Revista Cubana de Ciencias Informáticas (RCCI)
Vol. 6, No. 3, julio-septiembre, 2012
ISSN: 1994-1536 | e-ISSN: 2227-1899
RNPS: 0547 | e-RNPS: 2301
http://rcci.uci.cu

Tipo de artículo: contribución corta
Temática: Inteligencia artificial
Recibido: 4/07/2012 | Aceptado: 10/09/2012

Una mirada a las bases de datos difusas

A glance to the fussy databases

Yunior César Fonseca Reyna[1*], Oscar Gabriel Reyes Pupo[2], Mario Aballe Rodríguez[2], Alien Urquiza Jiménez[2]

[1] Departamento de Informática. Universidad de Granma, Carretera a Manzanillo, km 18 ½, Bayamo, Granma, Cuba. Correo-e: {fonseca@udg.co.cu, yuniorc}@uclv.edu.cu

[2] Departamento de Informática. Universidad de Holguín, Avenida XX Aniversario, Piedra Blanca, Holguín, Cuba. Correo-e: {oreyesp, maballer, aurquizaj}@ facinf.uho.edu.cu

* Correo para correspondencia: fonseca@uclv.edu.cu

Resumen: En este artículo se presenta una introducción general a las Bases de Datos Difusas comentando los modelos de implementación, la representación de la información, así como el manejo de las mismas.
Palabras clave: bases de datos difusas, conjuntos difusos, lógica difusa.

Abstract: *In this paper is present a general introduction to the Diffuse Databases commenting the implementation models, the representation of the information, as well as the handling of the same ones.*

Keywords: *fuzzy databases, fuzzy set, fuzzy logic.*

1. Introducción

Una de las características del lenguaje natural, que hace difícil su utilización en sistemas computacionales es su imprecisión. Por ejemplo conceptos como pequeño o grande, tienen significados diferentes de acuerdo al contexto en el que se estén utilizando, e incluso dentro del mismo contexto, pueden significar cosas diferentes para diferentes individuos. La teoría de los conjuntos difusos desarrollada por Zadeh (1965), provee una poderosa herramienta para la representación y manejo de la imprecisión por lo que actualmente está siendo utilizada en varios campos para el diseño de sistemas basados en reglas difusas.

La teoría de conjuntos difusos, extiende la teoría clásica de conjuntos al permitir que el grado de pertenencia de un objeto a un conjunto sea representada como un número real entre 0 y 1 en vez del concepto clásico en el que solo se tiene la posibilidad de pertenecer a un conjunto o no pertenecer al mismo; en otras palabras, el grado de pertenencia a un conjunto en la teoría clásica tiene solo dos valores posibles: 0 y 1 (Pradera *et al*., 2007; Cox, 1994). En el sentido más amplio, un sistema basado en reglas difusas es un sistema basado en reglas donde la lógica difusa es utilizada como una herramienta para representar diferentes formas de conocimiento acerca del problema a resolver, así como para modelar las interacciones y relaciones que existen entre sus variables. Debido a estas propiedades, los sistemas basados en reglas difusas han sido aplicados de forma exitosa en varios dominios en los que la información vaga o imprecisa emerge en diferentes formas (Galindo *et al.,* 2006).

Actualmente, el modelo relacional no permiten el procesamiento de consultas del tipo “Encontrar a todos los gerentes cuyo sueldo no sea muy alto” dado que ni el cálculo ni el álgebra relacional, que establecen el resultado de cualquier consulta como una nueva relación, tienen la capacidad de permitir consultas de una manera difusa.

En los últimos años, algunos investigadores han lidiado con el problema de relajar el modelo relacional para permitirle admitir algunas imprecisiones; esto conduce a sistemas de bases de datos que encajan en el campo de la Inteligencia Artificial, ya que permiten el manejo de información con una terminología que es muy similar a la del lenguaje natural. Una solución que aparece recurrentemente en los trabajos de investigación actuales en esta área es la fusión de los sistemas manejadores de bases de datos relacionales con la lógica difusa, lo que da lugar a lo que se conoce como Sistemas Manejadores de Bases de Datos Difusas o FRDBMS (del inglés, *Fuzzy Relational Database Management System*).

2. Desarrollo

Modelos de implementación

El problema de la implementación de los sistemas gestores de bases de datos difusas (SGBDR) ha sido tratado en dos vertientes principales (Galindo *et al*., 2006):

- Iniciar con un SGBDR con información precisa y desarrollar una sintaxis que permita formular consultas imprecisas, lo cual da origen a extensiones SQL, como Fuzzy SQL, con capacidades de manejar la imprecisión.
- Construir un gestor de bases de datos relacionales difusas (SGBDRD) prototipo que implemente un modelo concreto de base de datos relacional difusa en el que la información imprecisa pueda ser almacenada. Dentro de esta vertiente existen dos grandes ramas: Los modelos a través de unificación por relaciones de similitud y los modelos relacionales basados en distribuciones de probabilidades.

Representación de la información

Los elementos relacionados con la manipulación de información difusa pueden tener representaciones diferentes. Por ejemplo, una distribución normalizada de probabilidades puede ser representada por diferentes tipos de funciones (trapezoidal, triangular, intervalos, etc.). Lo más usual, es que se usen funciones de tipo trapezoidal. Lo mismo puede decirse de la forma en la que se modelan los operadores relacionales difusos así como los demás elementos difusos que aparezcan en el sistema.

El criterio empleado para seleccionar la forma de representación de los múltiples elementos difusos del sistema manejador de base de datos, puede afectar de manera determinante la funcionalidad y desempeño de la base de datos, por lo que debería ser uno de los puntos centrales en los que el experto ajuste la arquitectura del FRDBMS al problema específico a tratar mediante el mismo. Puede decirse entonces que este criterio de selección y ajuste constituye un paso entre la formulación de una base de datos relacional difusa y la implementación de un sistema basado en la misma. La información que se puede manejar en una base de datos difusa puede dividirse en dos tipos principales (Galindo *et al*., 2006; Wong y Flores, 2005; Rutkowski, 2004):

- **Datos Precisos**.

  Manejados usualmente mediante la representación provista por la base de datos relacional huésped.
- **Datos Imprecisos.**

Revista Cubana de Ciencias Informáticas (RCCI)
Vol. 6, No. 3, julio-septiembre, 2012
ISSN: 1994-1536 | e-ISSN: 2227-1899
RNPS: 0547 | e-RNPS: 2301
http://rcci.uci.cu

Los modelos usualmente consideran dos tipos de representación para los datos imprecisos además de la información desconocida o indeterminada que se maneja mediante los tipos *unknown*, *undefined* y *null*:

- **Datos imprecisos sobre dominios ordenados**

  Este grupo de datos contiene distribuciones de probabilidad definidas en dominios continuos o discretos, pero ordenados.

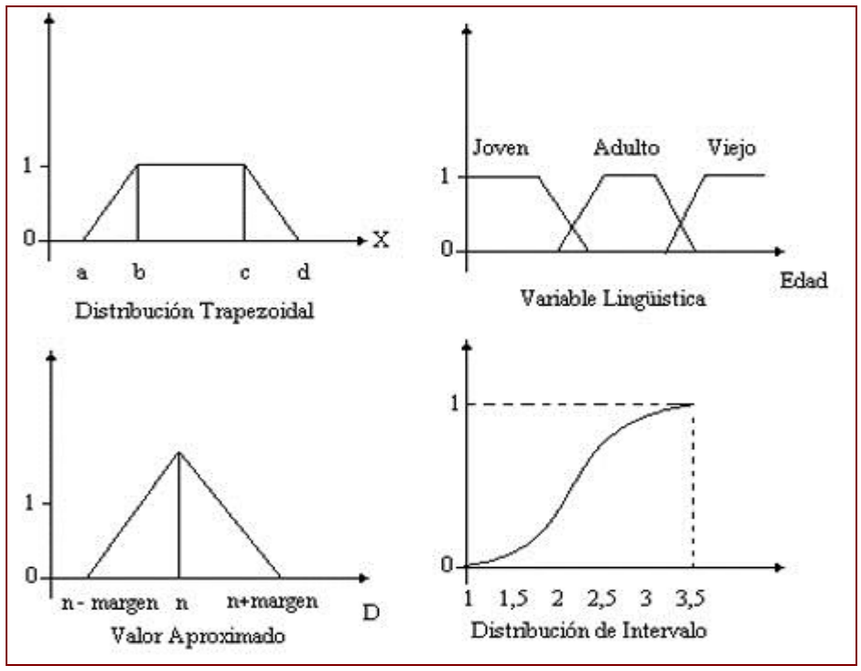

Figura 1. Datos imprecisos sobre dominios ordenados.

– **Datos con analogías sobre dominios discretos**

  Este grupo de datos se construye sobre dominios discretos en los que existen definidas relaciones de proximidad entre sus valores. En este caso se deberá almacenar la representación de los datos además de la representación de las relaciones de proximidad definidas para los valores en el dominio.

– **Tipo de dato Indefinido (undefined**)

  Cuando un atributo toma el valor **undefined**, esto refleja el hecho de que ningún valor de su dominio es permitido. Por ejemplo: el número de teléfono de alguien que no tiene teléfono.

– **Tipo de dato desconocido (unknown)**

  Los datos de este tipo expresan nuestra ignorancia sobre el valor que el atributo toma, sin embargo expresa también que puede tomar uno de los valores del dominio. Por ejemplo la fecha de nacimiento de alguien, la desconocemos pero tiene que tener alguna.

– **Tipo de dato nulo (null)**

  Cuando un atributo toma el valor nulo, esto significa que no tenemos información sobre él, ya sea porque no conocemos su valor o porque es imposible asignarle un valor del dominio. Por ejemplo el email de alguien es null si desconocemos su valor o si lo tiene o no.

Operaciones relacionales difusas

Los diferentes operadores de comparación que se utilizan para representar relaciones en la base de datos son los operadores relacionales. Para que estos funcionen sobre información imprecisa es necesario extender estos operadores.

Revista Cubana de Ciencias Informáticas (RCCI)
Vol. 6, No. 3, julio-septiembre, 2012
ISSN: 1994-1536 | e-ISSN: 2227-1899
RNPS: 0547 | e-RNPS: 2301
http://rcci.uci.cu

La representación adoptada se basa en el trabajo previo de Zadeh (Zadeh, 1965) y es la siguiente:

- **Igual a:**

  Este operador modela el concepto de igualdad para datos imprecisos.

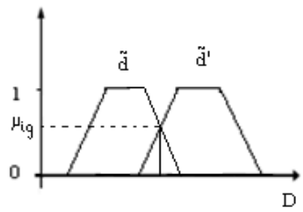

Figura 2. Igualdad para datos imprecisos.

- **Aproximadamente igual:**

  Este operador define el grado en el que dos valores numéricos (no difusos) son aproximadamente iguales de acuerdo a si la diferencia de sus valores se encuentra dentro de un límite preestablecido. Y se calcula mediante la expresión (1) :

$$margen \quad si \quad x - y > margen \qquad (1)$$

  A continuación se muestra la representación gráfica para este operador:

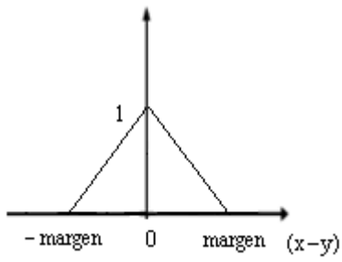

Figura 3. Representación del operador aproximadamente igual.

Manejo de las BDRD

Para el manejo de las bases de datos relacionales difusas (BDRD) se utiliza el lenguaje Fuzzy SQL (FSQL) que es un lenguaje que deriva de SQL, incorporando las siguientes novedades (Galindo, et al., 2006).

- **Etiquetas Lingüísticas**: En las sentencias FSQL las etiquetas van precedidas del símbolo $, para poder distinguirlas fácilmente.
- **Comparadores Difusos:** Permiten comparar dos atributos o un atributo con una constante.
- **Conectivas Lógicas**: Pueden usarse NOT, AND y OR, para enlazar condiciones difusas simples.
- **Umbral de Cumplimiento** (threshold) : Tras cada condición simple puede imponerse un umbral de cumplimiento mínimo (por defecto es 1), con el siguiente formato: <condición_simple> THOLD γ . La palabra reservada THOLD es opcional y puede sustituirse por un comparador tradicional (=, <, **≤**,...) modificando el sentido de la consulta. Por defecto es equivalente al comparador **≥**.
- **Constantes Difusas**: Pueden usarse en el SELECT todas las constantes difusas ya definidas: UNKNOWN, UNDEFINED y NULL, $[a,b,c,d] (Distribribución de posibilidad Trapezoidal), $label (Etiquetas), [n,m] (Intervalo) y #n (valores aproximados).
- **Función CDEG** (<atributo>): Usada en la lista de selección, la función CDEG calcula, para cada tupla, el grado de cumplimiento del atributo del argumento en la condición de la cláusula WHERE.

- **Función CDEG**(*): Calcula el grado de cumplimiento de cada tupla en la condición de forma global, para todos sus atributos y no sólo para uno de ellos en particular. La función CDEG usa, por defecto, los operadores típicos para la negación (1–x), conjunción (t-norma del mínimo) y disyunción (s-norma del máximo), pero pueden usarse otros (si se definen).
- **Carácter Comodín %:** Similar al carácter comodín * de SQL, pero este incluye además la función CDEG aplicada a todos los atributos de la condición. No incluye CDEG(*).
- **Condición con IS:** También admite condiciones del tipo: <atributo_difuso> IS [NOT] {UNKNOWN | UNDEFINED | NULL}
- **Cuantificadores Difusos**: Tiene dos modalidades que se aplican como condición en la cláusula HAVING que sigue a una cláusula GROUP BY: o "Q elementos de X cumplen A": $Cuantificador FUZZY[r] (condición_difusa) THOLD γ .

Ejemplos

a) "Dame todas las personas cuya edad es aproximadamente 20 años": (con grado mínimo 0.6):

```
SELECT *    FROM Personas WHERE Edad FEQ #20 THOLD 0.6;
```

b) "Dame todas las personas más o menos Rubias (con grado mínimo 0.5) cuya edad es posiblemente superior a Joven (con grado mínimo 0.8)":

```
SELECT * FROM Personas WHERE Pelo FEQ $Rubio THOLD 0.5 AND Edad FGT $Joven THOLD 0.8;
```

c) "Equipos que tienen muchos más de 3 (con grado mínimo 0.5) jugadores Altos" (con grado mínimo 0.75)":

```
SELECT Equipo, CDEG(*) FROM Personas GROUP BY Equipo HAVING $Muchos_Mas_Que[3] (Altura FEQ $Alto 0.75) 0.5;
```

Resumen de características

Ventaja:

- **Almacenar Imprecisión**: la información que tengamos de un atributo particular de un objeto, aunque esta información no sea el valor exacto. Suelen usar Etiquetas Lingüísticas con alguna definición asociada (por ejemplo, un conjunto difuso visto como una "Distribución de Posibilidad"), o sin ninguna definición asociada ("escalares" con una relación de similitud definida entre ellos).

Inconvenientes:

- Lenguaje de consulta **incómodo**, debido al gran número de parámetros que deben utilizarse.
- Comparadores **abstractos** que hacen difícil la decisión de cuál debemos usar.
- Falta de **estandarización**, derivado de la poca popularidad de este tipo de bases de datos.

3. Conclusiones

Las Bases de Datos Difusas permiten recuperar datos con tan solo una vaga descripción de lo que deseamos obtener. Esto es un importante avance al acercar el lenguaje natural a la organización formal de los datos. Puede implementarse, no sin dificultades, en SGBD ordinarios y el lenguaje de acceso a esta es una extensión de SQL. Sin

embargo, requiere una gran cantidad de parámetros, lo que hace el manejo muy incómodo, lo cual constituye la principal causa de su impopularidad.

Referencias

COX, E. The Fuzzy Systems Handbook. Chestnut Hill-United Kingdom. Academic Press Limited. 1994. 667.

GALINDO, J., *et al*. Fuzzy Databases: Modeling, Design and Implementation. Covent Garden-London. Idea Group Inc. 2006. 341.

LIMA, D. Modelo de diseño conceptual para una Base de Datos Relacional Difusa La Paz - Bolivia: MCAL Antonio José de Sucre; 2004.

PRADERA, A., *et al*. On Fuzzy Set Theories. Fuzzy Logic, a Spectrum of Theoretical and Practical Issues. New York. Springer-Verlag Berlin Heidelberg. 2007. p. 15-47.

RUTKOWSKI, L. FLEXIBLE NEURO-FUZZY SYSTEMS. Structures, Learning and Performance Evaluation. Czestochowa - Polonia. KLUWER ACADEMIC PUBLISHERS. 2004. p. 294.

WONG, C., *et al*. Fuzzy Queries. Un framework para realizar consultas difusas en Postgres desde aplicaciones Java. 2005. p. 10.

ZADEH, L. Fuzzy Sets. Information and Control. 1965. Vol. 8. p. 338-353.