BASES DE DATOS RELACIONALES CON CONJUNTOS INTERVALO-VALORADOS DIFUSOS

ANTONIO FRANCISCO MUNDO PÉREZ-CEA

MÁSTER EN INGENIERÍA INFORMÁTICA. FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

Trabajo Fin Máster en Ingeniería Informática

31 de enero de 2018

Convocatoria: febrero 2018

Calificación: 7

Director:

D. Ramón González del Campo Rodríguez Barbero

Agradecimientos

A mi familia, especialmente a mi madre María Guadalupe y a mi padre José Francisco, que en paz descanse, por haberme dado todo el apoyo moral y espiritual en la continuidad de mis estudios. De igual manera a mis hermanos Daniel y José.

A Ramón González del Campo Rodríguez Barbero por su disposición, apoyo y consejos que promovieron la realización de este trabajo.

A todos aquellos que de alguna manera han influido en la aportación de ideas, apoyo y aliento para lograr la culminación de una meta que abre la puerta para iniciar un nuevo camino en mi carrera profesional.

Autorización de difusión

Antonio Francisco Mundo Pérez-Cea

31 de enero de 2018

El/la abajo firmante, matriculado/a en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: "BASES DE DATOS RELACIONALES CON CONJUNTOS INTERVALO-VALORADOS DIFUSOS", realizado durante el curso académico 2017-2018 bajo la dirección de D. Ramón González del Campo Rodríguez Barbero en el Departamento de Sistemas Informáticos y Computación, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Resumen en castellano

Cuando se habla de bases de datos relacionales, se piensa en una colección de elementos de datos organizados, definidos y estructurados en un conjunto de tablas con columnas y filas. Además, las tablas pueden relacionarse con otras a través de claves foráneas, así como también debe existir una consistencia e integridad en los datos de la misma. Se entiende que en una base de datos relacional, los atributos están basados en las restricciones de su dominio, donde además, el concepto de dominio sigue los principios de teoría de conjuntos clásico, es decir, se basa en la hipótesis de que un valor de un atributo debe ser un elemento que pertenece al conjunto especificado en su dominio. Por su parte, cuando se habla de bases de datos relacionales difusas, se habla esencialmente de lo anterior mencionado, pero empleando las propiedades de la lógica y conjuntos difusos en en el dominio de integridad de los datos; se habla del tratamiento flexible de los datos de una base de datos.

Para explotar el hecho de que la membresía de un elemento a un conjunto difuso es relativo y parcial, y para flexibilizar aún más las condiciones y consultas hechas sobre una base de datos, se introduce en este trabajo los cimientos de las bases de datos relacionales con conjuntos intervalo-valorados difusos, dando origen a IVFSQL (Interval-Valued Fuzzy SQL). Para ello, se comenzarán a estudiar los conceptos básicos de conjuntos difusos y bases de datos difusas. Posteriormente, se introduce la noción de conjuntos intervalo-valorados difusos para luego relacionar estos con SQL, dando origen a FSQL, para por último dar paso a IVFSQL. Lo que se busca es que los datos puedan ser tratados de una forma más tolerante de lo que ya se tratan si se siguen los principios de conjuntos difusos sobre una bases de datos. La idea es, esencialmente, aumentar y permitir la incertidumbre en los datos lo más posible. Es precisamente por este motivo que los conjuntos intervalo-valorados difusos buscan tratar la incertidumbre de la información gradual y establecer criterios que midan el grado de pertenencia de un elemento a un conjunto intervalo-valorado difuso.

Palabras clave

Bases de datos difusas, Conjuntos Intervalo-Valorados Difusos, Bases de datos relacionales con conjuntos intervalo-valorados difusos, FSQL, IVFSQL.

Abstract

When talking about relational databases, we think of a collection of organized, defined and structured data elements in a set of tables with columns and rows. In addition, the tables can be related to others through foreign keys, as well as there must be a consistency and integrity in the data. It is well known that in a relational database, the attributes are based on the restrictions of their domain, where in addition, the concept of domain follows the principles of classical set theory, which is based on the hypothesis that a value of an attribute must be an element that belongs to the set specified in its domain. On the other hand, when talking about fuzzy relational databases, we essentially talk about the aforementioned, but using the properties of the logic and fuzzy sets in the data integrity domain, that is, we talk about the flexible treatment of the data of a database and the representation of a set of fuzzy data in it.

To take advantage of the fact that the membership of an element to a fuzzy set is relative and partial, and to further lighten up the queries and conditions, the foundations of the relational databases with interval-valued fuzzy sets are introduced, giving rise to IVFSQL (Interval-Valued Fuzzy SQL). For this, we will study the basic concepts of fuzzy sets and fuzzy databases. Subsequently, the notion of interval-valued fuzzy sets are introduced in order to make a relationship between this and SQL, giving rise to FSQL, to finally open to IVFSQL. What is sought is that the data can be treated in a more tolerant way than they are already treated if the principles of fuzzy sets are followed on a database, as well as allowing the expansion of being able to fulfill with certain conditions partially. The idea is, essentially, to increase and allow uncertainty in the data as much as possible. It is precisely for this reason that the interval-valued fuzzy sets seek to deal with the uncertainty of the gradual information and establish criteria that could measure the degree of belonging of an element to an interval-valued fuzzy set.

Keywords

Fuzzy Databases, Interval-Valued Fuzzy Sets, Fuzzy Logic, Interval-Valued Fuzzy Databases, FSQL, IVFSQL.

Índice general

Índice de figuras

Capítulo 1

Introducción

Cuando se definieron los conjuntos clásicos y sus propiedades debido y por la naturaleza innata del humano de clasificar objetos de acuerdo a sus características, quedó un vacío semántico de los valores de pertenencia de los elementos al conjunto, dado por la existencia de universos de discurso ambigüos. Se requirió entonces de establecer un mejor criterio de pertenencia y una redefinición de los conjuntos para aquellos elementos que no toman estrictamente valores de verdad absoluto o falso absoluto. Dichos conjuntos son los llamados conjuntos difusos. La lógica de los conjuntos difusos nació cuando el profesor L.A. Zadeh publicó un artículo titulado *Fuzzy Sets* o Conjuntos Difusos [13]. Los conjuntos difusos son aquellos cuyos elementos tienen un grado de pertenencia al conjunto de manera parcial, a diferencia de los conjuntos clásicos, cuyo grado de pertenencia de los elementos al conjunto es siempre absoluta en forma afirmativa o en forma negativa.

Después de esto nacen, por naturaleza propia, los Conjuntos Intervalo-Valorados Difusos, que son una extensión y generalización de los Conjuntos Difusos. En ellos, la función de pertenencia asigna un intervalo de valores en la imagen de cada elemento de su dominio. Por este motivo, sus valores son consecuencia del producto cartesiano de los números reales comprendidos entre 0 y 1.

Teniendo como base a estos dos conjuntos, nació por un lado el SQL para Conjuntos Difusos, o FSQL (Fuzzy SQL), y por otro, el SQL para Conjuntos Intervalo-Valorados Difusos, o IVFSQL (Interval-Valued Fuzzy SQL). El primero fue publicado por el profesor J. Galindo [1], y el segundo es la aportación del presente trabajo, el cual consiste en el establecimiento de las nociones básicas para IVFSQL, motivado en la gestión (almacenamiento y procesamiento) de la información en el «formato» de los Conjuntos Intervalo-Valorados Difusos, que permite la representación de información gradual con cierto margen de error.

1.1. Introduction

When classical sets and their properties were defined due to the innate human nature for classifying objects according to their characteristics, there remained a semantic emptyness of the membership values to the elements, due to the existence of ambiguous universe of discourse. It was then necessary to establish a better criteria of belonging and a redefinition of the sets for those elements that do not strictly take on values of absolute belonging or absolute displaced. These sets are called fuzzy sets. The logic of fuzzy sets was born when Professor L.A. Zadeh published an article entitled *Fuzzy Sets* [13]. The fuzzy sets are those whose elements have a degree of belonging to the set in a partial way, unlike the classical sets, whose degree of belonging of the elements to the set is always absolute in an affirmative or negative form.

After this, the interval-valued fuzzy sets took place, becoming an extension and generalization of the fuzzy sets. In them, the membership function assigns a range of values in the image of each element of its domain. For this reason, their values are a consequence of the cartesian product of real numbers between 0 and 1.

Based on these two sets, on one hand, SQL for Fuzzy Sets, or FSQL (Fuzzy SQL), took place, and on the other, SQL for Interval-Valued Fuzzy Sets, or IVFSQL (Interval-Valued Fuzzy SQL) is taking place in this Master Degree Project. The first one was published by Professor J. Galindo [1], and the second one is the contribution of this project, which consists on the establishment of the basic notions for IVFSQL, motivated in the management (storage and processing) of information in the «format» of the Interval-Valued Fuzzy Sets, which allows the representation of gradual information with a certain margin of error.

Capítulo 2

Conjuntos difusos

2.1. Definición

Un conjunto difuso A se define como una función que mide el grado de pertenencia de un elemento x a A mediante un número real comprendido entre 0 y 1. Esto quiere decir que se relaja la restricción de que el grado de pertenencia valga alguno de estos dos valores (como en los conjuntos clásicos), y se permite que tome algún valor comprendido entre 0 y 1. De esta forma, se define un conjunto difuso A como [14]:

$$A = \{\mu A(x)/x : x \in \Omega, \mu A(x) \in [0,1] \in \mathbb{R}\}$$

Donde:

$$\Omega = \{x_1, ..., x_n\}$$

Pueden darse 2 casos extremos:

1. $\mu A(x) = 0$

2. $\mu A(x) = 1$

El primero indica que x no pertenece en absoluto al conjunto difuso A. Por su parte, el segundo indica que x pertenece totalmente al conjunto difuso A.

$\mu A(x)$ es por tanto la función de pertenencia de x al conjunto difuso A. Dicha función se define como: $\mu A(x)$: $x \rightarrow$ [0,1]. x representa el universo de discurso del conjunto A. Como previamente se dijo, los conjuntos difusos pueden representarse de dos maneras: en su modo probabilista o en su modo gradual.

Existen dos tipos de interpretaciones de la función de pertenencia a un conjunto difuso:

1. La probabilista: esta mide el grado de probabilidad de que un elemento de hecho pertenezca a un conjunto difuso.

2. La gradual: esta noción mide el grado (total o parcial) de pertenencia de un elemento a un conjunto difuso. Será total cuando el elemento pertenezca del todo al conjunto difuso, y parcial cuando su pertenencia al conjunto no sea completa.

2.2. Operaciones con conjuntos difusos

Así como con los conjuntos clásicos, para los conjuntos difusos también existen operaciones y propiedades básicas [7]. Sean A y B dos conjuntos difusos, y x un elemento del universo de discurso X. Se definen algunas operaciones para ellos:

1. Unión: $(A \cup B)(x) = \max \{A(x),B(x)\}$

2. Intersección: $(A \cap B)(x) = \min \{A(x),B(x)\}$

3. Negación (complemento a uno): $\neg\, A(x) = 1$ - $A(x)$

También, se definen algunas propiedades básicas para los mismos:

1. Unión Conmutativa: $A \cup B = B \cup A$

2. Intersección Conmutativa: $A \cap B = B \cap A$

3. Unión Asociativa: $(A \cup B) \cup C = A \cup (B \cup C) = A \cup B \cup C$

4. Intersección Asociativa: $(A \cap B) \cap C = A \cap (B \cap C) = A \cap B \cap C$

5. Idempotencia: $A \cup A = A$

6. Distributiva uno: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

7. Distributiva dos: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

8. Doble negación: $\neg(\neg A) = A$

9. Condiciones frontera o límite: $A \cup \emptyset = A$; $A \cup X = X$; $A \cap \emptyset = \emptyset$; $A \cap X = A$

Sabiendo estas operaciones básicas, se han originado unos modelos genéricos para las operaciones de unión e intersección, que deben cumplir algunas de las propiedades mencionadas anteriormente [7].

- Norma triangular o t-norma: Es una operación binaria $\boldsymbol{t}$: $[0,1]^2 \rightarrow$ [0,1] que cumple las siguientes propiedades [7]:

1. Conmutativa: $x\ \boldsymbol{t}\ y = y\ \boldsymbol{t}\ x$
2. Asociativa: $x\ \boldsymbol{t}\ (y\ \boldsymbol{t}\ z) = (x\ \boldsymbol{t}\ y)\ \boldsymbol{t}\ z$
3. Monotonicidad: Si $x \leq y$ y $w \leq z$ entonces $x\ \boldsymbol{t}\ w \leq y\ \boldsymbol{t}\ z$
4. Condiciones frontera: $x\ \boldsymbol{t}\ 0 = 0$, $x\ \boldsymbol{t}\ 1 = x$

- Conorma triangular o s-norma: Es una operación binaria $\boldsymbol{s}$: $[0,1]^2 \rightarrow$ [0,1] que cumple las siguientes propiedades [7]:

1. Conmutativa: $x\ \boldsymbol{s}\ y = y\ \boldsymbol{s}\ x$
2. Asociativa: $x\ \boldsymbol{s}\ (y\ \boldsymbol{s}\ z) = (x\ \boldsymbol{s}\ y)\ \boldsymbol{s}\ z$
3. Monotonicidad: Si $x \leq y$ y $w \leq z$ entonces $x\ \boldsymbol{s}\ w \leq y\ \boldsymbol{s}\ z$
4. Condiciones frontera: $x\ \boldsymbol{s}\ 0 = 0$, $x\ \boldsymbol{s}\ 1 = x$

- t-norma del mínimo: La función min ($\wedge$) es una t-norma, que corresponde a la operación de intersección en conjuntos clásicos cuyos grados de pertenencia están en [0,1]. Esta función es la extensión natural de la intersección en conjuntos difusos [7].

- t-conorma o s-norma del máximo: La función max ($\vee$) es una s-norma, que corresponde a la operación de unión en conjuntos clásicos cuyos grados de pertenencia están en [0,1]. Esta función es la extensión natural de la unión en conjuntos difusos [7].

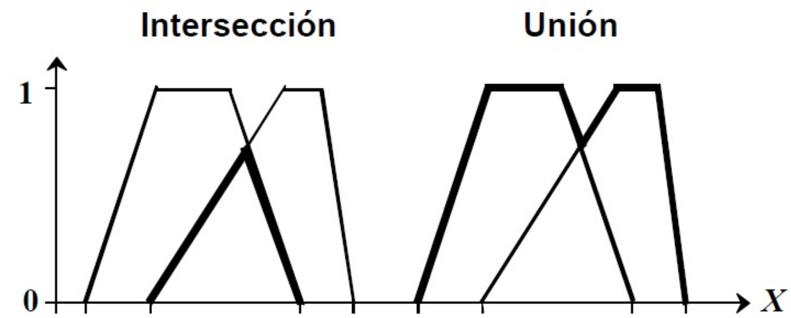

Figura 2.1: Función min y función max de la t-norma y s-norma [7].

- Relación entre t-normas y s-normas:

1. Para cada t-norma existe una s-norma dual o conjugada (y viceversa):

   - x $\boldsymbol{s}$ y = 1 - (1-x) $\boldsymbol{t}$ (1-y)
   - x $\boldsymbol{t}$ y = 1 - (1-x) $\boldsymbol{s}$ (1-y)

   Estas son las equivalentes a las leyes de De Morgan aplicada en teoría de conjuntos difusos, que en teoría de conjuntos clásicos aplican a la unión e intersección:

   - $A \cup B = \overline{\overline{A} \cap \overline{B}}$
   - $A \cap B = \overline{\overline{A} \cup \overline{B}}$

- Negación: Es el complemente de un conjunto difuso. $\boldsymbol{N}$: [0,1] $\rightarrow$ [0,1] el cual cumple las siguientes condiciones [7]:

1. Monotonicidad: $\boldsymbol{N}$ no es creciente.
2. Condiciones Frontera: $\boldsymbol{N}(0)$=1, $\boldsymbol{N}(1)$=0
3. Continuidad: $\boldsymbol{N}$ es una función continua.
4. Involución: $\boldsymbol{N}(\boldsymbol{N}(x))$=$x$, para $x \in$ [0,1]

2.3. Función de pertenencia, etiquetas y variables lingüísticas

Primero que nada hay que tener claro la diferencia entre etiquetas lingüísticas y variables lingüísticas. Las etiquetas lingüísticas identifican a un conjunto difuso. Las variables lingüísticas identifican a un universo de discurso. Por ende, las variables lingüísticas son variables que contienen etiquetas lingüísticas. A su vez, las funciones de pertenencia están atadas a las variables lingüísticas, las cuales son usadas como noción o concepto que se va a atribuir de forma difusa, como por ejemplo la «edad», la «altura», etc. Las etiquetas lingüísticas extienden a las variables lingüísticas, o universos de discurso, y son mayormente usadas para definir conjuntos difusos a partir de otros ya existentes.

Por ejemplo, la variable lingüística «edad» se puede extender con la etiqueta «viejo», y esta a su vez se puede extender usando modificadores de etiquetas lingüísticas para dar forma a la expresión «muy joven». Es el equivalente a los adverbios del lenguaje humano natural. En el fondo, una etiqueta lingüística lo que realmente hace es componer una función (la función de pertenencia) con la función de etiquetado lingüista, dando como resultado una nueva función compuesta. Más adelante se hablará un poco más sobre las etiquetas lingüísticas.

Figura 2.2: Diferentes etiquetas lingüísticas para representar la edad [1].

A continuación se muestra un cuadro con algunas de las funciones que se usan para componer a la función de pertenencia.

Nombre del modificador	Descripción del modificador
not	1-y
very (muy)	y^2
somewhat (algo)	$y^{\frac{1}{3}}$
more-or-less (más o menos)	$y^{\frac{1}{2}}$
extremely (extremadamente)	y^3

Un ejemplo de función de pertenencia sencillo puede ser el concepto de temperatura «alta». La función admite el hecho de que a mayor grados celsius de temperatura, más alta es la misma. También, establece que después de los 30 grados celsius, se considera que la temperatura es plenamente alta.

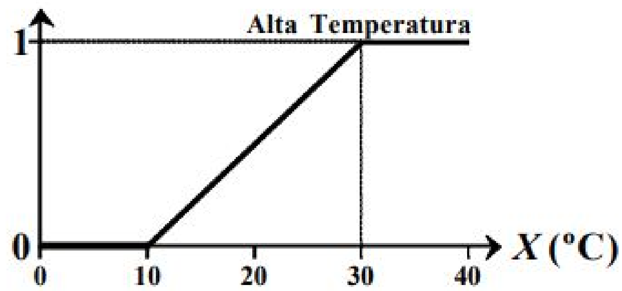

Figura 2.3: Función de pertenencia para la etiqueta «alta» de temperatura[1].

Existen también muchos tipos de funciones de pertenencia. Cada una de ellas nos ayuda a interpretar la relación de los elementos con el conjunto difuso.

- Función de pertenencia triangular: Definido por sus límites inferior a, superior b y el valor modal m, tal que $a<m<b$ [5]:

$$A(x) = \begin{cases} 0 & si \quad x \leq a \\ (x-a)/(m-a) & si \quad x \in (a, m] \\ (b-x)/(b-m) & si \quad x \in (m, b) \\ 0 & si \quad x \geq b \end{cases}$$

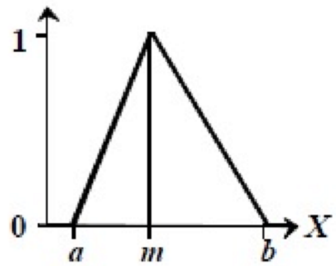

Figura 2.4: Función de pertenencia de tipo triangular[5].

- Función Gamma: Esta función se caracteriza por su rápido crecimiento a partir de a. También, se caracteriza por el hecho de que nunca toma el valor de 1, a pesar de tener una asíntota horizontal en 1 [5].

$$A(x) = \begin{cases} 0 & si \quad x \leq a \\ 1 - e^{-k(x-a)^2} & si \quad x > a \end{cases}$$

$$A(x) = \begin{cases} 0 & si \quad x \leq a \\ \frac{k(x-a)^2}{1+k(x-a)^2} & si \quad x > a \end{cases}$$

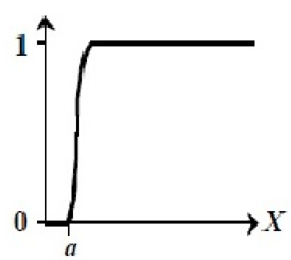

Figura 2.5: Función de pertenencia de tipo Gamma[5].

- Función S: Definida por sus límites inferior a y superior b, y el valor m, o punto de inflexión tal que a<m<b. El crecimiento es más lento cuanto mayor sea la distancia a-b [5].

$$A(x) = \begin{cases} 0 & si \quad x \leq a \\ 2\{(x-a)/(b-a)\}^2 & si \quad x \in (a, m] \\ 1 - 2\{(x-b)/(b-a)\}^2 & si \quad x \in (m, b) \\ 1 & si \quad x \geq b \end{cases}$$

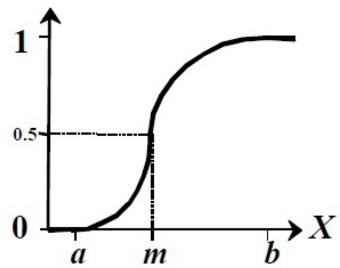

Figura 2.6: Función de pertenencia de tipo S[5].

- Función Gausiana: Definida por su valor medio m y el valor k>0. También conocida como Campana de Gauss. Cuanto mayor es k, más estrecha es la campana.

$$A(x) = e^{-k(x-m)^2}$$

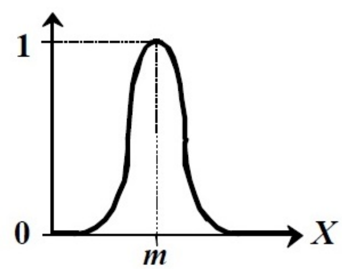

Figura 2.7: Función de pertenencia de tipo Gausiana[5].

- Función Trapezoidal: Definida por sus límites inferior a y superior d, y los límites de su soporte, b y c, inferior y superior respectivamente.

$$A(x) = \begin{cases} 0 & si \quad x \leq a \quad o \quad x \geq d \\ (x-a)/(b-a) & si \quad x \in (a, b] \\ 1 & si \quad x \in (b, c) \\ (d-x)/(d-c) & si \quad x \in (b, d) \end{cases}$$

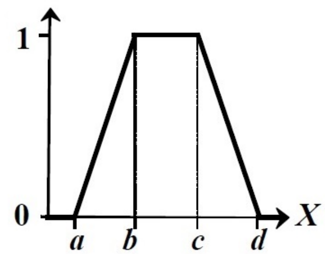

Figura 2.8: Función de pertenencia de tipo trapezoidal[5].

2.4. Cálculo de la función de pertenencia

Existen diversas maneras para calcular la función de pertenencia. En mayor medida ésta depende del modo en el que se quiera tratar la incertidumbre. Algunos de ellos son [5]:

1. Metodo Horizontal: Se basa en la respuesta que darían N *stakeholders*. Si A es un conjunto difuso y x es un elemento del universo de discurso, entonces este método se basa en resolver a la pregunta: ¿Es x un elemento que se considere compatible con el conjunto difuso A? La respuesta es estrictamente afirmativa o negativa. Por lo tanto, $A(x) =$ (Respuestas afirmativas)$/N$.

2. Metodo Vertical: Sea un α-corte definido como todos aquellos valores del universo de discurso X con grado mínimo, es decir:

   $$\alpha : A_\alpha = \{x \in X \mid A(x) \geq \alpha\}$$

   En este método se escogen varios valores para α, para construir sus α-cortes. Luego, la pregunta hecha para esos α pre determinados es: Identifique los elementos x de X que pertenecen al conjunto difuso A con grado no menor que α.

3. Método basado en la Especificación del Problema: Se requiere una función numérica que quiera ser aproximada. Luego, el margen de error se define como un conjunto difuso.

4. Método basado en la agrupación difusa: Los elementos del universo de discurso se clasifican en grupos no necesariamente disjuntos cuyos niveles de pertenencia de cada elemento a cada grupo representa un grado difuso. Uno de los algoritmos de agrupación difusa (*fuzzy clustering*

en inglés) más usados es el *Fuzzy Isodata*. El *fuzzy clustering* es una forma de *cluster* en la que cada elemento puede pertenecer a más de un *cluster* de modo que los elementos en el mismo *cluster* sean lo más similares posible, mientras que los elementos que pertenecen a diferentes *cluster* sean lo más diferentes posible [6].

Capítulo 3

Bases de datos difusas

Bien se sabe que el modelo relacional es un modelo de datos que se fundamenta en la teoría de conjuntos clásicos y en la lógica de predicados. Éste modelo es usado por el álgebra relacional basándose en su modelo para definir las operaciones necesarias para la manipulación de los datos, dejando bien en claro los pasos a seguir para computar datos relacionados. Dicho modelo, postulado en 1970 por Frank Codd, ha sido el más usado para la composición y la organización de las bases de datos clásicas o crisp. Actualmente existen varios modelos para hacer la misma organización y paradigma para las bases de datos difusas. La más usada y aceptada es el modelo GEFRED (GEneralized model for Fuzzy RElational Databases), ya que es la mejor considerada para representar impresición en una base de datos por, entre otros motivos, la tipología de los atributos difusos y por su adaptable implementación en una base de dato relacional difusa [9]. También han surgido otros modelos para bases de datos relacionales difusas, como el modelo Zemankova-Kandel, el modelo Umano-Fukami y el modelo Prade-Testemale, entre otros. Lo que todos estos modelos tienen en común es el afán de lograr el satisfactorio almacenamiento impreciso en una base de datos difusa, así como poder ejecutar operaciones sobre ellos de una manera lógica y razonable.

3.1. Características de los modelos de Bases de Datos Relacionales Difusos (BDRD)

Las BDRD poseen algunas características y particularidades que son dignas de estudiar y de sacar el mejor provecho. Todas tienen que ver con el hecho de que dichos modelos buscarán la mejor manera de modelar la imprecisión y para ello se valdrán de diferentes formas de manipulación y almacenamiento de datos. Algunas de ellos son:

- Se añade un grado del intervalo [0,1] bien sea en cada tupla de la relación, en cada valor de algún atributo o en un conjunto de valores [8]. Si es el primer caso, el grado pertenece a toda la tupla de la relación. Si se añade en cada valor de algún atributo entonces se considera el grado de ese valor concreto del atributo en la tupla a la que éste pertenezca. Si es la tercera opción, entonces el grado puede afectar a varios atributos al mismo tiempo, por lo tanto la información imprecisa debe ser considerada en todos ellos.

- La interpretación más común que se le da al grado difuso se basa o bien en el grado de probabilidad, que mide la probabilidad de pertenencia, o bien el grado de cumplimiento, también llamado grado gradual, que mide el grado de pertenencia de un elemento a un conjunto difuso.

Básicamente el modelo de BDRD nace de la necesidad de expresar información imprecisa debido a la naturaleza humana. Una forma muy común de hacerlo es usar etiquetas lingüísticas, como en capítulos anteriores se ha visto, para desde allí poder operar de una forma coherente y poco abstracta.

3.2. Modelo de BDRD de Buckles-Petry

Es un modelo de unificación mediante Relaciones de Similitud. Una Relación de Similitud establece en qué medida están relacionados (o son similares) los elementos en el dominio. Concretamente, mide la similitud entre un par de elementos en el dominio. Por ende, es una función S: $D \times D \rightarrow [0,1]$ [8]. En el caso extremo en que el valor de la función retorna *0*, implica que los elementos son absolutamente contrarios. Por su parte, si la función retorna *1* quiere decir que los elementos son iguales. Una Relación de Similitud cumple las propiedades Reflexiva, Simétrica y Transitiva. Las tuplas de la relación se agrupan en clases de equivalencia, ya que la propiedad transitiva hace que se puedan construir Clases de Equivalencia, las cuales son operadas en ellas mediante los operadores del álgebra relacional. El modelo de Buckles-Petry soporta varios dominios: escalares, numéricos y difusos.

3.3. Modelo de BDRD de Umano-Fukami

Es un Modelos Relacional sobre distribuciones de probabilidad, es decir, modelan la imprecisión de los datos mediante distribuciones de probabilidad [8]. Sobre el dominio X de un atributo, la distribución de probabilidad A indica que la probabilidad de que el atributo tome el valor $x \in X$ es $A(x)$. Un ejemplo de este modelo puede referir a cuatro etiquetas lingüísticas asociadas a cuatro distribuciones de probabilidad para el atributo «Altura» [8].

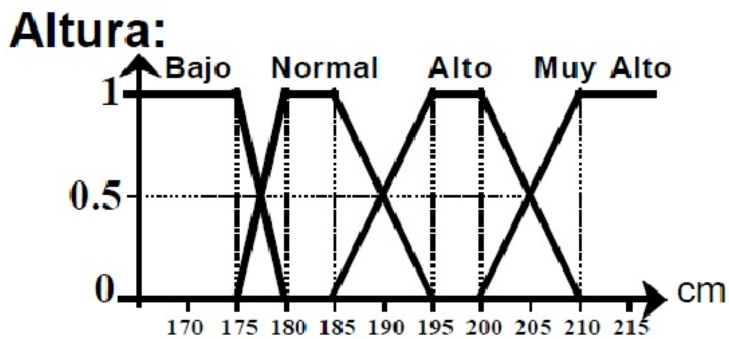

Figura 3.1: Ejemplo del modelo de BDRD de Umano-Fukami [8]

El modelo Umano-Fukami definen también las operaciones del Álgebra Relacional: Unión, Diferencia, Producto Cartesiano, Proyección y Selección, Intersección y División. Existen también valores especiales de distribuciones de probabilidad en el modelo:

- Unknown: Desconocido, pero aplicable: $A(x) = 1$, $\forall\ x \in X$.
- Undefined: Valor no aplicable: $A(x) = 0$, $\forall\ x \in X$.
- Null: Ignorancia total, no se sabe si es o no aplicable.

3.4. Modelo de BDRD de Prade-Testemale

Es un modelo posibilístico que añade un elemento e al dominio de todos los atributos, que representa el caso en el que el atributo no sea aplicable [8]. La forma en que manipula los datos es parecida a la empleada en el modelo Umano-Fukami. También establece medidas de probabilidad y necesidad para las condiciones en las consultas.

Información	Modelo Prade-Testemale	Modelo Umano-Fukami
Sabemos el dato y este es *crisp*: **c**	$A(e)=0; A(c)=1;$ $A(x)=0, \forall x \in X, x \neq c;$	$A(x) = \{1/c\};$
Desconocida (pero aplicable)	$A(e)=0;$ $A(x)=1, \forall x \in X;$	**Unknown**
No aplicable	$A(e)=1; A(x)=0, \forall x \in X;$	**Undefined**
Ignorancia total	$A(x)=1, \forall x \in X \cup \{e\};$	**Null**
Rango **[m,n]**	$A(x)=0, \forall x \notin [m,n]$ o $x = e;$ $A(x)=1, \forall x \in [m,n];$	$A(x)=0, \forall x \notin [m,n];$ $A(x)=1, \forall x \in [m,n];$
Distribución de Posibilidad ***B***	$A(e)=0;$ $A(x)=B(x), \forall x \in X;$	$A(x)=B(x), \forall x \in X;$
Posibilidad de que **no** sea aplicable es λ y si lo es vale ***B***.	$A(e)=\lambda;$ $A(x)=B(x), \forall x \in X;$	**No representable**

Figura 3.2: Modelo de BDRD de Prade-Testemale [8]

3.5. Modelo de BDRD GEFRED (GEneralized model for Fuzzy RElational Databases)

Es un modelo propuesto en 1994 por Medina J.M., Pons O. y Vila A., de la Universidad de Granada. Este modelo se basa en la admisión de dominios de distribuciones de probabilidad, sin dejar abierta la posibilidad de tratar dominios numéricos y escalares. El modelo también incluye tratamiento para los valores Null, Unknown y Undefined con el mismo sentido que en el modelo de Umano-Fukami [8]. Algunos ejemplos que se pueden tratar en el modelo son los siguientes:

- Un escalar: Comportamiento=bueno.
- Un número: Precio=150.
- Distribución de probabilidad sobre un dominio de los escalares Comportamiento = {0.8/bueno, 0.5/regular, 0.3/malo}.

Parte de los elementos de GEFRED que organizan toda esta información son el Dominio Difuso Generalizado, la Relación Difusa Generalizada y el Comparador Difuso Generalizado.

3.5.1. Dominio Difuso Generalizado

Sea D un dominio de discurso, y $P(D)$ el conjunto de todas las distribuciones de probabilidad definidas sobre D, el Dominio Difuso Generalizado (D_G) es aquel que cumple $D_G \subseteq P(D)$.

3.5.2. Relación Difusa Generalizada

La Relación Difusa Generalizada (R_{FG}) es una relación cuyos atributos tienen un Dominio Difuso Generalizado [8]. Está formada por dos conjuntos: una cabecera H y un cuerpo B. La cabecera es el nombre de cada uno de los n atributos, sus dominios y sus atributos de compatibilidad. El cuerpo incluye los valores de m tuplas. En una relación no pueden existir tuplas repetidas. De esta forma, la Relación Difusa Generalizada queda definida de la siguiente manera:

$$R = \begin{cases} H = \{(A_1 : D_1[, C_1]), ..., (A_n : D_n[, C_n])\} \\ B = \{(A_1 : \tilde{d}_{il}[, c_{il}]), ..., (A_n : \tilde{d}_{in}[, c_{in}])\} \end{cases}$$

3.5.3. Comparador Difuso Generalizado

El comparador usado por GEFRED es un comparador general que exige que se respeten los resultados de los comparadores clásicos cuando se comparan distribuciones de probabilidad, así como cuando se comparan valores numéricos o escalares [8]. Para conjuntos difusos, la igualdad difusa de necesidad y posibilidad es la usada por defecto y la más usada (sección 5.2.1 y 5.2.2). Sin embargo, también existen los comparadores de Mayor o Igual Difuso, Menor Difuso, etc.

3.6. Álgebra Relacional Difusa basada en GEFRED

El modelo GEFRED reseña un álgebra relacional difusa que intenta dominar la manipulación de las relaciones difusas generalizadas. Las operaciones de unión, intersección, entre otros, buscan operar sobre las relaciones difusas generalizadas de forma razonable. GEFRED se centra en cómo calcular los grados de compatibilidad de la relación difusa [8].

El modelo GEFRED define varias operaciones difusas, las cuales son:

3.6.1. Unión Difusa

Sean dos relaciones R y S. Para poder realizar una operación $R \cup S$ sobre ellas se requiere que ambas relaciones tengan la misma cantidad y tipo de atibutos. El resultado de aplicar dicha operación es otra relación difusa con el mismo encabezado y con las tuplas de ambas relaciones a excepción de las repetidas [8]. Si existe una redundancia entre alguna tupla de las relaciones, se toma como atributo de compatibilidad el máximo de ambos atributos de compatibilidad. Tiene la propiedad asociativa y conmutativa.

3.6.2. Intersección Difusa

Sean dos relaciones R y S. Para que pueda llevarse a cabo una operación $R \cap S$ se debe también cumplir el requisito de que ambas relaciones tengan la misma cantidad y tipo de atibutos. El resultado de aplicar esta operación es otra relación difusa con la misma cabecera y conteniendo las tuplas que existen a la vez en ambas relaciones. Si existen los dos atributos de compatibilidad de las dos relaciones, el atributo de compatibilidad de la relación resultante será el valor mínimo de ambos atributos de compatibilidad [8]. También tiene la propiedad asociativa y conmutativa.

3.6.3. Diferencia Difusa

Sean dos relaciones R y S. Igualmente, para que pueda llevarse a cabo una operación de intersección se debe también cumplir el requisito de que ambas relaciones tengan la misma cantidad y tipo de atibutos. El resultado es otra relación difusa con la misma cabecera y con las tuplas que existen en la primera relación y no existen en la segunda. El Grado de Compatibilidad para cada tupla del resultado se calcula como: $\min\{C_R, 1 - C_S\}$, donde C_R y C_S son los grados de compatibilidad de R y S respectivamente [8].

3.6.4. Producto Cartesiano Difuso

Sean dos relaciones R y S. El requisito a cumplir para poder llevar a cabo una operación cartesiana $R \times S$ es el mismo que para los anteriores operadores mencionados. El resultado es otra relación difusa teniendo como cabecera la unión de las cabeceras de R y S, y con cuerpo el producto cartesiano de los cuerpos de R y S [8].

3.6.5. Proyección Difusa

Sea una relación R. La operación $\Pi_X(R)$ no exige algún requisito previo desde el punto de vista del tipo de atributos. El resultado es otra relación difusa con la cabecera siendo los atributos de X, y con cuerpo los valores de esos atributos en las tuplas de R, sin dejar en ella las tuplas repetidas o redundantes.

3.6.6. Selección Difusa

Sea una relación R. Al aplicar una operación $\sigma_C(R)$ el resultado es otra relación difusa con la misma cabecera y con las tuplas que cumplen la condición C, con un umbral mínimo g establecido en la misma condición [8]. El grado de compatibilidad para cada atributo del resultado es el grado con el que cada atributo satisface la condición.

3.6.7. Join Difuso

Sean dos relaciones R y S. La operación de join difuso es el resultado de aplicar una selección sobre una condicion C al producto cartesiano de R y S [8]. Es decir, $R \bowtie_C S = \sigma_C(R \times S)$.

3.7. Cálculo Relacional Difuso

El cálculo Relacional Difuso fue definido con la intención de sentar las bases para poder definir un lenguaje, usando cuantificadores difusos, que permita realizar consultas de ámbito difuso. Actualmente el más aceptado en este ámbito es la propuesta de Cálculo Relacional para el modelo GEFRED.

3.7.1. Cálculo de Dominios

Un dominio es el conjunto de valores que pueden tomar uno o más atributos, en donde cada atributo se define basado en un dominio. En este ámbito, se define un formato de expresiones como sigue: $\{x_1, .., x_n | \Psi(x_1, .., x_n)\}$. Del lado izquierdo se fijan los atributos que pretenden aparecer en el resultado. Del lado derecho se definirá una Formula Bien Formada (FBF o *Well Formed Formula* en inglés) [8]. Las FBF expresan una condición que debe satisfacerse por todas las tuplas del resultado. A su vez, está basado usando los conceptos de Cálculo de Predicados de primer orden, es decir, predicados que admiten valores verdadero o falso, ya que una tupla puede estar completamente o no estar completamente en el resultado final de una consulta difusa. Esto no se debe confundir con el hecho de que, a pesar de que una tupla pertenezca al resultado de una consulta difusa, esta puede tener un grado de pertenencia parcial (o no necesariamente total). Las Fórmulas Bien Formadas están definidas partiendo de un Grado de Cumplimiento (Δ) y un Umbral (γ), donde $\Delta \geq \gamma$. En este contexto, el Umbral constituye el valor de verdad mínimo que debe tener el Grado de Cumplimiento para que este último sea verdad. El Grado de Pertenencia puede ser de dos tipos: de pertenencia o de comparación. En el caso de la primera, la FBF mide el grado con el que la tupla $\{x_1, .., x_n\}$ pertenece a la relación R, y en el caso en que sea comparativa, expresa el grado de cumplimiento en el que dos atributos se relacionan entre sí usando algún comparador difuso.

3.7.2. Cuantificadores difusos

Existen dos tipos definidos de cuantificadores difusos: los absolutos y los relativos. Los primeros son interpretados siguiendo una distribución de probabilidad trapezoidal en el intervalo [0,+∞] y los segundos como distribuciones de probabilidad trapezoidales en [0,1] [2]. Estos valores se almacenarán en la Base de Metaconocimiento Difuso. Ambos, tanto los cuantificadores difusos relativos como los absolutos, se pueden constituir de cualquiera de las siguientes maneras:

- Cuantificador Fuzzy[ρ] (condición-difusa) THOLD τ
- Cuantificador Fuzzy[ρ] (condición-difusa1) ARE (condición-difusa2) THOLD τ

τ es un umbral, como el definido para los comparadores difusos, en el intervalo [0,1].

Capítulo 4

Conjuntos Intervalo-Valorados Difusos

Los conjuntos intervalo-valorados difusos nacen como una natural extensión y por ende una generalización de los conjuntos difusos. Los denominaremos como IVFS por sus siglas en inglés (Interval Valued Fussy Sets).
Mientras que la función de pertenencia de los conjuntos difusos asignan un único valor de su dominio a la imagen de dicha función, en los IVFS asignan un subintervalo del intervalo [0,1] a cada elemento del universo.

Dado que un conjunto difuso generaliza el concepto de membresía de un elemento a un conjunto, un conjunto intervalo-valorado difuso intenta flexibilizar dicha generalización, otorgándole a la función de pertenencia más grado de incertidumbre e imprecisión. Por lo tanto, dado un universo de discurso U y una función de pertenencia de un conjunto intervalo-valorado difuso A, dicha función de pertenencia no es más que una función con recorrido en el producto cartesiano de los números reales comprendidos entre 0 y 1: $\mu A(x)$: $U \subseteq \mathbb{R} \rightarrow [0,1]\times[0,1]$

Su uso y aplicación ha venido en auge en las últimas décadas, haciendo un claro avance en todo lo relacionado con problemas de toma de decisiones; robótica, procesamiento de imágenes, economía, entre otros [4].

Existe, además de las dos mencionadas en la introducción a los conjuntos difusos (capítulo 2), una manera alterna de interpretar la función de pertenencia a un conjunto difuso. La misma se basa en los conjuntos difusos evaluados en intervalos difusos [5]. La interpretación que se le da a estos es: x_i pertenece al conjunto difuso A con un grado de pertenencia X, y la probabilidad de que eso sea cierto es de Y.

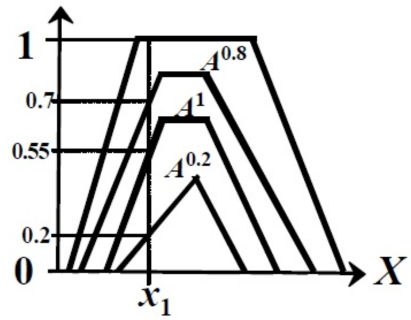

Figura 4.1: Ejemplo de conjunto difuso evaluados en intervalo difuso.

El ejemplo de la figura 4.1 se lee de la siguiente manera: x_1 pertenece al conjunto A con grado 0.8, y la certeza de que eso sea cierto es de 0.7.

4.1. La función IVCDEG para Conjuntos Intervalo-Valorados Difusos

En IVFS nos centraremos en la asunción de que existen elementos de uno o más conjuntos intervalo-valorados difusos en un universo de discurso finito o discreto, definido mediante los elementos:

$$\delta = \{x_1, x_2, ..., x_n\}$$

y los cuales, el conjunto intervalo-valorado difuso A sería de la forma:

$$A = \{\delta_1/x_1, \delta_2/x_2, ..., \delta_n/x_n\}$$

donde δ_i representa el intervalo de grado de pertenencia del elemento x_i, con i=1,..n, y donde «/» tiene el sentido de asociación de valores. δ define una función de pertenencia la cual llamaremos IVCDEG (Inter-Valued Compatibility Degree), el cual su argumento es un elemento y del que produce como resultado un par ordenado con el grado de compatibilidad o cumplimiento de la condición de la consulta para el elemento que se indica. Dicho par ordenado es de la forma (a,b), donde $a \in \{0,1\}$, $b \in \{0,1\}$ y $0 \leqslant a \leqslant b \leqslant 1$.

Por su lado, tomando en cuenta la función de pertenencia IVCDEG, la definición para un conjunto difuso intervalo-valorado A se establece de la siguiente manera:

$$A = \{\mu A(x)/x : x \in \Omega, \mu A(x) \in [0, 1] \times [0, 1]\}$$

Donde:

$$\Omega = \{x_1, ..., x_n\}$$

De esta forma pues, queda establecida que la función IVCDEG se define como:

$\mu A(x)$: $x \rightarrow [0,1]\times[0,1]$. Por lo tanto, la imagen de dicha función son pares ordenados (a,b) como los mencionados anteriormente.

4.2. Fundamentos de IVFS

En conjuntos difusos, una función $n : [0,1] \rightarrow [0,1]$ tal que $n(0) = 1$ y $n(1) = 0$, decreciente, es llamada *negación*. Si una función negación es estrictamente decreciente y continua, entonces se llama *negación estricta*. Si $n(n(x))=x$ para todo $x \in [0,1]$, entonces n es involutivo [20].

Se denotará por $FS_s(U)$ al conjunto de todos los conjuntos intervalo-valorados definidos en un universo finito U. Dado un conjunto difuso $A = \{(u, \mu_A(u))|u \in U\} \in FS_s(U)$, la expresión dada por $A_n = \{(u, n(\mu_A(u)))|u \in U\}$ tiene el significado de complemento del conjunto difuso A, donde n es una negación.

Se denota por $L([0,1])$ al conjunto de todos los subintervalos del intervalo [0,1], por lo tanto, $L([0,1]) = \{\boldsymbol{x} = [\underline{x},\bar{x}] \mid (\underline{x},\bar{x}) \in [0,1]^2$ y $\underline{x} \leq \bar{x}\ \}$ [20]. Sean $\boldsymbol{x}$, $\boldsymbol{y} \in L([0,1])$, $\boldsymbol{x} \leq_L \boldsymbol{y}$ si y sólo si $\underline{x} \leq \underline{y}$ y $\bar{x} \leq \bar{y}$. De esta forma, $L([0,1])$ es un conjunto parcialmente ordenado con respecto a la relación $\leq_L$ [15]. En dicha relación el elemento ínfimo es $0_L = [0,0]$ y el superior es $1_L = [1,1]$.

G. Deschrijver [21] define a los conjuntos intervalo-valorados como un conjunto A que se define como $A : U \rightarrow L([0,1])$, donde $U \neq \emptyset$. En este sentido, $A(u) = [\underline{A}(u), \bar{A}(u)] \in L([0,1])$ es el grado de membresía de $u \in U$.

4.3. Creación de un IVFS a partir de un conjunto difuso

Una de las maneras para crear un IVFS a partir de un conjunto difuso, propuesta por *Tizhoosh* [22], se basa en determinar el umbral de una imagen Q a partir de un elemento del dominio. La idea sería construir el umbral a partir de un intervalo $[(\mu_Q)^{\alpha}, (\mu_Q)^{\frac{1}{\alpha}})]$, donde $\alpha \in (1, \infty)$ y μ es el nivel de membresía del elemento, tal y como se muestra en la figura 4.2:

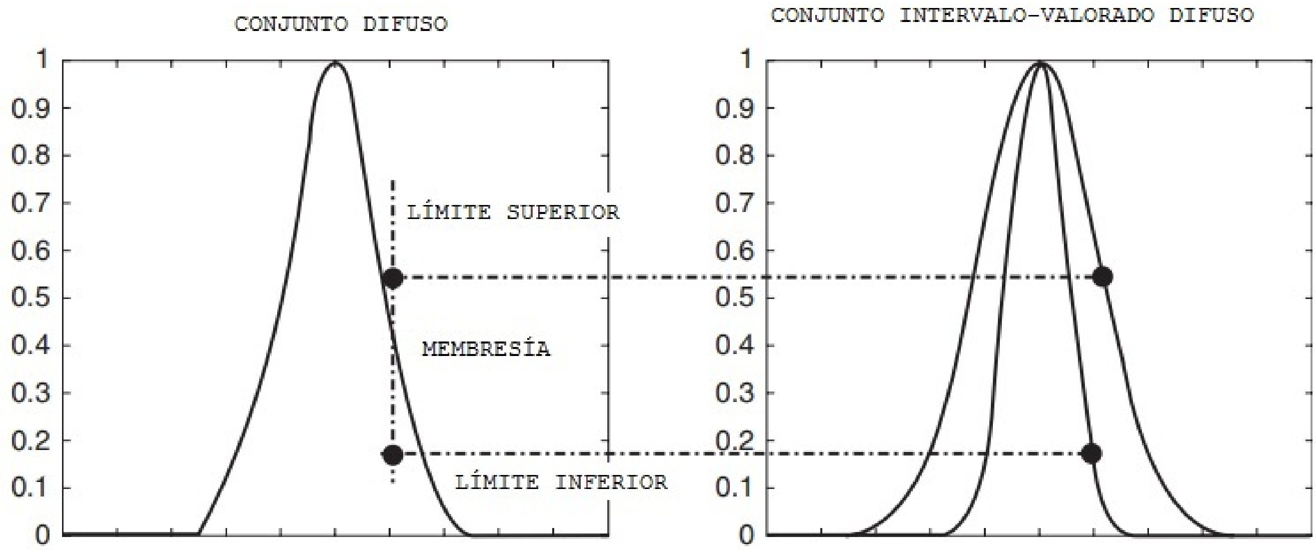

Figura 4.2: Creación de un IVFS a partir de un conjunto difuso [22]

4.4. Creación de un conjunto difuso a partir de un IVFS

Se ha intentado también sentar las bases para hacer posible lo inverso: crear un conjunto difuso a partir de un conjunto intervalo-valorado difuso. Para ello, se ha definido K_α, con $\alpha \in [0,1]$, como una función tal que $K_\alpha : L([0,1]) \rightarrow [0,1]$ y tal que satisface las siguientes condiciones [23]:

1. Si $\underline{x} = \bar{x}$, entonces $K_\alpha(\mathbf{x}) = x$.

2. $K_0(\mathbf{x}) = \underline{x}$, $K_1(\mathbf{x}) = \bar{x}\ \forall \mathbf{x} \in L([0,1])$.

3. Si $\mathbf{x} \leq_L \mathbf{y}$, con $\mathbf{x}, \mathbf{y} \in L([0,1])$, entonces $K_\alpha(\mathbf{x}) \leq K_\alpha(\mathbf{y})\ \forall \alpha \in [0,1]$.

4. $K_\alpha(\mathbf{x}) \leq K_\beta(\mathbf{x})$ si y sólo si $\alpha \leq \beta\ \forall \mathbf{x} \in L([0,1])$, con $\alpha, \beta \in [0,1]$.

El operador K_α permite asociar un conjunto intervalo-valorado *IVFS* a un conjunto difuso *FS*:

$$K_\alpha = \mathit{IVFS(U)} \rightarrow \mathit{FS(U)}$$

Dado por:

$$K_\alpha(A) = \{(u, \mu_{K_\alpha(A)}(u)) = K_\alpha(A(u)) = K_\alpha([\underline{A}(u), \bar{A}(u)])|u \in U\}$$

4.5. Conectores para conjuntos intervalo-valorados difusos

Las operaciones aquí expuestas utilizan y se basan en las definiciones hechas en las previas secciones. Sean A y B dos conjuntos intervalo-valorados difusos, las operaciones de unión, intersección y complemento se definen, para todo u perteneciente al universo U, de la siguiente manera [15]:

1. Unión: $(A \cup B)(u) = [\max(K_0(A(u)), K_0(B(u))), \max(K_1(A(u)), K_1(B(u)))]$
2. Intersección: $(A \cap B)(u) = [\min(K_0(A(u)), K_0(B(u))), \min(K_1(A(u)), K_1(B(u)))]$
3. Complemento: $A_n(u) = [1\text{-}K_1(A(u)), 1\text{-}K_0(A(u))]$

4.5.1. Negaciones para conjuntos intervalo-valorados difusos

Una Negación Intervalo-Valorada se define como una función $N : L([0,1]) \to L([0,1])$ tal que $N(1_L)$=0_L y $N(0_L)$=1_L [19]. Si para todo $x \in L$([0,1]), $N(N(x))$=x, entonces se dice que N es involutivo.

Partiendo de la definición, se define el siguiente teorema:

Sea la función $N : L([0,1]) \to L([0,1])$ dada por $N(\mathbf{x}) = [n(K_1(\mathbf{x})), n(K_0(\mathbf{x}))]$, donde además n : [0,1] $\to$ [0,1] representa una negación. Entonces, bajo estas condiciones, N es una Negación Intervalo-Valorada.

4.6. t-normas y t-conormas para conjuntos intervalo-valorados

Así como para los conjuntos difusos, las t-normas son usadas para modelar la intersección de dos conjuntos difusos, y las t-conormas para modelar la unión. Similarmente, se puede modelar la unión e intersección usando t-normas y t-conormas intervalo-valoradas.

A modo de definición, una función $\boldsymbol{T} : (L([0,1]))^2 \to L([0,1])$ se dice que es una t-norma intervalo-valorada si es conmutativa, asociativa, incremental y posee un elemento neutro tal que 1_L = [1,1]. Por su parte, una función $\boldsymbol{S} : (L([0,1]))^2 \to L([0,1])$ se considera una t-conorma si es conmutativa, asociativa, incremental y posee un elemento neutro tal que 0_L = [0,0] [16].

Sean A y B dos conjuntos intervalo-valorados, y u perteneciente al universo U, entonces:

1. $A \cap_{\mathbf{T}} B(u) = \boldsymbol{T}(A(u), B(u))$

2. $A \cup_{\mathbf{S}} B(u) = \boldsymbol{S}(A(u), B(u))$

Donde $\boldsymbol{T}$ es una t-norma y $\boldsymbol{S}$ es una t-conorma.

4.6.1. t-normas t-representables y t-conormas t-representables para conjuntos intervalo-valorados

La definición de t-norma y t-conorma para conjuntos intervalo-valorados permitió a Cornelis [17] y Deschrijver [18] introducir los conceptos de t-normas t-representables y t-conormas t-representables para conjuntos intervalo-valorados.

Una t-norma intervalo-valorada se dice ser t-representable si existen dos t-normas T_a y T_b en [0,1] tal que $\boldsymbol{T}(\mathbf{x},\mathbf{y}) = [T_a(K_0(\mathbf{x}),K_0(\mathbf{y})),\ T_b(K_1(\mathbf{x}),K_1(\mathbf{y}))]$ para todo $\mathbf{x}$, $\mathbf{y} \in L([0,1])$. Por su parte, una t-conorma intervalo-valorada se dice ser t-representable si existen dos t-conormas S_a y S_b en [0,1] tal que $\boldsymbol{S}(\mathbf{x},\mathbf{y}) = [S_a(K_0(\mathbf{x}),K_0(\mathbf{y})),\ S_b(K_1(\mathbf{x}),K_1(\mathbf{y}))]$ para todo $\mathbf{x}$, $\mathbf{y} \in L([0,1])$ tal que $\mathbf{x} = [K_0(\mathbf{x}),K_1(\mathbf{x})]$, $\mathbf{y} = [K_0(\mathbf{y}),K_1(\mathbf{y})]$.

4.6.2. Idempotencia y absorción

Para la idempotencia, se tienen las siguientes propiedades [15]:

1. $\boldsymbol{T}(\mathbf{x},\mathbf{x})=\mathbf{x}$ para todo $\mathbf{x} \in L([0,1])$ si y sólo si $\mathbf{T} = \mathbf{T}_{min,min}$

2. $\boldsymbol{S}(\mathbf{x},\mathbf{x})=\mathbf{x}$ para todo $\mathbf{x} \in L([0,1])$ si y sólo si $\mathbf{S} = \mathbf{S}_{max,max}$

Para la absorción, se tienen las siguientes propiedades [15]:

1. $\boldsymbol{T}(\mathbf{x},\mathbf{S}(\mathbf{x},\mathbf{y}))=\mathbf{x}$ para todo $\mathbf{x}$, $\mathbf{y} \in L([0,1])$ si y sólo si $\mathbf{T}$ es idempotente.

2. $\boldsymbol{S}(\mathbf{x},\mathbf{T}(\mathbf{x},\mathbf{y}))=\mathbf{x}$ para todo $\mathbf{x}$, $\mathbf{y} \in L([0,1])$ si y sólo si $\mathbf{S}$ es idempotente.

Capítulo 5

FSQL

FSQL son las siglas de Fuzzy SQL, que no es más que el lenguaje SQL difuso. Es una extensión del lenguaje SQL clásico que ha sido orientado al uso con tratamientos de bases de datos difusas, las cuales usan los conceptos y los principios de los conjuntos difusos y la lógica difusa. El lenguaje FSQL es un súper conjunto de SQL, pues contiene y maneja toda su sintaxis y semántica, y añade nuevos comparadores, cuantificadores difusos, nuevas constantes, nueva noción de atributos y etiquetas lingüísticas. También, FSQL trabaja sobre un servidor FSQL, que interactúa con una base de metaconocimiento difuso para establecer y definir todo el esqueleto y la estructura difusa que se van a emplear en las consultas, así como el manejo de los atributos difusos propiamente dicho. Una consulta FSQL es traducida por el servidor FSQL a una consulta SQL, y esta última ejecutada por un manejador de base de datos SQL.

5.1. Las consultas flexibles

Una consulta a una base de datos se considera flexible si permite *vaguedad* e *imprecisión* sobre la misma. Todo esto con la intención de proponer un modelo para poder representar y dar respuesta a propuestas que impliquen información y manipulación difusa. Su hilo conductor se basaría en la manipulación de la consulta a través de la elección del comparador y la proporción de semejanza difusa a aplicar. Sus bases se fundamentan en el álgebra y cálculo relacional difuso, principalmente en un modelo llamado GEFRED.

5.1.1. Consultas flexibles a Bases de Datos Difusas

Las consultas flexibles sobre bases de datos difusas es un área ya estudiada y planteada. Estas consultas mantienen el paradigma de la imprecisión, más allá de en los datos, en las consultas. El objetivo es poder alimentar la idea de arrojar resultados cercanos o aproximados a lo que la consulta difusa puede expresar. La idea es que tanto los atributos, los resultados y las condiciones permitan ser lo más flexible posible en un contexto donde se permita y se acepte este tipo de tratamiento de los datos [5].

5.1.2. Consultas flexibles a Bases de Datos Clásicas

Actualmente no está muy estudiada y propuesta esta posibilidad, pero un sistema gestor de bases de datos difusas debería considerar esta propuesta, pues la gran mayoría de los gestores de bases de datos actuales son gestores crisp (clásicos) [5]. Esto no quiere decir que sería necesario transformar el gestor de base de datos crisp a un gestor de bases de datos difusas; sólo adaptar lo que ya existe en el mismo tolerando la posibilidad de un sistema interno de consultas difusas.

5.1.3. Condiciones flexibles

Una de las mayores particularidades de las bases de datos difusas son las condiciones flexibles. Esto quiere decir que los elementos resultantes de la consulta no necesariamente cumplen con la condición a cabalidad; puede cumplirla de manera parcial. Lo que va a definir si se cumple la parcialidad (o totalidad) de la condición será la función de membresía o pertenencia que establece la medida de compatibilidad difusa.

5.2. Comparadores difusos para FSQL

Como se ha dicho, existen en FSQL nuevos comparadores que permiten manipular los atributos y la lógica difusa que concierne a las bases de datos difusas. Los aquí mostrados partirán de la base propuesta en [12]. Un operador de igualdad operando sobre conjuntos clásicos no es lo mismo que un operador de igualdad sobre conjuntos difusos; estos últimos realizan una consulta flexible de manera más vaga, más imprecisa, por lo que su resultado será más amplio y abierto y dependerá de los elementos que superen el umbral de pertenencia de la función de pertenencia. Los comparados difusos llevan una letra «F» en su inicio, denotando la palabra «Fuzzy». Por ejemplo, «FEQ» significa «Fuzzy Equal», o igualdad difusa. Estos nuevos comparadores son los comparadores de posibilidad y de necesidad mostrados en la siguiente imagen:

POSIBILIDAD	**NECESIDAD**	**SIGNIFICADO**
FEQ	NFEQ	Pos./Nec. igual
FGT (FGEQ)	NFGT (NFGEQ)	Pos./Nec. Mayor (o igual) que
FLT (FLEQ)	NFLT (NFLEQ)	Pos./Nec. Menor (o igual) que
MGT (MLT)	NMGT (NMLT)	Pos./Nec. Mucho Mayor (Menor) que

Además, existen también límites de restrictividad entre los operadores [5]. Hay operadores que son más *restrictivos* que otros. Si C1 y C2 son dos comparadores, si el comparador C2 recupera siempre mayor o igual cantidad de tuplas y el grado de cumplimiento de cada tupla es mayor o igual con C2 que con C1, entonces C1 es más restrictivo que C2.

FAMILIA	**DE MAYOR A MENOR RESTRICTIVIDAD**
Igual Difuso	NFEQ > FEQ
Mayor Difuso	NFGT > FGT > NFGEQ > FGEQ
Menor Difuso	NFLT > FLT > NFLEQ > FLEQ
Mucho Mayor Difuso	NMGT > MGT
Mucho Menor Difuso	NMLT > MLT

Sean A y B dos distribuciones de probabilidad que deseamos comparar dentro de cada uno de los siguientes comparadores, entonces:

5.2.1. El comparador de posibilidad difuso FEQ

Es el comparador *Posiblemente Igual*. Si se quiere aplicar el comparador FEQ a las distribuciones de probabilidad A y B, se tiene que A FEQ B:

$$Poss(A, B) = sup_{x \in X}\{min(A(x), B(x))\}[2]$$

Donde X es el dominio subyacente a ambas distribuciones de probabilidad, $A(x)$ es el grado de posibilidad para el valor $x \in X$ en la distribución de probabilidad A y $B(x)$ es el grado de posibilidad para el valor $x \in X$ en la distribución de probabilidad B.

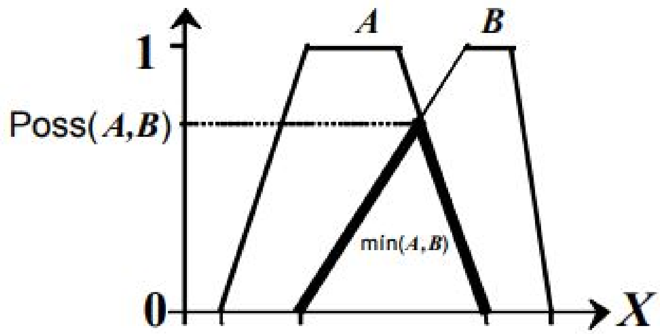

Figura 5.1: Comparador de posibilidad difuso FEQ

5.2.2. El comparador de necesidad difuso NFEQ

Es el comparador *Necesariamente Igual*. Si se quiere aplicar el comparador NFEQ a las distribuciones de probabilidad A y B, se tiene que A NFEQ B:

$$Nec(A,B) = inf_{x \in X}\{max(1 - A(x), B(x))\}[2]$$

Donde X es el dominio subyacente a ambas distribuciones de probabilidad, $A(x)$ es el grado de posibilidad para el valor $x \in X$ en la distribución de probabilidad A y $B(x)$ es el grado de posibilidad para el valor $x \in X$ en la distribución de probabilidad B. También, en los comparadores difusos de necesidad se niega la distribución de probabilidad de la parte izquierda de la comparación.

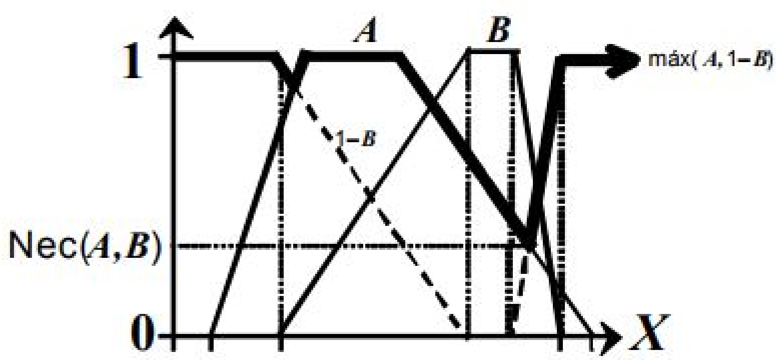

Figura 5.2: Comparador de necesidad difuso FEQ

Para el resto de comparadores de posibilidad se utiliza FEQ pero modificando la distribución de probabilidad de la derecha (B) según indique el comparador, mientras que para el resto de comparadores de necesidad el proceso es similar al anterior pero a su vez negando la distribución de probabilidad de la izquierda A [5].

5.2.3. El comparador de posibilidad difuso FGEQ

Es el comparador *Posiblemente Mayor o Igual*. Si se quiere aplicar el comparador FGEQ a las distribuciones de probabilidad A y B, se tiene que A FGEQ B [2]:

$$\begin{cases} 1 & si \quad \gamma_A \geq \beta_B \\ \frac{\delta_A-\alpha_B}{(\beta_B-\alpha_B)-(\gamma_A-\delta_A)} & si \quad \gamma_A < \beta_B \quad y \quad \delta_A > \alpha_B \\ 0 & en\ otro\ caso\ (\delta_A \leq \alpha_B) \end{cases}$$

5.2.4. El comparador de necesidad difuso NFGEQ

Es el comparador *Necesariamente Mayor o Igual*. Si se quiere aplicar el comparador NFGEQ a las distribuciones de probabilidad A y B, se tiene que A NFGEQ B [2]:

$$\begin{cases} 1 & si \quad \alpha_A \geq \beta_B \\ \frac{\beta_A-\alpha_B}{(\beta_B-\alpha_B)-(\alpha_A-\beta_A)} & si \quad \alpha_A < \beta_B \quad y \quad \beta_A > \alpha_B \\ 0 & en\ otro\ caso\ (\beta_A \leq \alpha_B) \end{cases}$$

5.2.5. El comparador de posibilidad difuso FGT

Es el comparador *Posiblemente Mayor*. Si se quiere aplicar el comparador FGT a las distribuciones de probabilidad A y B, se tiene que A FGT B [2]:

$$\begin{cases} 1 & si \quad \gamma_A \geq \delta_B \\ \frac{\delta_A-\delta_B}{(\delta_B-\gamma_B)-(\gamma_A-\delta_A)} & si \quad \gamma_A < \delta_B \quad y \quad \delta_A > \gamma_B \\ 0 & en\ otro\ caso\ (\delta_A \leq \gamma_B) \end{cases}$$

5.2.6. El comparador de necesidad difuso NFGT

Es el comparador *Necesariamente Mayor*. Si se quiere aplicar el comparador NFGT a las distribuciones de probabilidad A y B, se tiene que A NFGT B [2]:

$$\begin{cases} 1 & si \quad \alpha_A \geq \delta_B \\ \frac{\beta_A-\gamma_B}{(\delta_B-\gamma_B)-(\alpha_A-\beta_A)} & si \quad \alpha_A < \delta_B \quad y \quad \beta_A > \gamma_B \\ 0 & en\ otro\ caso\ (\beta_A \leq \gamma_B) \end{cases}$$

5.2.7. El comparador de posibilidad difuso FLEQ

Es el comparador *Posiblemente Menor o Igual*. Si se quiere aplicar el comparador FLEQ a las distribuciones de probabilidad A y B, se tiene que A FLEQ B [2]:

$$\begin{cases} 1 & si \quad \beta_A \leq \gamma_B \\ \frac{\delta_B-\alpha_A}{(\beta_A-\alpha_A)-(\gamma_B-\delta_B)} & si \quad \beta_A > \gamma_B \quad y \quad \alpha_A < \delta_B \\ 0 & en\ otro\ caso\ (\alpha_A \geq \delta_B) \end{cases}$$

5.2.8. El comparador de necesidad difuso NFLEQ

Es el comparador *Necesariamente Menor o Igual*. Si se quiere aplicar el comparador NFLEQ a las distribuciones de probabilidad A y B, se tiene que A NFLEQ B [2]:

$$\begin{cases} 1 & si \quad \delta_A \leq \gamma_B \\ \frac{\gamma_A-\delta_B}{(\gamma_B-\delta_B)-(\delta_A-\gamma_A)} & si \quad \delta_A > \gamma_B \quad y \quad \gamma_A < \delta_B \\ 0 & en\ otro\ caso\ (\gamma_A \geq \delta_B) \end{cases}$$

5.2.9. El comparador de posibilidad difuso FLT

Es el comparador *Posiblemente Menor*. Si se quiere aplicar el comparador FLT a las distribuciones de probabilidad A y B, se tiene que A FLT B [2]:

$$\begin{cases} 1 & si \quad \beta_A \leq \alpha_B \\ \frac{\alpha_A-\beta_B}{(\alpha_B-\beta_B)-(\beta_A-\alpha_A)} & si \quad \beta_A > \alpha_B \quad y \quad \alpha_A < \beta_B \\ 0 & en\ otro\ caso\ (\alpha_A \geq \beta_B) \end{cases}$$

5.2.10. El comparador de necesidad difuso NFLT

Es el comparador *Necesariamente Menor*. Si se quiere aplicar el comparador NFLT a las distribuciones de probabilidad A y B, se tiene que A NFLT B [2]:

$$\begin{cases} 1 & si \quad \delta_A \leq \alpha_B \\ \frac{\gamma_A-\beta_B}{(\alpha_B-\beta_B)-(\delta_A-\gamma_A)} & si \quad \delta_A > \alpha_B \quad y \quad \gamma_A < \beta_B \\ 0 & en\ otro\ caso\ (\gamma_A \geq \beta_B) \end{cases}$$

5.2.11. El comparador de posibilidad difuso MGT

Es el comparador *Posiblemente Mucho Mayor*. Si se quiere aplicar el comparador MGT a las distribuciones de probabilidad A y B, se tiene que A MGT B [2]:

$$\begin{cases} 1 & si \quad \gamma_A \geq \delta_B + M \\ \frac{\gamma_B + M - \delta_A}{(\gamma_A - \delta_A) - (\delta_B - \gamma_B)} & si \quad \gamma_A < \delta_B + M \quad y \quad \delta_A > \gamma_B + M \\ 0 & en\ otro\ caso\ (\delta_A \leq \gamma_B + M) \end{cases}$$

5.2.12. El comparador de necesidad difuso NMGT

Es el comparador *Necesariamente Mucho Mayor*. Si se quiere aplicar el comparador NMGT a las distribuciones de probabilidad A y B, se tiene que A NMGT B [2]:

$$\begin{cases} 1 & si \quad \alpha_A \geq \delta_B + M \\ \frac{\gamma_B + M - \beta_A}{(\alpha_A - \beta_A) - (\delta_B - \gamma_B)} & si \quad \alpha_A < \delta_B + M \quad y \quad \beta_A > \gamma_B + M \\ 0 & en\ otro\ caso\ (\beta_A \leq \gamma_B + M) \end{cases}$$

5.2.13. El comparador de posibilidad difuso MLT

Es el comparador *Posiblemente Mucho Menor*. Si se quiere aplicar el comparador MLT a las distribuciones de probabilidad A y B, se tiene que A MLT B [2]:

$$\begin{cases} 1 & si \quad \beta_A \leq \alpha_B - M \\ \frac{\beta_B - M - \alpha_A}{(\beta_A - \alpha_A) - (\alpha_B - \beta_B)} & si \quad \beta_A > \alpha_B - M \quad y \quad \alpha_A < \beta_B - M \\ 0 & en\ otro\ caso\ (\alpha_A \geq \beta_B - M) \end{cases}$$

5.2.14. El comparador de necesidad difuso NMLT

Es el comparador *Necesariamente Mucho Menor*. Si se quiere aplicar el comparador NMLT a las distribuciones de probabilidad A y B, se tiene que A NMLT B [2]:

$$\begin{cases} 1 & si \quad \delta_A \leq \alpha_B - M \\ \frac{\beta_B - M - \gamma_A}{(\beta_A - \alpha_A) - (\alpha_B - \beta_B)} & si \quad \delta_A > \alpha_B - M \quad y \quad \gamma_A < \beta_B - M \\ 0 & en\ otro\ caso\ (\gamma_A \geq \beta_B - M) \end{cases}$$

5.3. DML del Lenguaje FSQL: SELECT

FSQL utiliza una sintaxis DML similar a SQL, pero al que se le añaden nuevas características. Las más relevante de ellas es la incorporación de etiquetas lingüísticas, de nuevos comparadores (difusos) y de un umbral de cumplimiento. La primera y la segunda se han mencionado anteriormente, por lo que queda por definir qué es el umbral de cumplimiento o *threshold* τ [5].

En FSQL, para cada condición se puede establecer un umbral de cumplimiento de la misma, para así poder relajar la restricción de la propia condición. El formato que tiene es el siguiente: <*condición-simple*> THOLD τ [5]. La palabra reservada THOLD es opcional y puede sustituirse por un comparador crisp ($=$, $<$, $\leq$, ...).

De esta forma, la sintaxis de una consulta SELECT para FSQL queda como:

SELECT <lista-de-selección> FROM <lista-de-tablas> WHERE <condición-difusa>

Un ejemplo de una consulta FSQL puede ser:

Dame todas las personas cuya edad es aproximadamente joven con grado mínimo 0.6:

SELECT * FROM Personas WHERE Edad FEQ $\$joven$ THOLD 0.6.

5.4. El FIRST

Cuando se habla de consultas difusas, hay que tener claro que se tratan de consultas con alto nivel difuso, pero que en el fondo se resume y se interpreta como una consulta SQL clásica ya que se pretende construir un sistema de base de datos relacional difusa sobre un sistema gestor de base de datos clásico ya existente y que permita operar difusamente, arrojando resultados en base a los datos difusos ingresados. De acuerdo con los estudios realizados por una tesis doctoral hecha en la Universidad de Granada [11], una base de datos relacional difusa se construye definiendo un formato interno: el FIRST. El FIRST es un formato que define atributos difusos, un servidor FSQL para que el manejador de base de datos sobre el que se instala el sistema gestor de base de dato difusa pueda interpretar las consultas difusas, y una base de metaconocimiento difuso, la cual se discutirá en el próximo apartado. Es pues, la encargada de estructurar todo lo concerniente a la imprecisión de los atributos difusos y el tratamiento de cada uno de sus tipos: 1, 2 o 3. A continuación se detallan cada uno de ellos:

5.4.1. Atributo difuso tipo 1

Son aquellos atributos que contienen datos clásicos o «crisp». Almacenan datos exactos o concisos. A pesar de ser datos precisos, pueden tener tratamiento difuso y por ende se almacenará una etiqueta del mismo en la Base de Metaconocimiento Difuso. Esto implica también que se pueden realizar consultas difusas sobre ellos, aún cuando no admiten datos difusos, pues la idea es que los atributos con datos precisos no pierdan vigencia en las consultas difusas.

5.4.2. Atributo difuso tipo 2

Son aquellos atributos en condiciones difusas, por lo que están orientados a almacenar información imprecisa. Más allá de esto, los atributos tipo 2 soportan la representación de datos de tipo «NULL», «UNKNOWN», «UNDEFINED», además de etiquetas lingüísticas, intervalos y trapecios, mayormente usados para generar intervalos normalizados para el subsiguiente análisis de los datos en la realización de la consulta difusa. Algunos de ellos requieren representación de un solo parámetro en la Base de Metaconocimiento Difuso, como los valores constantes, clásicos y etiquetas. La etiqueta requiere dos parámetros para representar los dos extremos del intervalo. El trapecio requiere cuatro, para poder representar los valores de sus extremidades.

Tipos de Valores	FT	F1	F2	F3	F4
UNKNOWN	0	NULL	NULL	NULL	NULL
UNDEFINED	1	NULL	NULL	NULL	NULL
NULL	2	NULL	NULL	NULL	NULL
Crisp	3	NULL	NULL	NULL	NULL
Label	4	$FUZZY$	NULL	NULL	NULL
Intervalo[n,m]	5	N	NULL	NULL	m
Aproxidamente(d)	6	D	d-margen	D+margen	margen
Trapecio[α,β,γ,δ]	7	α	$\beta - \alpha$	$\gamma - \delta$	δ

5.4.3. Atributo difuso tipo 3

Son aquellos atributos que representan mayormente distribuciones de probabilidad sobre dominios escalares, constantes como las de tipo 2. Se representan mediante un valor «FT», que define el tipo de dato representado, y una o más parejas (FPi, Fi), con i=0,..,n, y tal que el par representa el grado de posibilidad y el valor, respectivamente.

Tipos de Valores	FT	FP_1	F_1	...	FP_n	F_n
UNKNOWN	0	NULL	NULL	...	NULL	NULL
UNDEFINED	1	NULL	NULL	...	NULL	NULL
NULL	2	NULL	NULL	...	NULL	NULL
Simple	3	p	d	...	NULL	NULL
Distr. Prob.	4	p_1	d_1	...	p_n	d_n

5.5. Base de Metaconocimiento Difuso

El FMB (Fuzzy Metaknowledge Base, o Base de Metaconocimiento Difuso) son un conjunto de tablas que son usadas para establecer la estructura difusa que se genera en el FIRST. Estas tablas son las que definen los objetos difusos a ser estudiados, las etiquetas, las funciones de aproximación o pertenencia, entre otros.

Tabla/Vista	Sinónimo
T.FUZZY_COL_LIST	FCL
T.FUZZY_OBJECT_LIST	FOL
T.FUZZY_LABEL_DEF	FLD
T.FUZZY_APROX_MUCH	FAM
T.FUZZY_NEARNESS_DEF	FND
T.FUZZY_COMPATIBLE_COL	FCC
T.FUZZY_QUALIFIERS_DEF	FQD
V.LABELS_FOR_OBJCOL	LFOC
V.LABELS_OBJCOL_T	LOCT3
V.ALL_COMPATIBLES_T	ACT3

Al final, la relación y la interacción entre el motor de base de datos tradicional, el servidor FSQL, el cliente FSQL y la Base de Metaconocimiento Difusa es similar a la siguiente imagen:

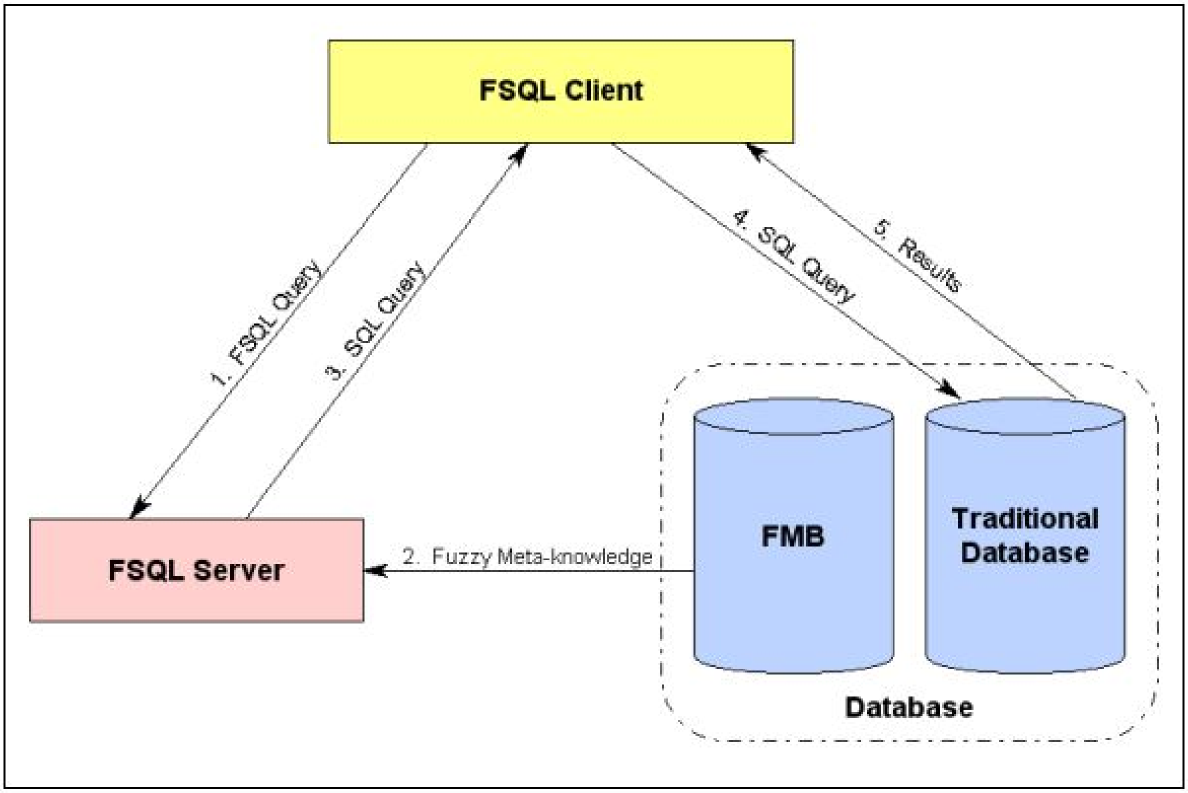

Figura 5.3: FSQL, BD clásica y Base de Metaconocimiento Difuso

5.6. Una consulta FSQL

Ahora se procederá a detallar todo el procedimiento de la realización de un ejemplo de consulta FSQL, desde el volcado de los datos difusos, hasta la realización de la propia consulta usando las funciones de operación definidas.

5.6.1. La compra de un piso en Madrid

Se quiere contar con una base de datos difusa que pueda arrojar resultados acerca de cuáles son las mejores opciones para la toma de decisión a la hora de comprar un piso. Para ello, y para hacer el ejemplo más simple, se toman en cuenta dos variables de peso: el precio del inmueble y la antigüedad.

1. Precio: La toma de decisión requiere saber como punto imprescindible el precio. Está definido en miles de euros, usando la siguiente escala:

    - Barato: 30 - 50 (miles de €)

- Regular: 45- 80 (miles de €)
- Costoso: 77 - 130 (miles de €)

2. Antigüedad: Será de muy buen criterio saber la antigüedad del inmueble a seleccionar. Se define en cantidad de meses, usando la siguiente escala:

   - Nuevo: 0 - 18 (meses de uso)
   - Usado: 15 - 60 (meses de uso)
   - Viejo: 55 - 300 (meses de uso)

5.6.2. Volcado en la Base de Metaconocimiento Difuso

Hay que ahora darle sentido a las etiquetas lingüísticas, a las tablas y atributos difusos de nuestra base de datos difusa a través de las tablas de la Base de Metaconocimiento Difuso. Los siguientes ejemplos de inserción muestran cómo se crean los datos en las columnas y objetos difusos en las tablas de la Base de Metaconocimiento Difuso.

−−INSERCIÓN EN FUZZY_COL_LIST :

insert into fuzzy_col_list values(1, 1, 2, 1, *null*, *null*, '*compra_piso_precio*');

insert into fuzzy_col_list values(1, 2, 2, 1, *null*, *null*, '*compra_piso_antiguedad*');

−−OBJETOS DIFUSOS:

Se definen las etiquetas lingüísticas para la variable linguistica (universo de discurso) «precio»:

insert into fuzzy_object_list values(1, 1, 0, '*Barato*', 0);

insert into fuzzy_object_list values(1, 1, 1, '*Regular*', 0);

insert into fuzzy_object_list values(1, 1, 2, '*Costoso*', 0);

Se definen las etiquetas lingüísticas para la variable linguistica «antigüedad»:

insert into fuzzy_object_list values(1, 2, 0, '*Nuevo*', 0);

insert into fuzzy_object_list values(1, 2, 1, '*Usado*', 0);

insert into fuzzy_object_list values(1, 2, 2, '*Viejo*', 0);

Se definen ahora los valores de los trapecios para generar intervalos normalizados, que son usados para modelar los datos de las etiquetas definidas arriba. Esto se hace de esta manera ya que se están considerando los atributos de tipo 2:

Se definen los datos de las etiquetas usadas para la variable lingüística «precio»:

«Barato»: insert into $fuzzy\_label\_def\ values(1, 1, 0, 30, 30, 45, 50)$;

«Regular»: insert into $fuzzy\_label\_def\ values(1, 1, 1, 45, 50, 77, 80)$;

«Costoso»: insert into $fuzzy\_label\_def\ values(1, 1, 2, 77, 80, 130, 130)$;

Por su parte, de esta manera se definen los valores de las etiquetas usadas para la variable lingüística «antigüedad»:

«Nuevo»: insert into $fuzzy\_label\_def\ values(1, 2, 0, 0, 0, 15, 18)$;

«Usado»: insert into $fuzzy\_label\_def\ values(1, 2, 1, 15, 18, 55, 60)$;

«Viejo»: insert into $fuzzy\_label\_def\ values(1, 2, 2, 55, 60, 300, 300)$;

5.6.3. La tabla difusa principal

Será ahora necesario crear una tabla general, que admite los valores difusos de los atributos difusos. Dicha tabla será la que el manejador de base de datos relacional usará para analizar las consultas difusas. En ella se representarán todas las etiquetas lingüísticas definidas en las tablas de la Base de Metaconocimiento Difuso. Para el ejemplo que se está mostrando, la tabla quedaría de la siguiente forma:

CREATE TABLE compra_piso (id varchar, numero_piso varchar, preciot numeric NOT NULL, precio1 numeric, precio2 numeric, precio3 numeric, precio4 numeric, antiguedadt numeric NOT NULL, antiguedad1 numeric, antiguedad2 numeric, antiguedad3 numeric, antiguedad4 numeric, PRIMARY KEY(id));

Nótese que, como se está trabajando con datos difusos tipo 2, y se están tratando como intervalos, se hace necesario crear cuatro campos adicionales para cada una de las variables lingüísticas.

5.6.4. Fusificación de la variable lingüística «Precio»

En la siguiente imagen se puede ver cómo se fusifica el universo de discurso «Precio» en tres trapecios probabilísticos, donde cada uno define las etiquetas usadas para definirla. Cada etiqueta define pues, un conjunto difuso.

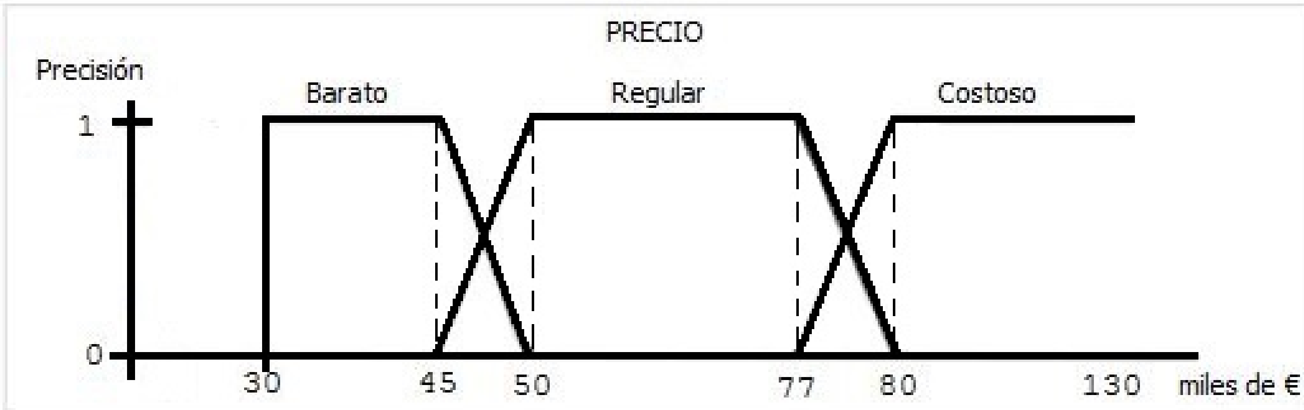

Figura 5.4: Modelo del universo de discurso *Precio*

5.6.5. Fusificación de la variable lingüística «Antigüedad»

En la siguiente imagen se puede ver cómo se fusifica el universo de discurso «Antigüedad» en tres trapecios probabilísticos, donde cada uno define las etiquetas usadas para definirla.

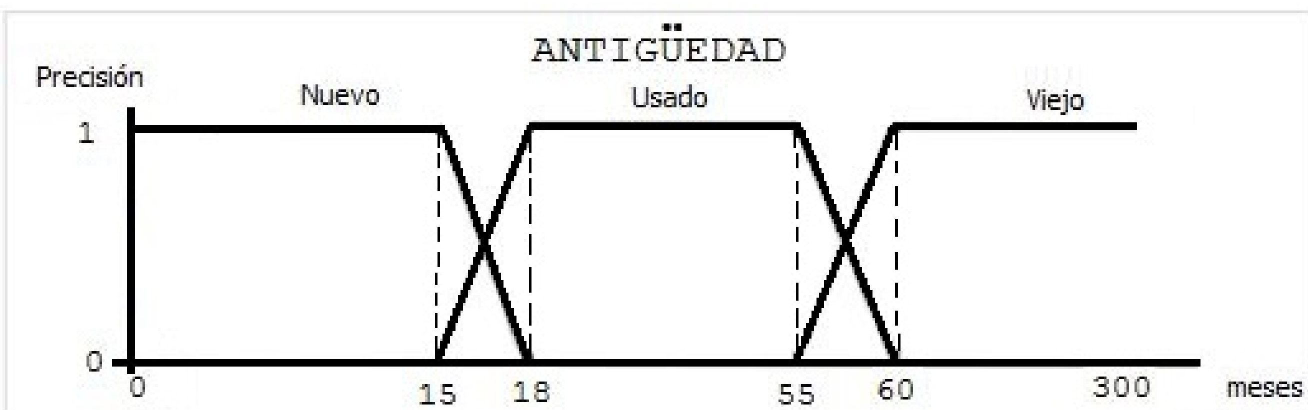

Figura 5.5: Modelo del universo de discurso *Antigüedad*

5.6.6. Recolección de datos de pisos candidatos

Vamos ahora a ver un ejemplo de cómo serían los datos en la tabla difusa principal llamada *comprapiso*. En ella se verán datos insertados que tienen que ver con el número de piso (el ID), su costo y su antigüedad.

ID	Costo (miles de €)	Antigüedad (meses de uso)
1	40	12
2	125	0
3	80	15
4	50	24
5	Barato	Usado
6	Costoso	90
7	70	Nuevo

5.6.7. La consulta

Por último, se mostrará el ejemplo de una consulta usando la notación de FSQL y su resultado. En el ejemplo se van a seleccionar los pisos baratos.

*/*Selecciona únicamente los pisos posiblemente baratos totalmente*/ :*

select id, costo fsql_functions_feq(null,null,null,40,30,30,45,50,preciot,precio1,precio2,precio3,precio4) as 'precio' from compra_piso

where fsql_functions_feq(null,null,null,40,30,30,45,50,preciot,precio1,precio2,precio3,precio4)=1;

En este ejemplo, los pisos retornados por la consulta serán aquellos con ID 1 y 5 de la tabla anterior.

Capítulo 6

FSQL para Conjuntos Intervalo-Valorados Difusos

Teniendo en consideración lo definido para SQL con conjuntos difusos (Fuzzy SQL o FSQL), pasaremos ahora a definir los principios y los fundamentos del FSQL para conjuntos intervalo-valorados difusos, de ahora en adelante denotado IVFSQL. Se van a definir las nuevas nociones de etiquetas lingüísticas, los nuevos operadores de comparación, el cómo queda definida la función de pertenencia, las constantes difusas y algunos condicionales.

Así como para FSQL, en IVFQL existirá la noción de variables lingüísticas, que definen un universo de discurso, y de etiquetas lingüísticas, que identificarían a un conjunto intervalo-valorado difuso. La función de pertenencia está atada a las variables lingüísticas, por lo que nos ayuda a interpretar la relación de los elementos con el conjunto intervalo-valorado difuso.

6.1. Fusificación de variables lingüísticas en IVFSQL

Así como en FSQL, se tratarán de forma análoga las etiquetas dentro de IVFSQL, teniendo un significado semántico con la intención de expresar a un conjunto difuso definido a través de la función de pertenencia IVCDEG discutida en la sección 4.1.

Siguiendo el ejemplo de la sección 5.6.4 para la fusificación de la variable lingüística «Precio», se extrapolará dicho ejemplo pero aplicado a etiquetas lingüísticas para IVFSQL. Los valores de cada una de las tres etiquetas (Barato, Regular y Costoso) estarán sujetas a la noción intervalo-valorada, por lo tanto cada uno de los elementos de las etiquetas tendrán dos valores posibles correspondientes a un par ordenado (a,b), con $a \in \{0,1\}$, $b \in \{0,1\}$ y $0 \leqslant a \leqslant b \leqslant 1$. En concreto, se puede ver en la figura que las etiquetas «Barato», «Regular» y «Costoso» tienen en su imagen dos valores diferentes en el rango; esto quiere decir que un elemento del eje x del universo de discurso «Precio» se puede encontrar entre dos números en el eje y, por lo que debe ser tratado como un elemento cuyo grado de pertenencia está comprendido en un rango de valores, con un cierto margen de error. Por ejemplo, el valor 35 en el eje x del precio, tiene un grado de pertenencia a la etiqueta «Barato» de entre 0.8 y 1. El margen de error dependerá de cuál sea el operador usado en la selección IVFSQL.

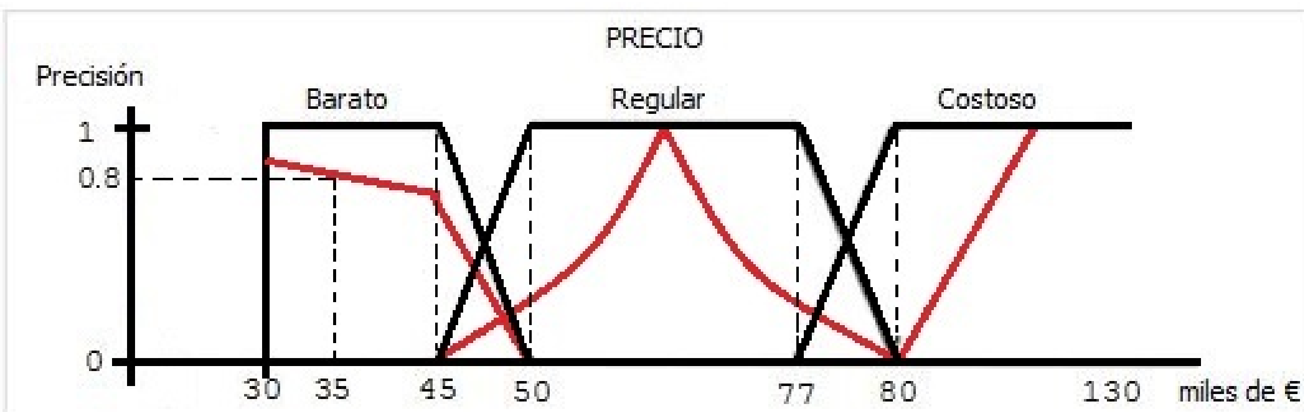

Figura 6.1: Fusificación de variables lingüísticas en IVFSQL

6.2. Modificadores de etiquetas lingüísticas para IVFSQL

En la sección 2.3 se discutió sobre los modificadores en las etiquetas, que no son más que aquellas que hacen el rol de adverbios del lenguaje humano natural: «Muy joven», «Extremedamente bueno», «No muy barato». Para los conjuntos intervalo-valorados, es posible también definir modificadores de este estilo. Basta con aplicar la función modificadora (ver figura 2.2 de la sección 2.3) al par ordenado (a,b) de cada uno de los elementos de las etiquetas. A continuación, se muestra una imagen con la aplicación del modificador «very» (muy) a la etiqueta «joven»:

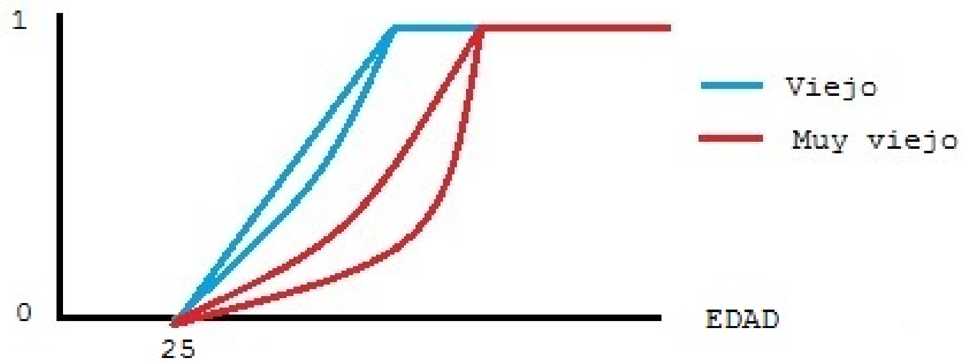

Figura 6.2: Modificadores de etiquetas lingüísticas para IVFSQL

La idea de aplicar un modificador a una etiqueta lingüística para IVFSQL sería aplicar la función modificadora a cada elemento del par ordenador (a,b), de forma tal que tanto el primer elemento como el segundo del intervalo apliquen a la modificación o transformación. Como ejemplo, se aplicará el modificador «muy» (very) para el ejemplo de la figura 2.2.

6.3. Comparadores para conjuntos intervalo-valorados difusos

Los comparadores que se sugieren usar para IVFSQL son esencialmente los mismos usados para FSQL, pero aplicado a conjuntos intervalo-valorados difusos, definido en un intervalo $[x,y]$, donde $0 \leqslant x \leqslant y \leqslant 1$. También se usa la misma notación de umbral de cumplimiento THOLD, el cual tras cada condición impone un umbral de cumplimiento mínimo.

Por su parte, las consultas, además de devolver los valores de acuerdo al umbral de cumplimiento, también devolverán un margen de error. Es decir, cada respuesta devuelta por la consulta, tendrá asociado un margen de error de certeza. Ese margen de error dependerá tanto del valor absoluto de la diferencia máxima entre entre el THOLD y los dos extremos del intervalo (a,b) del par ordenado como del comparador que se esté usando en la selección. Nótese que para que el margen de error sea cero, el valor mínimo y máximo del intervalo deben ser iguales.

La notación de consultas para IVFSQL quedaría igual que para FSQL, salvo que los operadores no serían los mismos; para el operador de igualdad difusa, se usará la igualdad difusa intervalo-valorada, entre otros.

6.3.1. El comparador IVFEQ

IVFEQ, del acrónimo Interval-Valued Fuzzy Equal, es el operador de igualdad de los conjuntos intervalo-valorados en SQL difuso. La idea de este comparador es que lo que compare sean condiciones cuyo umbral de comparación se comprenda entre el valor mínimo y el valor máximo del intervalo. Como se ha dicho, las tuplas devueltas en la consulta operadas con este operador tendrán un margen de error de su certeza.

Sea la siguiente consulta IVFSQL:

- SELECT * FROM personas WHERE edad IVFEQ $joven THOLD 0.5

Queremos seleccionar todas las personas cuya edad sea considerada «joven» usando el comparador IVFEQ, con un grado de pertenencia de 0.5. En este caso, para ilustrar el ejemplo, asumiremos de que para la edad de 40 años, el grado de pertenencia a la etiqueta «joven» está comprendido entre 0.4 y 0.7. Como para la edad de 40, el grado de pertenencia (0.5) está dentro del rango [0.4,0.7], se toma como cumplidora de la condición. Como estamos hablando de un intervalo de valores, [0.4,0,7], se debe hallar cuál será el margen de error de que la tupla devuelta por la selección, en efecto pertenezca a la selección. Para ello, se toma el valor absoluto de la diferencia entre el grado de pertenencia y del valor máximo entre el valor mínimo y el valor máximo del intervalo, en este caso, $\max(|0{,}5-0{,}4|,|0{,}5-0{,}7|)$=0.2. Por lo tanto, las tuplas con edad=40 serán devueltas por la selección con un margen de error de 0.2.

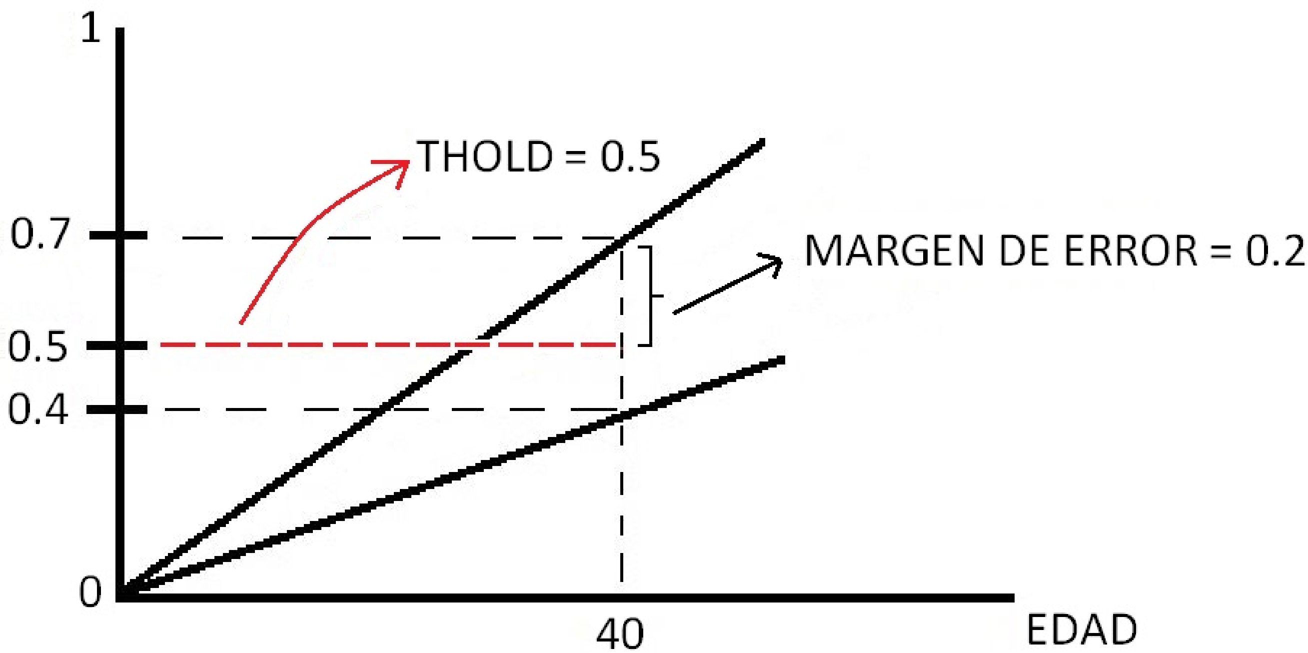

Figura 6.3: IVFEQ en detalle, en un ejemplo

Para que una tupla cumpla una condición usando el comparador IVFEQ, debe ocurrir que el umbral de cumplimiento esté comprendido entre el valor mínimo y el valor máximo del intervalo del elemento de la etiqueta.

Por otro lado, es conveniente establecer el criterio que indique cómo se calculan los valores de relevancia de las selecciones. Para ello, se propone usar el basado en la agrupación difusa intervalo-valorada o *interval-valued fuzzy clustering* [10]. Éste se fundamenta en un proceso de optimización del cual se obtienen particiones difusas intervalo-valoradas. Otra opción es considerar el método basado en la especifidad del problema, es decir, buscar una función numérica a ser aproximada, y luego el margen de error queda definido como un conjunto intervalo-valorado difuso.

Es importante y necesario considerar el hecho de que también se insertan las tuplas que definen los objetos difusos para conjuntos intervalo-valorados en la Base de Metaconocimiento Difuso, pero con novedades: la definición de las etiquetas lingüísticas para IVFSQL se mantienen como se propuso para los conjuntos difusos, ya que lo que realmente cambia son los valores de imagen a partir del dominio entre los conjuntos difusos y los conjuntos difusos intervalo-valorados (la imagen del primero es un numero, el del segundo es un intervalo). Como la imagen de cada elemento en el dominio estará definido en un intervalo cerrado, se hace necesario entonces definir dos tuplas por cada dominio de cada etiqueta difusa intervalo-valorada (para las etiquetas difusas se definía solo una tupla por etiqueta). Siguiendo esta idea, la definición de las etiquetas (proceso de fusificación), y siguiendo las consideradas en la sección 5.6.2, se harían de la siguiente manera:

«Barato»:

$insert\ into\ fuzzy\_label\_def\ values(1, 1, 0, 30, 30, 45, 50);$

$insert\ into\ fuzzy\_label\_def\ values(1, 1, 0, 30, 35, 50, 50);$

«Regular»:

$insert\ into\ fuzzy\_label\_def\ values(1, 1, 1, 45, 50, 77, 80);$

$insert\ into\ fuzzy\_label\_def\ values(1, 1, 1, 45, 55, 80, 80);$

«Costoso»:

$insert\ into\ fuzzy\_label\_def\ values(1, 1, 2, 77, 80, 130, 130);$

$insert\ into\ fuzzy\_label\_def\ values(1, 1, 2, 77, 90, 130, 130);$

Como se ve en este caso, la variable lingüística *precio* se ve representada por etiquetas con representación en conjuntos intervalo-valorados. La tabla difusa principal, discutida en la sección 5.6.3, no es necesario que modifique su estructura, pues ya la han modificado en la definición de las etiquetas en la Base de Metaconocimiento Difuso. Gráficamente, la fusificación anteriormente descrita se ve representada como sigue:

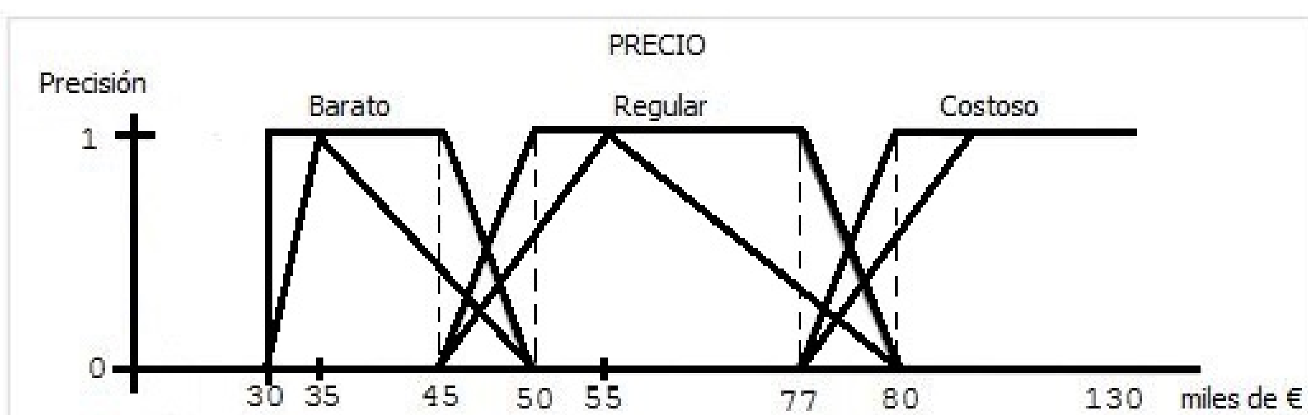

Figura 6.4: Fusificación intervalo-valorada del universo de discurso «Precio»

6.3.2. El comparador IVFGT

El operador IVFGT define el Inter-Valued Fuzzy Greater Than, es decir, el operador de mayor estricto para conjuntos intervalo-valorados difusos. La idea de este comparador es que lo que compare sean condiciones cuyo umbral de comparación esté más cercano al valor mínimo del intervalo que al valor máximo del mismo. En concreto, que la diferencia absoluta entre el valor máximo del intervalo y el umbral de cumplimiento sea estrictamente mayor a la diferencia absoluta entre el valor mínimo del intervalo y el umbral de cumplimiento.

Sea la siguiente consulta IVFGT:

- SELECT * FROM personas WHERE edad IVFGT $joven THOLD 0.5

Queremos seleccionar todas las personas cuya edad sea considerada «joven» usando el comparador IVFGT, con un grado de pertenencia de 0.5. Asumiremos nuevamente que para la edad de 40 años, el grado de pertenencia a la etiqueta «joven» está comprendido entre 0.4 y 0.7. Como para la edad de 40, la diferencia entre el valor máximo del intervalo y el umbral de cumplimiento es mayor que la diferencia entre el valor mínimo del intervalo y el umbral de cumplimiento, se toma como cumplidora de la condición. Es decir, $|0{,}7 - 0,5| > |0{,}4 - 0{,}5|$. Como estamos hablando de un intervalo de valores, [0.4,0.7], se debe hallar cuál será el margen de error de que la tupla devuelta por la selección, en efecto pertenezca a la selección. Para ello, se toma el valor absoluto de la diferencia entre el grado de pertenencia y el valor mínimo del intervalo (pues los valores que están entre el valor máximo del intervalo y el umbral de cumplimiento, ya superan el umbral de cumplimiento). Por lo tanto, el margen de error para este caso es de $|0{,}5 - 0{,}4|$=0.1.

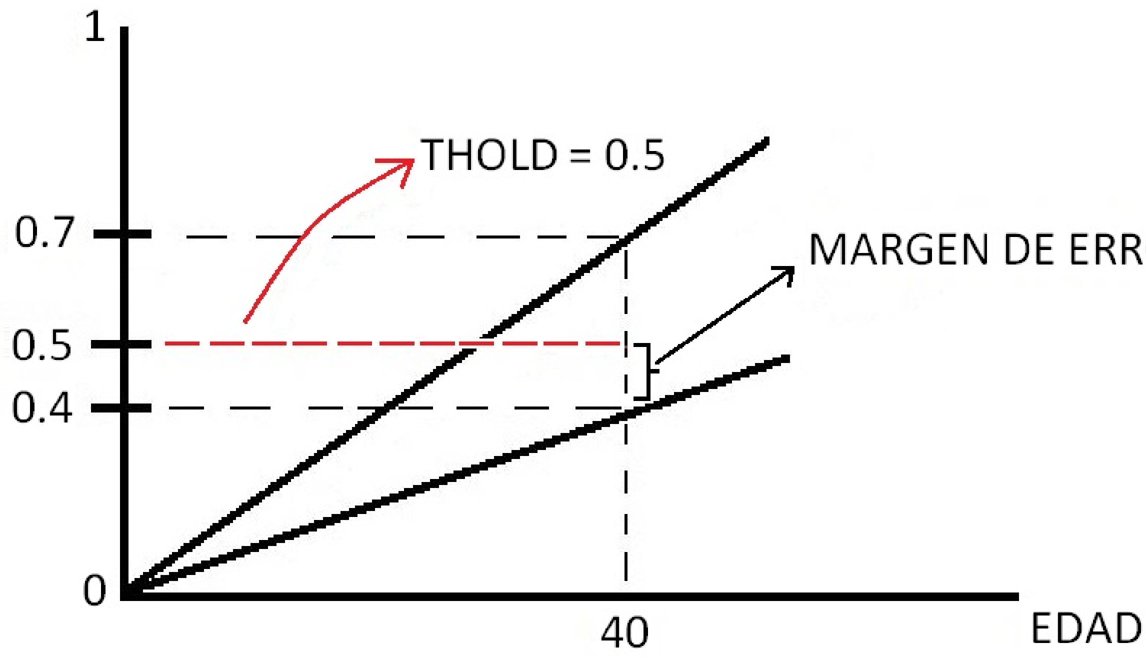

Figura 6.5: IVFGT en detalle, en un ejemplo

Si en este ejemplo se diera el caso de que el valor máximo del intervalo fuese 0.6 y el mínimo 0.4, con un umbral de cumplimiento de 0.5, entonces aquel elemento que cumpla con estos valores, no serían devueltos por el operador IVFGT.

6.3.3. El comparador IVFGEQ

El operador IVFGEQ define el Inter-Valued Fuzzy Greater or Equal, es decir, el operador de mayor o igual para conjuntos intervalo-valorados difusos. La idea de este comparador es que lo que compare sean condiciones cuyo umbral de comparación esté más cercano al valor mínimo del intervalo que al valor máximo del mismo, o al menos que sea igual de cercano tanto al valor mínimo como al valor máximo. En concreto, que la diferencia absoluta entre el valor máximo del intervalo y el umbral de cumplimiento sea mayor o igual a la diferencia absoluta entre el valor mínimo del intervalo y el umbral de cumplimiento.

Sea la siguiente consulta IVFGEQ:

- SELECT * FROM personas WHERE edad IVFGEQ $joven THOLD 0.5

$|0{,}7-0{,}5|$ Queremos seleccionar todas las personas cuya edad sea considerada «joven» usando el comparador IVFGEQ, con un grado de pertenencia de 0.5. En el caso en que para la edad de 40 años, el grado de pertenencia a la etiqueta «joven» está comprendido entre 0.4 y 0.7. Como para la edad de 40, la diferencia entre el valor máximo del intervalo y el umbral de cumplimiento es mayor o igual que la diferencia entre el valor mínimo del intervalo y el umbral de cumplimiento, se toma como cumplidora de la condición. Es decir, $|0{,}7-0{,}5| \geq |0{,}4-0{,}5|$. Como estamos hablando de un intervalo de valores, [0.4,0,7], se debe hallar cuál será el margen de error de que la tupla devuelta por la selección, en efecto pertenezca a la selección. Para ello, se toma el valor absoluto de la diferencia entre el grado de pertenencia y el valor mínimo del intervalo (pues los valores que están entre el valor máximo del intervalo y el umbral de cumplimiento, ya superan el umbral de cumplimiento). Por lo tanto, el margen de error para este caso es de $|0{,}5-0{,}4|$=0.1.

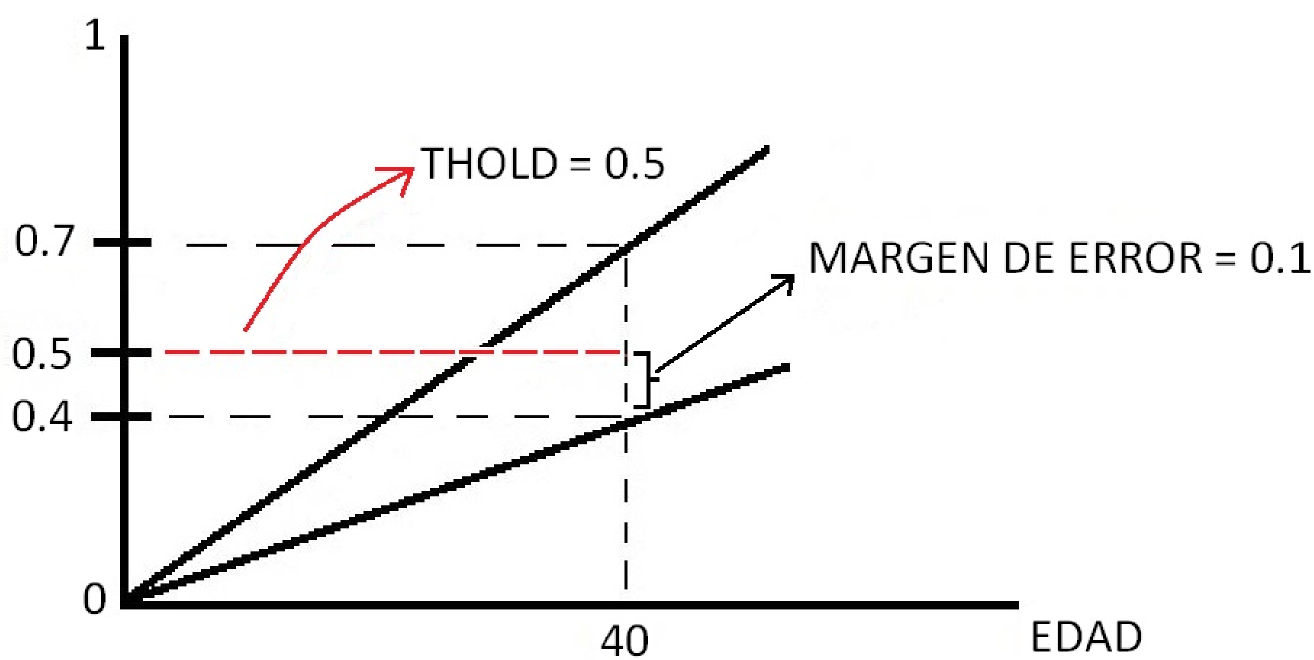

Figura 6.6: IVFGEQ en detalle, en un ejemplo

Notar que el principio es esencialmente similar a IVFGT.

6.3.4. El comparador IVMGT

El operador IVMGT define el Inter-Valued Fuzzy Much Greater Than, es decir, el operador de mucho mayor que, para conjuntos intervalo-valorados difusos. En la Base de Metaconocimiento

Difuso, se almacena un valor para cada tipo de atributo, 1 o 2, que indica cuál es la mínima distancia mínima para que 2 valores de ese atributo sean considerados muy separados. Sea este valor M. Si el valor absoluto de la diferencia entre el umbral de cumplimiento y el valor máximo del intervalo es mayor que dicho valor M, entonces el elemento cumple con el comparador IVMGT.

6.3.5. El comparador IVFLT

El operador IVFLT define el Inter-Valued Fuzzy Less Than, es decir, el operador de menor estricto para conjuntos intervalo-valorados difusos. La idea de este comparador es que lo que compare sean condiciones cuyo umbral de comparación esté más cercano al valor máximo del intervalo que al valor mínimo del mismo. En concreto, que la diferencia absoluta entre el valor máximo del intervalo y el umbral de cumplimiento sea estrictamente menor a la diferencia absoluta entre el valor mínimo del intervalo y el umbral de cumplimiento.

Sea la siguiente consulta IVFLT:

- SELECT * FROM personas WHERE edad IVFLT $joven THOLD 0.6

Queremos seleccionar todas las personas cuya edad sea considerada «joven» usando el comparador IVFLT, con un grado de pertenencia de 0.6. Asumiremos nuevamente que para la edad de 40 años, el grado de pertenencia a la etiqueta «joven» está comprendido entre 0.4 y 0.7. Como para la edad de 40, la diferencia entre el valor máximo del intervalo y el umbral de cumplimiento es menor que la diferencia entre el valor mínimo del intervalo y el umbral de cumplimiento, se toma como cumplidora de la condición. Es decir, $|0{,}7 - 0{,}6| < |0{,}4 - 0{,}6|$. Como estamos hablando de un intervalo de valores, [0.4,0,7], se debe hallar cuál será el margen de error de que la tupla devuelta por la selección, en efecto pertenezca a la selección. Para ello, se toma el valor absoluto de la diferencia entre el grado de pertenencia y el valor máximo del intervalo (pues los valores que están entre el valor mínimo del intervalo y el umbral de cumplimiento, ya son inferiores al umbral de cumplimiento). Por lo tanto, el margen de error para este caso es de $|0{,}7 - 0{,}6|$=0.1.

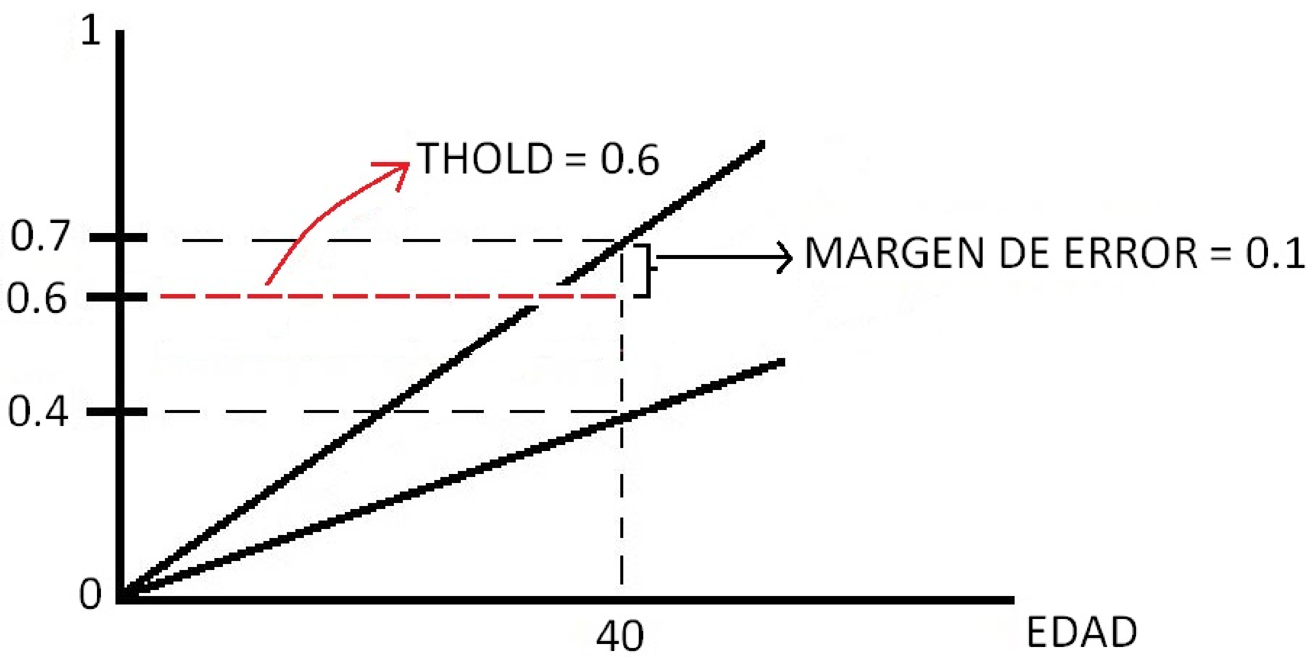

Figura 6.7: IVFLT en detalle, en un ejemplo

Si en este ejemplo se diera el caso de que el valor mínimo del intervalo fuese 0.5 y el máximo 0.7, con un umbral de cumplimiento de 0.6, entonces aquel elemento que cumpla con estos valores, no serían devueltos por el operador IVFLT.

6.3.6. El comparador IVFLEQ

El operador IVFLEQ define el Inter-Valued Fuzzy Less or Equal, es decir, el operador de menor o igual para conjuntos intervalo-valorados difusos. La idea de este comparador es que lo que compare sean condiciones cuyo umbral de comparación esté más cercano al valor máximo del intervalo que al valor mínimo del mismo o, al menos, tan cercano tanto al máximo como al mínimo. En concreto, que la diferencia absoluta entre el valor máximo del intervalo y el umbral de cumplimiento sea menor o igual a la diferencia absoluta entre el valor mínimo del intervalo y el umbral de cumplimiento.

Sea la siguiente consulta IVFLEQ:

- SELECT * FROM personas WHERE edad IVFLEQ $joven THOLD 0.6

Queremos seleccionar todas las personas cuya edad sea considerada «joven» usando el comparador IVFLEQ, con un grado de pertenencia de 0.6. Asumiremos nuevamente que para la edad de 40 años, el grado de pertenencia a la etiqueta «joven» está comprendido entre 0.4 y 0.7. Como para la edad de 40, la diferencia entre el valor máximo del intervalo y el umbral de cumplimiento es menor o igual que la diferencia entre el valor mínimo del intervalo y el umbral de cumplimiento, se toma como cumplidora de la condición. Es decir, $|0{,}7 - 0{,}6| \leq |0{,}4 - 0{,}6|$. Como estamos hablando de un intervalo de valores, [0.4,0,7], se debe hallar cuál será el margen de error de que la tupla devuelta por la selección, en efecto pertenezca a la selección. Para ello, se toma el valor absoluto de la diferencia entre el grado de pertenencia y el valor máximo del intervalo (pues los valores que están entre el valor mínimo del intervalo y el umbral de cumplimiento, ya son inferiores al umbral de cumplimiento). Por lo tanto, el margen de error para este caso es de $|0{,}7 - 0{,}6|$=0.1.

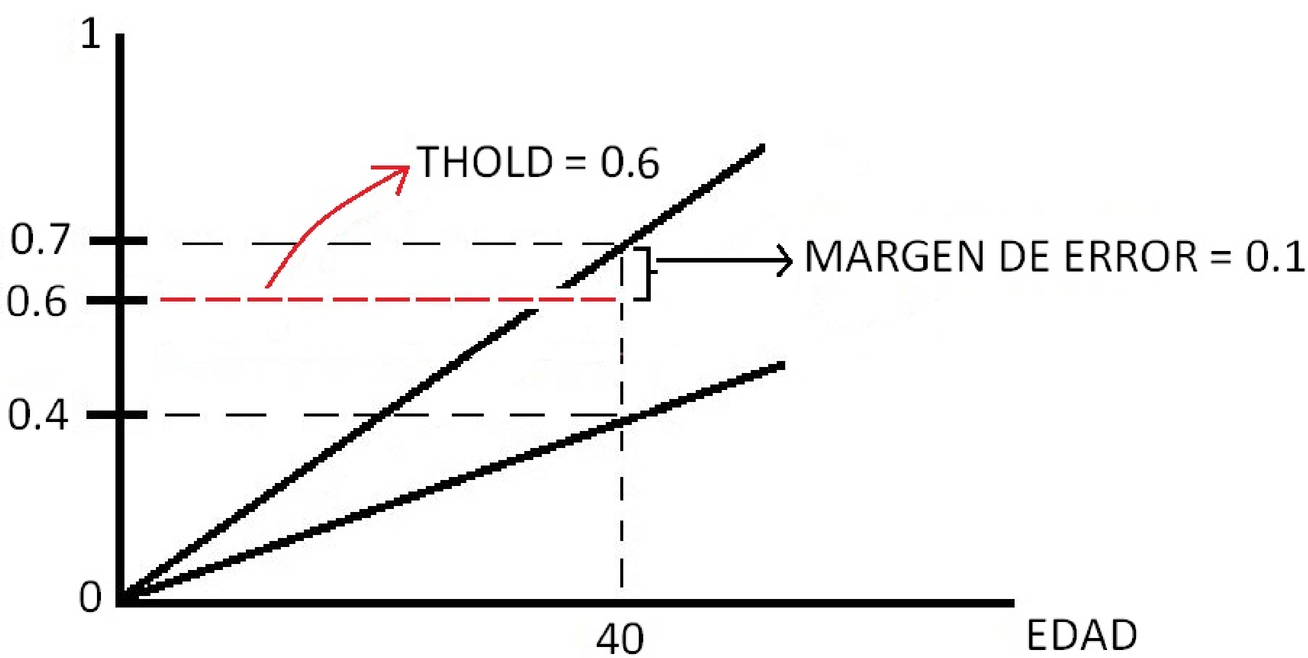

Figura 6.8: IVFLEQ en detalle, en un ejemplo

6.3.7. El comparador IVMLT

El operador IVMLT define el Inter-Valued Fuzzy Much Less Than, es decir, el operador de mucho menor que, para conjuntos intervalo-valorados difusos. En la Base de Metaconocimiento Difuso, se almacena un valor para cada tipo de atributo, 1 o 2, que indica cuál es la mínima distancia mínima para que 2 valores de ese atributo sean considerados muy separados. Sea este valor M. Si el valor absoluto de la diferencia entre el umbral de cumplimiento y el valor mínimo del intervalo es mayor que dicho valor M, entonces el elemento cumple con el comparador IVMLT.

6.4. Uso y alcance de IVFSQL

El tratamiento y manejo de la imprecisión es algo que cada vez es más usado, necesario e importante. Existen ya aplicaciones prácticas que utilizan la flexibilidad basada en bases de datos difusas, en concreto usando todo lo que ofrece FSQL relacionado a este tema. Todos estos se podrían extrapolar como ejemplos en el contexto de IVFSQL. Uno de ellos es una herramienta que sirve como clasificador difuso de imágenes, la cual a partir de datos extraídos de una imagen con los contornos de cierto objeto puede aplicarse un criterio de clasificación de la misma. La idea es que hallando la curva de la curvatura, mediante el nivel de curva que tiene el contorno en cada punto, se puedan clasificar las imágenes a través de algunos atributos como el número de vértices, la distancia entre los vértices, el tamaño de la curvatura, entre otros [1]. En esta aplicación, se podrían utilizar los conceptos y beneficios otorgados por IVFSQL para aumentar aún más la imprecisión y flexibilizar las consultas de clasificación. Los datos de los contornos contarían con un nivel de tolerancia suficiente como para tener un mejor conocimiento de las variables. Otro ejemplo es el uso de FSQL como herramienta de minería de datos. Existe la teoría de que un lenguaje de consulta difusa a bases de datos clásicas o difusas puede ser de gran utilidad en procesos de minería de datos. En caso de que se expandiera el uso en esta área, casi por transitividad se podría expandir el uso de IVFSQL, sobre todo en aspectos de inferencia y métricas de interés.

Por su parte, claro está que para llevar a cabo este uso, es necesario redefinir la arquitectura de manejadores de bases de datos difusos actuales. Hay que comenzar por replantear el esquema de la Base de Metaconocimiento Difuso y la redefinición de las etiquetas intervalo-valoradas difusas como se mencionó anteriormente, para que sea posible realizar las consultas con los nuevos operadores, constantes y atributos.

6.5. Constantes para IVFSQL

Ya se han definido constantes para FSQL. Todas ellas se mencionarán a continuación y se definirán otras dos constantes propias de IVFSQL

- Constantes básicas

6.5.1. UNKNOWN

Valor desconocido pero que puede aplicar en una consulta difusa

6.5.2. UNDEFINED

Valor indefinido. No es aplicable una consulta difusa sobre el atributo que posea esta constante.

6.5.3. NULL

No se sabe nada sobre el atributo, ni siquiera si es UNKNOWN o UNDEFINED. Posee valor nulo.

6.5.4. $[a,b,c,d]

Define una distribución de probabilidad trapezoidal.

$label

Define una etiqueta lingüística. Como $joven, $alto.

6.5.5. # n

Valor difuso que interpreta «aproximadamente n» en la consulta. Por ejemplo «aproximadamente 25 años».

- Constantes extendidas

6.5.6. n+-m

Aproximadamente n, con más o menos m. Usando para aproximar un radio de valores.

6.5.7. { P_1/L_1, P_2/L_2,..., P_n/L_n }

Distribución de probabilidad sobre etiquetas (labels, L_1,...,L_n). P_1,...,P_n son los posibles valores de probabilidad (entre 0 y 1).

6.5.8. { $P_1/\#N_1$, $P_2/\#N_2$,..., $P_n/\#N_n$ }

Distribución de probabilidad sobre números. P_1,...,P_n $\in$ {0,1}.

- Nueva constante para IVFSQL

6.5.9. { IVP_1/L_1, IVP_2/L_2,..., IVP_n/L_n }

Distribución de probabilidad sobre etiquetas usando conjuntos intervalo-valorados difusos. La idea es que la distribución de probabilidad no esté asociado a un valor fijo, sino a un rango de valores.

Por ejemplo, supongamos que definimos la etiqueta $llover, y queremos establecer que la posibilidad de que llueva está entre 0.4 y 0.6. Entonces, tenemos que $llover=[0.4,0.6]. Si definimos otra etiqueta $nevar, la cual $=[0.2,0.3], entonces tenemos la siguiente distribución de probabilidad:

{ [0.4,0.6]/$llover, [0.2,0.3]/$nevar }

Sea IVP_i, con i=1,...,n, si IVP_i = [a,b], entonces según la teoría de probabilidad difusa [3]:

$$IVP_i/L_i = \int_{\{x \in X/\delta(x) >= a\}} dP \tag{6.1}$$

donde δ(x) define la función del grado de pertenencia IVCDEG.

Capítulo 7

Conclusiones

Este trabajo de fin de máster se basó en la consolidación de una base sólida para poner en auge un sistema de base de datos relacionales con conjuntos intervalo-valorados difusos (IVFSQL, por sus siglas en inglés). El principal objetivo de esto es lograr un óptimo nivel de flexibilidad en las condiciones y consultas a bases de datos difusas.

Son los comparadores para IVFSQL uno de los principales aspectos propuestos para ello. Los mismos buscan establecer criterios lo suficientemente flexibles como para que el manejo de conjuntos intervalo-valorados difusos en ellos sea viable y provechoso. Al fin y al cabo, lo que busca cada uno de los comparadores es sacar provecho a la incertidumbre y usarla en favor de unas consultas tolerables a ella con el consiguiente margen de error de certeza. Un elemento de una base de datos será devuelto por el comparador si cumple la membresía definida por la función de pertenencia. Otro aspecto considerable es la fusificación de variables lingüísticas en IVFSQL, en donde para un universo de discurso se define un umbral de valores de pertenencia para cada etiqueta lingüística definida en él.

Los casos de estudio en este trabajo se han basado en la Base de Metaconocimiento Difusa para bases de datos difusas, ajustándolo a bases de datos difusas con conjuntos intervalo-valorados, creando así nuevos objetos y etiquetas dentro de ella. Uno de los principales puntos a favor de esto es la capacidad de inferencia que se puede aprovechar a partir del margen de error de la certeza de cada tupla devuelta por el comparador, o lo que es lo mismo, a partir de saber el margen de error de que la tupla devuelta por la selección, en efecto pertenezca a la selección.

7.1. Conclusions

This master degree project was based on the consolidation of a solid foundation for boosting a relational database system with interval-valued fuzzy sets (IVFSQL). The main purpose is to achieve an optimal level of flexibility in the conditions and queries to fuzzy databases.

The comparators for IVFSQL are one of the main aspects proposed. They seek to establish flexible enough criteria so that the management of interval-valued fuzzy sets in them would be viable and profitable. Basically, what each of the comparators seek is to take advantage of the uncertainty and use it in favor of tolerable queries with the consequent margin error of certainty. An element of a database will be returned by the comparator if it matches the membership defined by the membership function. Another important aspect is the merger of linguistic variables in IVFSQL, where for a universe of discourse, a threshold of belonging values is defined for each linguistic label defined in it.

The case of studies in this paper have been based following the Fuzzy Meta-knowledge Base for fuzzy databases, adjusting it to fuzzy databases with interval-valued fuzzy sets, thus creating new objects and labels within it. One of the main points in favor of this is the inference capacity that can be exploited from the margin of error of the certainty of each tuple returned by the comparator, or what is the same, from knowing the margin of error that the tuple returned by the selection, in effect belongs to the selection.

Bibliografía

[1] J. Galindo Gómez, *FSQL (Fuzzy SQL). A Fuzzy Query Language.*
Universidad de Málaga, 1999.

[2] Rafael Francisco Oliva Moreno, *Visual FSQL: gestión visual de bases de datos difusas en Oracle a través de internet usando FSQL.*
Universidad de Málaga, 2003.

[3] F.J. Montero de Juan, *Sobre la probabilidad difusa.* Disponible en internet: http://eprints.ucm.es/30423/1/Montero157.pdf
Universidad Complutense de Madrid, 1987.

[4] Ramón González del Campo Rodríguez Barbero, *Generalizaciones de las medidas de especificidad de la T-transitividad para conjuntos difusos intervalo valorados.* Disponible en internet: http://eprints.ucm.es/17918/1/T34138.pdf
Universidad Complutense de Madrid, 2013.

[5] J. Galindo Gómez, *Conjuntos y Sistemas Difusos.*
Universidad de Málaga, 1999.

[6] M. S. Yang, *A Survey of Fuzzy Clustering.*
Department of Mathematics. Chung Yuan Christian University, 1993.

[7] J. Galindo Gómez, *Operaciones con Conjuntos Difusos.*
Universidad de Málaga, 1999.

[8] J. Galindo Gómez, *Bases de Datos Relacionales Difusas (BDRD): Modelos Teóricos, Álgebra y Cálculo Relacional Difuso.*
Universidad de Málaga, 1999.

[9] Medina J.M., Pons O., Vila M.A., *GEFRED. A Generalized Model of Fuzzy Relational Data Bases. Information Sciences.*

[10] M. Pagola A. Jurio E. Barrenechea J. Fernández H. Bustince, *Interval-valued fuzzy clustering.*
Universidad Pública de Navarra.

[11] J.M. Medina Rodríguez, *Bases de Datos Relacionales Difusas: Modelo teórico y aspectos de su implementación.*
Universidad de Granada, 1991.

[12] H. Prade and C. Testemale, *Generalizing Database Relational Algebra for the Treatment of Incomplete/Uncertain Information and Vage Queries.*
Information Sciencies, 1984.

[13] L.A. Zadeh, *Fuzzy Sets.*
Information Control, 8, pp. 338-353, (1965).

[14] D. Dubois and H. Prade., *Fuzzy Sets and Systems: Theory and Applications.*
Academic Press. New York (1980).

[15] H. Bustince, J. Montero, M. Pagola, E. Barrenechea, and D. Gomez, *A Survey of Interval-Valued Fuzzy Sets.*
Handbook of Granular Computing, 2008

[16] H. Bustince, *Indicator of inclusion grade for interval-valued fuzzy sets. Application to approximate reasoning based on interval-valued fuzzy sets.*
Int. J. Approx. Reason. 2000.

[17] C. Cornelis, G. Deschrijver, and E. Kerre, *Intuitionistic fuzzy connectives revisited. In: Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems.*
IPMU'02, Annecy, Francia. 2002.

[18] G. Deschrijver, C. Cornelis, and E.E. Kerre, *On the representation of intuitionistic fuzzy T-norms and T-conorms.*
IEEE Trans. Fuzzy Syst. (2004).

[19] H. Bustince, J. Kacprzyk, and V. Mohedano, *Intuitionistic fuzzy generators. Application to intuitionistic fuzzy complementation.*
Fuzzy Sets Syst. (2000).

[20] J. Fodor and M. Roubens, *Fuzzy Preference Modelling and Multicriteria Decision Support, Theory and Decision.*
Kluwer Academic, Dordrecht, 1994.

[21] G. Deschrijver, *Arithmetic operators in interval-valued fuzzy set theory.*
Inf. Sci. 177(14) (2007)

[22] H.R. Tizhoosh, *Image thresholding using type-2 fuzzy sets.*
Pattern Recognit. 38 (2005)

[23] K. Atanassov, *Intuitionistic Fuzzy Sets. Theory and Applications.*
Physica-Verlag, Heidelberg, 1999.