



Bases de datos difusas

Inteligencia Computacional

Pedro Manuel Gómez-Portillo López

gomezportillo@correo.ugr.es

03 de diciembre de 2018

Índice

1. Resumen	3
1.1. Palabras clave	3
2. Introducción	4
2.1. No es lo mismo que probabilidad	6
3. Sistemas de Bases de Datos Difusas	7
3.1. Bases de datos relacionales	7
3.2. Bases de datos difusas	8
3.2.1. Conjuntos difusos	8
3.2.1.1. Funciones de pertenencia	9
Ejemplo práctico sobre ciudades para vivir	9
3.2.1.2. Variables lingüísticas	10
Ejemplo práctico sobre edades	10
3.2.2. Representación de la información	11
3.2.2.1. Datos precisos	11
3.2.2.2. Datos imprecisos	12
Datos imprecisos sobre dominios continuos	12
Datos con analogías sobre dominios discretos	12
Datos indefinidos	12
Datos desconocidos	12
Datos nulos	12
3.2.3. Implementación de las bases de datos difusas	13
3.2.4. Principales modelos de bases de datos difusas	13
3.2.4.1. Modelo relacional difuso básico	13
3.2.4.2. Modelo de unificación mediante relaciones de similitud	13
Ejemplo del modelo de unificación basado en relaciones de similitud	14
3.2.4.3. Modelos relacional mediante distribuciones de posibilidad	15
4. Lenguajes lógicos	16
4.1. FSQ	16
4.1.1. Valores lingüísticos	16
4.1.2. Comparadores difusos	17
4.1.3. Umbrales	17
4.1.4. Constantes difusas básicas	18
5. Conclusiones	19
6. Referencias	20

1. Resumen

En este ensayo se presenta una visión general a la Lógica Difusa, a las bases de datos difusas y a cómo funcionan. Se presentan y explican los cimientos de la Lógica Difusa en las que se apoyan las bases de datos difusas y por último se presenta el lenguaje de consultas difusas FSQL.

1.1. Palabras clave

fuzzy databases, fuzzy logic, fsql

2. Introducción

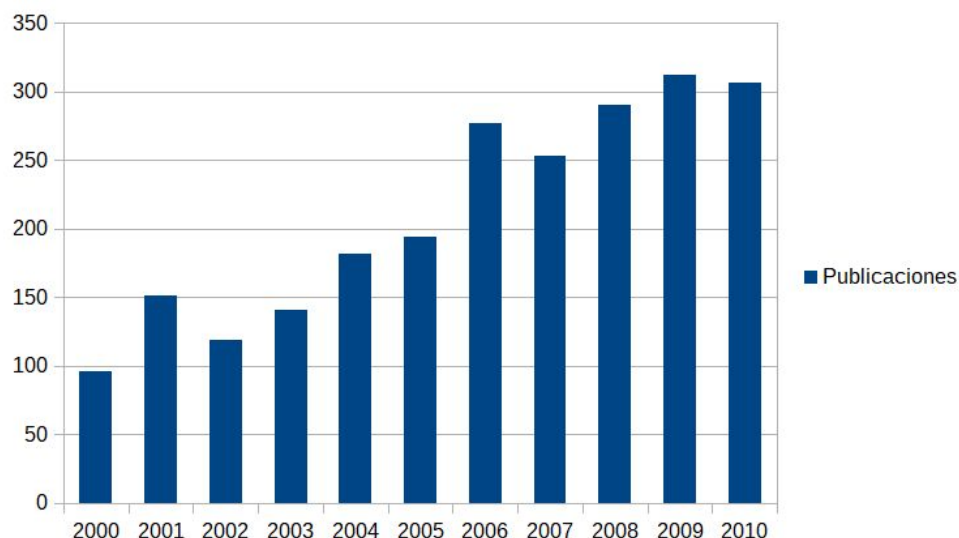
El ser humano comunica sus ideas de un modo muy vago. *Si llueve mucho cogeré el paraguas, sácalo del horno antes de que esté muy caliente o no le des tan fuerte o lo romperás* son solo algunos de los ejemplos que demuestran lo ambiguo que puede ser el lenguaje; estos valores cambian dependiendo del contexto al que nos refiramos.

Las personas no tenemos ningún problema manejando conceptos con fronteras tan **difusas**, pero los ordenadores, que trabajan con la lógica clásica, no entienden este tipo de enunciados. ¿Cuándo deja de estar caliente y pasa a estar muy caliente?

No fue hasta 1964 cuando Lofti A. Zadeh, catedrático de la Universidad de Berkeley y considerado el padre de la Lógica Difusa, presentó en su famoso artículo [\[4\]](#) los cimientos en los que en los años 70 se basarían los que vendrían detrás de él. En dicho artículo presentaba este paradigma como un sistema en el que los datos, en lugar de pertenecer a conjuntos de manera booleana, tienen un grado de pertenencia a los mismos.

Básicamente, la Lógica Difusa es una lógica multivaluada que permite representar matemáticamente la incertidumbre y la vaguedad, proporcionando herramientas formales para su tratamiento [\[3\]](#).

Las aplicaciones actuales de esta lógica son prácticamente infinitas, y un ejemplo que lo demuestra es el considerable aumento que ha sufrido el número de artículos publicados por año. En el gráfico inferior, adaptado de [\[2\]](#), puede verse que las publicaciones relacionadas con la Lógica Difusa han triplicado su número de 2000 a 2010.



Una recopilación reciente de la inmensa cantidad de investigaciones y aplicaciones de la Lógica Difusa puede verse en [9]. Una de estas aplicaciones ha sido la de las bases de datos difusas. Una base de datos es un almacén de información que debe permitir el acceso a su contenido a través de peticiones formales.

Si imaginamos una base de datos que contiene personas, y uno de sus atributos es su edad, podríamos obtener las personas con 20 años, abstrayéndonos de la tecnología utilizada, de un modo parecido al siguiente.

ENCUENTRA A LAS PERSONAS CON EDAD IGUAL A 20 AÑOS

De este modo, y desde un punto de vista matemático, se obtendrán las personas que pertenecen al grupo *tener 20 años*. Pero, ¿qué pasaría si quisiésemos también las que tiene 19? ¿Y 21? ¿Qué pasaría si estuviésemos buscando las personas *jóvenes*?

Y aquí es cuando aparecen las bases de datos difusas, que proponen *difuminar* los límites de pertenencia de los elementos a los conjuntos. De este modo, podríamos contestar nuestra pregunta anterior ejecutando la siguiente petición sobre una base de datos difusa.

ENCUENTRA A LAS PERSONAS JÓVENES

Una persona de 15 años sería totalmente joven, es decir, tendría un grado de pertenencia de 1 al grupo de personas jóvenes, mientras que otra de 30 años sería joven solamente a medias, es decir, tendría un grado de pertenencia de 0.5 al grupo de personas jóvenes.

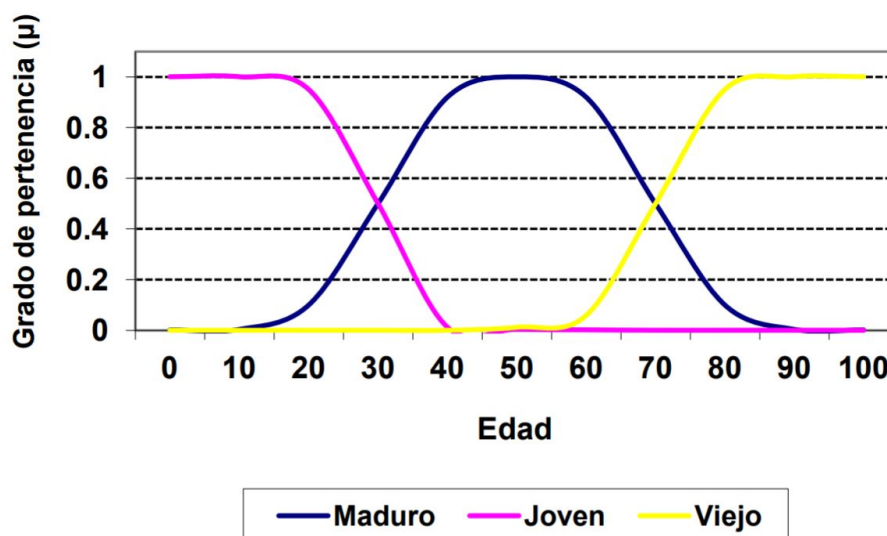


Imagen extraída de [8]

2.1. No es lo mismo que probabilidad

Aunque a primera vista los conceptos de *lógica difusa* y *probabilidad* puedan parecer relacionados, no tienen mucho que ver ya que, mientras la probabilidad presenta la frecuencia de ocurrencias de un suceso, la Lógica Difusa define el grado de pertenencia de un evento a un conjunto [\[3\]](#).

El ejemplo por antonomasia que se utiliza para diferenciarlas claramente es el de las botellas. Supongamos que tenemos dos botellas, una probabilística y la otra difusa, con etiquetas en las que se indica *potable al 0.8*. Esto significaría que en la primera botella, la probabilística, la probabilidad de que el líquido que contiene sea potable es del 80%, mientras que, en la difusa, el 80% del líquido que contiene es potable.

3. Sistemas de Bases de Datos Difusas

El ser humano siempre ha tenido la necesidad de almacenar información; al principio bastaba con con papel y lápiz para apuntarla, pero cuanto más amplia y compleja es esta información mayor es la necesidad de estructurarla y organizarla de manera adecuada.

Una base de datos es un conjunto de datos almacenados organizados a través de una estructura de datos. Cada base de datos soporta, como mínimo y por lo general como máximo, un único modelo lógico, siendo el más común el relacional (aunque hay otros, como el de red, el orientado a objetos o el jerárquico).

3.1. Bases de datos relacionales

El modelo relacional representa la segunda generación de las bases de datos, siendo la primera las que utilizan los modelos jerárquico o de red. En él, todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas.

Para trabajar con estas bases de datos se utilizan lenguajes relacionales, ya que cuentan con operadores dedicados a trabajar sobre relaciones. Todos se basan o bien en el álgebra relacional, si son procedurales, o en el cálculo relacional, si son no procedurales.

Tanto el álgebra relacional como el cálculo relacional representan distintos estilos de expresión del manejo de datos en el ámbito del modelo relacional. El álgebra relacional proporciona una serie de operaciones para construir relaciones. y el cálculo relacional proporciona una notación para formular la definición de una determinada relación. [\[6\]](#)

Las bases de datos relacionales se han extendido para almacenar nuevos tipos de datos, como imágenes, sonidos y vídeo. También se están utilizando técnicas para manipular grandes cantidades de información, como bancos de datos de genes humanos y sistemas de observación de señales del espacio. Además, se está trabajando con paradigmas como las bases de datos relacionales distribuidas heterogéneas, donde los datos provienen de múltiples fuentes. [\[8\]](#)

El problema del modelo relacional es que no contempla el procesamiento de consultas ambiguas como la del ejemplo de introducción *Encuentra a las personas jóvenes*,

dado que ni el cálculo ni el álgebra relacional, que establecen el resultado de cualquier consulta como una nueva relación, tienen la capacidad de permitir consultas de una manera difusa.

En los últimos años, los investigadores en este área han intentado relajar el modelo relacional para permitirle admitir algunas imprecisiones, lo que conduce a sistemas de bases de datos que encajan en el campo de la Inteligencia Artificial, ya que permiten el manejo de información con una terminología que es muy similar a la del lenguaje natural. [7]

El resultado de estas investigaciones ha sido la unión de las bases de datos relacionales con la teoría de la Lógica Difusa, lo que ha dado lugar a las bases de datos difusas.

3.2. Bases de datos difusas

El modelo de una base de datos difusa consiste, muy resumidamente, en añadir un grado de pertenencia en el intervalo $[0, 1]$ a cada elemento que se pretenda difuminar, que es mantenido en la base de datos en su forma original.

Las bases de datos difusas nacen de unir la teoría de bases de datos relacionales con la teoría de conjuntos difusos de Lofti A. Zadeh [4] para permitir, básicamente dos objetivos [13]:

- El almacenamiento de información difusa.
- El tratamiento y consulta de esta información de forma difusa o flexible.

3.2.1. Conjuntos difusos

Un conjunto es una colección de objetos bien especificados que poseen una propiedad común. Sin embargo, para un conjunto difuso la cuestión de pertenencia de un elemento al conjunto no es cuestión de todo o nada, sino que hay diferentes grados de pertenencia.

La definición formal de un conjunto difuso es la siguiente [8].

Sea X una colección de objetos cuyos elementos se denominan genéricamente x . Un conjunto difuso A en X es definido como un conjunto de pares ordenados tal que,

$$A = \{(x, \mu_A) \mid x \in X\},$$

Donde μ_A es la función de pertenencia para el conjunto difuso A , que asocia cada elemento del conjunto X un grado de pertenencia al conjunto difuso en el intervalo $[0, 1]$.

3.2.1.1. Funciones de pertenencia

La función de pertenencia de un conjunto difuso, que es la que determina con qué grado pertenece un elemento a un conjunto, puede tomar cualquier valor en el intervalo real $[0, 1]$.

A continuación se presentan la definición formal de esta función. [\[6\]](#)

- El Universo de Discurso es el conjunto X de posibles valores que puede tomar la variable x .
- La función de pertenencia $\mu_A(x)$ de un conjunto difuso A será la función

$$\mu_A(x) : X \rightarrow [0, 1].$$

De este modo, un conjunto difuso A queda definido del siguiente modo.

$$A = \{(x, \mu_A(x)) : x \in X, \mu_A(x) \in [0, 1]\}.$$

Por lo que cualquier elemento x del conjunto difuso A en X tiene un grado de pertenencia $\mu_A(x) \in [0, 1]$.

Ejemplo práctico sobre ciudades para vivir

J. Caderas Lucero, antiguo alumno de la UGR, presenta en su tesis doctoral [\[8\]](#) un ejemplo muy didáctico para entender este concepto, del cual se ha adaptado lo siguiente.

Sea X el conjunto discreto de ciudades en las que nos gustaría vivir tal que,

$$X = \{Madrid, Barcelona, Granada\}.$$

En este contexto no tendría sentido hacer uso de la lógica clásica, ya que la apetencia por vivir en cada ciudad no podría reflejarse de un modo booleano (sí o no).

Por lo tanto, se define el conjunto difuso A , *ciudad deseable para vivir*, y sus funciones de pertenencia μ_A tal que,

$$\begin{aligned}\mu_A(Madrid) &= 0.5, \\ \mu_A(Barcelona) &= 0.6, \\ \mu_A(Granada) &= 0.9,\end{aligned}$$

$$A = \{Madrid/0.5, Barcelona/0.6, Granada/0.9\}.$$

La función de pertenencia μ_A definida es muy subjetiva, ya que estas apetencias pueden cambiar en función de cada persona.

3.2.1.2. Variables lingüísticas

Los valores de una variable lingüística, como su nombre sugiere, son palabras o sentencias en un lenguaje natural (es decir, que no son números). El concepto de variable lingüística es clave en las aplicaciones de Lógica Difusa, ya que es la unión principal entre éstas y la manera en la que las personas nos comunicamos.

Siguiendo con el ejemplo de la introducción, *Edad* es una variable lingüística cuyos valores pueden ser *joven*, *maduro* o *viejo*. Cada uno de estos valores representa un conjunto difuso en un Universo de Discurso determinado.

En términos más formales [6], una variable lingüística es una quintupla

$$(N, T(N), X, G, M),$$

donde,

- N es el nombre de la variable lingüística.
- $T(N)$ es el conjunto de términos o valores de N .
- X es el Universo de Discurso donde se define $T(N)$.
- G es una regla sintáctica que genera los valores de $T(N)$.
- M es una regla semántica que asocia cada valor L su significado $M(L)$.

Ejemplo práctico sobre edades

Aplicando las definiciones previas en el ejemplo de introducción, *Edad* sería una variable lingüística N cuyo Universo de Discurso X , que representa la edad de una persona, podría ser definido como,

$$X = [0, 100] \text{ años}$$

Su conjunto de valores $T(N)$ sería,

$$T(Edad) = \{joven, maduro, viejo\}.$$

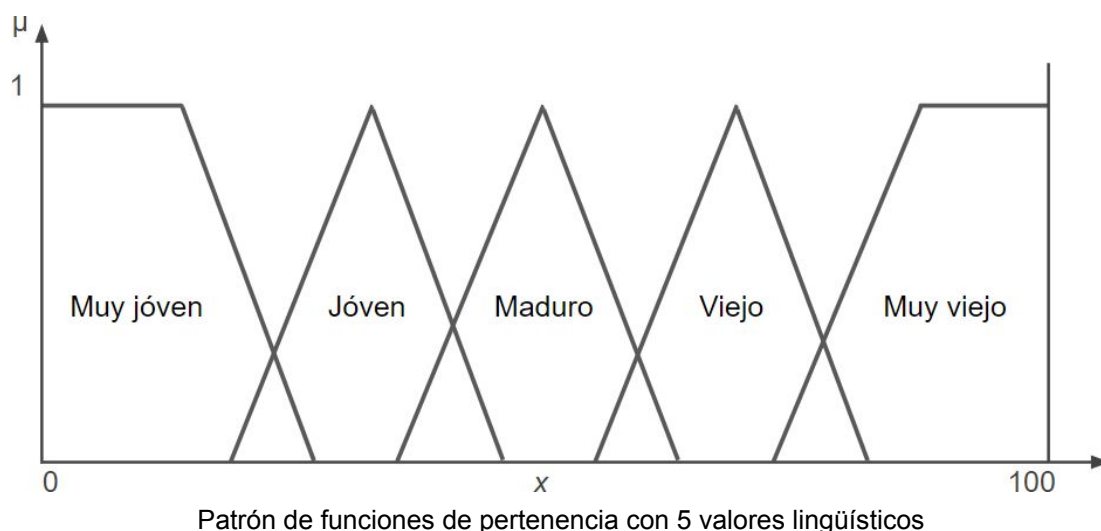
Cada uno de los término de $T(Edad)$ está definido por un intervalo difuso en el Universo de Discurso X .

La regla sintáctica G determina el orden de las palabras de los valores lingüísticos en $T(N)$. Por ejemplo, *muy* sería un modificador que precederían a la variable lingüística *joven* y la alterarían.

La regla semántica M asocia cada valor lingüístico con su significado. Por ejemplo, *joven* sería alrededor de 25 años.

Como se indicó antes, cada una de las variables lingüísticas tendría su propia función de pertenencia, $\mu_{joven}(x)$, $\mu_{maduro}(x)$ y $\mu_{viejo}(x)$, siendo x una edad tal que $x \in X$.

Según Lofti A. Zadeh [11], las aplicaciones basadas en la Lógica Difusa implementan entre 3 y 7 valores lingüísticos cuyas funciones de pertenencia son de tipo triangular o trapezoidal. La figura inferior, adaptada de [11], muestra un patrón de 5 valores lingüísticos donde el Universo de Discurso $X = [0, 100]$ es la edad de una persona.



3.2.2. Representación de la información

En una base de datos difusa pueden aparecer varios tipos de datos [5] [8] [10].

3.2.2.1. Datos precisos

Manejados normalmente mediante la representación provista por la base de datos relacional que estamos *difuminando*.

3.2.2.2. Datos imprecisos

Los datos imprecisos almacenados en una base de datos se pueden clasificar atendiendo a sus características.

Datos imprecisos sobre dominios continuos

Estos datos contiene distribuciones de probabilidad definidas en dominios continuos u ordenados.

Datos con analogías sobre dominios discretos

Estos datos se construyen sobre dominios discretos en los que existen ya definidas relaciones de proximidad entre sus valores. Se almacena la representación de los datos además de la representación de las relaciones de proximidad definidas para los valores en el dominio.

Datos indefinidos

Estos datos reflejan el hecho de que ningún valor de su dominio está permitido. Por ejemplo, el número de teléfono de alguien que no tiene teléfono.

Datos desconocidos

Este tipo de datos reflejan el desconocimiento sobre el valor que toma un atributo, aunque también expresan que puede tomar uno de los valores del dominio. Por ejemplo, la edad de una persona, que aunque desconocemos sabemos que tiene que tener.

Datos nulos

Estos datos expresan tanto que no conocemos su valor o que es imposible asignarle un valor del dominio. Por ejemplo, la dirección de correo de una persona puede ser nula si desconocemos tanto su valor como si tiene o no.

3.2.3. Implementación de las bases de datos difusas

El problema de la implementación de los sistemas de bases de datos difusas ha sido tratado en dos vertientes principales. [\[5\]](#) [\[7\]](#) [\[12\]](#)

- Partir de una base de datos con información precisa y desarrollar una sintaxis que permita formular consultas difusas con capacidades de manejar la imprecisión.
- Construir una base de datos difusas que permita que la información imprecisa pueda ser almacenada. Dentro de esta vertiente existen varios modelos alrededor de los cuales construir las bases de datos difusas.

3.2.4. Principales modelos de bases de datos difusas

Existen tres enfoques principales a la hora de diseñar una base de datos difusa [\[6\]](#) [\[8\]](#) [\[13\]](#) [\[15\]](#).

3.2.4.1. Modelo relacional difuso básico

Este modelo es el más básico y simplemente consiste en añadir un grado de pertenencia difuso a cada elemento de la base de datos. Este grado difuso pertenece a toda la tupla de elementos de cada relación, por lo que lo que se *difumina* es la relación propiamente dicha.

Sus principales inconvenientes son que no permiten información imprecisa distinta de un único grado difuso (por ejemplo, almacenar que cierta persona es *joven*) y que el manejo de este tipo de información se realiza de forma global; cada tupla asume su carácter difuso sin poder determinar cuál es la aportación difusa de cada atributo individualmente.

3.2.4.2. Modelo de unificación mediante relaciones de similitud

Este modelo, diseñado por [\[14\]](#), busca encontrar la mayor similitud entre los diferentes valores del dominio y el valor que se está definiendo en el intervalo $[0, 1]$, donde 0 se aplica a los elementos que son totalmente diferentes y 1 a los que son extremadamente parecidos.

Ejemplo del modelo de unificación basado en relaciones de similitud

Para intentar clarificar este modelo, se propone el siguiente ejemplo.

Se quiere trabajar con los colores de pelo de las personas almacenadas en una base de datos, por lo que se define la variable lingüística N , el conjunto de términos $T(N)$ y el Universo de Discurso X como sigue,

$$N = \text{color de pelo},$$

$$T(N) = \{\text{morelo}, \text{castaño}, \text{pelirrojo}, \text{rubio}\},$$

$$X = [0, 1].$$

A continuación, sobre estos valores se define la siguiente relación de similitud,

	Rubio	Pelirrojo	Castaño	Moreno
Rubio	1	0.5	0.25	0
Pelirrojo		1	0.5	0.25
Castaño			1	0.5
Moreno				1

Por otro lado, en nuestra base de datos existe la siguiente tabla,

Nombre	Color de pelo
Álvaro	Castaño
Jaime	Moreno
Marta	Pelirroja
Ana	Rubia

Y se realiza la siguiente consulta,

ENCUENTRA LAS PERSONAS CASTAÑAS CON GRADO 0.5

Comprobando la tabla de relaciones, los únicos colores de pelo que son parecidos al menos con grado 0.5 al castaño son el castaño y el pelirrojo, por lo que su respuesta sería,

{ÁLVARO, MARTA}

La principal ventaja de este modelo es que proporciona una herramienta lo suficientemente adecuada para representar la imprecisión de los elementos a pesar de que no modele totalmente bien todos los aspectos difusos de la información.

3.2.4.3. Modelos relacional mediante distribuciones de posibilidad

Este modelo es útil para representar la ambigüedad de los datos mediante el uso de distribución de posibilidades, donde se procesa la información del atributo que tenga el conflicto de ambigüedad .

Existen diferentes modelos basados en este enfoque, como el modelo *Umano-Fukami* [16], el modelo *Prade-Testemale* [17], el modelo *Zemankova-Kandel* [18] o el modelo *GEFRED* [19], desarrollado por profesores de la Universidad de Granada, aunque dichos modelos escapan del alcance de este ensayo.

4. Lenguajes lógicos

Actualmente existen varios lenguajes para trabajar con bases de datos difusas, aunque las principales alternativas son **SQLf** y **FSQL**. A continuación vamos a centrarnos en esta última, ya que ha sido desarrollada por un profesor de la Universidad de Granada.

Cabe destacar que, aunque la idea principal para este capítulo había sido realizar un ejemplo práctico en el que se mostrase la potencia de este lenguaje, me ha sido imposible configurar un entorno en el que poder ejecutarlo, así que lo presentaré y dejaré al lector la posibilidad de usar lo aprendido en un proyecto propio.

4.1. FSQL

FSQL son las siglas de *Fuzzy Structured Query Language*. Es una extensión del lenguaje SQL que permite el tratamiento de bases de datos relacionales difusas usando la Lógica Difusa. Inicialmente descrito por el profesor Juan Manuel Medina de la Universidad de Granada, ha sido más ampliamente definido e implementado por el profesor José Galindo de la Universidad de Málaga.

A continuación se presenta un resumen de sus características, su sintaxis y su uso [\[21\]](#) [\[22\]](#) [\[24\]](#).

4.1.1. Valores lingüísticos

FSQL permite definir fácilmente los valores de una variable lingüística (e.g. $\{alto, medio, bajo\}$), que en su uso irán precedidas el símbolo \$ para permitir ser identificados fácilmente. Los hay de dos tipos,

- Valores para atributos con un dominio difuso ordenado. Cada valor de este tipo tiene asociado una función de pertenencia trapezoidal.
- Valores para atributos con un dominio difuso no ordenado, que tienen definida una relación de similaridad entre cada una de las dos etiquetas en el dominio, cuyo grado de similaridad se encuentra en el intervalo $[0, 1]$.

4.1.2. Comparadores difusos

Además de los comparadores que ya vienen incluidos en SQL, como =, > y <, FSQl incluye los siguientes comparadores difusos.

Comparador FSQl		Equivalente SQL		Significado
FEQ	F=	EQ	=	Igual que
FDIF	F<>	DIF	<>	Diferente que
FGT	F>	GT	>	Mayor que
FGEQ	F>=	GEQ	>=	Mayor o igual que
MGT	F>>	X		Mucho mayor que
FLT	F<	LT	<	Menor que
FLEQ	F<=	LEQ	<=	Menor o igual que
MLT	F<<	X		Mucho menor que
INCL		X		Incluido en
FINCL		X		Difusamente incluido en

Como vemos, el lenguaje FSQl resulta ser más potente que SQL, ya que lo expande incluyendo cláusulas que su predecesor no.

4.1.3. Umbrales

Los umbrales, indicados con la palabra clave *THOLD* x, donde x es un número en el intervalo [0,1] representando el grado de pertenencia, sirven para realizar consultados difusas.

Por ejemplo, si tuviéramos una base de datos FSQl llamada `Personas` con los siguientes campos,

- `nombre` indicando el nombre de las persona,

- `color_pelo` indicando el color del pelo,
- `altura` indicando la altura

y quisiéramos obtener el nombre de las personas altas con grado 0.8 y rubias con grado al menos 0.8 podríamos emplear la siguiente sentencia FSQ.

```
SELECT nombre FROM Personas

WHERE color_pelo F= $rubio THOLD 0.8

AND altura FGEQ $alto THOLD 0.8
```

4.1.4. Constantes difusas básicas

Las constantes que se pueden usar en FSQ se resumen en la siguiente tabla, aunque algunas de ellas ya han sido presentadas en este trabajo con anterioridad.

Constante FSQ	Significado
UNKNOWN	Aunque el valor es desconocido, el atributo es aplicable
UNDEFINED	El atributo no es aplicable
NULL	El atributo será ignorado
$[a, b, c, d]$	Permite construir un trapecio difuso ($a \leq b \leq c \leq d$)
$\$variable$	Permite utilizar una variable lingüística
$[n, m]$	Permite definir un intervalo $n \leq m$ para el umbral
$\#n$	Constante <i>aproximadamente</i> para el umbral

Para más información acerca del lenguaje SQL pueden consultarse los libros [\[25\]](#) y [\[26\]](#).

5. Conclusiones

Las bases de datos difusas son el resultado de varias décadas de investigación y desarrollo en el campo de la Lógica Difusa, y prueba de ello son su complejidad y su profundidad.

En definitiva, me alegro de haber elegido las bases de datos difusas para el trabajo y haber tenido la oportunidad de investigar en este tema aunque, dada su profundidad y las características a las que ha habido que restringir el trabajo, haya sido imposible ahondar más en él. Creo que puede ser algo muy útil en mi vida laboral futura, y que seguramente llegue el momento en el que las necesite, por lo que podré reutilizar este trabajo para refrescar conceptos.

6. Referencias

- [1] Díaz, B. & Morillas, A. (2004) *Minería de datos y lógica difusa. Una aplicación al estudio de la rentabilidad económica de las empresas agroalimentarias en Andalucía.*
- [2] Patel, A., Gupta, S., Rehman, Q. & Verma, M. (2013). *Application of Fuzzy Logic in Biomedical Informatics.*
- [3] González, C. (2012) *Lógica Difusa: Una introducción práctica. Técnicas de Softcomputing (Técnicas de Softcomputing)*
- [4] Zadeh, L., A. (1964) *The concept of a linguistic variable and its applications to approximate reasoning.*
- [5] Fonseca, Y., Reyes, O., Aballe & M., Urquiza, A. (2012) *A glance to the fussy databases.*
- [6] Sanchis, D. (2015) *Bases de Datos Relacionales Difusas.*
- [7] Mata, T., Murillo, J. & Hernández, J. (2009) *Bases de datos deductivas y bases de datos difusas.*
- [8] Cadenas Lucero, T. (2015) *Sistemas de bases de datos difusas sensibles al contexto.*
- [9] Magdalena, L., Verdegay, J. & Esteva, F. (2015) *Enric Trillas: a passion for fuzzy sets: a collection of recent works on fuzzy logic.*
- [10] Vila, M., Cubero, J., Medina M., & Pons, O. (1996) *A conceptual approach for dealing with imprecision and uncertainty in object-based data models.*
- [11] Zadeh, L., A. (1994) *Soft computing and Fuzzy Logic.*
- [12] Galindo, J. et al. (2006) *Fuzzy databases: modeling, design and implementation.*

- [13] <http://www.koala-soft.com/bases-de-datos-difusas>
- [14] Buckles, B. & Petry, F. (1982) *A fuzzy representation for relational databases. Fuzzy Sets and Systems.*
- [15] Denisse, C. (2015) *Implementación de una base de datos relacional difusa.*
- [16] Umano, M. & Fukami, S. (1994) *Fuzzy relational algebra for possibility distributions fuzzy relational model of fuzzy data.*
- [17] Prade, H. & Testemale, C. (1984) *Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries.*
- [18] Zemankova, M. & Kandel, A. (1984) *Fuzzy relational bases: a key to expert systems.*
- [19] Vila A., Medina, J. & Pons, O. (1994) *GEFRED. A Generalized Model of Fuzzy Relational Databases.*
- [20] Urrutia, A., Galindo, J., Jiménez, L & Piattini, M. (2003) *Data Modeling Dealing With Uncertainty in Fuzzy Logic.*
- [21] Galindo, J. (2008) *Fuzzy Information Processing in Databases.*
- [22] Galindo, J. (1999) *Tratamiento de la Imprecisión en Bases de Datos Relacionales: Extensión del Modelo y Adaptación de los SGBD Actuales.*
- [24] <http://www.lcc.uma.es/~ppgg/FSQL>
- [25] Galindo, J., Medina, J., Pons, O., Cubero, J. (1998) *A Server for Fuzzy SQL Queries.*
- [26] Galindo, J. (2005) *"New Characteristics in FSQL, a Fuzzy SQL for Fuzzy Databases.*