

Aplicaciones de la Lógica Difusa

Oscar G. Duarte V.*

RESUMEN

El propósito de este artículo es el de presentar algunas de las aplicaciones conocidas de la Lógica Difusa, es por tanto un artículo de carácter divulgativo para no expertos en el tema. Los ejemplos seleccionados son un Controlador Difuso Adaptativo, un Identificador de Imágenes Aéreas que emplea un algoritmo de agrupamiento difuso, una Base de Datos Difusa, y un experimento de psicología cognocitiva inspirado en la lógica difusa. Todos estos ejemplos han sido tomados de la literatura técnica.

INTRODUCCIÓN

Pese a la sencillez conceptual y algorítmica de la Lógica Difusa[4,6,9,10,11] y a que los campos en que se ha aplicado con éxito son múltiples y variados [1,8,12], aún en el medio académico colombiano el término "*Lógica Difusa*" tiene un halo de misterio injustificado que ha dificultado su ingreso en ciertas áreas del conocimiento¹.

El propósito de este artículo es el de recoger unas pocas aplicaciones conocidas en la literatura especializada basadas en distintas estrategias de lógica difusa. Se han seleccionado aplicaciones de áreas muy diversas, buscando así resaltar que la Lógica Difusa puede entenderse como una herramienta matemática cuya amplia aplicabilidad se basa en la concepción de conjuntos con fronteras no exactas [4,7].

Se presentarán en este artículo un Controlador Difuso Adaptativo, un Identificador de Imágenes Aéreas que emplea un algoritmo de agrupamiento difuso, una Base de Datos Difusa, y un experimento de psicología cognocitiva inspirado en la lógica difusa. Todos estos ejemplos han sido tomados de la literatura técnica, específicamente de [16,15,12 y 13] respectivamente.

I. CONTROLADOR DIFUSO ADAPTATIVO

Quizás la aplicación en que la Lógica Difusa ha conseguido un éxito mayor, y por ende un mayor número de seguidores, se encuentra en el Control Industrial. Aún cuando existen numerosas versiones de controladores que emplean lógica

difusa, suele asignarse el término *Controlador Difuso* a un sistema de control cuya estructura interna corresponde a la de la figura ²1.

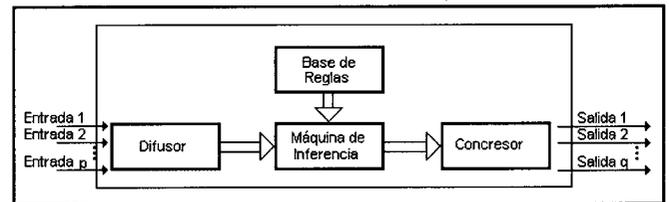


Figura 1 Estructura interna de un Controlador Difuso

Una de las ventajas que suelen mencionarse de los Controladores Difusos, frente a otro tipo de controladores, es que pueden diseñarse aunque no se tenga un modelo matemático exacto de la Planta a controlar, gracias a que están basados en reglas. Esta situación, sin embargo, no es del todo sencilla: el no tener un modelo matemático de la Planta implica no poder realizar simulaciones sobre la misma, de tal manera que los ajustes del controlador deben realizarse en línea.

Este hecho ha promovido la aparición de controladores autoajustables, que cuentan con algún algoritmo que les permite evaluar su desempeño, y de acuerdo con cierto criterio ajustar su diseño. Estos algoritmos de autoajuste (también conocidos como algoritmos de autosintonía) son muy útiles también en situaciones en las que la Planta a controlar varía en el tiempo, lo que implica que el controlador deba ajustarse a dichos cambios.

Cuando un Controlador Difuso cuenta con un algoritmo de autoajuste, se dice que es un *Controlador Difuso Adaptativo*. Sin embargo, existen Controladores con otros algoritmos que no son exactamente de autoajuste, y que podríamos llamar de entrenamiento, que algunos autores denominan también Controladores Difusos Adaptativos.

*Ingeniero Electricista M.Sc., Departamento de Ingeniería Eléctrica, Universidad Nacional de Colombia.

¹ El área en que ha penetrado con más fuerza es en la Automatización Industrial[1,5], indudablemente debido a la enorme fuerza que tiene a nivel internacional.

² Para una explicación sobre el funcionamiento de Sistemas con esta estructura, véase [4,6,11]

Los algoritmos de entrenamiento permiten ajustar el diseño del Controlador para que tenga un comportamiento deseado, pero fuera de línea. En general estos algoritmos no permiten la adaptación del controlador a cambios de la planta, pero son muy útiles para diseñar controladores en los que se combina información numérica con información lingüística [16,20].

A continuación se presenta un Controlador Difuso Adaptativo desarrollado por Wang & Mendel [20] que cuenta con un algoritmo de entrenamiento. Se ha seleccionado este controlador, porque el algoritmo empleado es uno de los más sencillos conceptualmente, y de mayor simplicidad algorítmica, tal como se presenta en el numeral siguiente.

A. ALGORITMO DE ENTRENAMIENTO DE WANG Y MENDEL

Este algoritmo parte de un cuadro que describe cuáles deben ser las salidas concretas, cuando se especifican las entradas concretas, es decir, de cuadros como la siguiente :

Cuadro 1. Parejas de Entrenamiento Entrada-Salida

Entrada 1	Entrada 2	...	Entrada p	Salida 1	Salida 2	...	Salida q
x_{11}	x_{21}	...	x_{p1}	y_{11}	y_{21}	...	y_{q1}
x_{12}	x_{22}	...	x_{p2}	y_{12}	y_{22}	...	y_{q2}
...
x_{1w}	x_{2w}	...	x_{pw}	y_{1w}	y_{2w}	...	y_{qw}

El cuadro anterior puede interpretarse como sigue :

“Se desea que el Sistema de Lógica Difusa responda con las salidas concretas $y_{11}, y_{21}, \dots, y_{q1}$, cuando las entradas concretas son $x_{11}, x_{21}, \dots, x_{p1}$; que responda con las salidas concretas $y_{12}, y_{22}, \dots, y_{q2}$, cuando las entradas concretas son $x_{12}, x_{22}, \dots, x_{p2}$; que responda con las salidas concretas $y_{1w}, y_{2w}, \dots, y_{qw}$, cuando las entradas concretas son $x_{1w}, x_{2w}, \dots, x_{pw}$ ”

Los w casos que se quieren obtener con el Sistema de Lógica Difusa pueden ser insuficientes para determinar completamente el diseño, o por el contrario, pueden ser inconsistentes entre sí.

El algoritmo de Wang & Mendel permite determinar la base de reglas a partir de la tabla de patrones de entrenamiento; el diseñador debe seleccionar los demás parámetros de Sistema de Lógica Difusa. El procedimiento es el siguiente para cada uno de los w casos:

1. Determinar los grados de pertenencia de $x_{1k}, x_{2k}, \dots, x_{pk}, y_{1k}, y_{2k}, \dots, y_{qk}$; a cada uno de los Valores Lingüísticos de las respectivas Variables Lingüísticas.
2. Seleccionar los Valores Lingüísticos $Lx_{1k}, Lx_{2k}, \dots, Lx_{pk}, Ly_{1k}, Ly_{2k}, \dots, Ly_{qk}$, para los cuales los grados de pertenencia respectivos son máximos.
3. Crear una regla de la forma IF Entrada 1 es Lx_{1k} AND Entrada 2 es Lx_{2k} AND ... AND Entrada p es Lx_{pk} THEN Salida 1 es Ly_{1k} AND Salida 2 es Ly_{2k} AND ... AND Salida q es Ly_{qk} .

4. Asignar a la regla anterior un factor de certeza, calculado como el producto de los grados de pertenencia a cada Valor Lingüístico.
5. Verificar si en la Base de Reglas existe ya una regla con el mismo antecedente (y quizás distinto consecuente); de ser así, dejar en la Base aquella que tenga un mayor factor de certeza. Si aún no hay en la Base de Reglas una de estas con el mismo antecedente, adicionar la nueva regla a la Base.
6. Complementar la Base de Reglas con la información lingüística disponible (si existe alguna).

• EJEMPLO 1

Presentamos aquí el ejemplo de un controlador aplicado al problema de llevar un vehículo con marcha hacia atrás a velocidad constante para que busque una cierta línea recta y la siga. El problema ha sido desarrollado por Wang en [16].

La figura 2 muestra gráficamente el problema. El vehículo está inicialmente ubicado a una distancia x de la línea, y formando un ángulo A con la recta normal a la línea. Se necesita diseñar un Controlador Difuso para que decida cuál es el ángulo de giro B que deben tener las ruedas del vehículo.

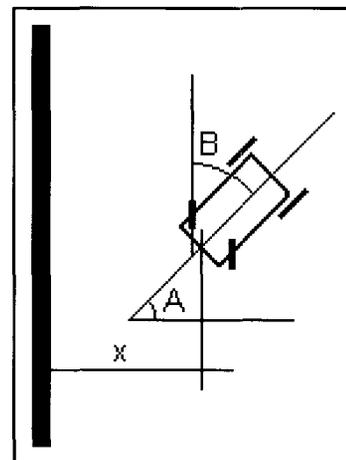


Figura 2. Planteamiento del problema

Wang encuentra una Tabla de Parejas Entrada-Salida (como el cuadro 1) a partir de su propia experiencia como conductor, y con ella entrena un Controlador Difuso que tiene 5 valores lingüísticos para la distancia x , 7 para el ángulo A , y 7 para el ángulo de salida B . Los resultados los compara con los obtenidos por Nguyen & Widrow con redes neuronales a partir de la misma Tabla Entrada-Salida, y las trayectorias seguidas por el vehículo son semejantes.

Wang también efectúa el entrenamiento con información numérica incompleta, es decir, considerando sólo una parte de la Tabla de entrenamiento; en estas condiciones el Controlador no es capaz de cumplir su objetivo. Sin embargo, al emplear

algunas reglas If-Then extraídas de su experiencia como conductor, Wang obtiene resultados que son iguales a los del primer caso.

Lo interesante de este último diseño es que allí se han combinado dos tipos de información de naturaleza diferente: por una parte está la información numérica proveniente de la Tabla, y por otra parte está la información lingüística contenida en las reglas If-Then.

II. IDENTIFICADOR DE IMÁGENES AÉREAS

El problema aquí planteado es el siguiente: Se tienen tres imágenes aéreas de la misma zona; las imágenes se han captado empleando cámaras a blanco y negro, de tal forma que muestran en cada pixel un cierto nivel de gris (nivel de luminancia); las tres fotografías no son iguales, porque cada una de ellas se ha tomado anteponiendo a la cámara un filtro que sólo permite captar una franja de colores (unas ciertas longitudes de onda), y los filtros para cada fotografía han sido diferentes.

En estas imágenes se ha captado una área extensa con zonas que se pueden clasificar así:

- Zonas de río.
- Zonas de construcciones humanas.
- Zonas dedicadas a la agricultura.
- Zonas boscosas.

El problema consiste en diseñar un algoritmo que, conociendo los niveles de luminancia para un cierto pixel en las tres fotografías, decida a cuál de las zonas anteriores corresponde ese pixel.

No existe una única combinación de luminancias que identifique a cada una de las zonas (no hay un único «verde» en la zona de bosques), y por tanto no se conoce el conjunto de todas las combinaciones de luminancias posibles asociadas a cada zona.

Nótese que el problema puede replantearse en términos de Conjuntos Difusos: Se puede definir como Universo de Discurso el conjunto de todas las combinaciones posibles de luminancias, con lo cual la tarea consiste en encontrar cuatro Conjuntos Difusos definidos sobre dicho universo, cada uno de los cuales debe representar a una de las zonas de las imágenes.

Cómo encontrar esos conjuntos? Trivedi [15] propone la utilización de un algoritmo de agrupamiento difuso, cuyos principios generales se exponen a continuación.

A. AGRUPAMIENTO "FUZZY C-MEANS"

Dado un Universo de Discurso \mathbf{X} , se define una c -partición como una colección de c Conjuntos Difusos definidos sobre \mathbf{X} , y con funciones de pertenencia

$$u_1(x), u_2(x), \dots, u_c(x)$$

siempre y cuando los Conjuntos Difusos cumplan con las siguientes características:

- Para todos los conjuntos debe existir al menos un elemento de \mathbf{X} tal que su función de pertenencia al conjunto sea distinta de cero.
- Para todos los elementos de \mathbf{X} , la suma de sus funciones de pertenencia a los c Conjuntos debe ser 1.

Bezdek[3] desarrolló un algoritmo para obtener una c -partición para un Universo de Discurso finito; dicho algoritmo se conoce como el *agrupamiento "fuzzy c-means"*, y es uno de los más populares, pese a que en la literatura se reportan muchos otros algoritmos distintos [2].

Bezdek parte de la existencia de un Universo de Discurso \mathbf{X} compuesto por n elementos:

$$\mathbf{X} = [x_1 x_2 \dots x_n]$$

donde cada uno de los n x_i es una d -upla de elementos

$$\begin{aligned} x_1 &= [x_{11} x_{21} \dots x_{d1}]^T \\ x_2 &= [x_{12} x_{22} \dots x_{d2}]^T \\ &\dots \\ x_n &= [x_{1n} x_{2n} \dots x_{dn}]^T \end{aligned}$$

Para encontrar los c Conjuntos de la partición, Bezdek define para cada uno de ellos un *centro* v_i , que identifique al conjunto, y define a \mathbf{V} como la colección de centros de dichos Conjuntos:

$$\mathbf{V} = [v_1 v_2 \dots v_c]$$

$$\begin{aligned} v_1 &= [v_{11} v_{21} \dots v_{d1}]^T \\ v_2 &= [v_{12} v_{22} \dots v_{d2}]^T \\ &\dots \\ v_c &= [v_{1c} v_{2c} \dots v_{dc}]^T \end{aligned}$$

También define la matriz \mathbf{U} como la matriz de orden $c \times n$ que contiene las funciones de pertenencia de cada uno de los n x_i casos a los c Conjuntos difusos identificados por los v_i centros.

$$\mathbf{U} = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ u_{c1} & u_{c2} & \dots & u_{cn} \end{pmatrix}$$

Como los c conjuntos son una c -partición de \mathbf{X} , debe cumplirse que:

$$\sum_{k=1}^n u_{ik} > 0; \forall i, i = 1, 2, \dots, c$$

$$\sum_{i=1}^c u_{ik} = 1; \forall k, k = 1, 2, \dots, n$$

Con las definiciones anteriores, el problema del agrupamiento de \mathbf{X} en c grupos puede formularse como la búsqueda de la pareja (\mathbf{U}, \mathbf{V}) que clasifique "mejor" los datos de \mathbf{X} . Para medir qué tan bien está hecha una clasificación, Bezdek emplea la función $J_m(\mathbf{U}, \mathbf{V})$

$$J_m(\mathbf{U}, \mathbf{V}) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^m \|x_k - v_i\|^2$$

en donde m es un real mayor que uno, y $\|\cdot\|$ es una norma diferenciable. Bezdek demuestra que la pareja $(\mathbf{U}^*, \mathbf{V}^*)$ que optimiza J_m debe satisfacer:

$$u_{ik}^* = \left(\sum_{j=1}^c \left(\frac{\|x_k - v_j^*\|}{\|x_k - v_i^*\|} \right)^{2/m} \right)^{-1}; \forall i, k \quad (1)$$

$$v_i^* = \frac{\sum_{k=1}^n (u_{ik}^*)^m x_k}{\sum_{k=1}^n (u_{ik}^*)^m}; \forall i \quad (2)$$

Con lo anterior puede plantearse un algoritmo iterativo para la búsqueda de la pareja $(\mathbf{U}^*, \mathbf{V}^*)$ así:

- 1) •Fijar c , un entero mayor que 1 y menor o igual que n .
 - Fijar m un real mayor que 1.
 - Seleccionar $\|\cdot\|$ una métrica inducida por producto interno para medir las distancias entre los miembros de \mathbf{X} y los miembros de \mathbf{V} .
 - Inicializar \mathbf{U}^0 .

$p=1$
- 2) Calcular \mathbf{V}^p empleando (1)
- 3) Calcular \mathbf{U}^p empleando (2)
- 4) Comparar \mathbf{U}^p con \mathbf{U}^{p-1} mediante alguna norma matricial. Si $\|\mathbf{V}^p - \mathbf{V}^{p-1}\|$ es menor que algún valor de tolerancia entonces terminar, de lo contrario $p=p+1$ e ir al paso 2.

El algoritmo anterior es el conocido como *agrupamiento fuzzy c-means*. Para ilustrar qué es lo que hace este algoritmo, se han desarrollado los ejemplos 2, 3, 4 y 5:

•Ejemplo 2

Se desea agrupar los números enteros del 1 al 20 en 3 conjuntos difusos empleando agrupamiento *fuzzy c-means*.

En este ejemplo $\mathbf{X} = [1 \ 2 \ 3 \ \dots \ 19 \ 20]$, con lo que

$$d=1$$

$$x_1=[1]; x_2=[2]; x_3=[3]; \dots; x_{19}=[19]; x_{20}=[20];$$

$$c=3$$

Las figuras 3 a 8 muestran como evolucionan los tres conjuntos difusos en cada una de las iteraciones. La partición inicial es aleatoria, y se ha escogido $m=1,5$.

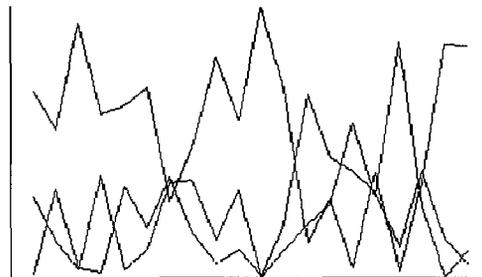


Figura 3. 3-partición para p=1



Figura 4. 3-partición para p=2

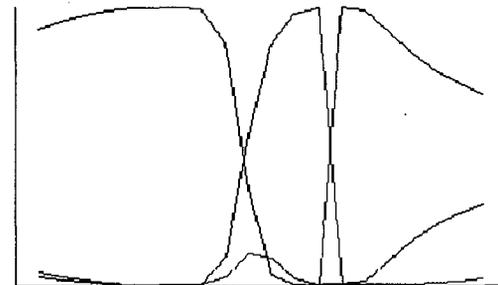


Figura 5. 3-partición para p=3

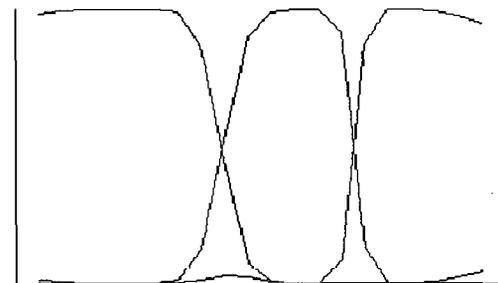


Figura 6. 3-partición para p=4

3 En las figuras 3 a 18 el eje horizontal corresponde al intervalo real [0,20], y el eje vertical a las funciones de pertenencia de los conjuntos definidos por el algoritmo *fuzzy c-means*

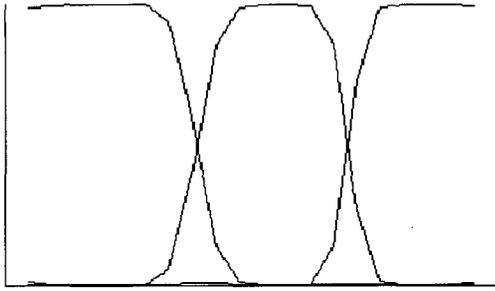


Figura 7. 3-partición para $p=5$

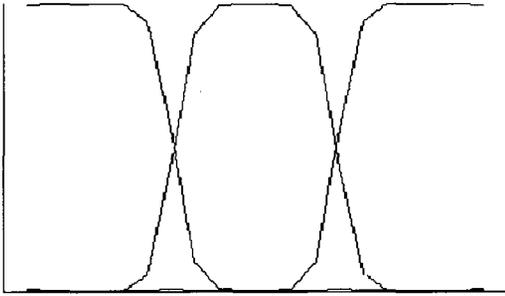


Figura 8. 3-partición para $p=27$

•EJEMPLO 3

Se desea conocer el efecto de cambiar el parámetro m en la partición del ejemplo 2.

Las figuras 9 a 14 muestran los resultados finales de las particiones, cuando se selecciona $m_1=1,2$, $m_2=1,5$, $m_3=2$, $m_4=3$, $m_5=5$, $m_6=10$. Nótese que aún cuando la forma de los conjuntos difusos varía fuertemente, el centro de éstos no.

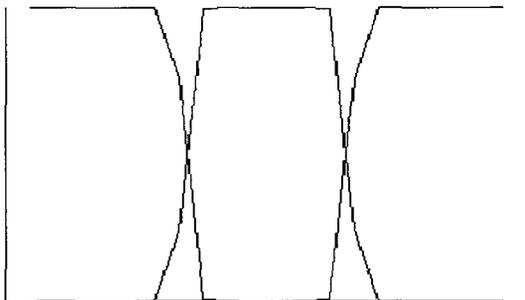


Figura 9. 3-partición para $m=1,2$

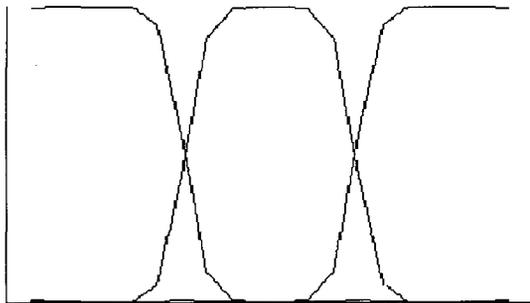


Figura 10. 3-partición para $m=1,5$

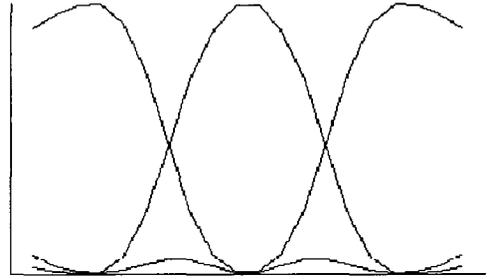


Figura 11. 3-partición para $m=2$

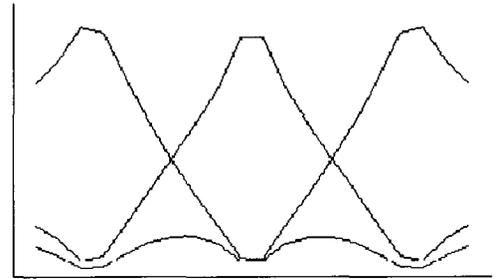


Figura 12. 3-partición para $m=3$

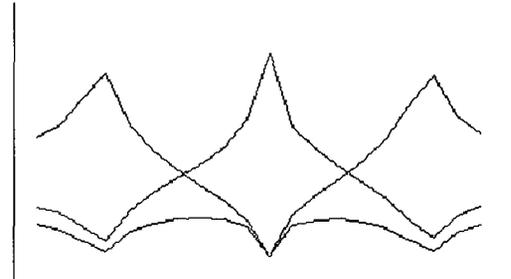


Figura 13. 3-partición para $m=5$

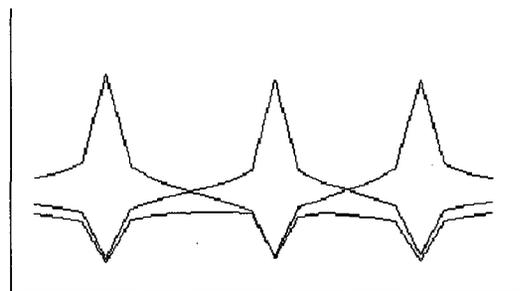


Figura 14. 3-partición para $m=10$

•EJEMPLO 4

Se desea conocer el efecto de cambiar el número de conjuntos difusos de agrupamiento, en el ejemplo 2.

Las figuras 15 a 18 muestran los resultados finales de las particiones, para $c_1=2$, $c_2=3$, $c_3=5$, $c_4=10$. En todos los casos se ha seleccionado $m=2$

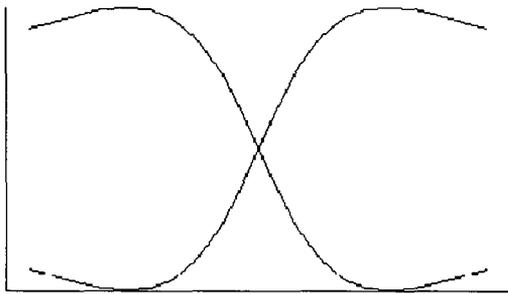


Figura 15. 2-partición

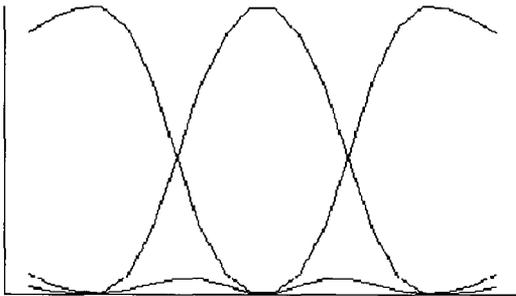


Figura 16. 3-partición

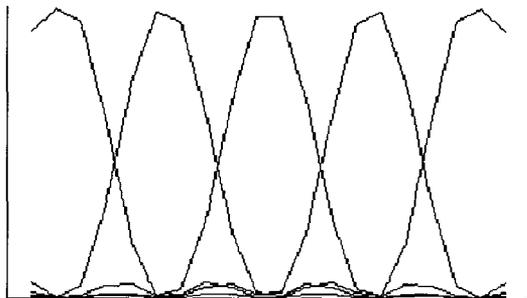


Figura 17. 5-partición

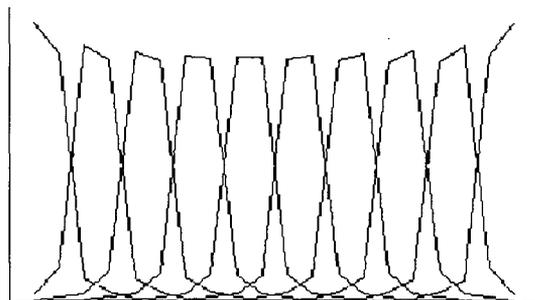


Figura 18. 10-partición

•Ejemplo 5

Dado el problema del Identificador de Imágenes Aéreas anteriormente presentado, se desea saber cómo debe plantearse el problema para emplear el agrupamiento *fuzzy c-means*, si cada una de las tres imágenes tiene 32*32=1.024 pixels.

Los tres grupos de 1.024 pixels pueden organizarse en 1.024 3-uplas así:

$$X = [x_1 \ x_2 \ x_3 \ \dots \ x_{1023} \ x_{1024}]$$

$$x_1 = [x_{1,1} \ x_{2,1} \ x_{3,1}]$$

$$x_2 = [x_{1,2} \ x_{2,2} \ x_{3,2}]$$

...

$$x_{1024} = [x_{1,1024} \ x_{2,1024} \ x_{3,1024}]$$

$x_{i,k}$ = luminancia del pixel k en la imagen i

Una vez determinado X se selecciona $c=4$ como el número de conjuntos que se desean obtener, debido a que se desean obtener cuatro zonas distintas en las imágenes, y se emplea el algoritmo *fuzzy c-means*.

Trivedi [15] emplea los centros de los conjuntos obtenidos con este procedimiento para caracterizar cada una de las cuatro zonas. Posteriormente, para decidir a que zona pertenece cada pixel, observa a cual de los cuatro conjuntos difusos hallados pertenece en mayor medida, con lo que se completa el proceso de identificación de la imagen.

III. BASE DE DATOS DIFUSA

La Lógica Difusa busca desarrollar un conjunto de procedimientos para manejar la información precisa y/o vaga. Ahora bien, los Sistemas de Bases de Datos tienen por propósito, hablando en términos muy generales, la organización de la información; por tanto no es de sorprendente que se haya intentado incorporar las técnicas de Lógica Difusa en el diseño de Bases de Datos.

Miyamoto & Umano [12] distinguen dos tipos de técnicas difusas en las Bases de Datos:

- Bases de Datos Difusas.
- Técnicas Difusas para la recuperación de la información.

En la primera de estas técnicas el concepto de Conjunto Difuso se incorpora en la estructura misma de la Base de Datos, mientras que en la segunda se emplea en las estrategias de recuperación de la información.

A continuación se presenta una de las técnicas del primer tipo, reseñadas en [12]; como en los anteriores ejemplos de este artículo, se ha seleccionado buscando un ejemplo conceptualmente sencillo. La técnica en cuestión se denomina *Búsqueda Difusa*, y fue propuesta inicialmente por Tahani [14].

En esta técnica la Base de Datos tiene dos componentes⁴: por una parte se tiene una Base de Datos tradicional, como la representada en el cuadro 2, y por otra parte se tiene una definición Difusa de las variables cuantificables de dicha cuadro, como la que se muestra en la figura 19.

Cuadro 2. Base de Datos tradicional del Ejemplo

Nombre	Edad	Salario	Año de Ingreso
Anderson	30	20.000	1995
Brown	30	15.000	1995
Long	25	40.000	1993
Nelson	55	20.000	1980
Smith	25	23.000	1996

⁴ El ejemplo presentado es extraído de [12], con algunas adiciones

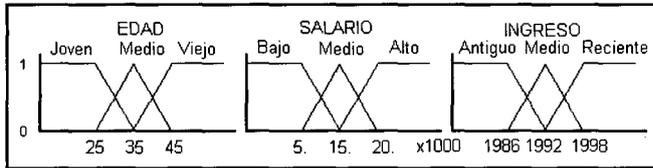


Figura 19. Variables lingüísticas para el ejemplo

Ante una consulta a la Base de Datos de la forma:

"Cuáles son los nombres de las personas jóvenes o recientemente empleadas pero con sueldo alto"

Tahani propone evaluar la función de pertenencia de cada registro a cada uno de los valores lingüísticos involucrados en la consulta, y entregar como resultado de la búsqueda un conjunto difuso con funciones de pertenencia obtenidas mediante la utilización de operadores AND, OR y NOT difusos [6]; los operadores empleados por Tahani son el mínimo para el AND, el máximo para el OR y el complemento para el NOT.

Con esta metodología la consulta anterior podría representarse por la operación

$$u_{búsqueda}(x, y, z) = \{u_{joven}(x) OR u_{reciente}(y)\} AND u_{alto}(z)$$

En donde x es la edad, y el año de ingreso, z el salario, $u_{joven}(x)$ es la función de pertenencia al conjunto *Joven* definido sobre la variable edad, $u_{reciente}(y)$ es la función de pertenencia al conjunto *Reciente* definido sobre la variable Año de Ingreso, $u_{alto}(z)$ es la función de pertenencia al conjunto *Alto* definido sobre la variable salario, y $u_{búsqueda}(x,y,z)$ es la función de pertenencia al conjunto resultante de la búsqueda.

Empleando la información de el cuadro 2 y de la figura 19 se puede construir el cuadro 3 que muestra los resultados de la búsqueda para cada registro.

Cuadro 3. Resultados de la búsqueda del ejemplo en cada registro

Nombre	$u_{joven}(x)$	$u_{reciente}(y)$	$u_{alto}(z)$	$u_{búsqueda}(x,y,z)$
Anderson	0,5	0,6	0,5	0,5
Brown	0,5	0,6	0,0	0,0
Long	1,0	0,2	1,0	1,0
Nelson	0,0	0,0	0,5	0,0
Smith	1,0	0,8	0,8	0,8

El resultado final del ejemplo sería el conjunto:

$$Búsqueda = \{0,5/Anderson, 1,0/Long, 0,8/Smith\}$$

IV. PSICOLOGÍA COGNOCITIVA: RECONOCIMIENTO DE PALABRAS

El ejemplo que se presenta a continuación difiere de los anteriores sensiblemente. Este caso no emplea los algoritmos asociados a la Lógica Difusa, sino el concepto mismo de los Conjuntos Difusos, y ha sido seleccionado para resaltar que la importancia de la Lógica Difusa radica en la noción de Conjuntos con fronteras no exactas, lo que implica gradualidad en los cambios.

Este ejemplo consiste en la definición de un experimento cuyo autor es Rueckl [13] para el reconocimiento de palabras, que es uno de los temas abordados por la Psicología Cognocitiva.

La pregunta que se desea contestar es :

¿Qué efectos tiene el contexto de una frase en el reconocimiento de palabras?

A esta pregunta existen dos respuestas opuestas, sustentadas cada una por dos teorías diferentes [13]:

- La teoría del modelo Interactivo sostiene que el contexto si influye en el reconocimiento de palabras.
- La teoría del modelo Autónomo sostiene que el contexto no influye.

Ambos modelos se apoyan en experimentos cuyos resultados son consistentes y robustos, con explicaciones consistentes con las respectivas teorías. Los experimentos son de la siguiente forma:

- Se selecciona una palabra objetivo, por ejemplo *desk*⁵.
- Se plantean frases con la palabra objetivo, unas en las que el contexto es congruente, y otras en las que el contexto es incongruente, por ejemplo:

Contexto congruente: *Mary's book were piled up on her _____*
 Contexto incongruente: *Last night Mary read a good _____*

- Se mide que tan fácilmente reconoce un individuo (un grupo de individuos) la palabra objetivo en los dos tipos de frases, para sacar conclusiones.

Hasta este punto las dos teorías coinciden, pero cuando las frases de contexto incongruente se remplazan por frases de contexto neutral, los experimentos dan resultados distintos, cada uno reforzando una teoría, dependiendo de lo que se entienda por "contexto neutral". Se han utilizado frases como las siguientes:

They said it was the _____
The the the _____
It was the _____
 Secuencias aleatorias de palabras.

La propuesta de Rueckl consiste en responder a la pregunta *¿Qué es un contexto neutral?* diciendo que hay *congruencias difusas*, es decir, diciendo que entre los contextos congruentes e incongruentes no hay un único tipo de contexto neutral, sino que la congruencia puede manejarse gradualmente.

⁵ En la explicación de este ejemplo se han mantenido las palabras originales en inglés, sin su traducción, para no desvirtuar la naturaleza del experimento.

El experimento de Rueckl utilizó dos palabras objetivo: *pair pain*. Estas palabras se insertaron en las siguientes frases:

The cardplayer had a _____ in his hand
The shoemaker had a _____ in his hand
The piano player had a _____ in his hand
The arthritic had a _____ in his hand

Claramente, el contexto varía para ambas palabras objetivo. Adicionalmente, en el experimento se manipulo la forma de la letra *r - n* que diferencia las dos palabras objetivo, en la forma que muestra la figura 20. Ante este experimento, las dos teorías predicen resultados distintos. Los resultados obtenidos concuerdan más con la teoría Interactiva.



Figura 20. Variaciones de la letra *r - n*

CONCLUSIONES Y COMENTARIOS

Se han presentado cuatro aplicaciones diferentes de la Lógica Difusa; las tres primeras recogen ejemplos sencillos de algunas de las más comunes aplicaciones en ingeniería, excluyendo otras igualmente comunes, como las relativas a la Investigación de Operaciones y toma de Decisiones, Filtrado de Señales, Diagnóstico de fallas, etc. La cuarta aplicación presentada cae en el área de la psicología, y se ha incluido aquí con el propósito de resaltar la importancia conceptual de los Conjuntos Difusos, ya que la existencia de fronteras no exactas puede emplearse en otros campos distintos a los de la ingeniería.

En el ámbito colombiano pueden sugerirse varias posibles aplicaciones de la lógica difusa; de hecho, ésta ya se ha empleado para desarrollar sistemas expertos que ayudan a la planificación del sector energético, así como a la negociación en la Bolsa de Energía. También se conocen algunas aplicaciones en el control industrial. Sin embargo, existe un gran potencial aún no explorado en aplicaciones socio-económicas, de dos tipos diferentes: por una parte están las aplicaciones directas en sistemas económicos (instituciones financieras, bolsa de valores, análisis macroeconómico) y por otra parte están los enfoques teóricos de las ciencias sociales, que pueden beneficiarse de la noción de "conjunto difuso" (definición del concepto de "Identidad Cultural", o el de "Paz Imperfecta").

BIBLIOGRAFÍA

1. ÁLVAREZ, Hernán D. "Control Difuso y Sistemas de Control Inteligentes" En: *Memorias del Segundo Congreso de la Asociación Colombiana de Automática*, Bucaramanga, Colombia, marzo de 1997, pp 331-340
2. BEZDEK, J. "Partition Structures: a tutorial" En "*Analysis of fuzzy information*", Bezdek, J. CRC Press, 1987, pp 81-108
3. _____ "Pattern recognition with fuzzy objective algorithms" *Plenum Press*, New York, 1981
4. DRIANKOV, Dimiter *et al.* "*An Introduction to Fuzzy Control*" Springer Verlag, Berlin, 1993
5. DUARTE, Oscar G. "UNFUZZY - Software para el análisis, diseño, simulación e implementación de Sistemas de Lógica Difusa" Tesis para optar el Título de Magíster en Automatización Industrial. Universidad Nacional de Colombia, Facultad de Ingeniería, 1997.
6. _____, "Sistemas de Lógica Difusa. Fundamentos" En *Ingeniería e Investigación* No. 42 Marzo de 1999,
7. KLIR, George and YUAN, Bo. "*Fuzzy Sets and Fuzzy Logic*" Prentice Hall, New Jersey, 1995
8. LAYNE, J. and PASSINO, K. "Fuzzy model reference learning control for cargo ship steering" en *IEEE Control Systems Magazine*, Vol 13, No. 6, dic 93, pp 23-24
9. LEE, Chuen Chien. "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part I" En *IEEE Transactions on Systems, Man, and Cybernetics*, V 20, No. 3 Marzo/Abril 1990, pp 404-418
10. _____, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part II" En *IEEE Transactions on Systems, Man, and Cybernetics*, V 20, No. 3 Marzo/Abril 1990, pp 419-435
11. MENDEL, Jerry. "Fuzzy Logic Systems for Engineering: A Tutorial" En *Proceeding of the IEEE*, V 83, No. 3 Marzo 1995, pp 345-377
12. MIYAMOTO, S. and UMANO, M. "Recent developments in fuzzy databases and applications", En "*Fuzzy logic and its applications to engineering information sciences and intelligent systems*", Editado por Bien, Z. & Min, K. C. Klumer Academic Publishers, pp. 367-378
13. Rueckl, J. "A fuzzy propositional account of contextual effects on word recognition" En "*Fuzzy Sets in psychology*", Zéténeyi, T. North-Holland Editors, 1988, pp 127-154.
14. TAHANI, V. "A conceptual framework for fuzzy querying processing - a step toward very intelligent database system" En *Information Processing Management* 13, 1977, 289 - 303
15. TRIVEDI, M.M. "Analysis of Aerial Images using fuzzy clustering" En "*Analysis of fuzzy information*", Bezdek, J. CRC Press, 1987, pp 133-152.
16. WANG, Li-Xin "*Adaptive Fuzzy Systems and Control. Design and Stability Analysis*" Prentice Hall, New Jersey, 1994
17. _____ "Fuzzy Systems are Universal Approximators" En *Proceedings of the IEEE International Conference on Fuzzy Systems*, San Diego, California 1992, pp 1163-1170
18. WANG, Li-Xin and MENDEL, Jerry. "Back Propagation Fuzzy System as Nonlinear Dynamic Systems Identifiers" En *Proceedings of the IEEE International Conference on Fuzzy Systems*, San Diego, California, 1992 pp 1409-1418
19. _____ "Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least-Squares Learning" En *IEEE Transactions on Neural Networks*, Vol 3, No 5, Septiembre 1992, pp 807-814
20. _____ "Generating Fuzzy Rules by Learning from Examples" En *Proceedings of the IEEE 1991 International Symposium on Intelligent Control* 13-15 August, Arlington, Virginia, U.S.A. pp 263-268