Aplicando Conjuntos Difusos en la Generación de Reglas de Asociación

Mariluz Martínez
Universidad del Valle, Escuela de Ingeniería de Sistemas y Computación
Cali, Colombia
mariluzm@libertad.univalle.edu.co

Gelver Vargas
Universidad del Valle, Escuela de Ingeniería de Sistemas y Computación
Cali, Colombia
gelvervb@libertad.univalle.edu.co

Andrés Dorado
Pontificia Universidad Javeriana, Carrera de Ingeniería de Sistemas y Computación
Cali, Colombia
adorado@puj.edu.co

Marta Millán
Universidad del Valle, Escuela de Ingeniería de Sistemas y Computación
Cali, Colombia
millan@eisc.univalle.edu.co

Abstract

The association rules model is one of most widely used in data mining. An association rule is an implication of the form $X \rightarrow Y$, where X and Y are set of items that satisfy two constraints given by the user called minimum support (*minsup*) and minimum confidence (*minconf*). Normally, the values of *minsup* and *minconf* are crisp. In this paper, we analize how the association rules mining is affected when these values are treated as fuzzy. An algorithm based on the Apriori algorithm is proposed in order to calculate frequent itemsets and to generate association rules using fuzzy sets.

Keywords: Frequent Pattern Mining, Association Rules, Data Mining, Fuzzy Systems

Resumen

El modelo de reglas de asociación es uno de los más ampliamente utilizados en minería de datos. Una regla de asociación es una implicación de la forma $X \rightarrow Y$, donde X e Y son conjuntos de items que satisfacen dos restricciones dadas por el usuario denominadas soporte mínimo (*minsup*) y confianza mínima (*minconf*). Normalmente, los valores *minsup* y *minconf* son exactos. En este artículo, analizamos la forma en que se afecta la minería de reglas de asociación cuando dichos valores son tratados como difusos. Un algoritmo basado en Apriori es propuesto, a fin de calcular conjuntos de items frecuentes y generar reglas de asociación usando conjuntos difusos.

Palabras Claves: Minería de Patrones Frecuentes, Reglas de Asociación, Minería de Datos, Sistemas Difusos

1 Introducción

Una tarea importante en minería de datos es la de reglas de asociación. Una regla de asociación es una expresión de la forma $X \rightarrow Y$ donde X e Y son conjuntos de items. Intuitivamente, la regla significa que las transacciones que contienen a X tienden a contener a Y. Por ejemplo, "El 86% de las personas que compran papas fritas y refresco compran pañales", es una de regla asociación.

Una aplicación típica de las reglas de asociación es el análisis de datos de la canasta de compras, en donde un registro consiste de una fecha de transacción y de los items comprados. Este tipo de reglas de asociación se conoce como reglas de asociación booleanas. Muchos algoritmos (i.e. Apriori, AprioriTID, AprioriHybrid, FP-Tree, DIC) han sido propuestos para el cálculo de reglas de asociación booleanas [1][10][11][12][4].

De otro lado, las reglas de asociación se pueden también generar a partir de conjuntos de datos relacionales, en los cuales los atributos pueden ser cuantitativos. Las reglas de asociación cuantitativas fueron introducidas en [10], en las cuales los dominios de los atributos se discretizan en intervalos disjuntos. Sin embargo, bajo este modelo no se tienen en cuenta los elementos cercanos a los límites de los intervalos y parece ser poco intuitivo, desde el punto de vista del usuario [9].

Para abordar este problema, se han introducido conceptos de conjuntos difusos. En [6] se define una regla de asociación difusa como SI X es A ENTONCES Y es B, donde A y B son conjuntos difusos que caracterizan a X y a Y. El uso de conceptos difusos permite, por ejemplo, que X pertenezca, en distinta proporción a varios conjuntos difusos. Por ejemplo, si se define la variable *estatura* utilizando los conjuntos difusos *baja* (0-160 cm), *mediana* (155-180 cm) y *alta* (175-230 cm), una persona con estatura de "178 cm" será considerada "no baja" y "entre mediana y alta". Para determinar el grado de pertenencia, del valor crisp que instancia la variable a cada conjunto, se utiliza una función característica. En el modelo propuesto en este artículo, el usuario debe ofrecer los conjuntos difusos para los atributos cuantitativos con sus funciones características o funciones de pertenencia correspondientes.

En [8] se propone un método para calcular los conjuntos difusos usando técnicas de *clustering*. Se propone, a partir de los datos, determinar los conjuntos difusos automáticamente usando técnicas de clustering y se presenta un algoritmo para encontrar las funciones de pertenencia para los conjuntos difusos calculados.

Una propuesta para calcular reglas de asociación difusas, en las cuales se incluyen tanto atributos categóricos como continuos, se presenta en [7]. Con el propósito de hacer las reglas más entendibles al usuario se introduce una nueva definición de regla cuantitativa que utiliza conceptos difusos.

En este artículo se propone utilizar inferencia difusa para tomar en consideración conjuntos de items frecuentes y reglas de asociación que son descartados en Apriori a pesar de estar muy cerca del límite (umbral) de aceptación. Esto se debe a la característica crisp del soporte y la confianza. Además, se ofrece un sistema de control difuso directo que le facilita al usuario definir los parámetros de confianza y soporte.

El resto del artículo está organizado en secciones. En la sección 2, se presentan los conceptos preliminares en los cuales se apoya el modelo propuesto. En la sección 3, se propone un modelo difuso para minería de reglas de asociación. En la sección 4, se presentan resultados experimentales. Finalmente, en la sección 5, se presentan las conclusiones.

2 Conceptos Preliminares

2.1 Reglas de Asociación

Uno de los objetivos de la minería de datos es encontrar un conjunto de reglas que describan propiedades de los datos. En particular, una regla de asociación[1] es una expresión de la forma $X \rightarrow Y$, donde X e Y son conjuntos de items. Intuitivamente, $X \rightarrow Y$ significa que las transacciones de la base de datos que contienen X tienden a contener Y. Un ejemplo de regla puede ser "el 98% de los clientes que compran llantas y accesorios para automóviles también compran servicios para automóviles".

Formalmente, una regla de asociación se define de la siguiente forma[1]:

> Sea $I = \{i_1, i_2, \ldots, i_n\}$ un conjunto de literales, llamados items. Sea D un conjunto de transacciones (base de datos a explorar), donde cada transacción $T \subseteq D$. Se dice que una transacción

T es un conjunto de items X, si $X \subseteq T$. Una regla de asociación es una implicación de la forma $X \rightarrow Y$, donde $X \subseteq I$ y $Y \subseteq I$, $X \cap Y = \emptyset$. La regla $X \rightarrow Y$ tienen soporte s, en el conjunto de transacciones T, si el $s\%$ de las transacciones en D contienen $X \cup Y$ y tiene una confianza c si el $c\%$ de las transacciones en D que contienen a X también contienen a Y.

2.2 Lógica Difusa

La lógica es el estudio de los métodos y principios de razonamiento en todas sus posibles formas. La lógica clásica trata con proposiciones que son verdaderas o falsas. En contraste con los sistemas de lógica clásica, la lógica difusa se propone como una formalización de los modos de razonamiento que son aproximados. Lo que se explota en la mayoría de las aplicaciones es la tolerancia de la lógica difusa de la imprecisión. De hecho, el principio operativo de la lógica difusa es: "la precisión es costosa y por lo tanto, se debe trabajar con la necesaria para realizar la tarea."

La teoría de conjuntos difusos, de la cual emergió la lógica difusa, extiende la teoría tradicional de conjuntos para resolver problemas generados, algunas veces, por la rigidez de la clasificación de "todo o nada" de la lógica Aristotélica. Tradicionalmente, una condición o expresión lógica podía estar solamente en uno de dos extremos: completamente verdadero o completamente falso. Sin embargo, en el mundo difuso, los valores se ubican dentro de un rango de 0 a 100% verdaderos o falsos. En otras palabras, todas las proposiciones difusas tienen algún grado de verdad real entre 0 y 1, inclusive.

2.2.1 Conjuntos Difusos

El uso de términos lingüísticos que involucran vaguedad es muy común en la vida diaria. Por ejemplo, es normal escuchar a alguien referirse a la temperatura ambiente como: "el día está frío" o "el cuarto está caliente". Es poco común escuchar algo como: "el día tiene una temperatura de 8°C." El término lingüístico frío es relativo al observador y al tratar de definirlo numéricamente se podría pensar en un rango de temperaturas que están "alrededor de" 10°C, por ejemplo.

Este dominio de valores posibles, asociados a un término lingüístico, conforman el *Universo del Discurso* de lo que se conoce como un *Conjunto Difuso*. La función que, dado un valor indica su grado de pertenencia (o de verdad) al conjunto, se conoce como *Función Característica o Función de Pertenencia*. Los valores de verdad no se restringen exactamente a los propuestos en la lógica clásica de Verdadero (1) o Falso (0), sino que se extienden al rango considerado por la lógica difusa [0,1].

2.2.2 Formas Típicas de Funciones de Pertenencia

Cada conjunto difuso se identifica con una etiqueta y cada variable difusa está compuesta por un grupo de conjuntos difusos. El significado de un término lingüístico difuso caracterizado por una función de pertenencia es asignado, intiuitivamente, por la persona que usa dicho término.

La distribución de valores de verdad para un conjunto difuso se puede representar por medio de una función gaussiana. Para disminuir la complejidad de las implementaciones software o hardware de aplicaciones que utilizan conjuntos difusos, se pueden usar funciones lineales (*piecewise linear functions*). Estas funciones deben cumplir con una serie de axiomas para considerarse válidas. Dentro de las más comunes se encuentran las funciones lineales: Zeta (Z), Lambda (Λ), Ese (S) y Pi (Π), cuyos nombres corresponden a la forma.

2.3 Sistemas Difusos

Hay personas que por sus conocimientos y habilidades, llegan a ser expertos en el control del sistema bajo su supervisión. Estas personas pueden formular las reglas básicas que describen el comportamiento del sistema y las características de cada variable. Es en el conocimiento de estos expertos en el que se centra la atención primaria para la definición de un sistema basado en lógica difusa compartiendo, de esta manera, la característica típica de los sistemas expertos de fundamentarse en la experiencia y en el comportamiento humano.

Los sistemas difusos se enmarcan dentro de los paradigmas que pretenden imitar funciones intelectuales o biológicas encontradas en el ser humano, y se categorizan dentro de lo que se denomina *Inteligencia Computacional* [5]. La Inteligencia Computacional depende de los datos numéricos suministrados por los usuarios y no se apoya completamente en el conocimiento (o en la codificación del mismo.) Por su parte, la Inteligencia Artificial estudia los mecanismos mediante los cuales opera el pensamiento humano y emplea porciones

del conocimiento para poder inferir o representar conclusiones.

Un sistema difuso involucra tres etapas primarias: "fusificación", inferencia difusa y "defusificación". Por medio de funciones de pertenencia, permite convertir expresiones lingüísticas en números que se pueden manipular fácil y eficazmente. La Figura 1 describe el esquema de un sistema difuso.

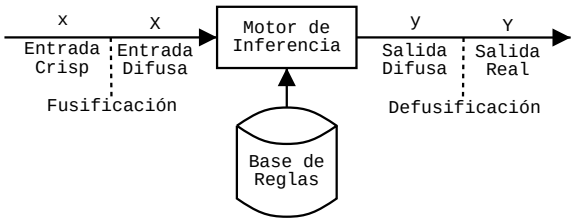


Figure 1: Esquema de un Sistema Difuso

2.3.1 Fusificación

En un sistema difuso cada variable se asocia con un grupo de términos lingüísticos que corresponden a conjuntos difusos. En la mayoría de los modelos, cada variable tiene entre tres y nueve etiquetas. Cuando la variable de entrada recibe un valor numérico, éste pasa por la etapa de fusificación, que consiste en obtener el grado de pertenencia de este valor a cada uno de los conjuntos difusos de la variable. El resultado de esta operación es un valor difuso.

Las funciones de pertenencia son provistas por el programador para dar un significado numérico a cada término. Cada función de pertenencia identifica el rango de valores de entrada que corresponden a una etiqueta. A diferencia de la lógica booleana convencional, los límites de estos rangos no son puntos de corte donde la etiqueta aplica completamente de un solo lado y del otro no. En su lugar, hay una región donde los valores de entrada cambian gradualmente de ser completamente aplicables a completamente inaplicables.

La etapa de fusificación relaciona valores numéricos concretos con expresiones lingüísticas vagas. Luego, las reglas escritas con estas expresiones lingüísticas se pueden evaluar con precisión matemática, en una Unidad Central de Procesamiento (CPU) normal.

2.3.2 Inferencia Difusa

Una de las formas de expresar el conocimiento del experto es mediante del uso de reglas de la forma

`SI situación ENTONCES acción.`

Este tipo de reglas posee dos características representativas: son cualitativas en lugar de cuantitativas y cada situación se relaciona con una acción apropiada.

La importancia del uso de este tipo de reglas se apoya en el hecho que mucho del conocimiento humano se presta para ser representado en forma de una jerarquía de reglas SI... ENTONCES... Además, los mecanismos de inferencia con estas reglas son relativamente simples y están en armonía con los modos de razonamiento humano, el cual es aproximado en lugar de exacto.

En la etapa de evaluación de reglas o inferencia difusa, se calculan los valores difusos de salida para las variables que correspondan, usando las relaciones entre variables de entrada y de salida que contiene la base de reglas lingüísticas suministradas por el experto. En este punto, varias reglas pueden ser verdaderas con diferentes grados, produciéndose competencia entre sus resultados.

Aunque las reglas parecen una forma libre del lenguaje natural, ellas están limitadas a un conjunto de términos lingüísticos y a una estricta sintaxis. Cada antecedente (Situación o Condición asociada al SI) de una regla corresponde a un valor específico de entrada difusa. Este valor de entrada difusa resulta de la etapa de fusificación. Cada consecuente (Acción o Conclusión asociada al ENTONCES) de una regla corresponde a una salida difusa.

Utilizando un operador llamado, por algunos autores, *Operador de Agregación*, se combinan los valores

de verdad o grados de pertenencia de los antecedentes de la regla y se calcula su valor de verdad. Este valor se aplica a todos los consecuentes de la regla. Es posible, para una variable de salida difusa, tener un conjunto difuso como consecuente en más de una regla. Cuando esto sucede, se toma como valor de verdad para esta variable de salida difusa, el resultado de aplicar un operador sobre los consecuentes que aparecen en dichas reglas, llamado *Operador de Composición*. El resultado de la evaluación de las reglas es un conjunto completo de valores de salida difusa las cuales reflejan el efecto de todas las reglas cuyos valores de verdad son mayores que cero.

2.3.3 Defusificación

La etapa de Defusificación consiste en combinar los valores de los conjuntos difusos, de las variables de salida involucradas en la instancia del proceso, por medio de una función para obtener un valor numérico real y aplicarlo a la variable de salida del sistema. En los primeros sistemas difusos se tomaba el mayor valor (máximo o más fuerte), pero actualmente es considerado como un método pobre, debido a que ignora la contribución de las reglas diferentes a la más fuerte. Existen, en la actualidad, muchos planteamientos para el cálculo de los valores para las variables de salida de un sistema difuso, uno de ellos es el que se conoce como *Centro de Gravedad*, que considera la contribución de todos los valores de salida difusos.

2.4 Sistemas de Control Difuso Directo

El problema de control se puede plantear como "El conjunto de acciones que se deben tomar para hacer que el comportamiento de un sistema, reflejado en las variables que lo caracterizan, cumpla con los requerimientos establecidos de forma tal que alcance los fines para los cuales ha sido creado."

A diferencia de las técnicas convencionales de control basadas en modelos matemáticos, la teoría de control difuso define un controlador del sistema directamente del dominio de conocimiento de expertos. Los sistemas de control difuso, producto de la aplicación de técnicas de Inteligencia Computacional combinadas con otras técnicas convencionales, proveen una platarforma confiable para el control de sistemas hardware o software.

Uno de los malentendidos más comunes con relación a los sistemas de control difuso, es pensar que no requieren el modelo del proceso. Si no se tiene algún conocimiento del proceso, resulta imposible controlarlo, incluso manualmente (el dominio del experto tiene implícito un conocimiento del proceso.)

Tres aspectos fundamentales para la implementación de un sistema de control basado en lógica difusa son: la representación del conocimiento (Cómo construir una estructura de reglas lingüísticas de la forma SI...ENTONCES... que pueda tener significado numérico), la estrategia de razonamiento (Cómo obtener acciones razonables ante las situaciones que se presenten) y la adquisición del conocimiento (Cómo definir claramente un conjunto de reglas de control).

3 Modelo Difuso para Minería de Reglas de Asociación

3.1 Algoritmo FUZZYC

En esta sección se describe el algoritmo FUZZYC [2], que implementa un controlador para sistemas basados en conjuntos difusos.

La rutina de fusificación recibe un valor para cada variable de entrada, calcula el grado de pertenencia del valor, para cada uno de los conjuntos difusos definidos para la variable, y lo registra en las estructuras de datos de los conjuntos o términos de la variable de entrada, si el grado de pertenencia es mayor que cero. Los grados de pertenencia, de cada término, se calculan teniendo en cuenta la forma de la función de pertenencia y el valor de la variable de entrada.

Una vez calculado el grado de pertenencia, la rutina de fusificación funciona de la siguiente manera: busca los antencedentes que contengan los términos actualizados, cuyos grados de pertenencia fueron diferente de cero en la base de reglas, y les asigna el valor obtenido. Si el valor, antes de la asignación en el antecedente es indefinido, se decrementa un contador en la regla, que representa el número de antecedentes con valor asignado. Si el contador llega a cero, se activa una bandera que indica que la regla está lista para participar en la inferencia difusa.

El valor de verdad de la regla será el resultado de aplicar un operador de agregacón sobre sus antecedentes.

Los operadores de agregación definidos para FUZZYC son: el mínimo de Mamdani, el producto algebráico, el producto de Larsen, el producto Acotado y el producto Drástico.

Una vez termina la rutina de fusificación, se procede a ejecutar la rutina de inferencia difusa, siguiendo estos pasos: Se aplica (o asigna) el valor de verdad de las reglas que tienen la bandera de participación activa a cada uno de sus consecuentes. Este valor se almacena en la estructura de datos de los términos de las variables de salida correspondientes. Para determinar el valor a asignar, cuando un término de salida aparece en más de una regla, se utiliza como consecuente un operador de composición. Los operadores de composición definidos en FUZZYC son: la unión, la suma algebráica, la suma acotada, la suma drástica y la suma disjunta. Todas las reglas se reinicializan para la siguiente iteración.

La rutina de defusificación sigue a la de inferencia difusa. Inicia recorriendo todas las variables de salida que se han marcado como activas (al menos uno de sus términos tiene un grado de pertenencia mayor que cero) en la etapa anterior. Para cada una de éstas variables activas la función calcula su valor final o de salida. El valor en el dominio de los números reales se calcula utilizando el método llamado *Centro de Area*. El resultado se aplica a las variables de salida del sistema.

3.2 F-Apriori: Extensión del Apriori utilizando FUZZYC

El cálculo de conjuntos frecuentes (conjunto de items cuyo soporte es al menos el mínimo definido por el usuario) usando el algoritmo Apriori [1], se inicia con la identificación de los conjutos frecuentes de tamaño $k = 1$, para lo cual se recorre la base de datos de transacciones. A partir de este conjunto de items frecuentes de tamaño k, se calculan los conjuntos de items de tamaño $k + 1$ y se verifica su condición de frecuencia en la base de datos. En cada iteración, el nuevo conjunto de items frecuentes de tamaño k sirve de semilla para el cálculo de los de tamaño $k + 1$.

Para el caso del algoritmo F-Apriori, el proceso de cálculo de conjuntos de items frecuentes, se realiza de manera similar. Se calculan los conjuntos de items frecuentes de tamaño $k+1$, con base en los de tamaño k. La diferencia en los dos procesos se basa fundamentalmente en la forma en la que se verifica la condición de frecuencia. En el caso de F-Apriori, ésta verificación se realiza utilizando el sistema de control difuso directo FUZZYC, en el cual el usuario ha configurado las condiciones de aceptación de items.

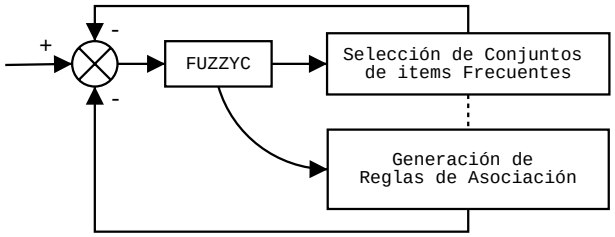


Figure 2: Sistema de Control Difuso utilizado por F-Apriori

La extensión de Apriori a F-Apriori consiste en modificar los dos subproblemas en los cuales éste se puede descomponer: cálculo de conjuntos de items frecuentes y generación de reglas de asociación. La extensión del cálculo de conjuntos de items frecuentes consiste en:

- Fusificar el mínimo soporte ofrecido por el usuario.
- Fusificar el soporte del conjunto de items.
- Determinar si el conjunto de items es frecuente, utilizando las reglas de inferencia difusas y la rutina de Defusificación.

De acuerdo con el modelo definido, al fusificar los valores crisp para las variables de entrada: `Mínimo soporte = 0.8` y `Soporte = 0.9`, se obtienen los valores difusos: Mínimo soporte `alto` con un grado de pertenencia al conjunto de 0.5 y `medio alto` con un grado de pertenencia al conjunto de 0.5 y soporte `alto` con un grado de pertenencia 1 al conjunto. La Figura 3 muestra la definición del soporte mínimo utilizada en el modelo.

Del conjunto de reglas de inferencia difusa definidas en la Figura 4 se activan las siguientes:

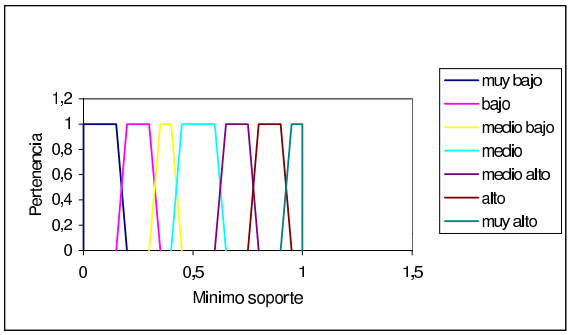


Figure 3: Definición del Soporte Mínimo Difuso

- SI `Mínimo soporte = Alto` Y `Soporte = Alto`
  ENTONCES el `Conjunto de items es Frecuente`
- SI `Mínimo soporte = Medio alto` Y `Soporte = Alto`
  ENTONCES el `Conjunto de items es Frecuente`.

soporte: Sop. Min:	Muy bajo	Bajo	Medio bajo	Medio	Medio alto	Alto	Muy alto
Muy bajo	frec	frec	frec	frec	frec	frec	frec
Bajo	no frec	frec	frec	frec	frec	frec	frec
medio bajo	no frec	no frec	frec	frec	frec	frec	frec
medio	no frec	no frec	no frec	frec	frec	frec	frec
medio alto	no frec	no frec	no frec	no frec	frec	frec	frec
Alto	no frec	no frec	no frec	no frec	no frec	frec	frec
Muy alto	no frec	no frec	no frec	no frec	no frec	no frec	frec

Figure 4: Reglas de Inferencia para determinar conjuntos de items frecuentes

Al realizar el proceso de inferencia difusa con dichas reglas, se obtiene un valor difuso de 0.5 para la variable de salida que corresponde, en este caso particular, al valor crisp de 0.5. Según la definición difusa de la variable de salida de la Figura 5, el conjunto de items es considerado frecuente.

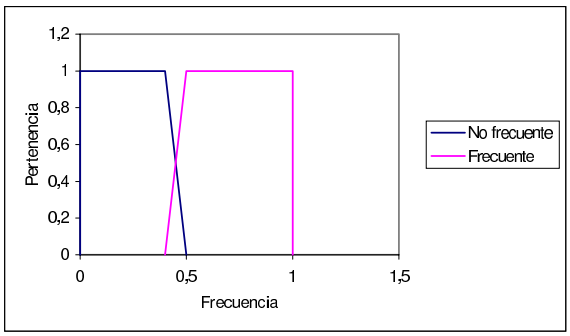


Figure 5: Definición de la Variable de Salida Difusa

El segundo subproblema que aborda Apriori es calcular las reglas de asociación a partir de los conjuntos de items frecuentes obtenidos. En este caso y para cada conjunto de items frecuente, Apriori calcula todas las reglas de la forma $a \rightarrow l - a$, donde l es un conjunto frecuente y $a \subset l$ y cuya confianza sea, al menos, igual a la definida por el usuario.

Para el caso del F-Apriori, el proceso de cálculo de las reglas de asociación es similar, excepto que ahora la verificación de la confianza mínima se trata de manera difusa. La extensión de Apriori consiste entonces en:

- Fusificar la mínima confianza definida por el usuario.
- Fusificar la confianza de la regla de asociación candidata.

- Determinar si se acepta la regla de asociación, utilizando las reglas de inferencia difusas y la rutina de Defusificación.

De acuerdo con el modelo definido, al fusificar los valores crisp para las variables de entrada: `Mínima confianza = 0.87` y `Confianza = 0.9`, se obtienen los valores difusos: Mínima confianza `alta` con un grado de pertenencia al conjunto de 0.4 y `media` con un grado de pertenencia al conjunto de 0.6 y la confianza `alta` con un grado de pertenencia 1 al conjunto. La Figura 6 muestra la definición de la confianza mínima utilizada en el modelo.

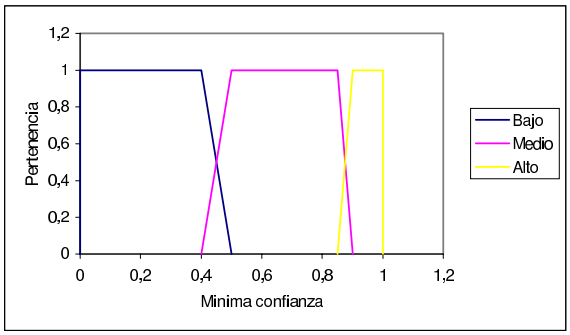


Figure 6: Definición de la Confianza Mínima Difusa

Del conjunto de reglas de inferencia difusa definidas en la Figura 7 se activan las siguientes:

- SI `Mínima confianza = Alta` Y `Confianza = Alta`
  ENTONCES la `Regla de asociación es Aceptada`
- SI `Mínima confianza = Media` Y `Confianza = Alta`
  ENTONCES la `Regla de asociación es Aceptada`.

Confianza: minConf:	Baja	Media	Alta
Baja	Acepto	Rechazo	Rechazo
Media	Acepto	Acepto	Rechazo
Alta	Acepto	Acepto	Acepto

Figure 7: Reglas de Inferencia para determinar reglas de asociación

Al realizar el proceso de inferencia difusa con dichas reglas, se obtiene un valor difuso de 0.6 para la variable de salida que corresponde, en este caso al valor crisp de 0.65. Según la definición difusa de la variable de salida de la Figura 8, la regla de asociación es aceptada.

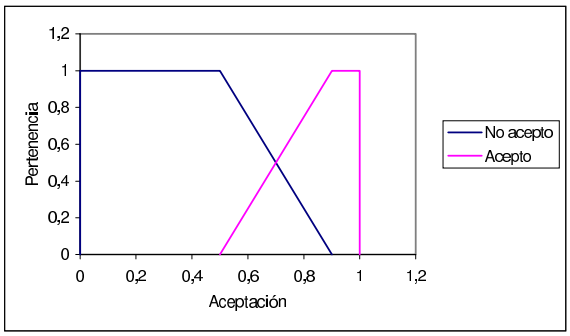


Figure 8: Definición de la Variable de Salida Difusa

4 Resultados Experimentales

F-Apriori ha sido implementado en Java (jdk1.2) con base en la versión de Apriori disponible en ARMINER [3]. Las pruebas se han hecho sobre la base de datos *mushrooms* del repositorio de la UCI[13], con el fin de analizar

cómo el uso de soporte y confianza difusos permite la generación de reglas que en el caso crisp no se hubieran podido extraer.

Por ejemplo, una de las pruebas realizadas cuando se utilizó Apriori con soporte mínimo = 0.9 y confianza mínima = 0.9, no generó el conjunto de reglas que se presentan en la Figura 9. Es importante observar que el soporte de cada una de las reglas est muy cercano al límite (i.e. 0.897095). La Figura 10 muestra resultados similares con soporte mínimo = 0.5 y confianza mínima = 0.9.

Estas reglas se podrían generar bajo un modelo de aceptación difuso, como el que se describe en la Figura 8.

Antecedente(s)	Consecuente(s)	Soporte	Confianza
ring-number=one (97)	veil-color=white (94)	0.897095	0.97329056
veil-color=white (94)	ring-number=one (97)	0.897095	0.91973746
veil-color=white (94) ring-number=one (97)	veil-type=partial (90)	0.897095	1.0
veil-type=partial (90) ring-number=one (97)	veil-color=white (94)	0.897095	0.97329056
veil-type=partial(90) veil-color=white (94)	ring-number=one (97)	0.897095	0.91973746
veil-type=partial (90)	veil-color=white (94) ring-number=one (97)	0.897095	0.897095
veil-type=partial (90)	gill-attachment=free (36) ring-number=one (97)	0.89807975 0.89807975	0.89807975 0.89807975

Figure 9: Reglas No Generadas por Apriori (*minsup=0.9, minconf=0.9*)

Antecedente(s)	Consecuente(s)	Soporte	Confianza
stalk-color-above-ring=white (79) veil-type=partial (90)	ring-number=one (97)	0.4894141	0.890681
stalk-color-above-ring=white (79)	90 veil-type=partial (90) 97 ring-number=one (97)	0.4894141	0.890681
gill-spacing=close (38) stalk-surface-below-ring=smooth (71)	veil-color=white (94)	0.49138355	0.9541109
stalk-color-above-ring=white (79) veil-color=white (94) ring-number=one (97)	veil-type=partial (90)	0.4894141	1.0
gill-spacing=close (38) gill-size=broad (41)	veil-color=white (94) ring-number=one (97)	0.49433777	0.88147503
gill-attachment=free (36) stalk-color-above-ring=white (79)	ring-number=one (97)	0.4894141	0.890681

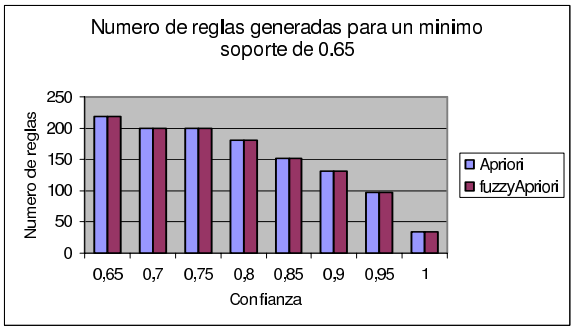
Figure 10: Reglas No Generadas por Apriori (*minsup=0.5, minconf=0.9*)

El comportamiento de las reglas de asociación generadas por Apriori y F-Apriori cuando se fusifican el soporte, la confianza o ambos se ilustran en las Figuras 11, 12 y 13.

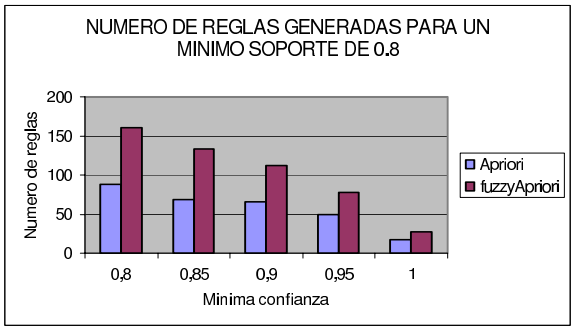
Como se observa en las figuras en las que el soporte es difuso, la cantidad de reglas generadas crece al ampliar el umbral en la variable de salida, es decir, al aumentar la pendiente de la función de pertenencia.

Para lograr un conjunto de reglas ms significativas (refinamiento de la salida) se deben incluir más conjuntos difusos que correspondan a "bordes lingüísticos". Al analizar el conjunto de reglas generadas se detectó que el modelo es más sensible a las variaciones del soporte que a las variaciones de la confianza.

En una etapa de post-procesamiento, el conjunto de reglas de asociación puede ser refinado por el usuario con base en el valor obtenido de la variable de aceptación.

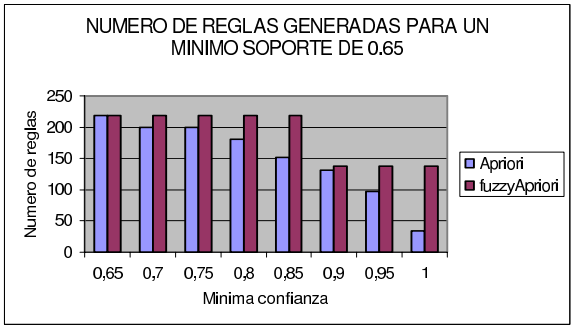


MinSup=0.65

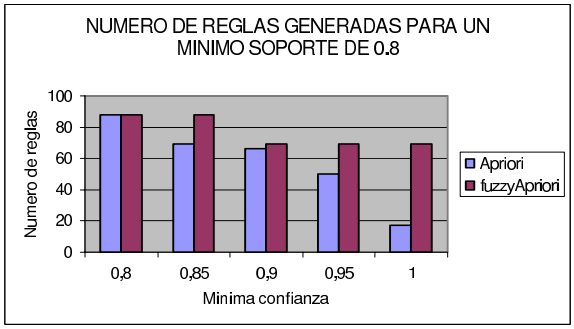


MinSup=0.8

Figure 11: Reglas Generadas con Soporte Difuso

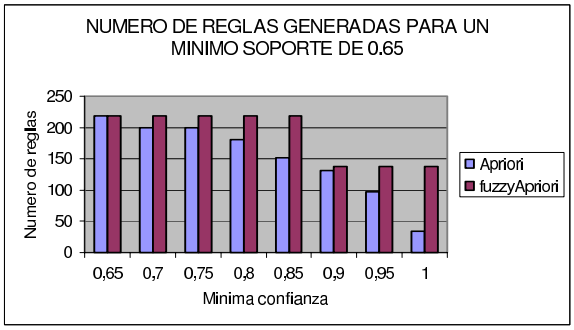


MinSup=0.65

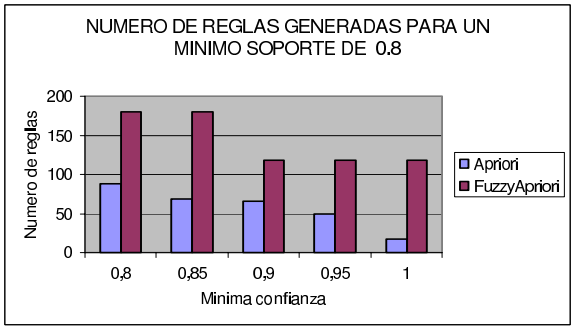


MinSup=0.8

Figure 12: Reglas Generadas con Confianza Difusa



MinSup=0.65



MinSup=0.8

Figure 13: Reglas Generadas con Soporte y Confianza Difusos

5 Conclusiones

En este artículo se ha propuesto una extensión al algoritmo Apriori, en la cual los parámetros de soporte y confianza se tratan como variables difusas.

El algoritmo propuesto F-Apriori, calcula reglas de asociación que no dependen ahora de los parámetros de soporte y confianza crisp, definidos por el usuario, sino que tiene en cuenta un modelo de aceptación tanto de conjuntos de items frecuentes como de reglas de asociación.

El sistema de control difuso directo FUZZYC permite simplificar las pruebas y el afinamiento -*tuning*- del modelo para la generación de reglas de asociación. El uso de un sistema de reglas de inferencia que está en armonía con el modo de razonamiento humano, hace que las aplicaciones sean entendibles y de fácil modificación. De esta forma, la definición de las restricciones de soporte y confianza mínimos se torna más intuitiva para el usuario al utilizar el modelo propuesto.

El algoritmo F-Apriori se está utilizando actualmente en la etapa de pre-procesamiento del proceso de descubrimiento de conocimiento en bases de datos como estrategia de limpieza de datos.

El funcionamiento de F-Apriori podría aproximarse con Apriori, si se considerara Apriori con soporte iterativo. Bajo este enfoque el algoritmo Apriori se ejecutaría variando ligeramente el valor del soporte y la confianza mínimos. Pero esta solución es ms costosa computacionalmente que F-Apriori al ser iterativa y no incremental.

Otra opción se presenta, redondeando los valores de soporte y confianza de los conjuntos de items. Sin embargo, es menos intuitiva para el usuario y no garantiza que se alcance la misma solución de F-Apriori.

Un ventaja que ofrece F-Apriori es la posibilidad de establecer un límite superior para el soporte y la confianza, definiendo un intervalo que puede utilizarse para combinar múltiples valores de soporte y confianza.

References

[1] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast Discovery of Association Rules, AAAI/MIT Press. Chapter 12, pp. 307-328, 1996.

[2] Andrés Dorado. FUZZYC: Algoritmo Basado en Conjuntos Borrosos. Conferencia Latinoamericana de Informática CLEI2001. Mérida, Venezuela. Sep. 2001.

[3] Cristofor Dana, Cristofor Laurentiu, Karatihy Abdelmajid, Xiaoyong Kuang, Long-Tsong Li; Arminer. 2000.

[4] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic item set counting and implication rules for market basket data. Proc. ACM SIGMOD Conference, pp. 255-264, 1997.

[5] A. Jaramillo-Botero, Inteligencia Computacional: Introducción y Aplicaciones en Visión y Control Senso-motriz, Colombia: Ciencia y Tecnología, vol. 12, no. 4, pp. 19-22, Oct.-Dic. 1994.

[6] Kuok C.M., Fu A, Wong M.H. Mining Fuzzy Association Rules in large Databases with quantitative attributes. ACM SIGMOD Record, 27(1), pp.41-46, 1998

[7] Gyenesei A. A Fuzzy Approach for mining Quantitative Association Rules. TUCS Techical Reports N335, University of Turku, Finland,March 2000.

[8] Fu A., Wong M.H., Sze S. C.,Wong W.C., Wong W.L., Yu W.K. Finding Fuzzy Sets for the Mining Fuzzy Association Rules for Numerical Attributes. En Proceeding of the 1st Internation Symposium on Intelligent Data Ingeneering and Learning, IDEAL98. pp. 263-268. 1998.

[9] ParkJ.S., Chen M-S., Yu P.S. An effective hash-based algorithm for mining association rules. En Proc. ACM SIGMOD, pp.175-186, 1995.

[10] Srikant R., Agrawal R. Mining Quantitive Association Rules in large relational tables. En Proc. ACM SIGMOD, pp.1-12, 1996

[11] Srikant R., Agrawal R. Fast Algorithms for mining Association Rules. En Proc. 20th VLDB Conference, pp.487-499, 1994

[12] Jiawei Han, Jian Pei, and Yiwen Yin. Mining Frequent Patterns without Candidate Generation. ACM SIGMOD, 2000.

[13] Blake, C. Keogh, E. Merz, C. J. UCI Repository of Machine Learning Databases. University of California, Dept. of Information and Computer Science. Irvine, CA. 1998.