. Frei, Chung-Ching Chen, "Fast Boundary Detection: A Generalization and a New Algorithm," IEEE Transaction on Computers, Vol.C-26 No.10 (988-998) October 1977.

[6] Robert M. Haralick, "Digital Step Edges from Zero Crossing of Second Directional Derivatives." IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.PAMI-6 No.1 (58-68) January 1984.

[7] J. S. J. Lee, R. M. Haralick, L. G. Shapiro, "Morphologic Edge Detection," IFAC Digital Image Processing in Industrial Applications, Espoo, Findland. (7-14).

[81] Sankar K. Pal, Robert A. King, "Image Enhancement Using Smoothing with Fuzzy Sets," IEEE Trans. on System, Man and Cybernetics, Vol. SMC-11, No.7. (494-501) July 1981

[9] J. K. Paik, J. C. Brailean, A. B. Katsaggelos, "An Edge Detection Algorithm using Multi-Scale Adalines," Pattern Recognition, Vol.24 No.12 December 1992.

[10] A. Larré, E. Montseny, "A Step Edge Detector Algortihm Based on Symbolic Analysis", Proceedings 12th ICPR, Inter. Conf. On Pattern Recognition, Octubre 1994. Jerusalen (Israel).

[11] W. Thompson, J. Kearney, "Inexact Vision". Proc. Workshop on motion, Kiawah Island (SC), 1986.

[12] Behrooz Kamgar-Parsi, Behzad Kamgar-Parsi, "Evaluation of Quantization Error in Computer Vision," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.PAMI-11 No.9 (929-940) September 1989.

Fuzzy Similarities in the Field of Image Analysis

Rainer Albersmann, Rudolf Felix, Thomas Kretzberg

Fuzzy Demonstrations-Zentrum Dortmund
Joseph-von-Fraunhofer-Str. 20
44227 Dortmund, Germany
Tel.: +Ger 231 9700 921, Fax: +Ger 231 9700 929

Abstract

For image data represented as fuzzy sets a special notion of similarity previously applied in fuzzy decision making is used for image comparison. Applikation examples are given.

Key words: fuzzy similarities, image analysis

1 Introduction

In the field of pattern recognition the syntactic and the decision-making-oriented approach are distinguished [16]. The syntactic approach considers the pattern recognition as solving of the word problem for formal languages. The decision-making-oriented approach uses weighted sums of image characteristics, the so-called utility functions.

Especially the industrial application of the approaches is almost impossible because of the high computational complexity of the analysis algorithms in real world problems with real time constraints [13].

In [2] and [3] an approach is presented which analyzes the interdependences between characteristics of decision situations expressed as decision goals. The approach is based on different fuzzy relations describing relationships between decision goals. Based on the relationships the decision making problems are solved in polynomial computational time.

Since the decision making approach mentioned above is an altermative to decision making methods using utility functions [4], the idea to compare images using the same fuzzy relations, now applied to image characteristics instead of goals, seems to be promissing.

2 Basic Definitions

Let C be a non-empty, finite set of image characteristics $C := \{c_1,\ldots,c_n\}$, $i \in \{1,\ldots,n\}$. For every image O two fuzzy sets are defined as follows:

$$\forall_i: \mu_S^O(c_i) := \begin{cases} \delta, & \text{if the presence of characteristic } c_i \text{ is observed with intensity } \delta \\ 0 & \text{else} \end{cases}$$

(1)

$$\forall_i: \mu_S^O(c_i) := \begin{cases} \delta, & \text{if the non-presence of characteristic } c_i \text{ is observed with intensity } \delta \\ 0 & \text{else} \end{cases}$$

(2)

Every image O can be represented as two fuzzy sets S and D. The fuzzy set S describes the degree of presence of the characteristics. The fuzzy set D describes the degree of their non-presence.

Starting with such a representation of an image, two images can be compared based on fuzzy inclusions and non-inclusions between the sets S and D of the corresponding images.

Let $\mu_{\subset}: \mu^{O_1} \times \mu^{O_2} \rightarrow [0,1]$

(3)

be an inclusion relation for $O_1 \subset O_2$.

The non-inclusion $\mu_{\not\subset}$ for $O_1 \not\subset O_2$ is defined as

$$\mu_{\not\subset} := 1 - \mu_{\subset}.$$

(4)

Based on the introduced definitions the notion of similarity called "analogy" is defined as follows:

$$\mu_{analogy}(O_1, O_2) := \min\left(\mu_S^{O_1} \subset \mu_S^{O_2}, \mu_S^{O_1} \not\subset \mu_D^{O_2}, \mu_S^{O_2} \not\subset \mu_D^{O_1}, \mu_D^{O_1} \subset \mu_D^{O_2}\right)$$

where O_1 and O_2 are two images to be compared.

Note that the analogy is not a similarity relation in the sense of [5], [6], [7], [8], [9], since it is not symmetric.

3 Determination of S and D

In order to describe in which way the fuzzy sets S and D are determined let us consider grey level images as an example.

In order to perform the comparision, the image is partitioned into a grid of sectors. For each sector an average of grey levels is used as characteristic in the sense of the definition of μ_S and μ_D. A characteristic c_i is completely absent if the average grey level is white. Using a normalization of values between black and white the value of $\mu_S{}^O$ and $\mu_D{}^O$ for every sector can be calculated. The image O is then represented by the two fuzzy sets S and D.

In a similar way additional characteristics, for example the number of grey level changes, can be considered.

Note that the computational complexity of the comparison of images based on the notion of analogy is $O(n)$ where n is the number of characteristics (sectors) under consideration.

4 Qualitative and Structural Image Analysis

One of the aspects when determining the sets S and D is the granularity of the grid structure which of course implies the number of sectors and therefore the number of characteristics to be considered when representing the image.

The higher the number of sectors, the more details of the image are represented explicitely by S and D. Since the location of the sectors within the image can be expressed by indexing, structural information of the images is represented explicitely, too. Therefore, with an increasing number of sectors the analysis of the image becomes increasingly structural

The lower the number of sectors the less structural information is represented explicitely and the more qualitative becomes the analysis of the image. The term "qualitative" refers to image analysis which does not concern information about the location of the characteristics of the image. Qualitative analysis refers rather to the presence or absence of image characteristics, to their distribution inside of the image and to their intensity.

In the extreme case, each sector corresponds to one pixel in case of structural analysis. The extreme case of qualitative analysis is reached when the whole image is contained in only one sector.

5 Application Fields

Both the structural and the qualitative image analysis have already been applied in numerous analyzing tasks. The possibility to analyze contours of any kind opens a variety of applications whenever information refering to contours is relevant. For example analysis of the quality of punched work pieces (see Fig. 2), quality analysis of radio button casings (see Fig. 1), quality of robot controlled placement and positioning actions in the process automation [11] are examples of successful applications.

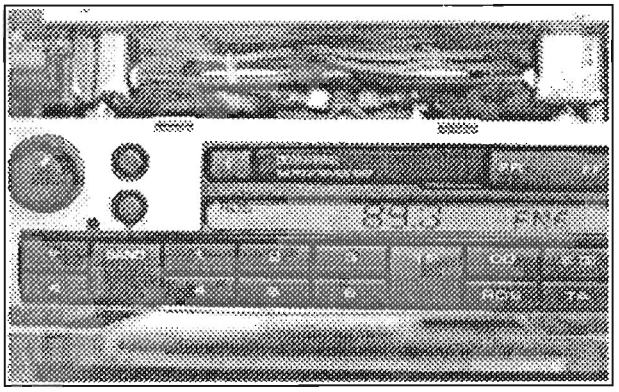


Fig.1

Qualitative image analysis has been applied in the field of surface classification [1], for example in the field of soot dispersion tests used in order to classify the quality (roughness) of gum workpieces used in the automotive industry (see Fig. 3).

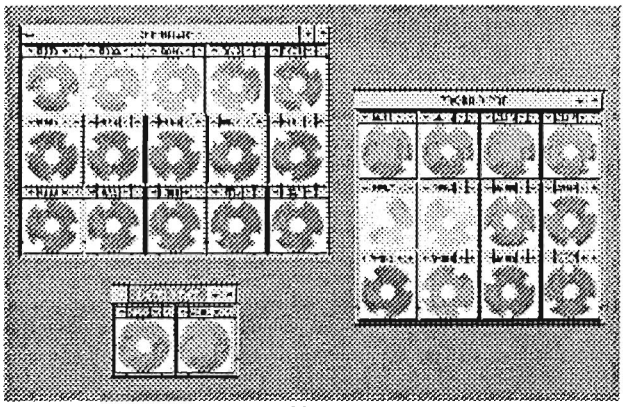


Fig.2

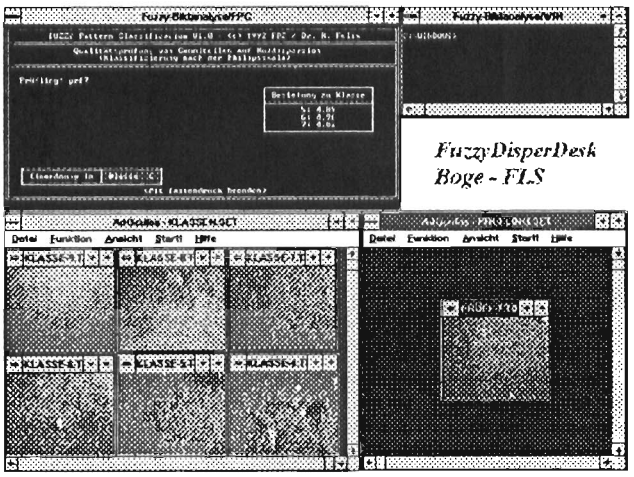


Fig. 3

6 Summary

A method for image analysis based on a notion of fuzzy similarity has been presented. Analogy is a fuzzy relation which is used to compare images represented as fuzzy sets. In contrast to other approaches, both the presence of image characteristics (for example grey level information) and their absence are used to represent an image. Based on the notion of analogy it is shown how structural and qualitative image analysis can be performed in linear computational time. Practical application examples show the efficency of the approach.

References

[1] Felix, R.; Reddig, S.: "Qualitative Pattern Analysis for Industrial Quality Assurance". *Proceedings of the 2nd IEEE Conference on Fuzzy Systems Vol I.* San Francisco, USA (March 28–April 1, 1993): 204–206.

[2] Felix, R.: "Entscheidungen bei qualitativen Zielen", Universität Dortmund, Fachbereich Informatik, Forschungsbericht Nr. 412, 1992

[3] Felix, R.: "Multiple Attribute Decision Making based on Fuzzy Relationships between Objectives", Proceedings of the 2nd International Conference on Fuzzy Logic and Neural Networks, Iizuka, Japan, 1992

[4] Felix, R.: "Goal-oriented Selection of Alternatives Based on Fuzzy Sets", IPMU Information Processing and Management of Uncertainty in Knowledge-based Systems, Paris, France, 1990

[5] Zadeh, L.A.: "Similarity Relations and Fuzzy Orderings", Yager, Ovchinnikov, Tong, Nguyen, FUZZY SETS AND APPLICATIONS, Selected Papers by L.A. Zadeh, Wiley-Interscience, 1987

[6] Ovchinnikov, S.: "Similarity relations, fuzzy partitions, and fuzzy orderings", Fuzzy Sets and Systems vol. 40, 1991, pp. 107-126, North Holland

[7] Ruspini, E.H.: "On the Semantics of Fuzzy Logic", International Journal of Approximate Reasoning, vol. 5, Number1, 1991, pp. 45 - 88

[8] Ruspini, E.H.: "ON TRUTH, UTILITY, and SIMILARITY", Fuzzy Engineering toward Human Friendly Systems, vol. 1, Nov. 1991, pp. 42 - 50

[9] Calvo, T.: "On fuzzy similarity relations", Fuzzy Sets and Systems vol. 47, 1992, pp. 121-123, North Holland

[10] Gonzales, R. C.;Thomason, M. G.: "Syntactic Pattern Recognition - An Introduction". *Addison-Wesley.* 1978.

[11] Felix, R.; Kretzberg, T.: "Qualitative und strukturelle Fuzzy Bildanalyse". In: Reusch, B (Editor): *Fuzzy Logic - Theorie und Praxis.* 3. Dortmunder Fuzzy Tage, Dortmund, 7.-9. Juni 1993. *Springer.* 1993.

[12] Felix, R., Kretzberg, T.; Wehner, M.: "Fuzzy Entscheidungsunterstützung und Bildanalyse", Tagungsband des Workshops *Fuzzy-Systeme '93 - Management unsicherer Informationen*, Braunschweig, 1993.

[13] to appear in: Kruse, R.: "Fuzzy Systems in Computer Science", Vieweg-Verlag 1994.

Aplicación de una metodología fuzzy a la evaluación de la calidad universitaria

Francisco Mesa Varela
Escuela Universitaria Politécnica
Universidad de Cádiz
C. Chile s/n
11001 Cádiz, España
e-mail: paco@sunca1.uca.es

Luis Miguel Marín Trechera
Escuela Universitaria Politécnica
Universidad de Cádiz
C. Chile s/n
11001 Cádiz, España
e-mail:quico@czv1.uca.es

Resumen

La Universidad en España experimenta en la actualidad un programa de evaluación que intenta conciliar los factores cuantitativos con los indicadores cualitativos. La dificultad de introducción de los aspectos cualitativos subyace en la imprecisión del concepto de calidad. La teoría fuzzy se aplica con éxito al dominio de los problemas imprecisos, y permite operar con variables lingüísticas. La evaluación de la calidad del Sistema Universitario se expresa en forma de relaciones lingüísticas, que facilita el estudio de la toma de decisión en la mejora de la calidad universitaria.

Palabras clave : Calidad. Variable Lingüística. Fuzzy. Toma de decisión.

1 Introducción

Se considera adecuado utilizar la evaluación conceptuada como "el proceso de identificar, obtener y proporcionar información útil y descriptiva acerca del valor y el mérito de las metas, la planificación y el impacto determinado, con el fin de servir de guía para la toma de decisiones, solucionar los problemas de responsabilidad y promover la comprensión de los fenómenos implicados" [6].

Dadas las características inherentes al proceso educativo el establecimiento de funciones de utilidad o la determinación y medición de las variables del proceso plantea algunos problemas. Normalmente se consideran únicamente los aspectos fácilmente mensurables y en la mayoría de los casos las variables son cuantitativas. Por la dificultad de introducir los aspectos cualitativos muchos de los estudios consideran únicamente los resultados cognitivos de la educación.

Además, debido a la insatisfacción de algún aspecto del desarrollo docente, se suelen realizar demostraciones sobre la importancia de alguna variable o factor característico del proceso.

Solmon [5] previene sobre los inconvenientes de los enfoques parciales. Señala que, en ocasiones, se ha definido la calidad atendiendo únicamente a los productos o resultados del proceso. Para originar estos resultados, las instituciones educativas emplean recursos con características no homogéneas. Si no son controlados, este criterio no aporta información acerca del papel de los factores en la determinación de los resultados.

De Weert [8] ha puesto de mani-fiesto algunas de las deficiencias:

- Es dificil encontrar indicadores significativos de la calidad de la docencia, por la complejidad de relaciones de los factores asociados a ella.

- El excesivo énfasis en el carácter cuantitativo hace que se generen indicadores sólo de lo que puede ser cuantificado.
- Al ser representaciones parciales de la calidad y no cubrir todas las dimensiones de la misma, pueden producir una imagen distorsionada.
- Si los indicadores no se orientan hacia la calidad, pueden incitar al simple incremento de la conducta medida, sin relación con la mejora.

El procedimiento clásico trata de establecer alguna hipótesis respecto al tipo de relación funcional que liga a las variables del proceso. Una vez se determinan las variables externas representativas de factores y productos, se intenta aplicar la lógica del análisis factorial, para concluir si hay variables ortogonales, tres o cuatro sólamente, en referencia a las cuales se pueden situar los elementos definitorios del proceso [3]. La incorporación de nuevos indicadores en forma de variable reinicia el proceso.

En la definición de un modelo conceptual de la calidad, los elementos definitorios se configuran en dimensiones por su similitud, más conceptual que técnica. Las dimensiones no son vectores ortogonales, independientes entre sí. Las dimensiones resumen las aportaciones al modelo de las variables externas e internas del sistema, lo que asegura la extensibilidad del modelo, y permiten experimentar tanto con indicadores de tipo cuantitativo como cualitativo. Las dimensiones de la calidad del sistema universitario que se consideran más adecuados para la comprensión del sistema son : *Docencia, Gestión e Investigación.*

2 Metodología de Evaluación del Sistema Universitario

Actualmente se experimenta, por acuerdo del Consejo de Universidades, un programa de evaluación del sistema universitario.

El programa pretende cubrir la globalidad del sistema universitario, identificando, a efectos metodológicos, tres áreas de evaluación.

Area de la enseñanza, cuyo núcleo de evaluación es la titulación;

Area de la investigación, centrada en el área de conocimiento:

Area de servicios, cuyo núcleos son los órganos intermedios de gestión administrativa.

La metodología de evaluación se desarrolla a partir del modelo de autoevaluación, con la incorporación de elementos de evaluación externa e indicadores de calidad.

El proceso de autoevaluación se basa en la opiniones de los responsables de los departamentos y de los centros universitarios, de los estudiantes y en el acceso a los datos universitarios que se recogen regularmente. El autoestudio interno se contrasta con expertos externos.

Las medidas de calidad que se utilizan en la evaluación, tanto las genéricas como las referidas a programas, son previamente consensuadas.

3 Representación Lingüística de la Información

El razonamiento de sentido común es cualitativo por naturaleza, en contraste con la mayor parte del razonamiento en las ciencias. La mayoría de los conceptos del razonamiento que conforma nuestro comportamiento son imprecisos.

La principal motivación para el concepto de *conjunto fuzzy* es extender la aplicabilidad de métodos matemáticos a dominios de problemas imprecisos que no pueden ser tratados con efectividad por los métodos convencionales basados en la teoría de probabilidades y lógica de predicados.

La representación gráfica o analítica de conjuntos definidos sobre un cierto universo de discurso suele ser bien comprendida por personas habituadas a trabajar con conjuntos, pero no ocurre así con los usuarios generales, más

acostumbrados a expresar información de modo lingüístico.

Aplicando el "Principio de Incompatibilidad" de Zadeh, parece adecuado el empleo de etiquetas como valoraciones lingüísticas para representar información imprecisa, modelo con el que los decisores pueden trabajar más cómodamente.

Zadeh [10] introduce el concepto de variable lingüística, como una variable referida a los elementos de un cierto universo y cuyos valores son palabras o sentencias en un lenguaje natural o artificial.

En consecuencia, el modelo más adecuado, para el manejo de la información lingüística es el de la variable linguística. La teoría de conjuntos difusos es la base del *enfoque lingüístico* de los problemas [2], bajo el cual las variables que intervienen se tratan como variables linguísticas, admitiendo que sus valores vienen dados linguísticamente.

4 Modelo Lingüístico de la Calidad

Un modelo lingüístico se expresa utilizando términos linguísticos en la estructura de la lógica fuzzy en vez de ecuaciones matemáticas o fórmulas de lógica convencional con símbolos lógicos. Existen diversos tipos de fuentes de información para un modelo lingüístico. De ellos, los más apropiados para la identificación del sistema son la observación basada en el conocimiento o la experiencia, y los datos numéricos [8].

En el caso estudiado, las reglas propuestas se estructuran de la forma siguiente:

Regla	Inves	Gestión	Docencia	Calidad
1	Neg	Neg	Neg	MNeg
2	Neg	Neg	Nor	Neg
3	Neg	Neg	Pos	PNeg
4	Neg	Nor	Neg	Neg
5	Neg	Nor	Nor	PNeg
6	Neg	Nor	Pos	PPos
7	Neg	Pos	Neg	PNeg
8	Neg	Pos	Nor	Pos
9	Neg	Pos	Pos	Pos
10	Nor	Neg	Neg	MNeg
11	Nor	Neg	Nor	Neg
12	Nor	Neg	Pos	PNeg
13	Nor	Nor	Neg	Neg
14	Nor	Nor	Nor	PNeg
15	Nor	Nor	Pos	PPos
16	Nor	Pos	Neg	PNeg
17	Nor	Pos	Nor	PPos
18	Nor	Pos	Pos	Pos
19	Pos	Neg	Neg	Neg
20	Pos	Neg	Nor	PNeg
21	Pos	Neg	Pos	PPos
22	Pos	Nor	Neg	PNeg
24	Pos	Nor	Pos	Pos
25	Pos	Pos	Neg	PPos
26	Pos	Pos	Nor	Pos
27	Pos	Pos	Pos	MPos

Neg = Negativa, MNeg = Muy Negativa, Nor = Normal , PPos = Poco Positiva, Pos = Positiva, MPos = Muy Positiva

Existen algunos problemas que aparecen en la determinación práctica de las relaciones lógicas entre dimensiones de la calidad y su evaluación [1]:

1) Documentación Linguística. Las relaciones lógicas pueden ser documentadas lingüísticamente después de ser analizadas en estudios específicos. En sustitución, deben tratarse por razonamiento de sentido común basado en la observación.

2) Extensión. El número de relaciones lógicas puede ser incrementado al modificar el conjunto de términos.

3) Evaluación. Las relaciones lógicas cualitativas se fundamentan en el juicio subjetivo. Las influencias culturales y sociales pueden hacer variar estas valoraciones.

Pueden minimizarse los riesgos reseñados si se considera la aproximación lingüística, tanto de las valoraciones de las variables lingüísticas como de las relaciones lógicas entre variables, en tres niveles:

Nivel 1: Aproximación con términos lingüísticos.
Nivel 2: Aproximación con términos lingüísticos y modificadores.
Nivel 3: Aproximación con términos lingüísticos, modificadores y conectivos.

5 Toma de Decisión con Valoraciones Lingüísticas

La Teoria de la Decisión es uno de los temas sujetos a mayor interés en las ciencias; su importancia justifica la existencia de muchos modelos de toma de decisiones. Uno de los postulados más aceptado es la posibilidad de representar las preferencias del decisor en algún tipo de escala. Ahora bien, es debatible que exista una única forma de estructuración de las preferencias subjetivas y aún asumiendo que ésta fuese única, ello no asegura la consecución de situaciones libres de incertidumbre o de modelos sin arbitrariedad en los conjuntos de acciones.

El concepto de evaluación adoptado, debe permitir la toma de decisión para la mejora de la calidad del sistema universitario. La necesidad de tomas de decisión sucesivas se justifica, si se toma como axioma la degradación contínua del sistema en funcionamiento autónomo.

La tendencia al enfoque parcial de los participantes en el proceso y el problema de la distribución de los recursos, derivado de cualquier toma de decisión, justifican la aplicación de la teoría de decisión en este modelo lingüístico.

Se considera la decisión sobre un problema para el cual tanto la ganancia o utilidad, como la información acerca del estado de la naturaleza están lingüísticamente valoradas.

A fin de generalizar la toma de decisión a cualquier modelo, se proponen acciones, expresadas como variables lingüísticas, de mejora en todas las dimensiones del modelo seleccionado. En este caso:

Mejora en la Investigación (MI)
Mejora en la Gestión (MG)
Mejora en la Docencia (MD).

Los esfuerzos de mejora en cada dimensión de calidad se establecen como

N = ± Positivo, PL = Positivo Leve
P = Positivo, MP = Muy Positivo

que posibilita un conjunto de 4^3 tomas de decisión.

Las experiencias previas permiten reducir el análisis al conjunto de decisiones de la tabla.

Decisión	MI	MG	MD
D_1	N	PL	PL
D_2	N	PL	P
D_3	N	PL	MP
D_4	N	P	PL
D_5	N	P	P
D_6	N	P	MP
D_7	PL	N	PL
D_8	PL	N	P
D_9	PL	N	MP
D_{10}	PL	PL	N
D_{11}	PL	PL	PL
D_{12}	PL	PL	P
D_{13}	PL	PL	MP
D_{14}	PL	P	N
D_{15}	PL	P	PL
D_{16}	PL	P	P
D_{17}	PL	P	MP

El conjunto de estados se obtiene considerando como estado cualquier combinación de las dimensiones, por lo que se asigna un estado a la combinación establecida por cada regla del modelo fuzzy.

Seleccionado el conjunto de decisiones lingüísticas, se obtiene la función de utilidad con los siguientes criterios de valoración de costes.

O El coste de actuación en Docencia es mayor que en Gestión y éste a su vez mayor que en Investigación. En esta valoración se incluye la dificultad y complejidad de los factores que influyen en cada dimensión de la calidad, el número de integrantes de la organización afectados por la decisión y el grado de tensión contrapuesta que soporta cada dimensión.

- Es necesario tener en cuenta, la degradación natural de los estados debido a estas tensiones, que confiere de carácter dinámico al proceso. Quiere decirse que se incorpora el concepto de tendencia al desorden y estados de menor coste por lo que, abandonada a sí misma, la evaluación de cada variable lingüística se deteriora en el transcurso del tiempo.

- Se precisa una mejora positiva leve en cualquier dimensión sólo para conseguir contrarrestar la evolución negativa natural del proceso.

- La ganancia se obtiene al considerar la calidad que consigue una decisión al modificar un estado.

En el caso de conocer la evaluación del estado actual de la calidad, por la determinación de la ocurrencia de una determinada combinación de las dimensiones, se puede establecer una relación de orden en las decisiones para cada estado; en función de la utilidad obtenida por las decisiones aplicadas a las dimensiones definitorias del estado en cuestión.

Sin embargo, puede ser de mayor interés la toma de decisión sin el conocimiento del estado actual, por ejemplo al inicio del sistema de evaluación con la metodología fuzzy. En este caso, la relación de orden se puede conseguir con la utilidad promedio calculada con todos los estados sometidas a una decisión.

Se mejora la prestación de la Toma de Decisión si se incluye la preferencia del decisor al establecer la relación de orden entre las decisiones. Se estiman las siguientes preferencias del decisor:

- Evitar, en cualquier caso, el empeoramiento de la calidad de algún estado.

- Intentar la obtención de la mayor relación entre el número de estados que mejoren su valoración de calidad, respecto al número de estados estables.

Si se tienen en cuenta estas preferencias, el conjunto de decisiones a tomar se reduce al conjunto $\{D_6 , D_{12} , D_{13} , D_{15} , D_{16} , D_{17}\}$.

Se fijan las relaciones binarias definidas en el conjunto de decisiones , que son representaciones de las preferencias subjetivas del decisor, con la estructura :

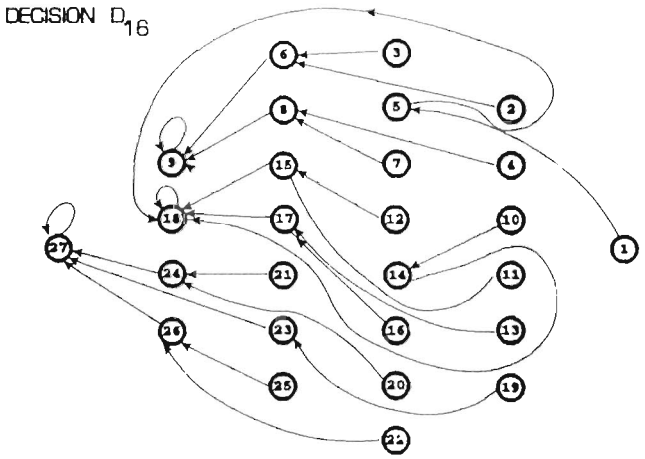
P preferencia cierta,
Q preferencia débil,
I equivalencia,
? imposibilidad de juicio.

	D_{13}				
D_{13}	I	D_{12}			
D_{12}	P	I	D_{15}		
D_{15}	P	P	I	D_{17}	
D_{17}	Q	Q	Q	I	D_6
D_6	P	Q	Q	P	I D_{16}
D_{16}	P	Q	Q	P	P I

Una estructura de preferencia es un preorden total si {P,I,?} verifican que P e I son transitivas y ? es vacío . Puede concluirse con los resultados de la tabla que la **estructura de preferencia** { P, I, ? } es **de preorden total** [4].

Lo que significa que la decisión óptima, con los criterios establecidos, es distribuir los recursos y esfuerzos institucionales en la mejora positiva en Gestión y Docencia, con mejora leve en Investigación. La figura presenta la evolución de

cada estado de calidad para esta decisión.



Esta relación de orden se puede condicionar a restricción por límites en los recursos disponibles, que modificaría las conclusiones establecidas.

6 Conclusiones

La metodología fuzzy es adecuada para la evaluación de la de calidad universitaria, por la complejidad del sistema y las numerosas variables indicadoras. La valoración lingüística de cada dimensión de la calidad, proporciona la determinación de estados como una clasificación fuzzy de las infinitas situaciones reales.

Se obtiene un sencillo modelo lingüístico de la calidad mediante un procedimiento iterativo, de extensión y evaluación de las relaciones lógicas entre las variables lingüísticas. Esto facilita la Toma de Decisión, con inclusión de las preferencias del decisor. Se propone extender la metodología lingüistica de la calidad a la determinación de las subdimensiones de las variables lingüísticas del modelo.

Referencias

[1] ALDASSING K.P , SHETHAUER W CADIAG-2 *Methods of Information in Medicine 24* (1984)

[2] DEPARTAMENTO DE C COMPUTACION - I.A. *Algunos aspectos del tratamiento de la información en Inteligencia Artificial* Universidad de Granada 1991

[3] DOCHY F , SEGERS M., WIJNEN W. Selecting Performance Indicators. *Peer Review and Performance Indicators*. Lemma B.V. Utrecht (1990)

[4] PUERTO ALBANDOZ, J. *Estructuración de las Preferencias*. Universidad de Sevilla 1994

[5] SOLMON, L.C. *The Quality of Education*. PSCHAROPOULUS. Economics of Education. Research and Studies. Pergamon Press. Oxford 1987

[6] STUFFLEBEAM, D.L. y SHINKFIELD, A.J. *Evaluación sistemática. Guía teórica y práctica*. Ed. Paidos/MEC. Barcelona 1987

[7] SUGENO M. , YASUKAWA T. Fuzzy-Logic-Based Approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems 1 Vol 1* (1993), 7-31

[8] WEERT E. DE. A macro-analysis of quality assesment in higher education. *Higher Education 19* (1990), 57-72

[9] WITOLD P. *Fuzzy Control and Fuzzy Systems*. John Wiley, 1989

[10] ZADEH L.A. *Linguistic Variables. Approximate Reasoning and Dispositions*. Medical Information 8 (1983)

ANÁLISIS DE TEXTURAS NATURALES MEDIANTE TÉCNICAS DIFUSAS

S. Alvarez*, D. Puig*

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

P. Sobrevilla**, E. Montseny*

**Dept. Matemàtica Aplicada II
Universitat Politècnica de Catalunya
Pau Gargallo 5
08028 Barcelona, España
e-mail: emontsen@etse.urv.es

Resumen

El algoritmo que aquí se presenta permite la segmentación de imágenes aéreas de entornos naturales, con el propósito de detectar las masas arbóreas que aparecen en ellas. Dicho algoritmo está basado en el análisis local de las texturas. Dado que las texturas que aparecen en los entornos naturales presentan unas características muy variables, especialmente si se las compara con las de los entornos artificiales, y que esta variabilidad supone una vaguedad en la determinación de las mismas, para realizar su análisis se han empleado técnicas difusas que permiten tratar dicha vaguedad.

Palabras clave: texturas, segmentación.

1 Introducción

La segmentación de imágenes aéreas es un problema que se empezó a tratar a principios de la década de los 70, cuando hubo que analizar las imágenes suministradas por el satélite LANDSAT [8], [11]. Esto motivó que, en esa década, se dedicaran grandes esfuerzos a la automatización del análisis de imágenes aéreas debido, en gran parte, a la gran cantidad de información que dicho satélite suministraba. Tras un período (década de los 80) de poca actividad investigadora en el tema, durante los últimos años ha vuelto a ser de gran interés para los investigadores, tanto en el análisis de texturas de entornos natures, como de entornos artificiales.

En el análisis de texturas todos los autores consideran distintas clases de texturas básicas, que se corresponden con los diversos elementos que aparecen en los entornos naturales.

Casi todos los trabajos realizados hasta ahora en el campo de las texturas, han estudiado por separado su componente estructural y su componente estadística. En el caso de las texturas naturales es difícil separar estas componentes, por lo que su estudio es más complejo.

2 Estudio realizado

Para la realización de nuestro trabajo hemos utilizado imágenes multinivel (256 niveles de gris). Dichas imágenes han sido tomadas desde satélite, con una resolución de 512x512 pixels, donde un pixel representa 2,5 m^2.

Se ha considerado un conjunto de imágenes donde se pueden encontrar los elementos siguientes:

Bosque.
Matorral.
Caminos.
Sembrados.
Campos de cultivo sin sembrar.
Arboles de cultivo.
Viña.

La variabilidad de las características que presentan estos elementos, nos ha llevado a sustituir los algoritmos y técnicas deterministas que

generalmente se utilizan para su análisis, por métodos de naturaleza cualitativa, los cuales han sido implementados con técnicas difusas.

2.1 Extracción de características

En principio todas las texturas naturales analizadas presentan una distribución normal de los niveles de gris. Sin embargo, la desviación y media de esta distribución varia según el tipo de textura de que se trate.

Por otra parte hemos observado que los bosques repoblados, así como los árboles, viñas y campos de cultivo presentan una estructura orientada según una cierta dirección. En cambio, los bosques y los campos sin cultivo no presentan ninguna direcionalidad en su estructura.

Otra diferencia clave entre las masas arboreas y las demás texturas naturales es que, en los bosques la ubicación de los niveles de gris presenta una distribucción aleatoria.

La última característica que nos ayudará a segmentar, es el rango de valores entre los que se mueven los niveles de gris dentro de los bosques.

Fruto de estas observaciones, hemos intentado encontrar unos parámetros que evaluen las siguientes características presentes en este tipo de texturas:

- direccionalidad,
- aleatoriedad,
- tonalidad.

Para ello hemos realizado:
- Análisis de los niveles de gris en la imagen.
- Análisis del módulo del gradiente.
- Análisis del argumento del gradiente.

Hay que comentar que la textura identificada como masa arbórea presenta una gran variabilidad dependiendo de la estación del año, del tipo de bosque de que se trate y las condiciones de iluminación en que hayan sido tomadas las imágenes a analizar.

En este primer estudio, hemos considerado bosques perennes para no añadir la dificultad del cambio de aspecto en función de la estación. En cambio hemos construido parámetros que miden las dos primeras caracteristicas y son robustos a las condiones de luz. Sin embargo la última característica es claramente dependiente de este factor.

2.2 Definición de operadores

a) Direccionalidad: Basándonos en los estudios realizados por Tamura[13], se analiza de forma local el histograma del argumento del vector gradiente de la función iluminación de los pixels en toda la imagen.

El módulo y el argumento del gradiente se expresan respectivamente como:

$$|\Delta G| = (|\Delta_H| + |\Delta_V|) / 2$$

$$\theta = \tan^{-1} (\Delta_V/\Delta_H) + \pi/2$$

donde Δ_H y Δ_V son operadores clásicos de extracción de las componentes horizontal y vertical.

Para estudiar el histograma del argumento del gradiente, se calcula la función:

$$H_D(\phi) = \sum_{\phi \in \omega} h(\phi)$$

donde ω es un entorno de ϕ, y $h(\phi)$ es el nº de pixels con argumento igual a ϕ.

La medida que refleja la direccionalidad es:

$$dir = \sum_{p=0}^{np-1} \sum_{\phi \in \omega} abs(\phi-\phi_p)\left((H_D(\phi_p)-H_D(\phi))/ \sum_{\phi=0}^{\pi} H_D(\phi)\right)^2$$

donde ϕ_p es el p-ésimo máximo de $H_D(\phi)$, y np es el nº de máximos.

b) Aleatoriedad: La evaluamos siguiendo tres criterios diferentes. Todos ellos operan de forma local, y para ello subdividimos la imagen en ventanas solapadas en las que aplicamos los siguientes operadores:

1. Definimos el parámetro dif2, donde p, a, y b son los valores de gris de la función iluminación de los pixels de la ventana a tratar.

$$dif2 = (p-a)^2+(p-b)^2,$$

p	a
b	

Y calculamos la suma de dif2 para todos los pixels de la ventana.

2. Dada una ventana, se calculan las diferencias entre el nivel de gris de la función iluminación del pixel central y el de todos sus vecinos. Los parámetros que miden el tipo de distribución son la media (m1) y la desviación (σ1) de estas diferencias.

3. Dada una ventana, calculamos el módulo del vector gradiente de la función iluminación de todos sus pixels, y definimos el parámetro dif:

dif= (p-a)+(p-b).

De la misma forma que en el operador (1) calculamos la suma de dif para todos los pixels de la ventana, y además calculamos su desviación (σ2).

c) Tonalidad: Dada una ventana, calculamos la media (m2) de los niveles de gris de la función iluminación de todos sus pixels.

2.3 Obtención de las medidas difusas

A partir de los parámetros analizados anteriormente, obtenemos un conjunto de reglas de clasificación. Cada regla determina el grado de pertenencia de cada uno de los pixels de la imágen a los conjuntos difusos: *masa arbórea* (bosque) y*no masa arbórea.* Las reglas obtenidas son las siguientes:

R1: La pertenencia de un pixel x al conjunto *masa arbórea* (representado por $\mu_{R1}(x)$), es mayor cuanto más cerca se encuentre la media m2(x) (esto es la media de los niveles de gris de su entorno) del intervalo (a1,b1). Esto es, μ_{R1} es una función convexa con

$$\mu_{R1}(x) = 1 \Leftrightarrow m2(x) \in (a1,b1).$$

R2: La pertenencia de un pixel x al conjunto *masa arbórea* (representado por $\mu_{R2}(x)$), es mayor cuanto más cerca se encuentre la característica suma de dif2(x) (calculada en los pixels de su entorno) del intervalo (a2,b2). Esto es, μ_{R2} es una función convexa con

$$\mu_{R2}(x) = 1 \Leftrightarrow \sum dif2(x) \in (a2,b2).$$

R3: La pertenencia de un pixel x al conjunto *masa arbórea* (representado por $\mu_{R3}(x)$), es mayor cuanto más cerca se encuentren los parámetros que miden la distribución de los niveles de gris, m1(x) (representado por $\mu_{R31}(x)$) del intervalo (a3,b3) y σ1(x)(representado por $\mu_{R32}(x)$) del intervalo (a4,b4). Esto es, μ_{R3} es una función convexa con

$$\mu_{R3}(x) = 1 \Leftrightarrow m1(x) \in (a3,b3) \text{ y } \sigma1(x) \in (a4,b4)$$

R4: Un pixel x pertenece al conjunto *masa arbórea* (representado por $\mu_{R4}(x)$), si el parámetro σ2(x) (calculada en los pixels de su entorno) es menor que b5. Esto es, μ_{R4} es una función convexa con

$$\mu_{R4}(x) = 1 \Leftrightarrow \sigma2\ (x) \in (0,b5).$$

R5: La pertenencia de un pixel x al conjunto *masa arbórea* (representado por $\mu_{R5}(x)$), es mayor cuanto más cerca se encuentre el parámetro suma de dif(x) (calculada en los pixels de su entorno) del intervalo (a6,b6). Esto es, μ_{R5} es una función convexa con

$$\mu_{R5}(x) = 1 \Leftrightarrow \sum dif\ (x) \in (a6,b6).$$

R6: Un pixel pertenece al conjunto *no masa arbórea* (representado por $\mu_{R6}(x)$), si la característica dir(x) (representado por $\mu_{R61}(x)$) es mayor que a7 y si el módulo del vector gradiente $|\Delta G|(x)$ es menor que b8. Esto es, μ_{R6} es una función convexa con

$$\mu_{R6}(x) = 1 \Leftrightarrow dir(x) \in (a7,+\infty) \text{ y } |\Delta G|(x) \in (0,b8)$$

Donde el conjunto correspondiente al concepto dir y al concepto $|\Delta G|$ es nítido.

Todas las funciones de pertenencia de todas las reglas, excepto la de la regla R6 son trapezoidales. Hemos escogido la función de pertenencia trapezoidal porque es la más fácil de calcular.

2.4 Obtención de los grados de pertenencia

El grado de pertenencia de un pixel al conjunto difuso *masa arbórea* se obtiene mediante una función de agregación definida sobre las medidas difusas obtenidas a partir de las reglas R1 - R5. Análogamente, el grado de pertenencia de un pixel

al conjunto difuso *no masa arbórea* se obtiene mediante la regla R6.
Diremos que un pixel pertenece a el conjunto difuso masa arbórea si cumple:

- En un entorno del pixel estructuralmente hay una distribución aleatoria, por lo tanto se tiene que cumplir que $\sum$dif2 en un entorno del pixel se encuentra dentro de un intervalo determinado ó los parámetros que definen la distribución m1 y σ1 entorno del pixel se encuentran dentro de un intervalo ó los parámetros que evaluan el módulo del gradiente en este intervalo: $\sum$dif (x) y σ2 se encuentran también dentro de ciertos intervalos.
- En un entorno del pixel no aparece direccionalidad, por lo tanto el parámetro dir se encuentra dentro de un intervalo.
- En un entorno del pixel, la tonalidad se sitúa dentro de unos valores determinados, es decir, la media de estos valores está entre unos valores calculados.

Por lo tanto la función de agregación utilizada es la siguiente:

$$\mu_R(x)=\text{mín}(\mu_{R1}(x),\mu_{R6}(x),\text{max}(\mu_{R2}(x),\mu_{R3}(x),\text{mín}(\mu_{R4}(x),\mu_{R5}(x)))).$$

3 Resultados obtenidos

En las figuras 1 y 2 mostramos las imágenes a analizar. La primera presenta zonas de bosque(zona oscura), además de caminos, zona urbanizada y sembrados. La segunda presenta zonas de cultivo, caminos,viñas y bosques.
Los resultados después de aplicar la función de agregación sobre estas imágenes se pueden ver en las figuras 3 y 4 repectivamente, donde se ha identificado en negro la zona de masa arbórea (bosque) y en blanco el resto de los elementos de la imagen.

fig.1:

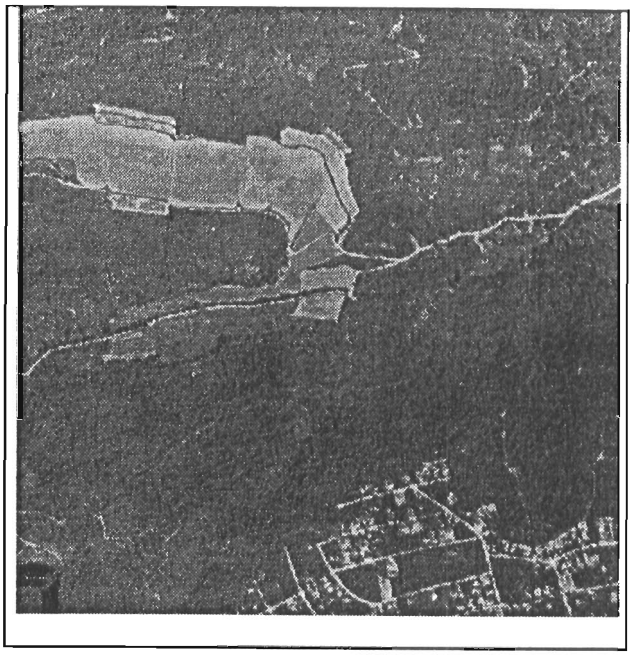


fig.2:

fig.3:

fig.4:

Bibliografía

[1] PAL, K. et al., "Image Description and Primitive Extraction Using Fuzzy Sets", IEEE Trans. SCM-13, January 1983.

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

[3] THOMPSON, W., KEARNEY, J., "Inexact Vision", Proc. Workshop on motion, Kiawah Island (SC), 1986.

[4] ZADEH, L. A., "Fuzzy Sets", Information and Control, n°8, pp. 338-353, Academic Press, New York,1965.

[5] SCHWEIZER, B., SKLAR, A., "Probabilistic Metric Spaces", Elsevier North-Holland, New York, 1983.

[6] ALSINA, C., FANK, M. J., SCHWEIZER, B., "Associative Functions on Intervals", 1987.

[7] ALSINA, C., TRILLAS, E., VALVERDE, L., "On Some Logical Connectives for Fuzzy Sets Theory", Journal of Mathematical Analysis and Aplications, Vol. 93, n° 1, pp. 15-26, 1983.

[8] WESZKA, J. S., DYER, C. R., ROSENFELD, A., "A Comparative Study of Texture Measures for Terrain Classification", IEEE Trans. on SMC-6, n° 4, April, 1976.

[9] TOMITA, F., "Computer Analysis of Visual Textures", Kluwer Academic Publishers, 1990.

[10] CONNERS, R. W., HARLOW, C. A., "A Theoretical Comparison of Texture Algorithms", IEEE Trans. PAMI-2, n° 3, May, 1980.

[11] HARALICK, R. M., SHANMUGAM, K., DINSTEIN, I., "Textural Features for Image Classification", IEEE Trans SMC-3, n° 6, November, 1973.

[12] HARALICK, R. M., "Statistical and Structural Approaches to texture", Proceedings of the IEEE vol. 67, n° 5, May, 1979.

[13] TAMURA, H., MORI, S. AND YAMAWAKI, T., "Textural Features Corresponding to Visual Perception". IEEE SMC-8, 1978, pp. 460-473.

Aprendizaje

UNA METODOLOGÍA DE REDES NEURONALES PARA RESOLVER ECUACIONES RELACIONALES*

A. Blanco, M. Delgado, I. Requena
Departamento de Ciencias de la Computación e I. A.
Universidad de Granada.
Facultad de Ciencias 18071 Granada España
e-mail: armando@robinson.ugr.es

Abstract

Presentamos un estudio sobre la identificación de sistemas de ecuaciones relacionales difusas, basado en la identificación de sistemas difusos mediante redes neuronales $max - min$, utilizando el método de la derivada suavizada y el método propuesto por Sayto-Mukaidono.

Keywords: Sistemas de ecuaciones relacionales difusas, redes neuronales $max - min$.

1 Introducción

Los sistemas difusos, estan siendo utilizados últimamente en una amplia variedad de problemas prácticos, como son el Control Automático, la Robótica y otras aplicaciones industriales. El estudio de los sistemas difusos, conlleva el análisis de las ecuaciones relacionales difusas, por lo que éstas han sido estudiadas por muchos autores. El problema de la resolución de ecuaciones difusas fué iniciado por Sanchez [1] y continuando posteriormente, muchos autores han realizado investigaciones para una gran variedad de problemas, usando diferentes métodos matemáticos.

Con el desarrollo de las investigaciones en el campo de las redes neuronales, algunos investigadores, han investigado el uso de esta herramienta, para resolver el problema de la identificación de los sistemas de ecuaciones relacionales difusa. En éste artículo vamos a profundizar en la identificación de los sistemas difusos mediante las redes neuronales difusas $max-min$, proponiendo tres métodos de aprendizaje de dichas redes.

El trabajo lo iniciamos con un breve resumen de los métodos de la derivada suavizada y del método propuesto por Sayto-Mukaidono [3] para la identificación, presentando a continuación tres métodos basados en los anteriores, para finalizar con los resultados experimentales realizados sobre algunas ecuaciones relacionales.

*Éste trabajo ha sido realizado bajo el proyecto PB 92- 0945 de DGICIT. MADRID

2 Método de identificación de Sayto-Mukaidono.

La ecuación relacional difusa que se considera es $X \circ R = Y$, donde $X \in [0,1]^{n\times m}$, $R \in [0,1]^m$, $Y \in [0,1]^n$, siendo $\circ$ la operación $max - min$, de forma más explícita

$$\begin{pmatrix} x_{11} & .. & x_{1j} & .. & x_{1m} \\ & .. & & .. & \\ x_{i1} & .. & x_{ij} & .. & x_{im} \\ & .. & & .. & \\ x_{n1} & .. & x_{ns} & .. & x_{nm} \end{pmatrix} \circ \begin{pmatrix} r_1 \\ .. \\ r_j \\ .. \\ r_m \end{pmatrix} = (\; y_1 .. y_i .. y_n \;)$$

$$i \in I = \{1, 2, \cdots, n\}, \quad j \in J = \{1, 2, \cdots, m\}$$

Obsérvese que ésto puede considerarse como un conjunto de ecuaciones yuxtapuestas de la forma

$$(\; x_{i1} \quad \cdots \quad x_{ij} \quad \cdots \quad x_{im} \;) \circ \begin{pmatrix} r_1 \\ \cdots \\ r_j \\ \cdots \\ r_m \end{pmatrix} = y_i$$

Para la resolución de este sistema en el conjunto de elementos de la matriz X, se consideran los siguientes subconjuntos:

$C_{mi} = \{j \in J / x_{ij} > y_i\}$ = posiciones de columnas de la fila i, donde aparecen elementos mayores que y_i,

$C_{ei} = \{j \in J / x_{ij} = y_i\}$ = posiciones de columnas de la fila i, donde aparecen elementos iguales a y_i,

$C_i = C_{mi} \cup C_{ei} = \{j \in J / x_{ij} \geq y_i\}$ = posiciones de columnas de la fila i, donde aparecen elementos mayores o iguales que y_i,

$C_{ui} = \bigcup_{s=i+1}^{n} C_{m_s}$ = posiciones de columnas de las filas $i+1, i+2, \cdots n$ donde aparecen elementos mayores o iguales que $y_{i+1}, y_{i+2}, \cdots y_n$ respectivamente,

$C_{k_i} = C_i - Cu_i$ = posiciones de columnas de la fila i donde aparecen elementos mayores o iguales que y_i menos posiciones de columnas de las filas $i+1, i+2, \cdots n$, donde aparecen elementos mayores o iguales que $y_{i+1}, y_{i+2}, \cdots y_n$ respectivamente,

$F_{mj} = \{i \in I / x_{ij} > y_i\}$ = posiciones de filas de la columna j, donde aparecen elementos mayores que y_i,

$F_{ej} = \{i \in I / x_{ij} = y_i\}$ = posiciones de filas de la columna j, donde aparecen elementos iguales a y_i,

$F_j = F_{mj} \cup F_{ej} = \{i \in I / x_{ij} \geq y_i\}$ = posiciones de filas de la columna j, donde aparecen elementos mayores o iguales que y_i,

$$\beta_j = \begin{cases} \min\limits_{j \in F_{mj}} y_i & \text{si} \quad F_{mj} \neq \emptyset \\ 1 & \text{si} \quad F_{mj} = \emptyset \end{cases}$$

$F_{k_j} = \{i \in F_j / y_i \leq \beta_j\}$.

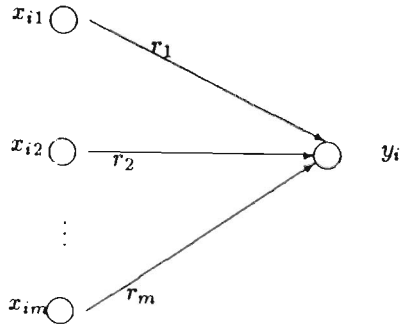
Sin pérdida de generalidad, se considera que los elementos de la matriz Y están ordenados, pues en caso contrario bastaría con reorganizar las filas de las matrices X e Y, de forma que $1 \geq y_1 > y_2 \cdots > y_n$

Notando por $\mathcal{R}$ el conjunto de todas las soluciones del sistema de ecuaciones relacionales difusas $X \circ R = Y$, se obtienen los teoremas

Teorema 1 $\mathcal{R} \neq \emptyset$ si y sólo si $C_{k_i} \neq \emptyset \ \forall i \in \{1, 2, \cdots, n\}$

Teorema 2 $\mathcal{R} \neq \emptyset$ si y sólo si $\bigcup_{j \in J} F_{k_j} = I$

Considerando una red neuronal de la forma:



A partir de éstos dos teoremas, se deduce un algoritmo de aprendizaje de tipo delta:
$\triangle w_{ij} = \mu(y_i - o_i) \star P$ donde μ es un factor de aprendizaje, siendo el valor más adecuado en la mayoría de los casos 1, y el valor de P se obtiene como

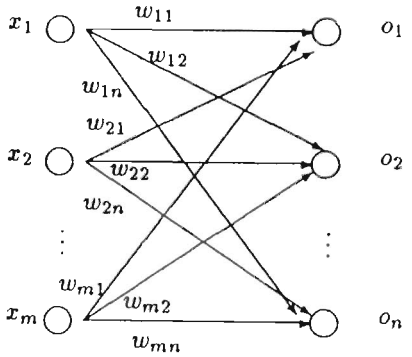
$$\begin{cases} o_i > y_i \begin{cases} x_{ij} > y_j \longrightarrow \text{P} = 1 \\ x_{ij} \leq y_i \longrightarrow \text{P} = 0 \end{cases} \\ o_i < y_i \begin{cases} x_{ij} \geq y_i \longrightarrow \text{P} = 1 \\ x_{ij} < y_i \longrightarrow \text{P} = 0 \end{cases} \\ o_i = y_i \longrightarrow \text{P} = 0 \end{cases}$$

3 Identificación de un sistema relacional difuso max-min mediante la derivada suavizada.

A continuación exponemos el método de la derivada suavizada [4], en el que nos basaremos fundamentalmente para el desarrollo de nuevos métodos de identificación de sistemas relacionales difusos.

El problema que se plantea, es el de resolver ecuaciones relacionales difusas mediante redes neuronales difusas $max - min$. Asumimos que existe un sistema descrito por una ecuación relacional difusa $X \circ R = Y$, $X \in [0,1]^m$, $Y \in [0,1]^n$, $R \in [0,1]^{n \times n}$, donde la operación $\circ$ es la $max - min$, del que sólo disponemos de una serie de ejemplos $[X^i, Y^i : i = 1 \cdots p]$ para resolver la ecuación relacional.

Para la identificación de la ecuación relacional difusa $X \circ R = Y$, usamos una red neuronal difusa $max - min$, que es entrenada con los ejemplos de que disponemos, dicha red tiene la topología que se muestra en la figura, con entradas y salidas $(x_1, \cdots, x_i, \cdots, x_m)$ y $(o_1, \cdots, o_j, \cdots, o_n)$ respectivamente, donde $o_j = max\ [min\ (x_i,\ w_{ij})]$, siendo $i = 1 \cdots m$, $j = 1 \cdots n$ y w_{ij} los elementos de la matriz de pesos.



Debido a la forma de obtener las salidas en la red neuronal, no es necesario incluir función de activación, ya que la propia función min es una función de umbral.

Al aplicar el método del gradiente descendente para desarrollar el método de aprendizaje de la red neuronal, se observa que la red sufre una parálisis, por

lo que en [4] demostramos que el problema aparece al hallar la derivada de las fuciones *min* y *max*, por tener éstas derivadas un comportamiento "crisp".

Éste problema lo resolvemos, considerando funciones con un comportamiento similar a las derivadas de *min* y *max*, pero con un comportamiento fuzzy, así para la derivada de *min*, la función considerada es el grado de inclusión de w en x.

El proceso de aprendizaje se realiza modificando los pesos de acuerdo con la siguiente regla de tipo delta: $\triangle w_{ij} = \mu \delta_j P$, donde $\delta_j = (y_j - o_j)$, y_j son las salidas esperadas, o_j son las salidas obtenidas y P viene dado por

$$x_s < w_{sj} \begin{cases} x_s \geq \underset{i \neq s}{\text{Max}} \ (Min(x_i, w_{ij})) & \rightarrow \quad P = x_s \\ x_s < \underset{i \neq s}{\text{Max}} \ (Min(x_i, w_{ij})) & \rightarrow \quad P = x_s * x_s \end{cases}$$

$$x_s \geq w_{sj} \begin{cases} w_{sj} \geq \underset{i \neq s}{\text{Max}} \ (Min(x_i, w_{ij})) & \rightarrow \quad P = 1 \\ w_{sj} < \underset{i \neq s}{\text{Max}} \ (Min(x_i, w_{ij})) & \rightarrow \quad P = w_{sj} \end{cases}$$

A continuación presentamos tres métodos de identificación de sistemas de ecuaciones relacionales difusas $X \circ R = Y$, modificando y combinando los dos anteriores.

4 Método I

La idea es partir del aprendizaje de la red neuronal mediante el método de la derivada suavizada, para obtener una nueva clasificación más fina con respecto a los pesos de la matriz W, con el objeto de mejorar la eficiencia del proceso de aprendizaje.

Para ello realizaremos una clasificación, distinguiendo los pesos correspondientes a neuronas candidatas a ser 'ganadoras' (pesos a modificar), de los correspondientes a neuronas 'no ganadoras' (pesos a mantener).

En el conjunto de elementos de la matriz X, vamos a considerar una clasificación a partir del vector y_i proporcionado por los conjuntos de entrenamiento, y de las salidas o_i obtenidas a través del sistema difuso, que podemos expresarlas como:

$$y_i = max\{\underbrace{[\underset{j \in C_i}{max} \ min(x_{ij}, wj)]}_{1}, \underbrace{[\underset{j \notin C_i}{max} \ min(x_{ij}, wj)]}_{2}\}$$

- $o_i > y_i$. Al estar en (2) los $x_{ij} < y_i$, la parte que produce o_i proviene de (1) donde los $x_{ij} \geq y_i$, por consiguiente el ajuste de los pesos para conseguir que $y_i = o_i$, se realizará sólo cuando $x_{ij} \geq y_i$

- $o_i < y_i$. La salida o_i, nunca puede provenir de (2), ya que al ser $x_{ij} < y_i$, siempre será $min(x_{ij}, w_j) < y_i$, por lo que la modificación de los pesos para que el elemento o_i coincida con y_i, tendrá que provenir de (1), es decir cuando $x_{ij} \geq y_i$

- $o_i = y_i$. En éste caso no será necesario modificar los pesos.

Finalmente, y como consecuencia de éstas consideraciones, obtenemos la siguiente clasificación:

$$\begin{cases} o_i > y_i \begin{cases} x_{ij} \geq y_j \longrightarrow \text{Modificaremos } w_j \\ x_{ij} < y_i \longrightarrow w_j \text{ no será modificado} \end{cases} \\ o_i < y_i \begin{cases} x_{ij} \geq y_i \longrightarrow \text{Modificaremos } w_j \\ x_{ij} < y_i \longrightarrow w_j \text{ no será modificado} \end{cases} \\ o_i = y_i \longrightarrow \text{El peso } w_j \text{ no será modificado} \end{cases}$$

Introduciendo esta clasificación en el método original de la derivada suavizada, obtenemos las siguientes asignaciones para P (ver clasificación 1)

Los resultados obtenidos en distintos ensayos mediante éste método, se presentan en la tabla 1.

5 Método II

La idea es combinar la clasificación del método de la derivada suavizada, con la propuesta por Sayto-Mukaidono.

Recordemos que las salidas obtenidas en la red neuronal difusa $max - min$ serán:

$$o_i = max\{\underbrace{[\underset{j \in C_{m_i}}{max} \ min(x_{ij}, wj)]}_{1}, \underbrace{[\underset{j \in C_{s_i}}{max} \ min(x_{ij}, wj)]}_{2},$$

$$\underbrace{[\underset{j \notin C_i}{max} \ min(x_{ij}, wj)]}_{3}\}$$

Comparando las salidas obtenidas, o_i, con las salidas esperadas, y_i, obtenemos los siguientes casos:

- $o_i > y_i$. Al estar en (2) los $x_{ij} = y_i$ y en (3) los $x_{ij} < y_i$, la parte que produce o_i proviene de (1) donde los $x_{ij} > y_i$, por consiguiente el ajuste de los pesos se realizará sólo cuando $x_{ij} > y_i$,

- $o_i < y_i$. La salida o_i, nunca puede provenir de (3), ya que al ser $x_{ij} < y_i$, siempre será $min(x_{ij}, w_j) < y_i$, por lo que la modificación de los pesos tendrá que provenir de (1) ó (2), es decir cuando $x_{ij} \geq y_i$,

- $o_i = y_i$. En éste caso no será necesario modificar los pesos.

Así pues

$$\begin{cases} o_i > y_i \begin{cases} x_{ij} > y_j & \longrightarrow \text{ Modificaremos } w_j \\ x_{ij} \leq y_i & \longrightarrow \ w_j \text{ no será modificado} \end{cases} \\ o_i < y_i \begin{cases} x_{ij} \geq y_i & \longrightarrow \text{ Modificaremos } w_j \\ x_{ij} < y_i & \longrightarrow \ w_j \text{ no será modificado} \end{cases} \\ o_i = y_i \longrightarrow \text{El peso } w_j \text{ no será modificado} \end{cases}$$

Combinando ésta clasificación con la del método de la derivada suavizada, obtenemos para P (ver clasificación 2)

Los resultados obtenidos mediante éste método para el aprendizaje de la red neuronal se presentan en la tabla 1.

6 Método III

Consideremos la ecuación relacional difusa $max - min$ $X \circ R = Y$, donde $X \in [0,1]^{n \times m}$, $R \in [0,1]^m$, $Y \in [0,1]^n$.

Los algoritmos de aprendizaje propuestos anteriormente modifican la matriz de pesos, mediante una regla proporcional a $(y_i - o_i)$. Ahora vamos a investigar la posibilidad de asignar a los pesos directamente el valor apropiado para que la salida esperada y_i, coincida con la salida obtenida o_i, de la siguiente forma:

$$\begin{cases} y_i < o_i \begin{cases} x_{ij} > y_i & \longrightarrow \quad w_j = y_i \\ x_{ij} \leq y_i & \longrightarrow \quad w_j = w_j \end{cases} \\ y_i > o_i \begin{cases} x_{ij} \geq y_i & \longrightarrow \quad w_j = y_i \\ x_{ij} < y_i & \longrightarrow \quad w_j = w_j \end{cases} \\ y_i = o_i \longrightarrow w_j = w_j \end{cases}$$

Obviamente, si los elementos y_i estuviesen ordenados, como hemos supuesto, con éste algoritmo, los pesos serían modificados una sóla vez, pero si no ocurre así, el número de modificaciones, dependerá del orden de presentación de los modelos de aprendizaje, pero siempre en número finito.

Obsérvese que esta forma de resolución es parecida a la de Sánchez [1] cuyo funcionamiento puede sintetizarse en

$$\begin{cases} X > Y \longrightarrow W = Y \\ X \leq Y \longrightarrow W = 1 \end{cases}$$

Ahora bien, nuestro método es superior porque no presenta problemas de convergencia en función de los valores iniciales de los pesos.

Con los pesos inicializados aleatoriamente, la red aprende, y aún más, cuando se inicializan a 1 se obtiene la misma solución que por el método clásico de resolución de ecuaciones relacionales difusas.

Éste método, no sólo tienen la ventaja, de que el algoritmo de aprendizaje, es más rápido, sino que el error cometido en el aprendizaje mediante la red neuronal, es nulo por lo que la matriz de pesos, coincide exactamente con la matriz relacional difusa, pudiendo obtener de entre las posibles soluciones, la mínima de las cotas superiores, así como una las soluciones mínimas, sin más que inicializar la matriz de pesos, a unos ó a ceros respectivamente.

7 Resultados

Los ensayos relativos a la identificación de las ecuaciones relacionales difusas $max - min$ se realizan considerando varios sistemas difusos con unas condiciones particulares que se exponen a continuación.

Los elementos de las matrices relacionales $R1$, $R2$, $R3$, $R4$, $R5$ y $R6$ de las ecuaciones relacionales difusas, en los distintos casos son números aleatorios, pertenecientes a un rango específico.

$R1$ está formado por elementos $w_{ij} \in [0, \epsilon]$.

$R2$ está formado por elementos $w_{ij} \in [0, 0.5]$.

$R3$ está formado por elementos $w_{ij} \in [0.5 - \epsilon, 0.5 + \epsilon]$.

$R4$ está formado por elementos $w_{ij} \in [0, 1]$.

$R5$ está formado por elementos $w_{ij} \in [0.5, 1]$.

$R6$ está formado por elementos $w_{ij}/w_{ij} = 0 \ \forall i \neq j$.

$R*$ y $R**$ representan una matriz relacional utilizada por Pedrycz [2], pero con distintos modelos de entrenamiento, el primero de los conjuntos de entrenamiento, es muy particular, mientras que el segundo ha sido escogido aleatoriamente.

Una vez escogidas las matrices relacionales con las condiciones indicadas, los modelos de entrenamiento se determinan para cada ecuación relacional de la siguiente forma.

A partir de una serie de vectores aleatorios X, de la matriz escogida y de la ecuación relacional $Y = max(min(X, R))$, se obtienen los vectores Y correspondientes a X

Los entrenamientos de las redes neuronales para la identificación de una ecuación relacional determinada por una R se realizan en base a los modelos de entrenamiento (X, Y), donde los vectores X son las entradas a la primera capa de la red neuronal e Y las salidas esperadas, adoptando el criterio de que el error cometido por la red neuronal para cada X, se expresa como la suma de los cuadrados de las desviaciones de las salidas obtenidas por la red y las salidas esperadas Y.

Los resultados obtenidos en los entrenamientos de la red neuronal difusa, en los distintos casos se expresan en la siguiente tabla, donde se consideran el método propuesto por Pedrycz en [2], la derivada suavizada [4], el propuesto por Sayto-Mukaidono [3] y los tres métodos propuestos en éste trabajo.

Bibliografía

[1] Sánchez, E. (1976). Resolution of composite fuzzy relation equations. *Inform. and Control*, 30, 38-48.

[2] Ikoma, N., Pedrycz, W., Hirota, K. (1993) Estimation of fuzzy relational matrix by using probabilistic descent method. *Fuzzy Sets and Systems*, 57, 335-349.

[3] Saito, T., Mukaidono, M. (1991). A learning algorithm for max-min network and its application to solve fuzzy relation equations. *Proceedings of the 2nd International Conference on Fuzzy Logic and Neural Networks* , Iizuka'92 184-187.

[4] Blanco A., Delgado M., Requena I. (1994). Solving fuzzy relational equations by max-min neural network. accepted to *Third IEEE International Conference on Fuzzy Systems.*

Tabla 1

Comparación de los resultados								
	Iteraciones							
Matriz	Pedrycz	Derivada	Mukaidono	I	II	III	Error	Modelos
R*	13	1	1	1	1	1	10^{-20}	5
R**	22	10	2	12	3	1	10^{-20}	25
R1	17	7	3	5	4	2	10^{-20}	16
R2	20	10	2	10	2	1	10^{-20}	20
R3	30	7	1	7	1	1	10^{-20}	16
R4	15	6	2	4	2	1	10^{-20}	36
R5	10	14	2	6	3	1	10^{-20}	20
R6	25	6	2	2	2	1	10^{-20}	16

Clasificación 1

$$x_s < w_{sj} \begin{cases} x_s \geq \max\limits_{i \neq s} (min(x_i, w_{ij})) \begin{cases} y_s > o_s \begin{cases} x_s \geq y_s \longrightarrow P = x_s \\ x_s < y_s \longrightarrow P = 0 \end{cases} \\ y_s < o_s \begin{cases} x_s \geq y_s \longrightarrow P = x_s \\ x_s < y_s \longrightarrow P = 0 \end{cases} \\ y_s = o_s \longrightarrow P = 0 \end{cases} \\ x_s < \max\limits_{i \neq s} (min(x_i, w_{ij})) \begin{cases} y_s > o_s \begin{cases} x_s \geq y_s \longrightarrow P = x_s * x_s \\ x_s < y_s \longrightarrow P = 0 \end{cases} \\ y_s < o_s \begin{cases} x_s \geq y_s \longrightarrow P = x_s * x_s \\ x_s < y_s \longrightarrow P = 0 \end{cases} \\ y_s = o_s \longrightarrow P = 0 \end{cases} \end{cases}$$

$$x_s \geq w_{sj} \begin{cases} x_s \geq \max\limits_{i \neq s} (min(x_i, w_{ij})) \begin{cases} y_s > o_s \begin{cases} x_s \geq y_s \longrightarrow P = 1 \\ x_s < y_s \longrightarrow P = 0 \end{cases} \\ y_s < o_s \begin{cases} x_s \geq y_s \longrightarrow P = 1 \\ x_s < y_s \longrightarrow P = 0 \end{cases} \\ y_s = o_s \longrightarrow P = 0 \end{cases} \\ x_s < \max\limits_{i \neq s} (min(x_i, w_{ij})) \begin{cases} y_s > o_s \begin{cases} x_s \geq y_s \longrightarrow P = w_{sj} \\ x_s < y_s \longrightarrow P = 0 \end{cases} \\ y_s < o_s \begin{cases} x_s \geq y_s \longrightarrow P = w_{sj} \\ x_s < y_s \longrightarrow P = 0 \end{cases} \\ y_s = o_s \longrightarrow P = 0 \end{cases} \end{cases}$$

Clasificación 2

$$x_s < w_{sj} \begin{cases} x_s \geq \max\limits_{i \neq s} (min(x_i, w_{ij})) \begin{cases} y_s > o_s \begin{cases} x_s \geq y_s \longrightarrow P = x_s \\ x_s < y_s \longrightarrow P = 0 \end{cases} \\ y_s < o_s \begin{cases} x_s > y_s \longrightarrow P = x_s \\ x_s \leq y_s \longrightarrow P = 0 \end{cases} \\ y_s = o_s \longrightarrow P = 0 \end{cases} \\ x_s < \max\limits_{i \neq s} (min(x_i, w_{ij})) \begin{cases} y_s > o_s \begin{cases} x_s \geq y_s \longrightarrow P = x_s * x_s \\ x_s < y_s \longrightarrow P = 0 \end{cases} \\ y_s < o_s \begin{cases} x_s > y_s \longrightarrow P = x_s * x_s \\ x_s \leq y_s \longrightarrow P = 0 \end{cases} \\ y_s = o_s \longrightarrow P = 0 \end{cases} \end{cases}$$

$$x_s \geq w_{sj} \begin{cases} x_s \geq \max\limits_{i \neq s} (min(x_i, w_{ij})) \begin{cases} y_s > o_s \begin{cases} x_s \geq y_s \longrightarrow P = 1 \\ x_s < y_s \longrightarrow P = 0 \end{cases} \\ y_s < o_s \begin{cases} x_s > y_s \longrightarrow P = 1 \\ x_s \leq y_s \longrightarrow P = 0 \end{cases} \\ y_s = o_s \longrightarrow P = 0 \end{cases} \\ x_s < \max\limits_{i \neq s} (min(x_i, w_{ij})) \begin{cases} y_s > o_s \begin{cases} x_s \geq y_s \longrightarrow P = w_{sj} \\ x_s < y_s \longrightarrow P = 0 \end{cases} \\ y_s < o_s \begin{cases} x_s > y_s \longrightarrow P = w_{sj} \\ x_s \leq y_s \longrightarrow P = 0 \end{cases} \\ y_s = o_s \longrightarrow P = 0 \end{cases} \end{cases}$$

Modelado borroso de procesos mediante un algoritmo de simplificación

Luciano Sánchez Ramos
ETSII de Gijón
e-mail: luciano@etsiig.uniovi.es

Resumen

En este artículo se propone un mecanismo de construcción automática de una descripción basada en reglas de un proceso arbitrario. Todos los parámetros que definen esta descripción pueden ser estimados, incluyendo el número de reglas borrosas asociado al modelo natural del proceso mencionado.

Este método se basa en que la relación entre la entrada y la salida del sistema que se modela es conocida en un número de situaciones. Este conocimiento puede ser impreciso hasta cierto grado. No se emplea una caracterización estocástica de esa imprecisión.

Palabras Clave: Sistemas con capacidad de aprendizaje, Control Borroso, Entropía, Longitud de Descripción Mínima, Complejidad de Sistemas

1 Introducción

La descripción de sistemas complejos mediante conjuntos borrosos fue propuesta por Zadeh [9]. Este tipo de descripciones tiene como objeto el explotar la *tolerancia a la imprecisión* en el ajuste del modelo, de modo que la complejidad de este último sea mínima.

Los modelos borrosos se basan normalmente en la definición de varios modelos lineales diferentes. Cada uno de estos modelos tiene un ámbito limitado de aplicación. El papel primario de los conjuntos borrosos en estos modelos consiste en caracterizar el ámbito de cada uno de los modelos lineales y la tolerancia admisible en su definición. La relación entre uno de estos conjuntos y su modelo lineal asociado se denomina *regla borrosa*. Alternativamente, el modelo lineal –el *consecuente* de la regla mencionada– puede reemplazarse por otro conjunto borroso. Este último se asocia a una distribución de posibilidad de la salida del sistema, condicionada a que la entrada pertenezca al ámbito correspondiente.

El diseño de modelos borrosos es una tarea parcialmente heurística, ya que hasta el momento no existe ningún procedimiento *completamente* automático de diseño [3]. Este artículo aporta una nueva metodología, basada en la teoría de la información. Aplicando ésta se deducirán todos los parámetros que componen el modelo borroso.

2 Exposición del Método

El método propuesto descansa sobre una idea bien conocida: el número de variables necesario para definir el modelo de un sistema crece con la calidad del ajuste exigido. Recíprocamente, el modelo de un sistema puede ser simplificado si se aumenta el grado de imprecisión de su definición [2]. De este modo, si existen varios modelos cuyo ajuste iguale o supere la calidad requerida, el modelo más adecuado –o modelo *natural*– será el más simple de este grupo.

Supóngase que se dispone de un número de pares entrada-salida del proceso, a los que se llamará *ejemplos*. A partir de estos ejemplos se operará en tres fases:

1. Elaboración de un modelo inicial. Este modelo contendrá un número de reglas suficientemente elevado como para incluir la tota-

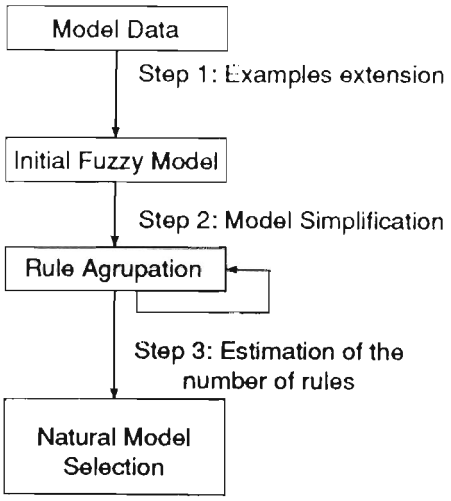


Figure 1: Esquema del método

lidad del conocimiento declarativo descrito en el conjunto de ejemplos. Como primera aproximación, se estima una regla por cada ejemplo.

2. Simplificación iterativa del modelo. Esta simplificación reducirá progresivamente, de la mejor forma posible, el número de grados de libertad del modelo inicial. Se termina cuando el sistema simplificado esté compuesto por una única regla. Con este paso se consiguen tantos modelos borrosos distintos como ejemplos. Estos modelos son los candidatos al modelo natural mencionado antes.

3. Se cuantifican las complejidades y los grados de ajuste de cada uno de estos modelos. A partir de estos valores se decide cuál de los candidatos es el modelo natural del sistema.

2.1 Teoría de la Longitud de Descripción Mínima Borrosa

Dado un modelo compuesto de N reglas y un conjunto S de ejemplos, la cantidad de información conjunta de sistema y ejemplos es

$$I(S,N) = I(N) + I(S|N)$$

donde $I(N)$ se denomina *información sintáctica* e $I(S|N)$ *información semántica*. $I(N)$ mide la complejidad estructural del modelo. $I(S|N)$ se define sobre la distribución de los errores de ajuste y es una de las medidas posibles de la similaridad entre el modelo y el proceso.

La suma de ambas cantidades se denomina *longitud de descripción* del modelo. Según la teoría de la longitud de descripción mínima, si existen varios modelos distintos que se ajustan al mismo conjunto de ejemplos S, el modelo natural es el que minimiza $I(S,N)$.

El tercer paso del método expuesto se resuelve definiendo una extensión borrosa de esta teoría [5], donde las expresiones de $I(N)$ e $I(S|N)$ son las siguientes:

$$I(N) = \log_2(N) \tag{1}$$

–es decir, en ausencia de otra información, se aproxima la incertidumbre estructural del modelo mediante la entropía de Hartley del número de parámetros de éste– y

$$I(S|N) = \sum_{i=1}^{N} n_{pi} I(r_i) \tag{2}$$

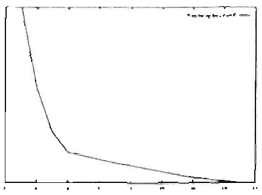
donde $I(r_i)$ es una extensión al caso continuo de la U-incertidumbre de Klir [2]:

$$I(r_i) = \int \log_2 |A_{i\rho}| d\rho \tag{3}$$

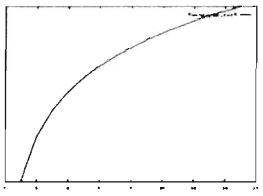
y $|A_{i\rho}|$ es la medida de Lebesgue de un ρ-corte de un conjunto borroso A_i asociado a la regla i del modelo.

Esta última expresión se aclara a continuación: En el proceso de extracción de reglas, los n ejemplos se clasificarán en N clases. Cada una de las reglas se estimará a partir de los ejemplos de una sola de estas clases. La regla descendiente de la clase i-ésima se construirá a partir de la información aportada por los n_{pi} puntos de la clase i; evidentemente $\sum_1^N n_{pi} = n$.

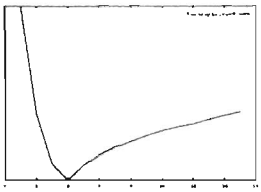
El desajuste medio entre un punto de la clase i y el modelo formado exclusivamente por la regla i se mide mediante $I(r_i)$. A partir de los valores de $I(r_i)$ se aproxima la información semántica de la distribución de ejemplos respecto al modelo. De este modo, la longitud total de descripción es



Complejidad Semántica



Complejidad Sintáctica



Longitud de Descripción

Figure 2: Estimación del número de reglas

$$\text{TDL} = I(N) + I(S|N) = log_2 N + \sum_{i=1}^{N} n_{pi} I(r_i) \tag{4}$$

En la figura 2 se grafican los valores de las complejidades semántica, sintáctica y longitud de descripción, frente al número de reglas. La complejidad sintáctica es necesariamente creciente con el número de reglas. La complejidad sintácica es, en principio, decreciente con este número. La suma de ambas tiene un mínimo en algún punto entre 1 y N; ese punto corresponde al modelo natural.

2.2 Tipos de reglas borrosas

Los sistemas basados en reglas borrosas emplean, como se indicó en la introducción, uno de los siguientes tipos de reglas:

1. Antecendente y consecuente son proposiciones borrosas. Estas reglas se nombrarán en lo sucesivo "reglas tipo I" o reglas *tipo Mamdani.*

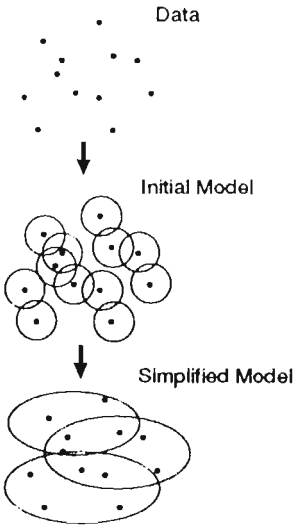


Figure 3: Extensión de ejemplo

2. El antecedente es una proposición borrosa y el consecuente una función afín de las variables del antecedente. Estas se denominarán "reglas tipo II" o reglas *tipo Sugeno.*

El método de extracción de reglas es válido en ambos casos, con pequeñas diferencias.

2.3 Extensión de los ejemplos

Si las dimensiones de los espacios de entrada y de salida son m y n, un ejemplo se podrá representar mediante un vector de dimensión $n = m + p$. Cada uno de los ejemplos, definidos de esta forma, se extiende con un pequeño conjunto borroso, según se muestra en la figura 3. De este grupo de conjuntos puede extraerse un sistema de control formado por tantas reglas como ejemplos, aplicando el procedimiento explicado en la sección 2.6.

La magnitud de esta extensión es un parámetro de diseño. Los valores altos tienden a suavizar la respuesta del modelo [5].

2.4 Operaciones de agrupación de reglas

En el paso segundo del método, se generan sucesivamente modelos simplificados del proceso. Estos modelos se obtienen a partir de colecciones de conjuntos borrosos por medio de una operación de *agrupación.*

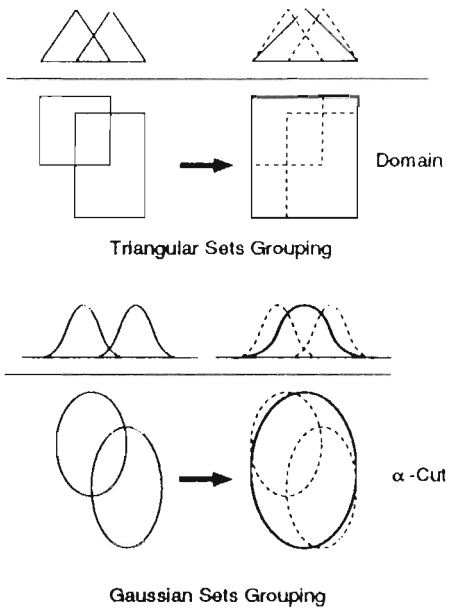


Figure 4: Operaciones de Agrupación

Las operaciones de agrupación extienden la descripción de dos conjuntos borrosos a un área que incluye los dominios de ambos. Se proponen las siguientes definiciones de agrupaciones para parametrizaciones triangular y gaussiana de conjuntos (ver figura 4):

Parametrización triangular

Dados dos conjuntos $A^1 \subset X_1^n$, $A^2 \subset X_2^n$ donde

$$A = A_{X_1 \times \ldots X_n}(x_1, \ldots x_n) = A_{X_1}(x_1) \wedge \ldots A_{X_n}(x_n) \quad (5)$$

$$A_{X_i} = A_{X_i}(c_i^A, \alpha_i^A, \beta_i^A)$$

donde α y β son las coordenadas de los extremos inferior y superior de los números borrosos A_X, el conjunto $A^1 \bullet A^2$ se define como

$$(A^1 \bullet A^2)_{X_1 \times \ldots \times X_n}(x_1, \ldots x_n) = (A^{1\bullet 2})_{X_1}(x_1) \wedge \ldots \wedge (A^{1\bullet 2})_{X_n}(x_n) \quad (6)$$

$$A_{X_i}^{1\bullet 2} = A_{X_i}^{1\bullet 2}(c_i^{1\bullet 2,A}, \min(\alpha_i^{1,A}, \alpha_i^{2,A}), \max(\beta_i^{1,A}, \beta_i^{2,A}))$$

Los dos centros $c_i^{1\bullet 2,A}$, $c_i^{1\bullet 2,C}$ se determinan mediante una combinación convexa de los centros iniciales. Si el conjunto A^1 está asociado a n_1 ejemplos y A^2 a n_2:

$$c_i^{1\bullet 2,A} = \omega_1 c_i^{1,A} + \omega_2 c_i^{2,A}$$
$$c_i^{1\bullet 2,C} = \omega_1 c_i^{1,C} + \omega_2 c_i^{2,C} \quad (7)$$

donde ω_1, ω_2 son

$$\omega_1 = \frac{n_1}{n_1 + n_2}; \quad \omega_2 = \frac{n_2}{n_1 + n_2} \quad (8)$$

Parametrización gaussiana

Dados dos conjuntos $A^1 \subset X_1^n$, $A^2 \subset X_2^n$, definidos por las funciones de pertenencia

$$\mu_{A_1}(\mathbf{x}) = e^{-k(\mathbf{x}-\mathbf{x}_{c1})^T C_1^{-1}(\mathbf{x}-\mathbf{x}_{c1})}$$
$$\mu_{A_2}(x) = e^{-k(\mathbf{x}-\mathbf{x}_{c2})^T C_2^{-1}(\mathbf{x}-\mathbf{x}_{c2})} \quad (9)$$

el conjunto $A^1 \bullet A^2$ se define como

$$\mu_{A_{1\bullet 2}}(\mathbf{x}) = e^{-k(\mathbf{x}-\mathbf{x}_{c1\bullet 2})^T C_{1\bullet 2}^{-1}(\mathbf{x}-\mathbf{x}_{c1\bullet 2})} \quad (10)$$

donde

$p = \frac{n_1}{n_1+n_2}$

$\mathbf{x}_{c\ 1\bullet 2} = p\ \mathbf{x}_{c1} + (1-p)\ \mathbf{x}_{c2}$

$C_{1\bullet 2} = p\ C_1 + (1-p)\ C_2 +$

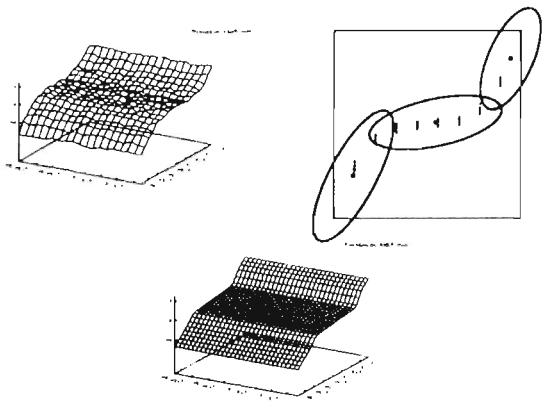
$p(1-p)(\mathbf{x}_{c1} - \mathbf{x}_{c2})(\mathbf{x}_{c1} - \mathbf{x}_{c2})^T$

2.5 Procedimiento de simplificación

A partir de los n ejemplos que se definieron en 2.1 se realizará $n-1$ veces una agrupación entre ellos, hasta llegar a un único conjunto.

Para realizar esta tarea, se forma un árbol binario, en el que las hojas son los conjuntos iniciales y cada nodo es la agrupación de sus dos ramas. Esta disposición reduce el problema de simplificación a la realización de un clustering jerárquico sobre estos conjuntos.

Cada nodo del árbol es un conjunto borroso, asociado a una regla de uno de los n modelos que se generan en el proceso -precisamente, el conjunto mencionado en la sección 2.1-. Puede justificarse que el árbol óptimo será aquél que minimize una función de energía calculada como la suma de las entropías de todos sus nodos.



Datos del modelo Reglas Estimadas Modelo Borroso

Figure 5: Resultados con planta de test

2.6 Extracción de reglas

Una vez construido el árbol, puede deducirse de él una colección de n grupos de conjuntos. Cada uno de estos grupos origina un modelo del proceso.

Cada conjunto de un grupo se convierte en una regla borrosa de forma inmediata. En primer lugar, se aplican las operaciones de agrupación indicadas en el árbol a los conjuntos derivados de las m primeras coordenadas de cada ejemplo extendido. Los conjuntos resultantes forman los antecedentes. Si las reglas son del tipo I, los consecuentes se calcular al repetir la operación con las n últimas coordenadas de cada ejemplo. Si son del tipo II, cada consecuente se estima ajustando un modelo lineal a los n_{p_i} ejemplos que forman las hojas del árbol cuya raíz es el conjunto borroso que se procesa.

2.7 Planta de ejemplo

Una de las plantas empleadas en los tests tiene como función de transferencia la parábola cúbica $z = x^3$. A partir de 100 valores generados por simulación, se estimó según este método un modelo borroso de la planta. Según el criterio expuesto, el número natural de reglas fue de 3. En la figura 5 se muestran los gráficos de las funciones de transferencia del proceso simulado y del modelo deducido. Entre las dos, se representa la proyección en el plano XZ de los α-cortes al 90 por ciento de los conjuntos borrosos asociados a las tres reglas.

2.8 Comparación con otros métodos

El sistema no lineal

$$y = (1 + x_1^{0.5} + x_2^{-1} + x_3^{-1.5})^2$$

planteado por Sugeno, ha sido empleado por otros autores para comparar las características de varios sistemas con capacidad de aprendizaje.

La prueba emplea un conjunto de 40 datos de tres entradas y una salida, la mitad de los cuales se emplean para identificar la planta y los restantes para evaluar el modelo identificado. Adicionalmente, se contamina este conjunto de entrenamiento introduciendo una cuarta entrada x_4 no correlacionada con la salida del sistema.

El ajuste se mide con la función de error

$$E = \frac{1}{N} \sum_{i=1}^{N} \frac{|y_i - y_i^*|}{y_i} \times 100$$

donde N es el número de datos, y_i es la respuesta deseada y y_i^* es la respuesta ofrecida por el modelo.

Los resultados obtenidos se cotejarán con los obtenidos por una red neuronal funcionalmente equivalente a un modelo borroso de Sugeno y entrenada por descenso de gradiente. Las prestaciones de esta red fueron comparadas favorablemente por su autor con los métodos más habituales de estimación de reglas [1].

El tiempo de cálculo empleado por el proceso propuesto es muy inferior al necesitado para entrenar la red neuronal, lo que justifica el algoritmo de simplificación en problemas reales.

Nótese que el resultado obtenido indica que el modelo más adecuado tiene 2 reglas. Esta conclusión coincide con la razonada experimentalmente por Horikawa, según los resultados mostrados en la tabla 6. En esta tabla se denomina E_1 al error obtenido en los ejemplos de entrenamiento, y E_2 al error obtenido en los puntos de test.

Los resultados segundo y tercero de la tabla 7 han sido obtenidos bajando el margen del ruido

Reglas	E_1	E_2
2	0.48	21.36
3	0.00	31.29
4	0.00	53.26
5	0.00	55.74

Figure 6: Precisión de la FNN de Horikawa

Reglas	E_1	E_2
2	12.69	19.07
5	8.76	19.07
5	5.63	19.06

Figure 7: Precisión de este método

inicial en el proceso de entrenamiento. Comparando ambas tablas, puede deducirse que la red neuronal ha realizado un ajuste menos correcto que el sistema propuesto para los modelos de 3,4 y 5 reglas, dado que el error E_2 es superior al del modelo más simple, que es un caso particular de los demás. La causa de este ajuste radica en que ese modelo neuronal no está realizando una distinción entre información relevante y ruido.

3 Conclusiones

Este método emplea un proceso basado en teoría de la información, que determina automáticamente todos los parámetros que definen el modelo borroso de un sistema. Se ha comprobado que el modelado obtenido es bueno incluso cuando los datos están distorsionados por gran cantidad de ruido.

La implementación del método es fácil y su coste computacional es relativamente bajo, por lo que es adecuado para ser empleado en tiempo real. El número de ejemplos necesitado y el tiempo preciso para deducir modelos arbitrariamente no lineales es muy inferior al que se emplearía si se empleasen arquitecturas conexionistas y entrenamiento por descenso de gradiente.

Bibliografía

[1] Horikawa, S., Furuhashi, T., Okuma, S., Uchikawa, Y. (1990) "Composition Methods of Fuzzy Neural Networks". Department of Electronic Mechanical Engineering, Nagoya University.

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

[3] Lee, Ch, (1990). Fuzzy Logic in Control Systems: Fuzzy Logic Controller. IEEE Transactions on SMC, Vol 20, no. 2, pp 404-418 and 419-435

[4] Sánchez, L. (1993). Extracción Automática de Reglas en un Sistema de Control Borroso. III Congreso Español de Tecnologías y Lógica Fuzzy. Santiago de Compostela. pp 243-250.

[5] Sánchez, L. (1994). *Control Difuso de Procesos Industriales mediante una Arquitectura Paralela y Distribuida, tipo Red Neuronal.* Tesis Doctoral. Universidad de Oviedo.

[6] Sánchez, L. (1994). An estimator of the number of rules in fuzzy models. Admitido en SICICA'94. Budapest, Junio 1994.

[7] Wallace, R.S. (1989). *Finding Natural Clusters through Entropy Minimization.* Ph. D. Thesis. School of Computer Science. Carnegie Mellon University.

[8] Xuecheng, Liu. (1992). Entropy, distance measure and similarity measure of fuzzy sets and their relations. Fuzzy Sets and Systems, **52**, 305-318. North-Holland.

[9] Zadeh, L. (1973). Outline of a New Approach to the Analysis of Complex Systems and Decision Process. IEEE trans. on SMC, vol SMC-3, no. 1, pp 28-44

Un Método para Validar Procesos de Clustering Difuso

M. Delgado M. A. Vila
Universidad de Granada
18071 Granada, España
e-mail: mdelgado@ugr.es avila@robinson.ugr.es

A. F. Gómez-Skarmeta
Universidad de Murcia
30071-Espinardo, Murcia, España
e-mail: skarmeta@dif.um.es

Resumen

Nuestra propuesta pretende aportar algunas alternativas al problema de la validación del clustering difuso, introduciendo la realización de un análisis previo de los datos mediante el uso de clustering jerárquicos para establecer un ranking entre las posibles particiones como preproceso al clustering difuso. Hemos definido una serie de medidas de validez a aplicar a un cluster jerárquico, con el objeto de detectar aquellas particiones que nos den unos clusters que reflejen la estructura interna de los datos. El interés del cluster jerárquico frente al particional es que nos permite trabajar con múltiples posibles particiones mediante algoritmos de poca complejidad computacional, permitiéndonos reducir el espacio de búsqueda de un clustering particional difuso.

Palabras Claves: clustering difuso, medidas validez, relaciones similitud.

1 Modelado Difuso y Clustering

En general la idea que subyace en el modelado difuso es la representación de propiedades de las relaciones entre entidades por medio de conjuntos difusos (o valores linguísticos). Entre las diferentes de modelos difusos uno de los más prometedores es aquel que se apoya en el análisis cluster. Clusters difusos son utilizados para determinar la partición del espacio de una determinada variable o un conjunto de ellas. Dos ventajas obvias de éstas técnicas son:

1. la formación automática de conjuntos difusos basándose en que las técnicas clustering son esquemas de aprendizaje no supervisado y como tales no requieren ningún conocimiento inicial sobre la estructura de los datos.
2. la posibilidad de extraer toda la información contenida en los datos como consecuencia del enfoque del análisis cluster que desarrolla toda posible estructura durante el proceso de clustering.

En la identificación de sistemas los procedimientos de clustering difuso han sido utilizados de diferentes modos[10][11]. Uno de los problemas fundamentales en el clustering y en particular en el difuso es el de la validación. Este aspecto ha sido tratado por diversos autores y en general todos los enfoques consisten en la introducción de una función de validación. Entre los métodos más empleados en el clustering difuso están, los grados de separabilidad, los coeficientes de partición y la entropía de la clasificación, los cuales han sido estudiados en profundidad por Bezdek[2][4]. Esta variedad de medidas de validez dependientes de los métodos y en otros casos ad hoc, sugieren que es en general muy difícil el obtener soluciones generales al problema de la validación.

Nuestra propuesta pretende aportar algunas alternativas a este problema, introduciendo la realización de un análisis previo de los datos mediante el uso de clustering jerárquicos que permitan establecer un ranking entre las posibles particiones como preproceso al clustering difuso. Así en la sección 2 daremos un repaso a la técnica de clustering difuso más extendida, para a continuación en la sección 3 estudiar el uso del clustering jerárquico como apoyo al clustering difuso. Finalmente en la sección 4 veremos algunos ejemplos numéricos de aplicación de nuestra propuesta para finalizar en la sección 5 con las conclusiones.

2 Clustering Difuso

2.1 Algoritmo de las C-medias Difusas FCM

Una de las funciones objetivos más conocidas y utilizadas para el clustering en un conjunto de datos X, es la clásica función de suma de errores cuadráticos dentro de un grupo (within groups sum of squared errors WGSS), notada comúnmente por J_1 :

$$J_1(U, v : X) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik} \|x_k - v_i\|^2$$

donde $v = (v_1, v_2, ..., v_c)$ es un vector de centroides de los clusters(pesos o prototipos) en principio desconocidos, con $v_i \in R^p$ para $1 \leq i \leq c$ y $U \in M_{cn}$ una $c-$partición crisp o convencional de X. Las particiones óptimas U^* de X forman parte de los pares (U^*, v^*) que son mínimos locales de J_1. Como un medio explícito para obtener particiones óptimas de unos datos, J_1 fue popularizada formando parte del algoritmo ISODATA por Ball y Hall en 1967. Dunn fue el primero en generalizar $J_1()$[7], permitiendo a U ser difusa y a la norma ser una norma arbitraria basada en un producto escalar cualquiera. Bezdek generalizó

la función de Dunn a la familia infinita descrita por[2]:

$J_m(U, P : X) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m D^2_{ik,A}$
$m \in [1, \infty)$ es un exponente de peso sobre cada pertenencia difusa
$U \in M_{fcn}$ es una c – partición difusa restringuida de X
$P = v = (v_1, v_2, ..., v_c)$ son c vectores prototipos en R^p
$A =$ cualquier matriz$(p \times p)$ positiva definida
$D_{ik,A} = \|x_k - v_i\|_A = \sqrt{(x_k - v_i)^T A (x_k - v_i)}$

$D_{ik,A}$ es la distancia (asociada a la norma $\| \ \|_A$) de x_k a v_i. Las c–particiones del conjunto X obtenibles por la aplicación de un algoritmo de clustering se caracterizan como conjuntos de (cn) valores $\{u_{ik}\}$ que satisfacen las condiciones siguientes:

$$\begin{aligned} &\text{i) } 0 \le u_{ik} \le 1 \quad 1 \le i \le c, \ 1 \le k \le n \\ &\text{ii) } 0 < \sum_{k=1}^{n} u_{ik} < n \quad 1 \le i \le c \\ &\text{iii) } \sum_{i=1}^{c} u_{ik} = 1 \quad 1 \le k \le n \end{aligned} \tag{1}$$

Generalizaciones a J_m y aplicaciones de las misma han permitido grandes progresos[4][9][5][2][10][11]. Las condiciones necesarias que definen los algoritmos iterativos para minimizar aproximadamente J_m son conocidas:

Teorema 1 (Fuzzy C-means (FCM))
Supongamos $D_{ik,A} > 0 \ \forall i, k$. (U, v) minimizará J_m sujeto a las condiciones señaladas anteriormente para $m > 1$ sólo si:

$$\begin{aligned} u_{ik} &= \left[\sum_{j=1}^{c} \left(\frac{\|x_k - v_i\|_A}{\|x_k - v_j\|_A} \right)^{\frac{2}{m-1}} \right]^{-1} \quad \forall \ i, k; \\ v_i &= \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m} \quad \forall \ i \end{aligned} \tag{2}$$

Estas ecuaciones convergen a las del algoritmos HCM (versión crisp) a medida que $m \longrightarrow^+ 1$, y para el caso de que $m > 1$ U es una partición difusa, es decir $U \in (M_{fcn} - M_{cn})$. Los algoritmos HCM/FCM son simples iteraciones de Picard a través de las condiciones necesarias señaladas, de forma que una breve descripción de los mismo sería:

Algoritmo

P.1. Dado un conjunto de datos no etiquetados $X = \{x_1, x_2, ..., x_n\}$. Fijese $c, T, \| \ \|_A$ y $\epsilon > 0$.

P.2. Inicializar $U_0 \in M_{fcn}$. Elegir $m = 1$ (HCM) o $m > 1$ (FCM). Calcular los vectores prototipos $v_{i,0}$ según (2) para FCM.

P.3. Para $t = 1, 2, ..., T$

1. *Computar los (cn) grados de pertencia $\{u_{ik,t}\}$ según (2) para FCM*
2. *Actualizar los (c) vectores prototipos $\{v_{i,t+1}\}$ según (2) para FCM*

P.4 Si $E_t = \|v_{t+1} - v_t\| = \sum_{i=1}^{c} \|v_{i,t+1} - v_{i,t}\| \le \epsilon$ parar, en otro caso ir a paso P.3

Es conocido que este procedimiento converge desde cualquier inicialización a un mínimo local o un punto de silla de J_m que notaremos como (U^*, v^*).

2.2 Validación de Clusters Difusos

Durante el proceso de clustering de los datos no es posible en general realizar presunciones acerca del número de clusters subyacentes. Cuando no existe información a priori sobre la estructura interna de los datos, o en el caso de evidencias conflictivas sonbre el número óptimo de grupos, es necesario formular medidas de rendimiento para comparar la bondad de las particiones obtenidas para los diferentes números de clusters. Existen pues tres grandes dificultades durante el proceso del clustering difuso de unos datos reales:

1. el número de cluster no puede ser siempre determinado a priori, necesitándose por tanto un criterio de validez del clustering para determinar el número óptimo de clusters presentes en los datos
2. el caracter y localización de los centroides de los clusters no está normalmente determinado a priori, y es necesario realizar una elección inicial sobre ellos
3. es posible la existencia de grandes variabilidades en la forma, densidad y número de puntos en los diferentes clusters

Esto ha llevado en los últimos tiempos a la presentación por diferentes autores de una serie de criterios tanto de validez de los clusters, como de modificación del propio proceso de clustering, haciéndolos orientados al objetivo. Se trata de proceso de personalización u objetivización de los algoritmos en el sentido de hacerlos procesos de aprendizaje no supervisados de un determinado problema y no de una generalidad de ellos. En este sentido tenemos diferentes propuestas[4][8][9], cuyas características pueden sintetizarse en:

1. búsqueda de inicializaciones adecuadas para aproximar la posición de los centroides
2. posibilidad de un segundo ajuste(debido a utilizar algoritmos más inestables es deseable una buena inicialización obtenida por el paso primero) buscando el óptimo de los centroides y las particiones en una zona más reducida. Para este ajuste se ha empleado:

(a) distancias de tipo exponencial que manejan una zona más restringuida del espacio de soluciones

(b) enfoque posibilístico de las particiones, relajando la condición iii) de (1)

(c) variación de m que al controlar el grado de fuziness de las particiones puede irse variando con las iteraciones para acercarse a la visión crisp

3. necesidad de criterios de validez que permitan determinar entre una amplia gama de posibles valores de c. Entre estos criterios podemos destacar: el índice de separación D definido por Dunn[4] que identifica clusters compactos y separados, y que se debe intentar maximizar; el coeficiente de partición F introducido por Bezdek[2] para medir la cantidad de solapamiento entre los clusters, y que se busca maximizar; el índice S propuesto por Xie y Beni[4], el cual estudia la validez de la separación y compactez del clustering, y que es deseable ver minimizado; o el índice SS utilizado por Sugeno y Yasukawa[11], una medida que involucra la varianza en cada cluster y la varianza entre clusters y que se busca minimizar.

El principal problema asociado al empleo de estas medidas es la complejidad que conlleva tener que repetir los cálculos para diferentes valores de c. Además, dada la inicialización aleatoria del algoritmo FCM para los valores de (U, v), es necesario probar con diferentes inicializaciones para comprobar que se ha llegado a un mínimo local.

2.3 Implicaciones

Ante estos enfoques nuestra propuesta es el desarrollo de una variante que combine algunas de estas ideas para sucesivamente:

1. obtener un conjunto de posibles valores de c y sus particiones asociadas para dar así una aproximación a los centroides iniciales de los clusters

2. partiendo de estas inicializaciones aplicar un algoritmo FCM pero variando los valores de m, o bien dando un segundo ajuste con los algoritmos PCM o FMLE[9][8]

3. tomando diferentes medidas de validez para los posibles números de clusters considerados en la primera fase, determinar cual o cuales podrían ser los casos más aceptables.

3 Clustering Jerárquico

El clustering jerárquico nos brinda la oportunidad de trabajar con múltiples posibles particiones (desde los casos triviales hasta agrupar todos los datos en un único cluster). Por tanto se puede analizar el comportamiento de los datos probando sobre ellos diferentes medidas de validez y en consecuencia analizar posibles estructuras en los mismos.

Se van a considerar tres posibles criterios de optimalidad basados en : 1) concepto de distancia intra e inter grupos: 2) relación existente entre particiones y la matriz de similaridad asociada la jerarquía de particiones; 3) comparaciones entre conjuntos basadas en la teoría de conjuntos difusos.[12][13]

3.1 Relaciones de Similitud

Asociado con un cluster jerárquico existe un distancia ultramétrica que podemos representar por una matriz: $U = (u_{ij}), i = 1, 2, \quad , n, j = 1, 2, \ldots, n$, que puede normalizarse siempre en [0,1]. En ella se satisface la desigualdad ultramétrica, sólo en caso de que los clusters se hayan formado de una manera monótona a medida que la disimilaridad aumenta. Por esto al cumplimiento de la desigualdad ultramétrica también se le denomina monotonicidad. No todos los métodos de agrupamiento jerárquico son monótonos, y entre los que sí lo son se encuentran los más utilizados del mínimo, el máximo y el del método Ward[6].

Zadeh estableció [14] que la transitividad $\max - \min$ es equivalente a la desigualdad ultramétrica para la función $r_{ij} = 1 - u(x_i, x_j)$, donde $R = [r_{ij}]$ es una matriz transitiva max-min, también llamada relación de similaridad o en general una F-indistinguibilidad con $F = Min$. [3]

Notaremos por $A^b = (a^b_{ij})$ con $b \in [0, 1]$, a la matriz de relaciones de equivalencias crisp que se obtienen para los diferentes b-cortes ($\alpha-$cortes) de la relación de similaridad R.

La secuencia $S_k (k = 1, \ldots, h)$, de todos los posibles b-cortes de la relación de similaridad, en orden ascendiente, será de gran utilidad en los desarrollos posteriores: $\forall k (k = 1, ..., n), \ \exists \ r_{ij}/r_{ij} = S_k$ y $0 = S_1 < S_2 < \ldots < S_h$

Finalmente para un nivel $b \in \{S_k\}$ de similaridad, notaremos por $\{C^b_i, i = 1, \ldots, m(b)\}$, los clusters o particiones que se inducen en ese nivel, de forma que:

$$\forall i, j \ \ \exists \ e/ \ i, j \in C^b_e \Longleftrightarrow a^b_{ij} = 1$$

$$\bigcup_i C^b_i = \{1, \ldots, n\} \quad C^b_e \bigcap C^b_{e'} = \emptyset$$

siendo $\mid C^b_e \mid$ el cardinal del conjunto $C^b_e \subset \{1, \ldots, n\}$.

3.2 Medidas basadas en distancias intra-inter clusters

En función de la distancia entre elementos de cada cluster y la distancia entre clusters, se pueden definir dos funciones de distancia $d^b$and D^b. La primera nos dará una medida global de las distancias entre los elementos de cada cluster, mientras que la segunda representará una medida global de las distancias entre los clusters. Ambas estarán definidas sobre la sucesión $\{S_k\}$ y estarán valuadas en [0,1].

Definición 1 . *Sea $b \in \{S_k\}$ y $\{C^b_i, i = 1, \ldots, m(b)\}$ los clusters inducidos por el b-corte. Para cualquier*

$e = 1, \ldots, m(b)$. *La distancia global media inter clusters es.*

$$d_1^b = \sum_{e=1}^{m(b)} d^b(e)/m(b)$$

$$con\ d^b(e) = \begin{cases} \frac{\sum_{i,j \in C_e^b, i \neq j} 2u_{ij}}{|C_e^b| * (|C_e^b| - 1)} & si\ |C_e^b| > 0 \\ 0 & en\ otro\ caso \end{cases}$$

La distancia global media entre los clusters será $\forall e, e' \in \{1, 2, \ldots, m(b)\}$:

$$D_1^b = \begin{cases} \frac{\sum_{e=1}^{m(b)-1} \sum_{e'>e} 2D^b(e,e')}{m(b) * (m(b)-1)} & si\ m(b) > 1 \\ 0 & en\ otro\ caso \end{cases}$$

$$con\ D^b(e, e') = u_{ij} \quad i \in C_e^b\ j \in C_{e'}^b$$

Definición 2 *Bajo las mismas hipótesis anteriores podríamos definir:*

$$d_2^b = \max_{\{e=1,\ldots,m(b)\}} \left\{ \max_{i,j \in C_e^b} u_{ij} \right\}$$

$$D_2^b = \min_{\{e,e'=1,\ldots,m(b)\}, e \neq e'} D^b(e, e')$$

Es evidente que d_2^b es la distancia máxima entre los elementos de un cluster, y D_2^b es la mínima distancia entre los clusters. En estas condiciones para obtener la partición óptima proponemos maximizar la combinación convexa:

$$H_{pq}(S_k) = pD^{Sk} - qd^{Sk} \quad p, q \in [0, 1]$$

donde p y q son los pesos subjetivos que se dan al hecho de que los clusters esten separados y de que los puntos de un clusters esten cercanos respectivamente. Caso de que estos pesos sean iguales se tomaría $H(S_k) = D^{Sk} - d^{Sk}$

Proposición 1 (Propiedad de Estabilidad) *Sea la función objetivo $H(S_k)$ con funciones de distancia D_2 y d_2. Se demuestra que si S_t es el punto de la sucesión $\{S_k\}$ donde se alcanza el máximo de $H(S_k)$, entonces los clusters inducidos por dicha partición son los que se mantienen en un rango de similaritud mayor dentro del clustering jerárquico, es decir es la partición más estable:*

$$\max_k \left(D_2^{Sk} - d_2^{Sk}\right) = \max_k \left(S_k - S_{k-1}\right)$$

3.3 Medidas basadas en distancias matriciales

Cabe pensar que, un corte en un determinado nivel de la jerarquía de particiones, será óptimo si la distancia entre su matriz de partición A^b, y la matriz de similitud es mínima. Podemos dar entonces un criterio de partición óptima tomando como distancia $d(A, B) = \max_{i,j} \mid a_{ij} - b_{ij} \mid$, siendo $A = (a_{ij})$ y $B = (b_{ij})$ matrices de la misma dimension. Si consideremos una relación de similitud R, su sucesión asociada $\{S_k\}$ y la sucesión de matrices correspondientes a las relaciones de equivalencias crisp correspondientes $\{A^k\}$ $k = 1, \ldots, h$, (Nota: por comodidad $A^k \equiv A^{Sk}$). En estas condiciones podemos utilizar como función objetivo $H(S_k) = d(R, A^k)$ que deberá minimizarse.

Se demuestra que el valor mínimo de H se alcanza en el intervalo de similitud que contenga al punto 0.5. En el caso en que exista $S_e = 0.5$, el valor mínimo se dará en todo el intervalo $(S_{e-1}, S_{e+1}]$, y existirán dos particiones óptimas.

3.4 Medidas basadas en los conjuntos difusos asociados a una partición

Para todo $e \in \{1, 2, \ldots, m(b)\}$, podemos definir un subconjunto difuso F_e^b asociado a C_e^b con función de pertenencia: $F_e^b(i) = \min_{j \in C_e^b} r_{ij} \quad \forall i \in \{1, 2, \ldots, n\}$. Esta función puede interpretarse como el mínimo nivel en el cual se unirán el cluster e y aquel al que pertenece i. Así para cualesquiera $i, j \in C_h^b, h \in \{1, 2, \ldots, m(b)\}$ y para todo e, se verifica $F_e^b(i) = F_e^b(j)$. Por tanto cualquier condición o tratamiento que se aplique a F_e^b, no se verificará punto a punto sino globalmente, conjunto a conjunto, y por tanto $F^b(e, h) = F_e^b(i) \quad \forall i \in C_e^b$ para cualquier $e, h \in \{1, 2, \ldots, m(b)\}$, y cualquier nivel b. Además $F^b(e, h) = F^b(h, e)$, debido a la simetría de la relación R.

Definidos los subconjuntos difusos F_e^b, construímos el conjunto crisp $X_e^b = \{i / F_e^b(i) \geq P_e^b\}$ que denominaremos "más próximo" a F_e^b, con $P_e^b = \{\max_i F_e^b(i) - \min_i F_e^b(i)\}/2$, el valor medio de la similaridad del cluster. X_e^b establece los elementos que se unirán al cluster en un nivel de similaridad que corresponde a la media del actual. Es inmediato que $C_e^b \subset X_e^b$ y que $C^0, X^0 = \{1, 2, \ldots, n\}$.

En estas condiciones parece lógico pensar que C_e^b será tanto más coherente con la información inicial cuanto más parecido sea a X_e^b. Una medida de la diferencia entre X_e^b y C_e^b podría formularse como:

$$H_e(b) = \frac{\sum_{C_h^b \subset X_e^b} |C_h^b| \left(F^b(e, e) - F^b(e, h)\right)}{|X_e^b| - |C_e^b|} \qquad (3)$$

Como el objetivo es determinar un nivel b óptimo, se hace necesario establecer una medida global de las

diferencias entre $\{C_e^b, e = 1, 2, \ldots, m(b)\}$ y $\{X_e^b, e = 1, 2, \ldots, m(b)\}$ para todo b.

A partir de (3) dicha medida global puede definirse como:

$$H(b) = \sum_{e=1}^{m(b)} H_e(b)/m(b)$$

que deberemos minimizar sobre $\{S_k\}$. Se trata de medir para cada nivel lo que le cuesta pasar al siguiente, es decir cuanta variación en la similaridad supone dicho salto y $H(b)$ mide la tendencia del nivel b a cambiar de estado. En particular se demuestra que si S_t es tal que $S_{t-1} < S_t/2$, entonces $H(S_t) = 0$.

La búsqueda de un valor mínimo para $H()$, choca con el hecho de que esta función es siempre nula en los dos primeros niveles de la sucesión. Por ello para elegir el valor de b óptimo, tomaremos b óptimo= max $\{b/H(b) = 0\}$

3.5 Utilización de las medidas

Un aspecto interesante en relación con el uso de estas medidas es que su principal función será darnos una información acerca de cuales son los niveles de similitud que generan particiones que mejor reflejen la estructura interna de los datos. Debemos recordar que las particiones que generan son crisp y por tanto puede resultar que una partición que no sea la óptima sea la que posteriormente de un mejor resultado a la hora de realizar con ella un partición difusa de los datos.

Por otro lado señalamos que cada medida puede dar un nivel óptimo distinto, lo que apoya más la utilización de estas medidas: 1) como una forma de ordenar en función de su bondad los posibles niveles, los cuales posteriormente pasarían a ser evaluados en ese orden mediante medidas de validez difusas correspondientes a las particiones difusas obtenidas mediante el algoritmo FCM; 2) además tenemos como ventaja el que es posible utilizar la partición crisp correspondiente como inicialización del algoritmo FCM, con lo que se deberá tener como resultado una reducción en el número de iteraciones necesarias para el mismo

Con este enfoque por tanto superamos la mayor dificultad asociada a la validación de los clustering difusos, como es el tener que ejecutar el algoritmo FCM para un número considerable de valores de c, además de diferentes veces para cada valor de c con objeto de obtener el mejor mínimo local (U^*, v^*).

4 Ejemplos

Para comprobar la efectividad de las medidas anteriores hemos utilizado diversos ejemplos de los que a continuación comentamos dos debido a su especial significación:

1. análisis de los datos experimentales de los IRIS de Anderson, el cual se trata de un conjunto de datos que se ha utilizado extensamente para ilustrar diseños de clusterings y clasificadores[3][4][8]. Con este ejemplo hemos comprobado los dos objetivos fundamentales de nuestra propuesta: detectar diferentes particiones optimales; utilización de las particiones crisp como inicializaciones del algoritmo FCM y analizar las consecuencias sobre su rendimiento

   La aplicación sobre el conjunto de datos del clustering jerárquico con diferentes tipos de métodos de agrupación monótonos como el de distancia mínima entre cluster (single-link), distancia máxima entre cluster (complete-link), o el método de Ward (mínima varianza), generan todos para las diferentes medidas que los óptimos se encuentran entre los 2 ó 3 clusters. Esto coincide plenamente con la visión gráfica de los datos, o con los resultados de las medidas de validez difusas aplicadas a este caso concordando con los resultados señalados en diferentes estudios[2][4][8].

   Si consideramos el caso de 3 clusters, y comparamos los resultados de un caso sin inicialización concreta y aquella en que se utiliza la partición crisp como medida de inicialización obtenemos los siguientes valores:

   Caso FCM
   Valores $S = 0.1369$ $F = 0.7833$
   Valores $SS = -444.5192$ $D = 0.0110$
   No. de Iteraciones : 28

   Caso FCM con Clustering Jerárquico
   Valores $S = 0.1369$ $F = 0.7833$
   Valores $SS = -444.5298$ $D = 0.0110$
   No. de Iteraciones : 17

   Como puede observarse en la aplicación de la inicialización basada en el resultado de la partición del clustering jerárquico se han visto reducido el número de iteraciones, sin que se vea afectado la determinación de los centroides que coinciden, como también las medidas de validez difusas.

2. se refiere a un ejemplo propuesto en [9], consistente en introducir dos datos de tipo ruido sobre un conjunto de 12 datos que forman dos cluster bien definidos. Estos dos datos se encuentran equidistantes de los anteriores, y uno de ellos mucho más alejado que el otro de los dos clusters. Su principal interés se refiere a la capacidad de detectar esta situación anómala, que estaría formada por cuatro clusters con dos de ellos referidos a cluster de un único punto con lo que se podría detectar que son ruido, o bien la existencia de tres clusters con uno de ellos agrupando a los datos que son ruido. La aplicación de los clustering jerárquico del mínimo y el máximo, no da como posibles soluciones optimales para las diferentes medidas la existencia de 2 ó 4, correspondiente en los casos que se da como solución 2 al caso del punto más alejado y teniendo en estos casos además como segunda alternativa optimal el valor de 4 clusters. Si se utiliza el método Ward en el clustering, se obtienen para todas las medidas como solución óptima primera el valor de 3 clusters, agrupándose los dos puntos ruidos en uno de ellos. Esta solución coincide además con

el enfoque propuesto por Dave[5], referido a introducir un cluster adicional para englobar en él a los datos que sean ruido de los problemas.

5 Conclusiones

Hemos definido una serie de medidas de validez a aplicar a un cluster jerárquico, con el objeto de detectar aquellas particiones que nos den unos clusters que, de alguna forma, parezcan reflejar la estructura interna de los datos. El interés del cluster jerárquico frente al particional es que nos permite trabajar con múltiples posibles particiones mediante algoritmos de poca complejidad computacional. Así reducimos el espacio de búsqueda de un clustering particional difuso posterior, ya que nos permite considerar:

1. un conjunto de posibles valores de c, es decir de número de clusters información fundamental en el algoritmo particional difuso más extendido el del FCM
2. además al obtener particiones crisp, podemos tomar a estas como bases para determinar los valores iniciales de los centroides, que se situarían en el centro de las distribuciones de las particiones crisp
3. al disponer de una valores previos de c esto permite centrar el estudio del comportamiento de estos posibles clusters, mediante otras medidas de validez obtenidas sobre las particiones difusas de los clustering realizándose por tanto un segundo estudio de la validez según los resultados del clustering particional del FCM

Abordamos por tanto el problema del clustering, con un enfoque similar a las aproximaciones más recientes, consistentes en realizar diversas pasadas sobre los datos que nos permitan centrar los resultados obtenidos y utilizar múltiples informaciones a la hora de guiar la selección de las particiones más adecuadas.

Referencias

[1] Bezdek, J.C.; Harris, J.D., Fuzzy partitions and relations; An axiomatic basis for clustering, Fuzzy Sets and Systems, vol 1, pp. 111-127 (1978)

[2] Bezdek, J.C., Pattern Recognition with Fuzzy Objective Function Algorithms, Plenun, New York (1981)

[3] Bezdek, J.C.; Tsao, E.C.; Pal, N.R., Fuzzy Kohonen Clustering Networks, Proc. First IEEE Conf. on Fuzzy Systems, San Diego, pp. 1035-1043 (1992)

[4] Fuzzy Models for Pattern Recognition, Bezdek, J.C.; Pal, S.K. Eds., New York, IEEE Press (1992)

[5] Dave, R.N., Robust Fuzzy Clustering Algorithms, Proc. IEEE Conf. on Fuzzy Systems, pp. 1281-1286 (1993)

[6] Dubes, R.; Jain, A., Algorithms that Cluster Data, Prentice Hall, Englewood Cliffs (1988)

[7] Dunn, J., A Fuzzy Relative of the ISODATA Process and its use in Detecting Compact Well Separated Clusters, J. Cybernet 3, pp. 32-57 (1974)

[8] Gath, I., Geva, A., Unsupervised Optimal Fuzzy Clustering, IEEE Trans. Pattern Anal. Machine Intell., vol 11, pp. 773-781 (1989)

[9] Krishnapuram, R.; Keller, J.M., A Possibilistic Approach to Clustering, IEEE Trans. Fuzzy Systems, vol. 1, No 2, pp. 98-110 (1993)

[10] Yoshinari, Y , Pedrycz W.; Hirota K., Construction of Fuzzy Models through Clustering Techniques, Fuzzy Sets and Systems, 54, pp. 157-165 (1993)

[11] Sugeno, M., Yasukawa T., A Fuzzy-Logic-Based Approach to Qualitative Modeling, IEEE Trans. Fuzzy Systems, vol 1, No 1, pp. 7-31 (1993)

[12] Vila Miranda, A., Criterios absolutos para el Cálculo de Particiones Optimas en Conglomerados Estratificados, Actas de la XII Reunion Nacional de I.O. (1980)

[13] Vila Miranda, A., Nota sobre el calculo de particiones optimas, Actas de la XI Reunion Nacional de I.O. (1979)

[14] Zadeh, L.A., Similarity relations and fuzzy ordering, Information Sciences, vol 3, pp. 177-200 (1971)

Modelo para Codificación de Información Difusa sobre Memorias Asociativas Discretas [1].

A. Blanco, W. Fajardo, I. Requena.

Dpto. Ciencias de la Computación e Inteligencia Artificial.
Universidad de Granada. E.T.S.I. Informática. 18071 Granada.
Tlfno.: 958 243100 e-mail: aragorn@robinson.ugr.es

Resumen

Presentamos un modelo de codificación para implementar un sistema de relaciones difusas mediante una memoria asociativa discreta, en particular se propone como ejemplo una BAM. La presentación termina con un ejemplo que pretende ilustrar la aplicación del mismo.

1. INTRODUCCIÓN.

Un sistema de memoria es un dispositivo capaz de almacenar información y recuperarla en un momento determinado. Tradicionalmente las memorias se han clasificado bajo tres etiquetas, en función de como se accede a la información almacenada en ellas; estos tres tipos son los siguientes: memorias de acceso directo (RAM), memorias direccionables por su contenido (CAM) y memorias asociativas (AM).

Tras esta breve clasificación y sabido que una memoria asociativa almacena información memorizada mediante correlación o asociación y accede a la misma a partir de datos o información, o sea, sus mapas de acceso relacionan datos con datos. Es fácil concluir sobre su particular adecuación para tratar información parcialmente alterada, bien sea por causa de ruido o por perdida de detalle en el contenido. la mayoría de las arquitecturas conocidas de sistemas neuronales artificiales (ANS) pueden ser usados como memorias asociativas, siendo sin duda esta capacidad una de las que dotan a las redes neuronales de parte de su interés.

En el campo de investigación de los sistemas neuronales artificiales, y en particular dentro de los múltiples modelos desarrollados, denominaremos como memorias asociativas a aquellos que han sido específicamente ideados con el fín de almacenar pares de patrones para su posterior recuperación a partir de alteraciones más o menos razonables de los mismos.

Otro campo, relacionado con el manejo de información y conceptos vagos es el del razonamiento aproximado. Este pretende, partiendo de una colección de premisas constituidas por afirmaciones o hechos imprecisos, obtener consecuencias (posiblemente 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 [9], [13], [14].

Tal y como vemos, en el razonamiento aproximado no estamos interesados en modo alguno en el sistema de almacenamiento de la información, únicamente pretendemos establecer una relación causa-efecto entre un par de cláusulas, para posteriormente y poder obtener un determinado efecto en función de una determinada causa relativamente cercana a la inicial.

Parece pues que si identificamos de alguna forma a las cláusulas del razonamiento aproximado con los patrones tratados por las memorias asociativas, podríamos pensar fundamentalmente en dos cosas:

1. Trabajo financiado por el proyecto PB 920946 de la DGICYT

a. Una memoria asociativa es un sistema que intenta inferir información a partir de un patrón más o menos cercano a uno dado.
b. El razonamiento aproximado pretende a partir de un par de patrones iniciales, dado uno de ellos parcialmente alterado, obtener el otro de la forma más perfecta posible.

De alguna forma no parece descabellado pensar que ambos sistemas tienen bastante en común, por lo menos en lo referido a origen y objetivos. Si el razonamiento aproximado se fundamenta en la teoría de conjuntos difusos, parece lógico combinar las memorias asociativas con los conjuntos difusos, para obtener un método perfeccionado de almacenamiento y proceso de la información vaga e imprecisa.

Así hablaremos de memorias asociativas difusas. Pero previamente, y antes de proponer una aproximación, daremos un repaso al concepto de memoria asociativa; y en particular a los modelos en los que nos fundamentaremos.

2.- Modelos de Memorias Asociativas.

Las memorias asociativas son uno de los grandes bloques temáticos de las redes neuronales. Hopfield [4] presentó un modelo autoasociativo monocapa unidireccional cuya potencia se encuentra más que probada. Este modelo fue posteriormente extendido a un modelo bicapa, bidireccional por Kosko [7]. La extensión así obtenida permite heteroasociación. Además de Hopfield y Kosko, otros autores [2], [5], [10] han tratado sobre el tema con más o menos fortuna. Veamos brevemente cual es el estado actual y en que forma se relacionan o distinguen los diversos modelos.

Centrándonos en modelos discretos, bidireccionales debemos comenzar hablando de la memoria asociativa bidireccional (BAM) introducida por Kosko. La BAM representa la extensión lógica de los modelos autoasociativos a los heteroasociativos. La clave del sistema se encuentra en el hecho de que el feedback entre las capas alcanza estabilidad en un determinado punto, en el que es posible almacenar un patrón. Una BAM es una memoria asociativa que codifica, de forma binaria o bipolar, pares de patrones espaciales utilizando aprendizaje Hebbiano. O sea, pretende almacenar m pares de patrones $\{(A_1, B_1), (A_2, B_2), \ldots, (A_i, B_i), \ldots, (A_m, B_n,$ donde

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

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

encontrándose a_{ij}, b_{ij} en estado ON u OFF

En función de que hablemos de codificación binaria o bipolar hablaremos de pares de la forma (A_i, B_i) ó (X_i, Y_i) respectivamente.

La codificación de la información en una BAM se lleva a cabo mediante una matriz de correlación, fruto de cualquier método de aprendizaje (supervisado o no supervisado), siendo el más popular el aprendizaje Hebbiano bipolar sobre los m pares de patrones a memorizar.

$$M = \sum_{i=1}^{m} X_i^T Y_i \qquad (1)$$

De esta forma la BAM pretende conseguir m atractores que coincidan con los modelos a memorizar. Esto no siempre es posible y en ocasiones es necesario hacer modificaciones sobre los patrones con el fin de que posteriormente sean recuperados de forma adecuada.

La BAM pretende recuperar la información en ella depositada partiendo de información incompleta o parcialmente errónea. Esto es posible gracias a que al presentar una información al sistema, esta caerá en uno de los entornos de atracción de los m atractores generados tras el aprendizaje.

Como es normal, las matrices de conexión sináptica son capaces de codificar una cantidad limitada de información, excedida esta, el sistema *olvida* o *confunde* algunos patrones. El limite en el cual ocurre esto es lo que denominaremos **capacidad de la memoria**. El sistema tiende a superar su capacidad conforme m se aproxima a n ó p, o al $\min(n, p)$.

En redes feedback no supervisadas, la literatura asegura, que la capacidad se encuentra severamente acotada por la dimensión. Estos análisis asumen codificación Hebbiana simple (1) y como límite de capacidad el de McEliece et al. [8]

$$\frac{n}{2\ \log_2 n} \tag{2}$$

Para otras codificaciones distintas al producto exterior bipolar (1), se demuestra que la capacidad de la BAM puede llegar incluso a superar $\min(2^n, 2^p)$ para codificaciones tales como el producto exterior Booleano (Hecht-Nielsen [3]).

Básicamente la BAM presenta dos problemas fundamentales: la precisión de almacenamiento y la capacidad.

El problema de la precisión de almacenamiento viene originado por el tipo de aprendizaje. El aprendizaje Hebbiano simple (1) conlleva el problema de que los atractores generados no siempre coinciden con los m patrones presentados al sistema, con lo cual se puede presentar el problema de que a pesar de que la información sea correcta, la recuperación de la asociación no de el resultado deseado. Para solucionar este problema se nos presentan fundamentalmente dos aproximaciones: la de Kamp & Hasler [5]; y la de Wang, Cruz & Mulligan [11], [12].

Para tratar el problema del entorno de atracción, Kamp y Hasler definen el concepto de **base y radio de atracción**, utilizando para ello la distancia de Hamming para vectores bipolares, definida por:

$$d(x_i,x_i') = \frac{1}{2}\sum_{j=1}^{n} |x_{ij} - x_{ij}'| \tag{3}$$

y definiendo entorno de atracción de x_i con radio r como:

$$E_r(x_i) = \left\{x_i' \in \{-1,1\},\ \text{t.q. } d(x_i,x_i') \le r\right\} \tag{4}$$

Enunciado de esta forma, el problema se reduce a codificar los m patrones (o pares de patrones) en el sistema mediante un método de aprendizaje que garantice su condición de base de atracción y para ello propone utilizar o bien una ponderación de la regla de Hebb

$$M = \frac{1}{m}\sum_{i=1}^{m} X_i^T X_i \tag{5}$$

o la regla de la pseudo-inversa

$$M = X(X^T X)^{-1} X^T \tag{6}$$

ya que con el método de aprendizaje Hebbiano no se garantiza un correcto almacenamiento y posterior recuperación de la información, o lo que es lo mismo, los atractores generados por el método de aprendizaje Hebbiano no siempre coinciden con los m pares de patrones proporcionados al sistema en el aprendizaje.

Tal y como ha sido definido el modelo de BAM, la aproximación de Kamp y Hasler puede considerarse como un caso particular del mismo, modificado de forma más o menos severa con el fin de solucionar los problemas de precisión en la recuperación.

Wang, Cruz & Mulligan proponen la solución del problema de la precisión en el almacenamiento de la información mediante la aplicación de dos estrategias de codificación en memorias asociativas bidireccionales. Se fundamenta en el hecho de que si la energía E calculada a partir de las coordenadas del par (A_i, B_i) ;

$$E = -A_i M B_i^T \tag{7}$$

no constituye un mínimo local, entonces el punto del espacio no puede ser recuperado a partir del un estado inicial de la forma $\alpha = A_i$. Wang et al. proponen un método de incremento en el aprendizaje y otro con respecto a la codificación con los cuales garantizan que la matriz M memoriza la información de forma que los pares de patrones asociados (equivalentes a puntos del espacio n+p dimensional) recaen sobre mínimos de energía locales (7).

Teniendo en cuenta las ventajas e inconvenientes del modelo y las distintas estrategias para superar los problemas que se presentan en el estado inicial, pasamos a hablar de memorias asociativas difusas.

3. Memorias Asociativas Difusas.

Tal y como comentábamos en la introducción, conocidos los objetivos del razonamiento aproximado y de las memorias asociativas, no parece descabellado la idea de combinar ambos conceptos, con el fin de obtener un método mejorado de almacenamiento y proceso de información vaga e imprecisa.

Según Kosko [6] podemos enfocar los sistemas difusos $S: I^n \longrightarrow I^p$ como una función que aplica familias de conjuntos difusos I^n en familias de conjuntos difusos I^p. Estos sistemas difusos continuos se comportan como memorias asociativas, trasladan entradas cercanas a la asociación original en salidas cercanas a la asociación original. Se denomina a dichos sistemas FAM (*fuzzy associative memorie*).

La FAM más simple codifica una **FAM regla** (A_i, B_i), que asocia el conjunto difuso n-dimensional A_i con el conjunto difuso p-dimensional B_i. Esta FAM es comparable a un ANS muy simple.

En general un sistema **FAM** $F: I^n \longrightarrow I^p$ codifica y procesa en paralelo un **FAM banco** de m FAM reglas $(A_1, B_1), \ldots, (A_m, B_m)$. Cada entrada A al sistema activa en un determinado grado cada una de las reglas almacenadas. El correspondiente conjunto difuso de salida B combina las correspondientes conjuntos difusos $B_1', \ldots, B_m'$ ponderándolas por un factor w_i que refleja la credibilidad, frecuencia, o fuerza de la asociación difusa (A_i, B_i). De esta forma tenemos

$$B = w_1 B_1' + \ldots + w_m B_m' \qquad (8)$$

Cada una de las FAM reglas memoriza la información mediante una matriz M fruto de una codificación por *correlación de mínimos*, que en definitiva no es más que una transformación del aprendizaje Hebbiano en la cual se ha substituido la norma del producto por la del mínimo.

$$m_{ij} = \min(a_i, b_j) \qquad (9)$$

que empleando notación matricial para el producto exterior, tendremos:

$$M = A^T \cdot B \qquad (10)$$

Posteriormente Kosko modifica la FAM convirtiéndola en una BIOFAM (memoria asociativa difusa con entradas y salidas binarias) con el fin simplificar el método de tratamiento de información para determinados sistemas.

4. Modelo Propuesto.

A pesar de todo, los modelos propuestos por Kosko adolecen de necesitar gran cantidad de memoria a la hora de codificar información. En particular en el modelo difuso, que es en el que estamos interesados, la necesidad de memoria requerida por el sistema, no solo depende de las dimensiones de los espacios de entrada y salida n y p sino que se incrementa con un factor multiplicativo m del número de reglas difusas a codificar, puesto que necesita una FAM por regla, quedando compuesto el sistema final por el conjunto de las correspondientes FAM reglas.

Sabemos que un sistema difuso se puede identificar mediante un sistema basado en reglas (Blanco [1]), y que una regla no es más que una asociación de patrones, los cuales pueden representarse en forma discreta por múltiples y diferentes protocolos. Una representación de una regla con antecedente y consecuente difuso, de la forma (A, B) puede expresarse en función de vectores de características que tengan por componentes las n ó p etiquetas lingüísticas sobre las que pueden evaluarse. Así mismo, cada componente del vector podría a su vez representarse como una cadena de bits, más o menos larga en función de la sutileza de las posibles apreciaciones de las descripciones. O sea, que 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 de $n \times j$ elementos y $p \times k$ respectivamente. En nuestro caso particular representaremos las etiquetas lingüísticas en función de la inclusión de sus α-cortes, por ejemplo, supongamos que pretendemos modelizar un sistema de frenado. La idea es detener un vehículo en un determinado punto de una línea. El conjunto de reglas que caracteri-

zan el sistema son las siguientes:

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.**

Si suponemos que sobre cada etiqueta se puede tomar un valor perteneciente al intervalo `[0,1]`, la codificación resultante para, por ejemplo, la componente Y_1 del conjunto de patrones de entrenamiento, 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.

Con esta representación, parece lógico el hecho de que ateniéndonos a las limitaciones de la BAM en cuestión de codificación y capacidad, podemos implementar sistemas difusos mediante una BAM, utilizando las técnicas suministradas por Wang, Cruz y Mulligan para garantizar la recuperación de los patrones suministrados en el entrenamiento. Asi pues tendremos la información difusa representada de forma discreta y podremos memorizarla mediante la matriz M construida mediante el producto exterior (1) (aprendizaje Hebbiano) de los patrones, en nuestro caso reglas, a memorizar. Dada la capacidad de almacenamiento de asociaciones de la BAM y sus aplicaciones para la eliminación de ruido y recuperación de información a partir de patrones parciales, podremos recuperar el consecuente a partir del antecedente y mediante un proceso inverso al de la codificación obtener el resultado de la aplicación de la regla o conjunto de reglas difusas correspondientes.

Una vez codificada la información según la Tabla I aplicamos aprendizaje no supervisado, bien sea producto exterior (1) o de otro tipo (codificación por pseudo-inversa (6) o variaciones del aprendizaje Hebbiano, por ejemplo (5)), sobre los *m* ejemplos, obteniendo de esta forma la M de nuestro sistema, al cual presentaremos 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 esto 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

0000000000 1100000000 1111111100 0000000000 0000000000
1111111000 1110000000 0000000000 0000000000 0000000000

al cual el sistema nos devolvería la correspondiente solución (en función del aprendizaje utilizado).

Las principales ventajas aportadas por el modelo son su simplicidad y su economía frente a otras memorias asociativas difusas tales como la BIOFAM

Tabla I

FUERZA		VELOCIDAD				
		NM	NB	CE	PM	PA
POSICION	NM			PM		
	NB		PB		CE	
	CE			CE		
	PB		CE		NB	
	PM			NM		

Es fácil de observar que el modelo de codificación es elemental, por lo cual y teniendo en cuenta que se monta sobre una BAM (elemento de memoria simple y suficientemente bien estudiado) su complejidad a nivel computacional es bastante reducida, con lo que conseguimos velocidades elevadas.

La otra ventaja citada es la economía a nivel de memoria requerida. Dado que en nuestro caso eliminamos el problema de tener que utilizar una matriz de dimensión n por p para cada una de las m FAM reglas, obteniendo un sistema con una única matriz M de dimensión por p.

El único problema que se presenta es el de la aparición de algunas respuestas algo confusas, en particular las que tienen sub-cadenas con unos fuera de lugar, por ejemplo:

1110001000

pero el problema no es tal, ya que se soluciona con facilidad, pasando los vectores descompuestos en los subvectores de 10 bits correspondientes a las etiquetas, por un elemento de filtrado (memoria asociativa entrenada en la α-inclusión) de forma que obtenemos la correspondiente corrección

1110000000.

References.

[1] Blanco, A.; (1993). *Identificación de Sistemas Mediante Redes Neuronales.* Tesis Doctoral. Universidad de Granada.

[2] Gelenbe, E.; (1992). *Generalised Associative Memory and the Computation of Membership Functions.* Neural Networks: Adv. and Applic., 2 (Eds. E. Gelenbe). Elsevier Science Publishers B. V. 129-140.

[3] Haines, K.; Hecht-Nielsen, R.; (1988). *A BAM with Increased Information Storage Capacity.* Proceedings of the IEE International Conference on Neural Networks ICNN-88, vol. I, 181 - 190.

[4] Hopfield, J. J.; (1982). *Neural Networks and Physical Systems with Emergent Collective Computational Abilities.* Proceedings of the National Academy of Sciences U.S.A., 79, 2554 - 2558.

[5] Kamp, Y., Hasler, M. *Recursive Neural Networks for Associative Memory* Wiley-Interscience series in systems and optimization. 1990.

[6] Kosko, B., (1986). *Fuzzy Knowledge Combination.* International Journal of Intelligent Systems, vol. 1, 293 - 320.

[7] Kosko, B., (1988). *Bidirectional Associative Memories.* IEEE Trans. on Systems, Man, and Cybernetics, SMC-18, 42 - 60.

[8] McEliece, R. J.; Posner, E. C.; Rodemich, E. R., Venkatesh, S S.; (1987). *The Capacity of the Hopfield Associative Memory* IEEE Trans. on Information Theory, vol. IT-33, 461 - 482.

[9] Mizumoto,M., Zimmermann, H. J.; (1982). *Comparison of Fuzzy Reasoning Methods.* Fuzzy Sets & Syst, 8, 253 - 283.

[10] Schneider, A., Sigillito, V. G.; (1991). *Two-Layer BAM.* Progress in Neural Networks, Vol. 1 (Eds. Omid M. Omidvar). Ablex publishing corp. 87 - 104.

[11] Wang, Y.-F., Cruz, J. B.; Mulligan, J. H.; (1990). *Two Coding Strategies for Bidirectional Associative Memory.* IEEE Transaction on Neural Networks, vol. 1, 81 - 92.

[12] Wang, Y.-F., Cruz, J. B.; Mulligan, J H.; (1991). *Guaranteed Recall of All Training Pairs for BAM.* IEEE Transaction on Neural Networks, vol. 2, 559 - 567.

[13] 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*, vol. 8, 301 - 357, *Part III*, vol. 9, 43 - 80.

[14] Zadeh, L. A., (1979). *A Theory of Approximate reasoning. Machine Intelligence 9 (J. E. Hayes, D. Michie, L. I. Mikulich, eds.) Elsevier, 149 -194.*

UN PROCEDIMIENTO EMPÍRICO PARA OBTENER REGLAS DIFUSAS USANDO REDES NEURONALES[1]

J.M. Benítez
A. Blanco
I. Requena

Dpto. Ciencias de la Computación e I.A.
Universidad de Granada
E.T.S. Ingeniería Informática. 18071
GRANADA
e-mail: requena@robinson.ugr.es

RESUMEN

Hemos realizado una implementación propia de la Máquina de Boltzmann (MB) para aprendizaje que utilizamos, entre otros problemas, para aprender el comportamiento de un sistema gobernado por reglas difusas, mediante una codificación en binario de las etiquetas de las variables que intervienen en las reglas.

Entrenando la MB con ejemplos sacados de las reglas, conseguimos comportamientos equivalentes. A partir de esta idea, se propone un procedimiento para obtener las reglas difusas de un sistema cuyo comportamiento se aprende a partir de ejemplos.

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

1. INTRODUCCIÓN.

Las Redes Neuronales Artificiales (RNA) son capaces de aprender y reproducir el comportamiento de un sistema, a partir de ejemplos en los que se conocen las entradas y las respectivas salidas al sistema, aun desconociendo las reglas o procesos que lo gobiernan.

Hoy día están de actualidad el control de sistemas, generalmente efectuado por expertos humanos, y los intentos de aplicar reglas difusas para ello.

En muchas ocasiones, sólo conocemos las situaciones o valores que corresponden a las variables de entrada al sistema, y las salidas correspondientes, es decir, las actuaciones a realizar para controlar el sistema, según cada situación concreta. En definitiva, conocemos muchos ejemplos del comportamiento del control del sistema.

La Máquina de Boltzmann (MB) es un modelo de RNA que se ha utilizado en problemas de Optimización (combinatoria), como clasificador y en problemas de aprendizaje (MBA), de acuerdo con la fijación de algunas neuronas de la red, y con la evolución o no de los pesos de las conexiones. Se utilizan ajustes estocásticos y es convergente asintóticamente en probabilidad [1]. Esto implica que tenemos que utilizar aproximaciones en tiempo finito, considerando distintos parámetros para el Enfriamiento, Selección y Aceptación cambios de estado de neuronas, parámetro de control (temperatura) y criterios de parada del algoritmo.

Hemos realizado implementaciones propias de la MB, para ser utilizada en las tres acepciones señaladas, y las hemos aplicado a distintos

[1] Este trabajo ha sido financiado en parte por el Proyecto PB92-0496 de la DGICYT M.E.C. Madrid.

problemas, con y sin información difusa. El comportamiento de la implementación sobre los problemas considerados ha sido excelente.

2. APRENDER UN SISTEMA DIFUSO.

Nos planteamos entonces, la posibilidad de usar la MBA para aprender el comportamiento de sistemas controlados por un conjunto de reglas difusas.

Para ello hemos considerado necesario lo siguiente:

- Las variables difusas que intervienen en las reglas toman valores según etiquetas (consideramos 7 etiquetas).
- Como la MB es un modelo de neuronas binarias, las etiquetas han de ser codificadas en binario para poder utilizarlas.
- Como conjunto de entrenamiento, usamos todos los casos posibles, que se obtienen de las reglas que manejan el sistema, y ya codificados.

Hemos utilizado esta idea sobre el clásico problema del péndulo invertido [7]. A partir de las 7 reglas que lo gobiernan obtenemos el conjunto de entrenamiento.

Las configuraciones de parámetros de la MBA (hemos usado 72 combinaciones distintas) han proporcionado diferentes resultados, pero hemos obtenido varios casos donde el aprendizaje es completo.

3. OBTENER LAS REGLAS DIFUSAS DE UN SISTEMA.

Nuestra implementación aprende bien el comportamiento del sistema a partir de las reglas. Entonces nos planteamos la posibilidad de obtener las reglas que lo controlan, si lo único que conocemos son ejemplos del comportamiento del sistema.

Lo primero que hemos comprobado, es que si entrenamos nuestra MBA con ejemplos del comportamiento del sistema, codificándolos en forma similar a las reglas, el aprendizaje obtenido es correcto y prácticamente equivalente al obtenido con las reglas directamente.

Pensamos entonces, que es posible obtener las reglas difusas que gobiernan un sistema, si disponemos de ejemplos del comportamiento del mismo, de forma suficiente para entrenar una MB adecuadamente. Para ello proponemos la siguiente metodología:

0. Suponemos que tenemos un conjunto de ejemplos del comportamiento del sistema, con los que hemos entrenado adecuadamente nuestra MBA (MBEJ).

   Suponemos también, que tenemos identificadas las variables que intervienen en el control del sistema, y que éstas toman como valores etiquetas. Si no es así, incluso podemos hacer una hipótesis previa sobre las mismas, para su evaluación.

1. Construimos todas las reglas posibles, según las etiquetas (valores) de las variables identificadas.

2. Entrenamos una RNA (MBA) con cada una de estas reglas, para comparar el comportamiento con la MBEJ. De acuerdo con lo indicado anteriormente, esto es equivalente a comprobar si cada regla se adapta o no a la red entrenada globalmente con los ejemplos (MBEJ).

   Descartamos aquellas reglas que claramente no se adaptan al sistema (MBEJ). Con esto tenemos una primera selección de reglas que posiblemente intervienen en el control del sistema.

3. Formamos ahora las combinaciones posibles (para $i = 1 .. N$) con estas reglas seleccionadas, y entrenamos una MBA para cada una de las combinaciones, que notamos MBCR(i).

4. Comparando el comportamiento de cada MBCR(i) con el de MBEJ, seleccionamos como conjunto de reglas definitivo las que

corresponden a la MBCR(i) que mejor se adapta a la MBEJ.

4. COMENTARIOS FINALES.

La metodología propuesta puede tener algunas dificultades a priori:

a) Si el número de variables es grande, tendremos muchas reglas posibles para analizar, y si en el paso 2 no se descartan una gran cantidad de ellas, los pasos 3 y 4 pueden complicarse bastante.

De todas formas, en la práctica, los resultados obtenidos en el paso 2, junto con la experiencia real de los propios expertos (que proporcionan los ejemplos de entrenamiento), pueden ayudar mucho a simplificar el proceso, y reducir el número de combinaciones posibles a comprobar empíricamente en el paso 4.

Además hemos comprobado que cuando una red ha sido bien entrenada, su funcionamiento en modo de utilización (recuperación de salidas a partir de entradas) es rapidísimo, por lo que se puede comprobar la adecuación (al comportamiento del sistema) de un número elevado de reglas en tiempos aceptables.

b) El hecho de que la MB sea un modelo neuronal binario, no plantea problemas para el aprendizaje cuando se hace a partir de las reglas conocidas codificadas. Cuando no es así, la codificación de los ejemplos del comportamiento para usarlos en el entrenamiento no es a priori fácil.

En la práctica, cada sistema concreto puede ser tener más o menos facilidad para realizar la codificación. En cualquier caso, la metodologái no está ligada al modelo de RNA comentado (la MB) y podemos utilizar otros modelos en su lugar, como el algoritmo de Backpropagation sobre redes multicapa feedforward, que permitan la utilización de valores continuos.

En este sentido hemos realizado experimentos utilizando un modelo de RNA de propagación hacia adelante con Backpropagation (8 neuronas de entrada, una capa oculta con 3 neuronas y 4 neuronas de salida) para el modelo del frenado de un coche [6]. Las etiquetas difusas se procesan como se indica en [3, 4], considerando 4 neuronas (un número difuso trapezoidal) en la capa de salida.

Los primeros resultados obtenidos se refieren a la evaluación de la concordancia entre la red entrenada con los ejemplos y la red entrenada con las reglas. Se han probado ejemplos de reglas sobre la red entrenada con ejemplos y ejemplos del comportamiento del sistema sobre la red entrenada con reglas. Los resultados obtenidos en las pruebas realizadas han sido excelentes, ya que en todas ellas las salidas calculadas por ambas redes ante las mismas entradas concuerdan en el 100% de los casos probados. Lo que constituye un fuerte argumento en favor de la metodología propuesta.

Se puede obtener también con esta metodología la combinación mínima de reglas que gobiernan adecuadamente el sistema. Este es el objetivo de los pasos 3 y 4.

BIBLIOGRAFÍA.

[1] AARTS, E.H.L. AND KORST, J.H.M. *Simulated Annealing and Boltzmann Machines.* Wiley & Sons. Chichester. (1990).

[2] ZADEH, L.A. The Concept of a Linguistic Variable and its Applications to Approximate Reasoning. Part I, II and III. *Information Sciences* (1975), Vol 8, pp. 199-249, pp. 301-357 and vol. 9, pp. 43-80.

[3] REQUENA, I., DELGADO, M., AND VERDEGAY, J.L. Automatic Ranking of Fuzzy Numbers with the Citeria of a Decision-

maker Learnt by an Artificial Neural Network. To appear in *Fuzzy Sets and Systems*.

[4] REQUENA, I., DELGADO, M., AND VERDEGAY, J.L. Un Índice Personal de Decisores para Números Difusos basado en Redes Neuronales Artificiales. *Actas III Congreso Español de Tecn. y Lógica Fuzzy*. Santiago de Compostela. (1993), 81-88.

[5] YAMAOKA, M. AND MUKAIDONO, M. A Learning Method of Fuzzy Inference Rules with a Neural Network. *Fourth IFSA Congress*, Engineering (1991), 222-225.

[6] WOLF, T. Das Fuzzy-Mobil. *mc 3/91*, (1991) pg. 50-63.

[7] YAMAKAWA, T. Stabilization of a Inverted Pendulum by a High-Speed Fuzzy Logic Controller Hardware System. *Fuzzy Sets and Systems, 32*, (1989), 161-180.

DEMOSTRACIONES DE SOFTWARE

FuzzyControlDesk

Rainer Albersmann, Rudolf Felix

Joseph-von-Fraunhofer-Str. 20
44227 Dortmund, Germany
Tel.: +Ger 231 9700 921, Fax: +Ger 231 9700 929

Abstract

The features of the software tool FuzzyControlDesk are described and applikation examples are given.

FuzzyControlDesk is a powerful tool for developing fuzzy controllers. It provides for a broad spectrum of engineering features and is independent of any additional hard- and software.

FuzzyControlDesk generates a fuzzy controller in C-code. The user specifies the controller either in the rule-oriented language FuzzyTalk or by using a graphical user interface (see Fig. 1).

FuzzyControlDesk generates both the ANSI and the Kernighan & Ritchie C-standard code. Using an appropriate C-compiler any target architecture, for example 80C51 or 80C166 micro controller, can be supported. Test functions can be generated optionally for both tuning and error detection.

Special requirements regarding the performance and memory space are met for example by automatical renunciation of float variables. The inference methods to be used are the max-min and the max-prod inference, which can be chosen by the user. For the defuzzification the centre of gravity method is applied. If required, additional defuzzification methods can be provided.

Key features of FuzzyControlDesk:

- textual or graphical specification of the controller
- table-oriented representation of linguistic variables and membership functions
- function-oriented representation of linguistic variables and membership functions
- max-min inference
- time-optimized max-prod inference
- centre of gravity defuzzification
- optimized code generation for 8 bit micro controller
- memory-optimized integer arithmetic
- generation of ANSI and Kernighan & Ritchie C-standard
- generation of test functions
- C-code integrability by automatically generated header files
- on-line help functions
- DIF interface to standard tools (for example MS-Excel or Lotus, see Fig. 2)
- MS-Windows graphical user interface
- flexible simulation of the controller

FuzzyControlDesk has been successfully applied in the following fields:

- heating and air conditioning technics
- testing technics in the automotive industry
- controling of processes in the chemical industry

Demonstration objects developped with **FuzzyControlDesk:**

- filling plant
- inverted pendulum

FuzzyControlDesk is available for IBM compatible PCs with Microsoft Windows 3.1. If required **FuzzyControlDesk** can be adapted to any hardware or be integrated as a part of a problem specific system solution.

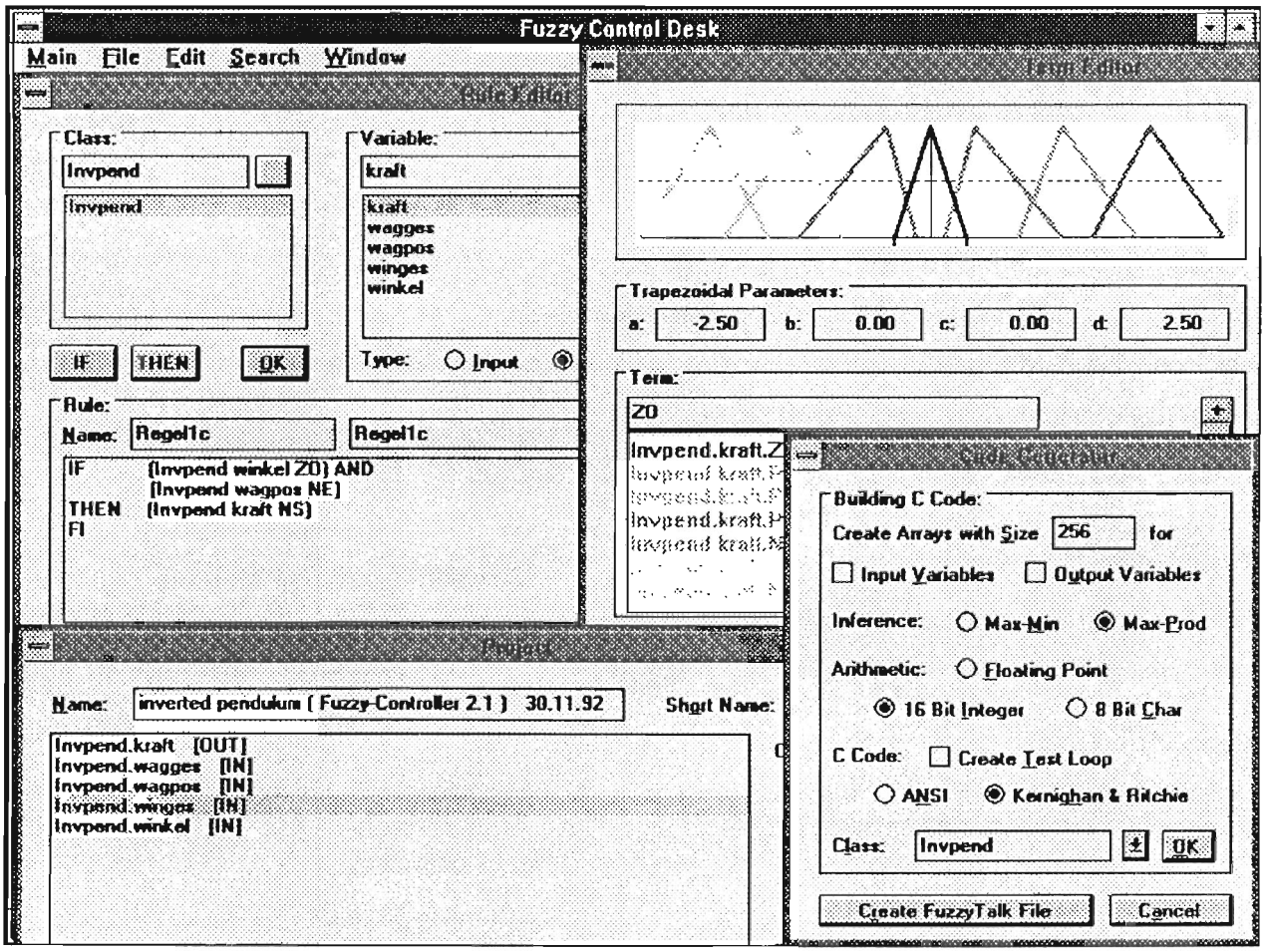


Fig. 1: Graphical User Interface of **FuzzyControlDesk.**

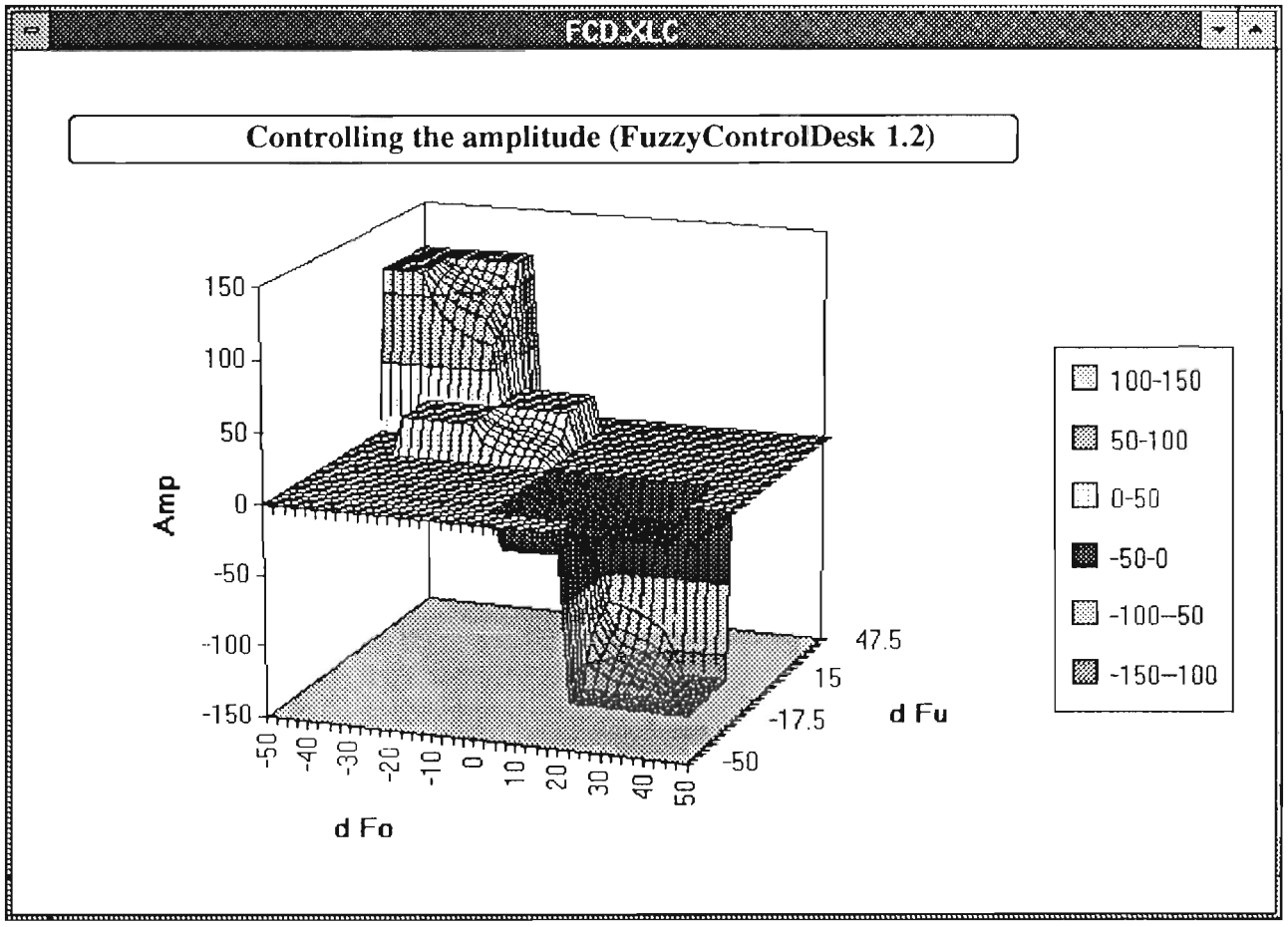


Fig. 2: Controllers input / output visualisation.

FuzzyDecisionDesk

Rainer Albersmann, Rudolf Felix

Fuzzy Demonstrations-Zentrum Dortmund
Joseph-von-Fraunhofer-Str. 20
44227 Dortmund, Germany
Tel.: +Ger 231 9700 921, Fax: +Ger 231 9700 929

Abstract

The features of the software tool FuzzyDecisionDesk are described and applikation examples are given.

In case of optimization of industrial tasks, fuzzy logic has proved itself to be an efficient method of solving planning and classification problems.

The tool **FuzzyDecisionDesk** is based on a universal problem independent approach. The tool can be used for solving complex optimization problems in both technical and management oriented applications.

Through the orientation on both qualitatively and quantitatively specified goals (see Fig. 1) and their relationships **FuzzyDecisionDesk** provides for analysis and solutions easy to understand even in case of very complex optimization problems.

The optimization algorithm of **FuzzyDecisionDesk** refers to a powerful model of the optimization process. The model is based on fuzzy estimates of the impact of optimization and decision alternatives (short actions) on the goals and their priorities. **FuzzyDecisionDesk** first analyzes the consequences of each action and derives relationships (see Fig. 2) between the goals (for instance competition or cooperation). Based on the relationships either a single optimization alternative or a fuzzy set of optimization alternatives is determined.

FuzzyDecisionDesk has been successfully applied in the following fields:

- process analysis
- forming technology (beam bending)
- VLSI design
- production planning
- operations research (OR)
- image recognition
- management decision making
- optimization problems

FuzzyDecisionDesk is available for SUN compatible workstations with OS Release 4.1.2 and Motif-user interface and for IBM compatible PCs with Microsoft Windows 3.1. If required **FuzzyDecisionDesk** can be adapted to any hardware or be integrated as a part of a problem specific system solution.

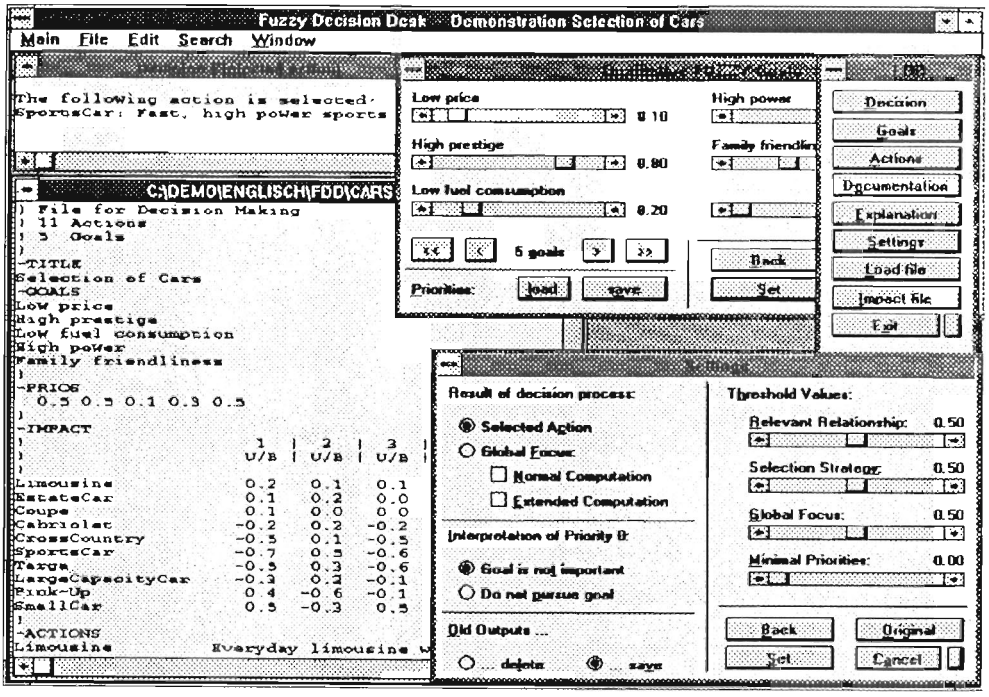


Fig. 1: Selection of the best car with **FuzzyDecisionDesk.**

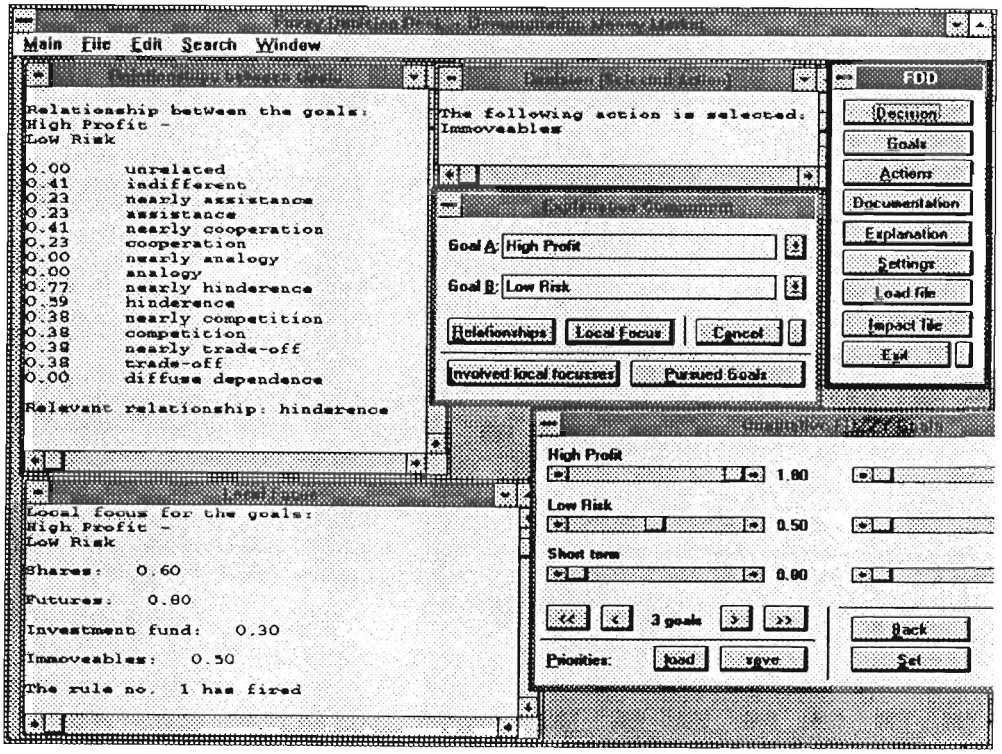


Fig. 2: Graphical user interface of **FuzzyDecisionDesk.**

Milord II

Josep Puyol Gruart, Carles Sierra

Institut d'Investigació en Intel·ligència Artificial (*IIIA*)
Camí de Santa Bàrbara, 17300 Blanes (Girona)
e-mail: {puyol, sierra}@ceab.es

Abstract

Milord II is a programming language and a tool for the development of expert systems. **Milord II** has been designed and developed at the *IIIA* and it works as a testbed for the implementation of the theoretical developments of our group. Several applications from medicine to biology, pedagogy and pig farms are being developed. **Milord II** is a free prototype available for research purposes. This is a brief summary of the main concepts of **Milord II** in order to the understanding of a software demonstration.

1 Introduction

The main goal of **Milord II** is the programming of real world expert systems, that is, those dealing with real problems and programmed by real experts. Programming in the large and imperfect information treatment are one of the most important characteristics of **Milord II**.

2 A Modular Language

The decomposition of a whole problem into simpler parts is a good and natural programming methodology. **Milord II** is then a homogeneous language based on modules. A **Milord II** program is a hierarchy of modules.

Modules are the basic unit of programming in **Milord II**. All the other components of expert systems as facts or rules belong to modules. Every module can be considered as a specialist on a concrete domain, that is, every module is a complete expert system containing both the domain and control knowledge. In the following we will briefly describe the main components of modules using the simple example of Figure 1.

2.1 Interface

The interface of a module has two components: the *import* and the *export* interface. The export interface of a module is its output result. It represents the facts the module is able to deduce. For instance the module *Gram* exports a set of germs. The import interface of a module is the set of facts provided by the user (*Penicillin* in the module *Gram*).

Modules can also import information from other modules via their declaration as submodules. The module *Gram* has three submodules (*Respiratory Diagnosis*, *Type of Infection* and *Gram of Sputum*). It can import the fact *DCGP* exported by the module *Gram of Sputum* by means of the path *S/DCGP*[1].

```
Module Gram =
    Begin
    Module D= Respiratory_Diagnosis
    Module T= Type_of_Infection
    Module S= Gram_of_Sputum
    Import Penicillin
    Export Pneumococcus, Haemophilus, Staphylococcus, Enterobacteria
    Deductive knowledge
        Dictionary: ; not defined here.
        Rules:
        R001 If S/DCGP then conclude Pneumococcus is possible
        R002 If S/DCGP and D/Bact_Pneumonia
             then conclude Pneumococcus is very_possible
        R003 If S/BGN and D/Aspiration_Pn and T/Nosocomial
             then conclude Enterobacteria is quite_possible
        R004 If S/CBGN and Penicillin
             then conclude Haemophilus is sure
        Inference system:
        Truth values= (impossible, few_possible, sligh_possible, possible,
             quite_possible, very_possible, sure)
        Renaming
             D/False ==> impossible
             D/True ==> sure
             T/False ==> impossible
             T/True ==> sure
             ...
        Connectives:
             Conjunction =
                  ((impossible impossible impossible impossible,
                       impossible impossible impossible)
                  ...
                  (impossible few_possible sligh_possible possible
                       quite_possible very_possible sure))
        end deductive
    Control knowledge
        Evaluation Type: Lazy

        end control
    end
```

Figure 1: Example of module declaration.

2.2 Deductive Knowledge

Milord II deals with imperfect knowledge, that is, uncertain, imprecise and vague knowledge. The use of a symbolic multi-valued logics allows us to give to the concepts a graduated truth value. For instance, we can say that *fever is possible*.

The deductive knowledge is composed by the declaration of the dictionary, the rules and the inference system of the module.

Dictionary: It defines the fact identifiers and some of their attributes. The most important attribute of

[1] *S* is the local name of the submodule *Gram of Sputum*. Local names are defined by **Module** *local_name* = *module*.

facts is their type. There are different types of facts: boolean, logic, enumerated and numeric facts[2]. The value of a logic fact is an interval of truth–values. In Figure 2 there is an example of a numeric fact declaration. Enumerated facts are fuzzy sets.

```
Dictionary:
    Predicates:
    ...
    Temperature =
        Name: "Patient's Temperature (in centigrade)"
        Question: "Which is the patient's Temperature?"
        Type: Numeric
        Relation: less-relevant-than AIDS
    ...
;; within the Temperature predicate definition the next
;; meta-predicate instance is present:
;; less-relevant-than(Temperature, AIDS)
```

Figure 2: Example of predicate declaration.

Rules: This component of modules represent the relational knowledge. For instance:

> **If** temperature $>$ 39' **then conclude** fever **is** definite

Inference System: It is the declaration of the local logic of a module. It consists of the declaration of a total ordered set of linguistic terms and the connective operators used to combine and propagate the truth- values when making inference. In the example of Figure 1 we can see a declaration of a logic with seven terms and the conjunction operator.

The inference system declaration of a module also includes the mapping (translation) of the terms of the different logics of its submodules to the own logic[3].

In the example of Figure 1 the renaming declaration D/False ==> *impossible* means that the term *False* from the module *D* is translated to the local term *impossible*.

2.3 Control Knowledge

The current implementation of the meta-language allows the definition of meta rules, and the type of module execution, which can be lazy or eager. *Lazy* means that facts are evaluated, at the object-level, only when needed, i.e. imported facts and exported facts of submodules are asked only if they may be useful to compute the export interface of the module. On the contrary an *eager* module execution obtains, first of all, values for the imported facts and for the exported facts of the submodules and then the deductive knowledge is used. The interaction between the object level (rules) and the meta level (metarules), reification/reflection step, is made after every import information is obtained, after every submodule execution and after every time object level is extended by deduction.

An example of meta–rule:

M001 **If** K(X/$y.$v) then conclude K($y.$v)

Given any fact of the submodule *X (X/y)* with any value *v*, then we give to the fact *y* of the current module the same value *v*.

3 The Tool

Milord II is composed by two programs, the compiler and the interpreter. These programs have been implemented using C and Common Lisp respectively. Now we have only a Macintosh version because of the dependency of the window system to the machine.

The operation on **Milord II** interpreter can be summarized as follows:

1. The list of modules or a graphical representation is presented. The user can choose the starting point by selecting a module. Notice that it is not necessary to start on the root module of the application.
2. The list of the exported facts of the module is presented. The user can ask a module for the value of a fact or of a set of facts.
3. The module will generate questions to the user and to its submodules. The local control strategy of every module executed will determine which is the information needed to reach the goals.
4. Finally the answer or conditioned answers[4] to the initial questions are presented to the user.

4 Applications

We are developing several expert system applications in different domains.

Terap–IA: It is a medical application for pneumonia treatment. It is the natural extension of a previous expert system named *Pneumon- IA* for pneumonia diagnosis.

Ens–AI: It is an intelligent tutorial system directed to the diagnosis and orientation assistance in pedagogical processes.

Spong -IA: It is an expert system for marine sponge classification.

Porc -IA: It is an expert system for the supervision of a pig farm. It is an example of the use of temporal reasoning in **Milord II**.

5 Future Work

We are working on an intelligent editor for **Milord II**. We are also working in the connection of **Milord II** with databases and its distribution on a Mac network.

[2]For the sake of simplicity we avoid to explain *temporal* facts that allows to implement temporal reasoning and facts of type *array* used to implement bayesian reasoning.

[3]These translations of linguistic terms should preserve the deduction among modules with some criteria.

[4]The inference engine of **Milord II** works by *specialization*. The conditioned answers are obtained by the specialization (partial deduction) of the object level of modules.

Índice de autores