Universidad de las Ciencias Informáticas

Facultad 3

Título:

Implementación de un algoritmo Apriori-like-1 para el minado de reglas de asociación difusas.

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

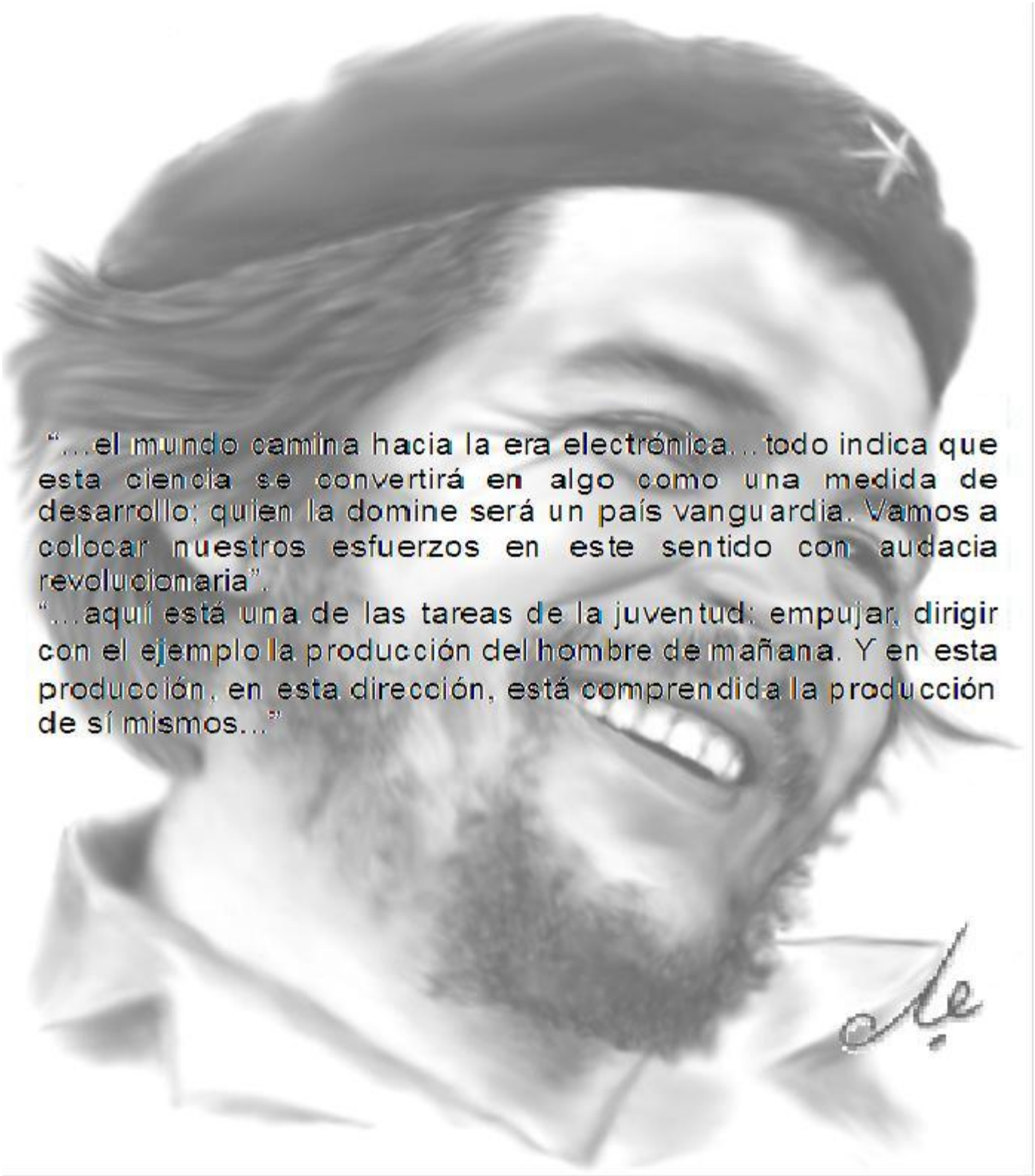
Autor: Liset Díaz Llerena

Tutores: Msc. Julio Cesar Diaz Vera

Ing. Andy Fernandez Garabote

La Habana, Cuba

Junio, 2013


"...el mundo camina hacia la era electrónica... todo indica que esta ciencia se convertirá en algo como una medida de desarrollo; quien la domine será un país vanguardia. Vamos a colocar nuestros esfuerzos en este sentido con audacia revolucionaria".
"...aquí está una de las tareas de la juventud: empujar, dirigir con el ejemplo la producción del hombre de mañana. Y en esta producción, en esta dirección, está comprendida la producción de sí mismos..."

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

_____	_____	_____
Liset Díaz Llerena	Msc. Julio Cesar Diaz Vera	Ing. Andy Fernandez Garabote

AGRADECIMIENTOS

A mis padres, por su apoyo, por darme todo el cariño de este mundo y todas sus fuerzas para que pudiera seguir adelante durante toda mi vida estudiantil.

A mi abuela querida Digna, por quererme tanto y siempre confiar plenamente en mí.

A mi tío Claro, por ser ese otro padre que tanto quiero y por ayudarme en mis estudios.

A mi hermano querido Michel, que ha sido un ejemplo a seguir, y gracias a él he logrado terminar mis estudios.

A mi prima del alma Diurmy y mi cuñada Yelena, a las cuales considero que son las hermanas que nunca tuve, por apoyarme en todo y darme tan buenos consejos.

A todos los miembros de mi familia, por darme ánimo en todo momento.

A los profesores que han contribuido con sus conocimientos y experiencia a mi preparación como estudiante.

A mis tutores, Julio Cesar y Andy Fernández, por ser tan atentos, por su ayuda y su empeño en la realización de este trabajo.

A todos mis compañeros de aula y amigos, en especial a Yadelis, Reiniel, Dailenis, Aylen, por estar siempre ahí cuando los necesito.

A todos los que de una forma u otra han tenido que ver con la realización de esta tesis.

A la Revolución y a Fidel.

DEDICATORIA

A mis padres, hermano y mi abuelita querida.

RESUMEN

El presente documento propone la implementación de un algoritmo para la extracción de reglas de asociación difusas. Dicho algoritmo está basado en la propuesta realizada en (Garabote, 2012) bajo el principio de “Clausura descendente del soporte de ítemsets o conjuntos de elementos”, el cual plantea que todo subconjunto de un ítemset frecuente es frecuente, mientras que cualquier supraconjunto de un ítemset no frecuente tampoco es frecuente (Medina Pagola, 2007) . Para el correcto funcionamiento del mismo se ajustó el mecanismo de conteo del soporte de un ítemset difuso bajo el principio propuesto en (Delgado, 2005). La presente solución fue validada utilizando la técnica de demostración pues esta se adapta correctamente a las características del trabajo siendo ideal para corroborar la validez del mismo pues no existen soluciones anteriores para este tipo de problemas.

PALABRAS CLAVES

Algoritmo, Apriori, ítemsets, minería de datos, reglas de asociación.

ÍNDICE

ÌNDICE DE TABLAS

ÌNDICE FIGURAS

INTRODUCCIÓN

La Minería de Datos (MD) consiste en extraer conocimiento interesante a partir de conjuntos de datos del mundo real y es el paso central del proceso de Extracción de Conocimiento a partir de Base de Datos (BD).

La extracción de reglas de asociación es uno de los campos de investigación, en MD, más activos de los últimos años. Uno de los principales problemas que ha sido objeto de investigación en el área radica en la gran cantidad de reglas que son generadas y en la dificultad de utilizar una gran parte de las mismas dentro del proceso de toma de decisiones ya sea porque son obvias, demasiado generales, demasiado específicas o porque no tienen interés para el usuario final.

La mayoría de las investigaciones en esta área están orientadas a disminuir la complejidad computacional del minado de reglas de asociación y a aumentar la calidad de las reglas extraídas. Comúnmente los esfuerzos realizados en la temática siguen tres direcciones diferentes:

- Definir mecanismos más eficientes para cubrir el espacio de búsqueda.
- Explotar estructuras de datos más eficientes.
- Utilizar conocimiento previo del dominio particular.

Normalmente los trabajos que pretenden obtener reglas de mayor calidad se centran en la etapa de post-procesamiento mientras que los que pretenden disminuir la complejidad computacional trabajan directamente sobre los algoritmos o en la etapa de pre-procesamiento.

El uso de técnicas de softcomputing en particular la utilización de lógica difusa es una opción válida con vista a la reducción de complejidad en los modelos obtenidos a partir de reglas de asociación debido a que acercan los mismos a la forma de razonar de un experto humano (Universidad de Granada, 2011).

Se define como **Problema** a tratar cómo contribuir a la reducción de complejidad de los modelos de reglas de asociación con el uso de lógica difusa.

El **Objeto de Estudio** estará enmarcado en el minado de reglas de asociación difusas, defendiéndose como **Campo de Acción** la implementación del algoritmo Apriori sobre modelos de datos difusos.
Objetivo General la implementación de un algoritmo Apriori-like para la extracción de reglas de

asociación difusas. Para cumplimentar este objetivo se definen los siguientes **Objetivos específicos:**

1. Establecer el marco conceptual de referencia.
2. Definir los requisitos especiales de los datos.
3. Ajustar el algoritmo Apriori para mejor rendimiento.
4. Definir el modelo de componentes.
5. Probar la validez del resultado.

De los anteriores objetivos específicos se derivan las **tareas** de investigación que se presentan a continuación:

1. Recopilación de la bibliografía referente al tema.
2. Selección de la bibliografía.
3. Análisis de la bibliografía.
4. Determinar qué tipos de reglas de asociación se ajustan al problema
5. Establecer el mecanismo de extracción de Itemsets frecuentes.
6. Establecer el mecanismo de generación de reglas.
7. Implementar las funciones asociadas al algoritmo.
8. Validar el resultado obtenido.
9. Presentar los resultados.

Se espera obtener como **resultado** de la investigación la Implementación de un algoritmo Apriori-like que permita minar reglas de asociación difusas.

En el **Capítulo 1** se desarrolla el marco conceptual, se presentan los principales conceptos de minería de datos, extracción de conocimiento y lógica difusa. Se enmarca el objetivo de las reglas de asociación. Definiendo estas últimas haciendo énfasis en las reglas difusas. Por último se presenta el algoritmo Apriori básico, detallando sus dos procesos principales a partir de los cuales se obtienen las reglas de asociación relevantes.

En el **Capítulo 2** a través de un ejemplo sobre el censo realizado en Los Ángeles en los años comprendidos entre 1970,1980 y 1990, se describe todo el proceso que conllevará a la propuesta de solución aclarando los principales inconvenientes que se pretenden resolver. Se proponen los ajustes al algoritmo de extracción de reglas de asociación con lo que se pretende mejorar la complejidad

computacional y la reducción de las reglas de asociación finales, con una mayor claridad en las mismas. Se presenta la solución definiendo las entradas correspondientes al algoritmo, los procesos básicos del mismo así como sus salidas.

En el **Capítulo 3**, se presentan los diferentes mecanismos utilizados para validar los resultados de una investigación. Entre ellos se encuentran los experimentos, modelos matemáticos, el razonamiento lógico y la demostración. Este último es el seleccionado para validar la solución propuesta en esta investigación por adaptarse a las características de la misma. Se describen los recursos computacionales utilizados para desarrollar la demostración. Se incluye además la descripción de todos los elementos que conforman el dominio de la demostración. Por último se presentan los resultados obtenidos y se realiza una detallada discusión de los mismos, demostrando la validez de la presente investigación.

Capítulo 1. Fundamentación teórica.

1. 1 Introducción

El creciente aumento del volumen y la variedad de la información que se encuentra informatizada en bases de datos digitales ha crecido gradualmente en las últimas décadas. Todos esos archivos contienen normalmente gran cantidad de datos que serían de utilidad si fuera posible aprovecharlos mediante procesos que arrojarían información útil. Las áreas de sistemas han venido trabajando para crear extractos de información de las bases de datos y almacenar estos datos en archivos, tratando de responder a las peticiones de los usuarios que necesiten obtener información que les ayude a tomar mejores decisiones.

Tradicionalmente el análisis de los datos que están contenidos en bases de datos se efectúa utilizando lenguajes generalistas de consultas. Algunos lenguajes como el estructurado de consultas (SQL por sus siglas en inglés) permiten generar información resumida en informes. Esta solución es muy poco flexible y sobre todo poco escalable ante grandes volúmenes de datos (Puente, 2010).

Otras herramientas utilizadas para analizar los datos son las estadísticas. Algunos paquetes estadísticos son capaces de inferir patrones (“dados una serie de hechos (datos) D, un lenguaje L, y una medida de certeza C, un patrón P es un enunciado en L que describe las relaciones (asociaciones) entre varios subconjuntos de D con una certeza dada mediante C, tal que P es más sencillo (en algún sentido) que la enumeración de todas las relaciones entre dichos subconjuntos” (Ruiz, 2010)) a partir de los datos. El problema radica en que generalmente estos no funcionan bien para las bases de datos actuales con millones de registros, además no se integran bien con los sistemas de información existentes (Delgado, 2005).

Producto a todos los problemas que existían surgió la necesidad de nuevas herramientas y técnicas para la extracción de conocimiento que realmente fuera útil desde la información disponible en las base de datos actuales.

1.2 Minería de Datos

El concepto de minería de datos y el de extracción de conocimiento en bases de datos (KDD por sus siglas en inglés) ambos están estrechamente relacionados pero no significan exactamente lo mismo. La primera definición de extracción de conocimiento y una de las más referenciadas es la propuesta en (Fayyad, 1996) donde se define como un proceso no trivial de identificación de patrones válidos, novedosos, potencialmente útiles, entendibles y comprensibles que se encuentran ocultos en los datos. Es importante destacar algunos vocablos utilizados por Fayyad: el término “no trivial” implica que el proceso no es superficial, los patrones que se obtengan no deben ser evidentes, deben tener la capacidad de producir algún efecto, novedosos y potencialmente útiles refiere que sean poco conocidos o totalmente nuevos y produzcan beneficios considerables, además destaca que deben ser factibles al entendimiento humano (Garabote, 2012).

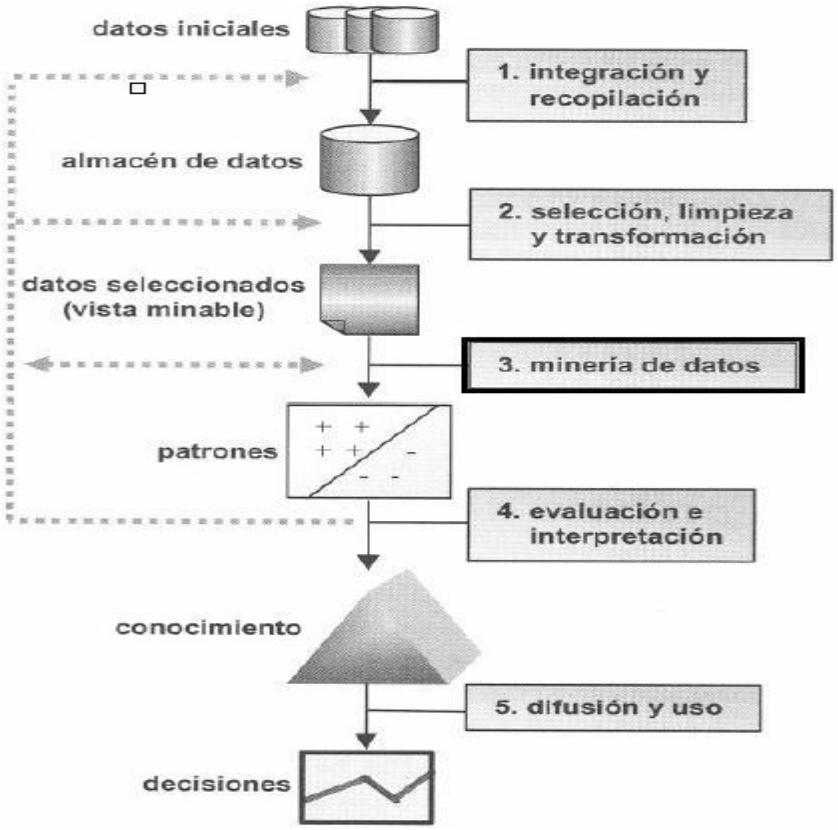


Figura 1. Fases del proceso de extracción de conocimiento en bases de datos (Fayyad, et al., 1996).

En la Figura 1 se muestran las fases del proceso de extracción de conocimientos, una de ellas es la minería de datos, esta se encarga de extraer los patrones ocultos en los datos a partir de la vista minable,

datos previamente procesados con el fin de eliminar los elementos no deseables que puedan existir en ellos. Esta es la fase de mayor importancia y de mayores dimensiones dentro de todo el proceso, razón por la que muchos autores usan los términos de manera indistinta. En esta investigación se enmarca la minería de datos como fase dentro del proceso de extracción de conocimiento, definiéndola como una familia de métodos computacionales que tienen como objetivo recolectar y analizar datos relacionados a un sistema de interés con el propósito de ganar un mejor entendimiento del mismo (Triantaphyllou, 2010) .

El proceso de extracción de conocimiento persigue dos objetivos fundamentales, la predicción y la descripción. El aprendizaje automático es la disciplina que se encarga de estudiar todo lo referente al primero. El segundo se ajusta a la definición de minería de datos propuesta en (Triantaphyllou, 2010) donde el propósito no es otro que describir un sistema de interés.

Asociadas a estas dos vertientes existen una serie de tareas que se encargan del uso de varias variables conocidas para intentar determinar los valores desconocidos de otras y de buscar patrones interpretables que expliquen la naturaleza de los datos, refiriéndose a la predicción y la descripción respectivamente. Algunas de ellas son:

- **Clasificación**: tiene como objetivo pronosticar el valor o la clase que puede tomar un atributo en función de los valores que toman otros atributos.
- **Regresión**: pretende pronosticar el valor numérico que tiene un determinado atributo.
- **Agrupamiento**: intenta crear grupos de individuos en función de sus similitudes de manera que los objetos de un grupo son muy similares entre sí y muy diferentes a los de otro grupo.
- **Reglas de asociación**: identifican relaciones no explícitas entre atributos categóricos (Garabote, 2012).

El hecho de que existan dos objetivos bien marcados dentro del proceso de KDD no quiere decir que estos no se relacionen. Para hablar de predicción es necesario contar primeramente con una descripción detallada del dominio sobre el que se actúa.

El termino Minería de Datos es una etapa dentro de un proceso mayor llamado extracción de conocimiento en bases de datos, ambos términos se usan de manera indistinta según algunos autores,

mientras que otros no lo creen así. Una definición tradicional de minería de datos es la siguiente: Un proceso no trivial de identificación válida, novedosa, potencialmente útil y entendible de patrones comprensibles que se encuentran ocultos en los datos (Fayyad, 1996).

Por lo tanto, dos son los retos de la minería de datos: por un lado, trabajar con grandes volúmenes de datos, procedentes mayoritariamente de sistemas de información, con los problemas que ello conlleva (ruido, datos ausentes, intratabilidad, volatilidad de los datos...), y por el otro usar técnicas adecuadas para analizar los mismos y extraer conocimiento novedoso y útil. En muchos casos la utilidad del conocimiento minado está íntimamente relacionada con la comprensibilidad del modelo inferido (Orallo Hernández, 2004).

La Minería de Datos denota el proceso de extracción de la información implícita previamente desconocida y potencialmente útil (tales como reglas, restricciones y regularidades) desde los datos en una Base de Datos. El término de minería ha sido fundamentalmente utilizado por los estadísticos, los analistas de datos y la comunidad de los administradores de información (Puente, 2010).

1.3 Reglas de asociación

Minería de Reglas de asociación es el proceso de encontrar asociaciones, correlaciones o estructuras casuales entre conjuntos de ítems u objetos en una base de datos de transacciones, relacional o un almacén de datos (Castro, 2008).

Las reglas de asociación son una importante herramienta de la minería de datos que ha recibido mucha atención y estudio desde su primera aparición en los trabajos de Agrawal en (Agrawal, 1993) donde plantea que siendo $I = \{ i_1, \ldots, i_n \}$ un conjunto finito de ítems, D una base de datos que contiene las transacciones o subconjuntos propios de ítems no vacíos que pertenecen a I, asociándosele además un identificador único. Se define una regla de asociación como una implicación de la forma $X \rightarrow Y$, donde son conjuntos de elementos o ítems (son atributos, variables o campos de una Base de Datos (Ruiz, 2010)), llamados ítemsets, nombrando a X como antecedente, Y como consecuente y cumpliendo que $X \cap Y = \emptyset$.

Es importante aclarar que un ítemset o conjunto de ítems es aquel cuyos elementos son atributos, variables o campos de cualquier base de datos. Su tamaño se determina de acuerdo a la cantidad de

ítems que contiene, definiéndose como k-ítemset, donde k es igual al número de ítems (Medina Pagola, 2007).

La siguiente expresión es un ejemplo de regla de asociación $queso, dulce \rightarrow \{agua\}$, el antecedente de la regla sería el conjunto de ítems formado por $queso, dulce$, el consecuente sería el formado por $\{agua\}$ y la intersección entre ambos sería nula.

El descubrimiento de reglas de asociación, en grandes volúmenes de datos, suele generar un gran número de ellas y se corre el riesgo de que no todas sean realmente relevantes. Para evaluar el cumplimiento o grado de verdad de una regla de asociación es habitual utilizar las llamadas medidas de interés. Las dos medidas más utilizadas, aunque no por ello las que tienen mejores propiedades, son el soporte y la confianza. (Orallo Hernández, 2004)

Soporte y Confianza:

Se define como soporte de un ítemset la probabilidad de ocurrencia del mismo en una transacción t de la base de datos D. Calculándose de la forma siguiente:

$$sop(X) = \frac{|\{t \in D \mid X \dot{\subseteq} t\}|}{|D|}$$

En este caso se calcula el soporte del ítemset X contando las transacciones en que este aparece y dividiéndolo entre el total de transacciones de la base de datos D.

El soporte de una regla de asociación se calcula de manera similar al de un ítemset como se muestra a continuación:

$$Sop\ X\ = \frac{|\ t \in D\ X \cup Y\ \dot{\subseteq} t\}|}{|D|}$$

Con la particularidad de que se cuenta la cantidad de apariciones conjuntas entre el antecedente X y el consecuente Y en las transacciones t de la base de datos D, luego se divide entre el total de estas últimas.

La confianza está dada de acuerdo al porcentaje de transacciones t en D que contienen $X \cup Y$ dividido entre todas aquellas que contienen al antecedente, como se muestra a continuación:

$$Conf\ X\ = \frac{|\{t \in D | X \cup Y \dot{\subseteq} t\}|}{|\{t \in D | X \dot{\subseteq} t\}|}$$

Las reglas de asociación cuyo soporte y confianza sea mayor o igual que los definidos por los interesados o especialistas en los datos analizados, se les denomina reglas fuertes (Jimenez Ruiz, 2010).

La interpretación de una regla de asociación teniendo en cuenta el soporte y confianza podría ser:

- El 75% de los clientes de una cafetería que compran dulces, compran también refresco; un 10% de todas las ventas contienen estos elementos.

En este caso particular el 75% se refiere a la confianza de la regla y el 10% a su soporte.

1.4 Lógica difusa

Una de las disciplinas matemáticas con mayor número de seguidores actualmente es la llamada lógica difusa o borrosa, que es la lógica que utiliza expresiones que no son ni totalmente ciertas ni completamente falsas, es decir, es la lógica aplicada a conceptos que pueden tomar un valor cualquiera de veracidad dentro de un conjunto de valores que oscilan entre dos extremos, la verdad absoluta y la falsedad total (Delgado, 2008).

La lógica difusa es un tipo de lógica que reconoce más que simples valores verdaderos y falsos. Con lógica difusa, las proposiciones pueden ser representadas con grados de veracidad o falsedad. Por ejemplo, la sentencia "hoy es un día soleado", puede ser 100% verdad si no hay nubes, 80% verdad si hay pocas nubes, 50% verdad si existe neblina y 0% si llueve todo el día.

La Lógica Difusa ha sido probada para ser particularmente útil en sistemas expertos y otras aplicaciones de inteligencia artificial. Es también utilizada en algunos correctores de voz para sugerir una lista de probables palabras a reemplazar en una mal dicha. La Lógica Difusa, que hoy en día se encuentra en constante evolución, nació en los años 60 como la lógica del razonamiento aproximado, y en ese sentido podía considerarse una extensión de la Lógica Multivaluada. La Lógica Difusa actualmente está relacionada y fundamentada en la teoría de los Conjuntos Difusos. Según esta teoría, el grado de pertenencia de un elemento a un conjunto va a venir determinado por una función de pertenencia, que puede tomar todos los valores reales comprendidos en el intervalo [0,1] (Yuliana Corzo, 2009).

En 1965 L.A. Zadeh introduce una lógica infinito valorada caracterizando el concepto de conjunto difuso y por extensión la lógica difusa (Diaz, 2010). La idea es hacer que el rango de valores de pertenencia de un elemento a un conjunto pueda variar en el intervalo [0,1] en lugar de limitarse a uno de los valores del par {0,1} (o lo que es lo mismo Falso, Verdadero).

A partir de la Teoría de Conjuntos Difusos (un conjunto difuso A se caracteriza por una función de pertenencia: $\mu A: U \rightarrow 0,1$ que asocia a cada elemento X de U un número $\mu A(X)$ del intervalo [0,1], que representa el grado de pertenencia de X al conjunto difuso A (2011)) Zadeh introduce la Lógica Difusa como una extensión de las lógicas polivaloradas. Lo que justifica el desarrollo de la Lógica difusa es la necesidad de un marco conceptual donde tratar la incertidumbre no probabilística y la imprecisión léxica.

Para un conjunto difuso $A = \frac{x, u_A\ x}{x} \in X$, se tiene que el elemento x pertenece al conjunto A con un grado de pertenencia $u_A\ (x)$ que puede variar entre 0 y 1. Por lo tanto, una variable puede ser caracterizada por diferentes valores lingüísticos, cada uno de los cuales representa un conjunto difuso (Diaz, 2010).

1.4.1 Características de la Lógica Difusa según Zadeh

- En Lógica Difusa (LD) todo es cuestión de grado
- El Razonamiento Exacto es un caso límite del Razonamiento Aproximado
- En LD el conocimiento se interpreta como una colección de restricciones elásticas (difusas) sobre un conjunto de variables
- En LD la inferencia puede verse como la propagación de un conjunto de restricciones elásticas.
- Sistema Difuso (SD): resultado de la “fuzzificación” de un sistema convencional
- Los Sistemas Difusos operan con conjuntos difusos en lugar de números
- En esencia la representación de la información en Sistemas Difusos imita el mecanismo de Razonamiento Aproximado que realiza la mente humana

1.4.2 Reglas de asociación difusas

Los primeros estudios relacionados con las reglas de asociación difusas propuestos en (Agrawal, 1993) se centraban en el uso de etiquetas lingüísticas para generar reglas de asociación cuantitativas. Las reglas cuantitativas se basan en dividir el dominio del atributo en intervalos para luego descubrir reglas cuyos ítems son los pares <atributo; intervalo> en lugar de <atributo; valor>. Al realizar este procedimiento en un número fijo de intervalos aparece el llamado "problema de la frontera", que se traduce en la posibilidad de excluir intervalos de interés por estar muy cerca de los extremos. Ante esta situación se introduce el uso de conjuntos difusos en lugar de intervalos, de manera que cierto elemento está presente en un conjunto de este tipo con cierto grado de pertenencia que oscila entre [0,1]. Los conjuntos difusos permiten el uso de variables lingüísticas que facilitan la interpretación de las reglas descubiertas (Delgado, 2008).

Según (Kuok, 1998) un ítemset difuso es un conjunto de ítems de la forma $A_i; a_i \cup \ldots \cup A_i; a_i$ que se denota como {X, A} donde X contiene todos los elementos o ítems y A los conjuntos difusos asociados a cada atributo en X.

Una regla de asociación difusa de acuerdo a (Gyenesei, 2001) es una expresión de la forma:

$$si\ X = \{x_1, \ldots, x_p\}\ es\ A = \{a_1, \ldots, a_p\}\ entonces\ Y = \{y_1, \ldots, y_p\}\ es\ B = \{b_1, \ldots, b_p\}$$

$$a_i \in \{conjuntos\ difusos\ asociados\ a\ x_i\}$$
$$b_i \in \{conjuntos\ difusos\ asociados\ a\ y_i\}$$

Siendo $D = \{t_i, \ldots, t\ \}$ la base de datos que contiene todas las transacciones con atributos y los conjuntos difusos asociados a esos atributos. Además $\{X, Y \in I\}$ son conjuntos disjuntos de ítems y {A, B} contienen los conjuntos difusos correspondientes a los elementos de X e Y respectivamente. A la primera parte de la regla {X es A} se le llama antecedente y la segunda parte {Y es B} es el consecuente. Para abreviar la forma de escribir la regla desde ahora se denotará como: $X, A \rightarrow \{Y, B\}$

Para calcular el soporte de los ítemsets difusos, según el enfoque que se propone en (Gyenesei, 2001) y (Kuok, 1998), se utilizan las fórmulas siguientes:

$$fsop(X,A) = \frac{\sum_{t_i \in D} \prod_{x_i \in X} \mu_{a_i}^{i}(x_i)}{|D|}$$

Donde |D| es el número de transacciones de la base de datos, $\mu_{ai}^{i}(X)$ es el grado de pertenencia del atributo $x_j \in X$ en la transacción i-ésima al conjunto difuso $a_j \in A$.

Partiendo del resultado del soporte de un ítemset se puede calcular el soporte y la confianza de las reglas de asociación difusas a partir de las siguientes formulas:

$$fSop(\{X,A\} \rightarrow \{Y,B\}) = fsop(Z,C)$$

$$fConf(\{X,A\} \rightarrow \{Y,B\}) = \frac{fsop(< Z,C >)}{fsop(< X,A >)}$$

$$Z = [X \cup Y] \, y \, C = [A \cup B].$$

Otro concepto de reglas de asociación difusas es la propuesta en (Delgado, 2003), para el término transacción difusa plantea lo siguiente:

$$\tilde{T} \subseteq I \text{ donde, } I = \{i_1, \ldots, i_m\}$$

una transacción difusa es un subconjunto difuso no vacío representa un conjunto finito de ítems.

La evaluación de las reglas de asociación difusas puede realizarse a partir de la generalización del soporte y la confianza utilizando sentencias cuantificadas (Delgado, 2003), (Delgado, 2005). Una sentencia cuantificada es una expresión de la forma “Q de los F son G” donde F y G son dos subconjuntos difusos de un conjunto finito X y Q es un cuantificador difuso relativo. Los cuantificadores relativos son etiquetas lingüísticas que representan porcentajes difusos y que pueden expresarse como conjuntos difusos en [0,1]. De manera particular la propuesta de (Delgado, 2003) plantea el uso de una familia de cuantificadores relativos denominada cuantificadores coherentes (Garabote, 2012).

El soporte de una regla de asociación difusa en el conjunto de transacciones difusas $X \rightarrow Y$ en el conjunto de transacciones difusas D es $sop(X \cup Y)$, lo que equivale a la evaluación de la sentencia:

$$Q \text{ de los } D \text{ son } \tilde{\Gamma}_{X \cup Y} = Q \text{ de los } D \text{ son } (\tilde{\Gamma}_X \cap \tilde{\Gamma}_Y)$$

La confianza de la regla de asociación difusa $X \rightarrow Y$ en el conjunto de transacciones difusas es la evaluación de la sentencia cuantificada:

$$Q\ de\ los\ \tilde{\Gamma}_X\ son\ \tilde{\Gamma}_Y$$

El soporte y la confianza dependerán del método de evaluación que sea escogido. La propuesta de (Delgado, 2003) es utilizar el método GD (Delgado, 2000) que ha demostrado tener buenas propiedades y mejor desempeño que otros. La evaluación de la sentencia "Q de los F son G" con el método GD se define como:

$$GD_Q\left(\frac{G}{F}\right) = \sum_{\alpha_i \in \Delta(G/F)} (\alpha_i - \alpha_{i+1}) Q\left(\frac{|(G \cap F)_{\alpha_i}|}{|F_{\alpha_i}|}\right)$$

Donde $\Delta\ \frac{G}{F} = \wedge\ G \cap F\ \cup \wedge(F)$, con para toda $a_i > a_{i+1}$ para toda $i \in \{1, \dots, p\}$. El conjunto F se asume está normalizado de lo contrario debe normalizarse y el factor de normalización aplicarse a $G \cap F$.

Este enfoque establece medidas de soporte y confianza que dependen del método de evaluación y el cuantificador elegido. En (Delgado, 2003) y (Delgado, 2005) se justifica el uso del método GD y el cuantificador Q_M que cumple: $Q_M\ \ X = X$

Para el minado de reglas de asociación se han propuesto una serie de algoritmos que pretenden descubrir la mayor cantidad de reglas relevantes utilizando la menor cantidad de recursos computacionales, entre ellos se encuentra el SEMT, AIS, Partition, Eclat, (Garabote, 2012). La mayoría de ellos surgen a partir del propuesto en (Agrawal, 1993) bajo el nombre: Apriori.

1.5 Algoritmo Apriori

Uno de los algoritmos más propagados en el campo de la minería de reglas de asociación es el Apriori propuesto en (Agrawal, 1994), este se basa en el principio de "Clausura descendente del soporte de ítemsets", el cual plantea que el soporte de cualquier subconjunto de un ítemset es mayor o igual que el soporte de ese ítemset. Además, todo subconjunto de un ítemset frecuente es frecuente, mientras que cualquier supraconjunto de un ítemset no frecuente tampoco es frecuente (Medina Pagola, 2007). Un ítemset es frecuente cuando cumple con un soporte mínimo definido con anterioridad. Es importante

destacar que Apriori asume que los ítems de cualquier transacción están ordenados lexicográficamente, para una correcta generación de los k-ítemsets candidatos (Motoda, 2009).

Este algoritmo está compuesto por 3 secciones para la obtención de los *k*-ítemset frecuentes:

- **Generación de candidatos**: se generan los candidatos de nivel k (k-1 ítemset) haciendo uso de la información referida a los conjuntos frecuentes de la iteración anterior (k-1 ítemset frecuentes).
- **Poda**: se garantiza que no pase al conteo de soporte los ítemset que se conoce a priori que no van a ser frecuentes, eliminando a todos aquellos candidatos que no cumplan con que todos sus subconjuntos sean también frecuentes.
- **Conteo de soporte**: se cuenta la cantidad de apariciones de los k-ítemsets candidatos en la base de datos y se determina, según el soporte mínimo establecido por el usuario, el conjunto de los k-ítemsets frecuentes.

Estas secciones se repiten hasta que ya no haya más conjuntos frecuentes. Los 1-ítemsets frecuentes son obtenidos por el sistema al inicio del proceso (Mesa Rodriguez, 2009). Como se muestra a continuación:

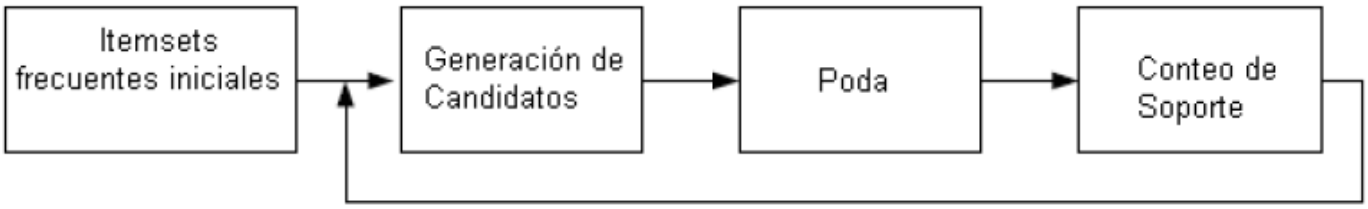


Figura 2. Estructura funcional de Apriori para obtener k-ítemsets frecuentes.

Apriori Básico

Entrada: **I,D, minsup, minconf**

Salida: **Conjunto de reglas de asociación con soporte y confianza ≥ que minsup y minconf.**

Algoritmo:

1. **Generar todos los ítemsets frecuentes con soporte ≥ minsup**
2. **Dado un ítemsets frecuente $X = x_i \dots x_k \ con\ k \geq 2$ generar todas las reglas de la forma $X|\{x_j\} \rightarrow \{x_j\}$, siendo el soporte de dicha regla el soporte de X y la confianza, el cociente entre el soporte de este y el soporte de $X|\{x_j\}$. (Jimenez Ruiz, 2010)**

1.6 Conclusiones parciales

En el presente capítulo se da una breve descripción sobre los conceptos de minería de datos donde el objetivo de este no es más que convertir los datos en conocimiento, se explica que es lógica difusa donde esta ha demostrado ser una excelente alternativa para sistemas de control, ya que imita a la lógica de control humana, también se aborda sobre las reglas de asociación y los tipo de reglas que existen entre las cuales se encuentra la de asociación difusa que es una de las cosas más importantes en este presente trabajo y se explica que es un algoritmo a priori y como funciona básicamente.

Capítulo 2. Análisis y Diseño del Algoritmo.

2.1 Introducción

El censo es la lista oficial de los habitantes de un país, una provincia o una ciudad, donde figuran sus datos personales, sus propiedades o bienes y otras informaciones. El mismo se realiza cada 10 años, en ocasiones se detectan errores en los datos recogidos lo que puede afectar los resultados finales.

El proyecto IPUMS tiene una gran colección de datos del censo federal, que ha estandarizado los sistemas de codificación para hacer comparaciones a través del tiempo (IPUMS, 1999). En este trabajo se ha escogido el ejemplo del censo Los Ángeles - área de Long Beach para los años 1970, 1980 y 1990. En el mismo los datos no ponderados son una muestra de 1 en 100 de las respuestas dadas. La familia y los registros individuales se acoplan en una sola tabla. En este se utilizaron todas las variables disponibles para las 3 décadas en cuestión. A continuación se muestran algunos datos, de los mismos se puede obtener patrones ocultos que pueden ser útiles al utilizar la minería de reglas de asociación.

Lugar Nacimiento	Estado Civil	Edad	Raza	Sexo
Colorado	Separado	30	Blanco	Masculino
Canadá	Viudo	50	Blanco	Femenino
Arizona	Soltero	9	Amarillo	Masculino
Colorado	Separado	30	Negro	Femenino
California	Soltero	15	Blanco	Masculino
Canadá	Divorciado	49	Negro	Masculino
Arizona	Soltero	19	Amarillo	Masculino
Canadá	Viudo	50	Blanco	Femenino
Colorado	Separado	35	Negro	Femenino
Canadá	Viudo	48	Blanco	Femenino

Tabla 1. Muestra de datos asociados al censo de los Ángeles.

En la tabla anterior se muestran algunas de las variables registradas en el censo antes mencionado. El lugar de nacimiento se refiere al lugar de origen de una persona determinada. La columna raza define el color de la piel de dicha persona. En el presente ejemplo se definen tres tipos de razas solamente: blanca, negra y amarilla. La edad se recoge en la columna edad. El estado civil es la columna encargada de archivar si una persona esta soltera, casada, separada, divorciada o viuda; en el momento de realización del censo. Por último se muestra el sexo de las personas encuestadas en la columna correspondiente.

Los algoritmos utilizados para el minado de reglas de asociación, específicamente el algoritmo Apriori básico al cual se hace referencia en el capítulo anterior, están diseñados para el caso conocido como "cesta de compras", es decir, toma como instancia de entrada, entre otras, una serie de transacciones donde cada una de ellas constituye una compra realizada por un cliente donde aparecen todos los productos que fueron adquiridos por él. Cada una de estas transacciones además de contar un identificador guarda uno en aquellos productos adquiridos por el cliente y cero en los que no, de la forma siguiente:

Id	Pan	Jamón	Queso	Refresco	Mantequilla
1	0	1	0	0	1
2	1	0	1	1	0
3	0	0	0	0	1
4	1	0	1	0	1

Tabla 2. Transacciones correspondientes a la compra.

Utilizando este mecanismo, en la muestra anterior (Tabla 1) donde cada uno de los atributos presenta valores diferentes a 0 y 1, habría que modificar la forma en que están escritos, transformando los atributos en la sucesión de cada uno de sus distintos valores ajustándose a la entrada del algoritmo en cuestión, de la siguiente forma:

Lugar de Nacimiento	California	Canadá	Arizona	Colorado

Tabla 3. Transformación del atributo Lugar de Nacimiento.

De manera que en vez de utilizar un sólo campo para el lugar de nacimiento se utilizarían 4, procediendo de la misma forma con los demás atributos. Una vez ajustados todos los atributos que contienen la declaración del censo a la entrada utilizada por Apriori, se procede a obtener los conjuntos de ítemsets frecuentes.

Para conocer si un ítemset es frecuente es necesario calcular el soporte asociado al mismo. Para esto, Apriori propone recorrer cada una de las transacciones contando aquellas en las que aparezca el ítemset, dividiendo este resultado entre el total de transacciones. A partir del resultado de este soporte se define si el ítemset en cuestión es frecuente o no. Esta tarea se repite mientras se puedan crear nuevos ítemsets candidatos. Es lógico pensar que el costo computacional de esta operación es bastante alto cuando se procesan miles de ítems (Garabote, 2012).

Una vez obtenidos los ítemsets frecuentes se procede a determinar las reglas de asociación para cada uno de ellos. Al obtener dichas reglas, se procede a descartar aquellas que no cumplan con los umbrales de soporte y confianza definidos por el usuario. En la muestra anterior (Tabla 1), al aplicar el algoritmo, se obtienen reglas como:

- El 100% de las mujeres blancas viudas de origen Canadá se clasifica de manera incorrecta, con soporte = 30%.
- El 100% de los hombres solteros de origen Arizona son de raza amarilla, con soporte = 20%.
- El 77% de los hombres separados de origen Colorado tienen 30 años, con soporte = 30%.

La tabla 1 cuenta con solo 10 filas puesto que es una muestra tomada del censo pero a la hora de manejar el 100% de los datos referentes al mismo el trabajo se hace mucho más engorroso. El algoritmo *Apriori-Like* propuesto en (Garabote, 2012) obtiene un número de reglas de asociación muy difícil de manejar por los especialistas en estos temas, las salidas de dicho algoritmo supera las dos mil reglas de asociación. Un mecanismo con el cual pudiese reducirse esta cantidad es la lógica difusa. Partiendo del principio de que la cantidad de reglas generadas por *Apriori-Like* depende directamente de la cantidad de ítems y transacciones se puede afirmar que al reducir alguna de estas entradas el número de salidas también se reduce. En el ejemplo anterior, el atributo edad se comporta de forma difusa, es decir, se divide en tres etiquetas difusas: niño (1-10), joven (12-20), adulto (20-50). Al realizar las transformaciones

mencionadas en (Tabla 3) se reduce considerablemente la cantidad de campos, como se muestra a continuación:

Edad	30	50	9	15	49	35	19	48

Tabla 4. Atributo edad transformado sin aplicar lógica difusa.

Edad	Niño	Joven	Adulto

Tabla 5. Atributo edad transformado aplicando lógica difusa.

Como se puede apreciar en (Tabla 4) y (Tabla 5) se reduce de 8 nuevos ítems a solo 3 para un 66,5% solo para el ejemplo. En la práctica, donde las personas pueden tener edades entre 0 y 100 años el porciento de reducción es mucho mayor. La complejidad de Apriori está dada por $O_{n} = m * n^2$, donde m es el total de transacciones y n la cantidad total de ítems, por lo que a medida que estas aumentan, el costo computacional asociado a su procesamiento también aumenta (Garabote, 2012) . En este caso se reduce la cantidad de ítems por lo que se puede afirmar que la complejidad computacional también es mejorada al aplicar técnicas de lógica difusa en el minado de reglas de asociación.

2.2 Propuesta de Solución

Partiendo de la situación planteada anteriormente se propone *Apriori-like-1*, el cual es un ajuste al algoritmo *Apriori-Like* presentado en (Garabote, 2012), con el que se pretende añadir la capacidad de obtener reglas de asociación a partir de datos difusos y con esto minimizar la cantidad de reglas generadas.

Al estudiar el funcionamiento de *Apriori–Like* (Garabote, 2012) se concluyó que uno de los principales problemas que impiden minar datos difusos está localizado en el mecanismo de conteo del soporte de los ítemsets. Este se basa en el conteo probabilístico convencional definido en (Agrawal, 1994) que plantea que al encontrar un Itemset en la transacción aumenta en uno el contador de las transacciones que contienen al Itemset que se le desea calcular el soporte y luego se divide este contador entre la cantidad de transacciones. Para el caso de los datos difusos este mecanismo no es válido, puesto que este tipo de datos tienen la particularidad de no establecer veracidad al 100%. Por esta razón cada ítem contiene

además del dato concreto un valor de veracidad que oscila en el intervalo de $[0-1]$. En estos casos se han propuesto una serie de soluciones (M.Sc., 2009), (Delgado, 2005). Esta última es una de las más referenciadas definiendo el cálculo de los Itemset difusos a partir de la fórmula GD_q que se muestra a continuación:

$$GD_Q\left(\frac{G}{F}\right) = \sum_{\alpha_i \in \Delta(G/F)} (\alpha_i - \alpha_{i+1}) Q\left(\frac{|(G \cap F)_{\alpha_i}|}{|F_{\alpha_i}|}\right)$$

Para un mejor entendimiento se propone el siguiente ejemplo donde se evidencia como se realiza el mecanismo de conteo del soporte basados en la propuesta de (Delgado, 2005). La siguiente tabla muestra 4 columnas y 6 filas donde cada una de las filas contiene los valores de veracidad asociados a su respectiva columna.

	Sexo	**Edad-niño**	**Edad-joven**	**Raza-blanca**
T1	0	0.8	0.9	0.7
T2	1	1	1	0.5
T3	0	0.5	0.2	0
T4	1	0	0.3	1
T5	1	0.5	0.5	0.2
T6	0	0.6	0.6	0.1

Tabla 6.Transacciones de atributos.

Evaluando los datos de la tabla en la ecuación propuesta anteriormente para realizar el mecanismo de conteo de soporte sobre los diferentes ítemsets se obtienen los siguientes resultados:

Tomando la columna de la Raza-blanca como punto de partida podemos determinar los $\alpha_i = [1;\ 0{,}7;\ 0{,}5;\ 0{,}2;\ 0{,}1]$, una vez obtenidos estos valores se procede a sustituirlos en la ecuación:

$\alpha_i = 1$ $G \cap F\ \alpha_i = 1$ *(1-0.7)*(1/6)=0.048*

$\alpha_i + 1 = 0.7$ $F_{\alpha i=6}$

$\alpha_i = 0.7$ $G \cap F\ \alpha_i = 1$ *(0.7-0.5)*(1/6)=0.032*

$\alpha_i + 1 = 0.5$ $F_{ai=6}$

$\alpha_i = 0.5$ $G \cap F\ \alpha_i = 1$ *(0.5-0.2)*(1/6)=0.048*

$\alpha_i + 1 = 0.2$ $F_{ai=6}$

$\alpha_i = 0.2$ $G \cap F\ \alpha_i = 1$ *(0.2-0.1)*(1/6)=0.016*

$\alpha_i + 1 = 0.1$ $F_{ai=6}$

$\alpha_i = 0.1$ $G \cap F\ \alpha_i = 1$ *(0.1-0)*(1/6)= 0.016*

$\alpha_i + 1 = 0$ $F_{ai=6}$ $GD_q = 0.16$

En el caso anterior se ejemplificó el cálculo del soporte para un Itemset de tamaño 1. En el caso de los ítemsets de mayor tamaño el procedimiento es el mismo. A continuación se evaluarán ítemsets de tamaño 2 y 3 respectivamente en aras de demostrar la afirmación anterior. En el primer caso se toman las columnas Edad-niño y Edad-joven:

$\alpha_i = 1$ $G \cap F\ \alpha_i = 2$ *(1-0.9)*(2/6)=0.03*

$\alpha_i + 1 = 0.9$ $F_{ai=6}$

$\alpha_i = 0.9$ $G \cap F\ \alpha_i = 1$ *(0.9-0.8)*(1/6)=0.016*

$\alpha_i + 1 = 0.8$ $F_{ai=6}$

$\alpha_i = 0.8$ $G \cap F\ \alpha_i = 1$ *(0.8-0.6)*(1/6)= 0.032*

$\alpha_i + 1 = 0.6$ $F_{ai=6}$

$\alpha_i = 0.6$ $G \cap F\ \alpha_i = 2$ *(0.6-0.5)*(2/6)= 0.03*

$\alpha_i + 1 = 0.5$ $F_{ai=6}$

$\alpha_i = 0.5$ $G \cap F\ \alpha_i = 3$ *(0.5-0.3)*(3/6)= 0.1*

$\alpha_i + 1 = 0.3$ $F_{ai=6}$

$\alpha_i = 0.3$ $G \cap F\ \alpha_i = 1$ *(0.3-0.2)*(1/6)= 0.016*

$\alpha_i + 1 = 0.2$ $F_{ai=6}$

$\alpha_i = 0.2$ $G \cap F\ \alpha_i = 1$ *(0.2-0)*(1/6)= 0.032*

$\alpha_i + 1 = 0$ $F_{ai=6}$ $GD_q = 0.256$

Por último se evalúa un Itemset de tamaño 3 que contiene los ítems: Raza-Blanca, Edad-Joven y Sexo. En este caso se combinan atributos difusos y no difusos indistintamente con el objetivo de demostrar la validez del cálculo del soporte con el mecanismo GD_q en ambos tipos de datos:

$\alpha_i = 1$	$G \cap F \ \alpha_i = 5$	*(1-0.9)*(5/6)= 0.083*
$\alpha_i + 1 = 0.9$	$F_{ai=6}$	
$\alpha_i = 0.9$	$G \cap F \ \alpha_i = 1$	*(0.9-0.8)*(1/6)= 0.016*
$\alpha_i + 1 = 0.8$	$F_{ai=6}$	
$\alpha_i = 0.8$	$G \cap F \ \alpha_i = 1$	*(0.8-0.6)*(1/6)= 0.032*
$\alpha_i + 1 = 0.6$	$F_{ai=6}$	
$\alpha_i = 0.6$	$G \cap F \ \alpha_i = 2$	*(0.6-0.5)*(2/6)= 0.03*
$\alpha_i + 1 = 0.5$	$F_{ai=6}$	
$\alpha_i = 0.5$	$G \cap F \ \alpha_i = 3$	*(0.5-0.3)*(3/6)= 0.1*
$\alpha_i + 1 = 0.3$	$F_{ai=6}$	
$\alpha_i = 0.3$	$G \cap F \ \alpha_i = 1$	*(0.3-0.2)*(1/6)= 0.016*
$\alpha_i + 1 = 0.2$	$F_{ai=6}$	
$\alpha_i = 0.2$	$G \cap F \ \alpha_i = 1$	*(0.2-0)*(1/6)= 0.032*
$\alpha_i + 1 = 0$	$F_{ai=6}$	$GD_q = 0.309$

Mediante las evaluaciones realizadas se obtuvo el soporte de determinados itemsets de la tabla 5, esta solución permite calcular el soporte de los atributos tanto difusos como categóricos, ya que el algoritmo *Apriori-Like* expuesto por (Garabote, 2012) solamente puede trabajar datos duros, es decir, categóricos. Por lo que no es factible para extraer reglas de asociación provenientes de datos.

El *Apriori-like-1* propone extraer la mitad de las reglas de asociación a partir de un ítemsets dado puesto que la otra mitad, se refiere a las mismas reglas lo que con la particularidad de que se invierten el antecedente y el consecuente como se muestra a continuación:

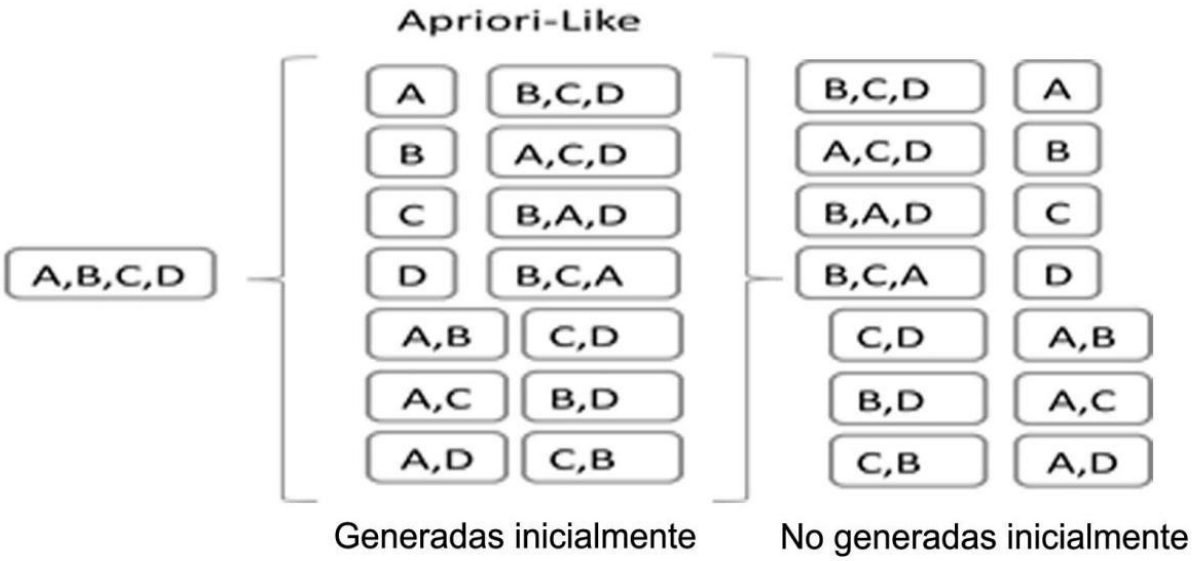


Figura 3. Reglas de asociación generadas y no generadas según *Apriori-like-1*.

Una vez que son generadas las reglas iniciales, *Apriori-like-1* invierte el antecedente con el consecuente de la misma, y vuelve a calcular el soporte y la confianza, en caso de cumplir con los parámetros establecidos para estos indicadores se guarda dicha regla, de lo contrario se descarta.

2.2 Implementación

La implementación del algoritmo está basada en el *Apriori-Like*, puesto que lo que se pretende realizar es una mejora al algoritmo expuesto en (Garabote, 2012), donde los cambios están dirigidos al mecanismo de conteo del soporte. Por lo tanto el algoritmo recibe cuatro parámetros al igual que *Apriori-Like*. El primero es el conjunto de transacciones, las mismas se encuentran almacenadas en una base de datos (D) que una vez procesada la información contenida en ella, se escribirá en un fichero en texto plano. Los datos recopilados en dicho fichero deben cumplir con el formato:

$$tabla - atributo: valor(grado\_pertenencia), tabla - atributo: valor(grado\_pertenencia), \ldots,$$

Figura 4. Formato de entrada de las transacciones de *Apriori-like-1.*

Donde *tabla* es el nombre que llevan las tablas en la base de datos. Este formalismo surge debido a la necesidad de distinguir campos iguales de tablas diferentes. Por lo tanto es opcional puesto que no siempre los datos son extraídos desde diferentes tablas lo que permite que los nombres de los atributos no se repitan. Luego del nombre de la tabla se escribe *atributo,* que es el nombre de los campos contenidos en las tablas y *valor* almacena los valores correspondientes a cada campo. Por último *grado_pertenencia* es el valor asociado a la función de pertenencia del elemento en cuestión, es decir, el grado de veracidad del mismo. Es importante seguir el orden propuesto y los separadores deben ser los definidos anteriormente, de otra forma el algoritmo no devuelve los resultados esperados. A continuación se muestra un ejemplo de la (Tabla 1) antes descrita, transformado al formato requerido:

```
wkswork2:1(1),yrlastwk:20(1),workedyr:2(1),inctot:001250(1),
labforce:2(1),occ1950:employee(0.85),occscore:22(1),sei:18(1),
labforce:2(1),occ1950:employee(0.63),occscore:30(1),sei:24(1),
wkswork2:6(1),yrlastwk:10(1),workedyr:2(1),inctot:008450(1),
incwage:999999(1),incbus:999999(1),incfarm:999999(1),incss:99999(1),
```

Figura 5. Base de transacciones del censo en formato de entrada.

El segundo parámetro que recibe es el conjunto de todos los ítems que pueden estar contenidos en las transacciones. Estos ítems deben estar escritos en el formato descrito anteriormente, exceptuando la definición de la función de pertenencia y deben estar escritos en la primera línea del fichero donde se guardan las transacciones.

$$tabla - atributo: valor, tabla - atributo: valor, \ldots,$$

Figura 6. Formato de entrada de los ítems de *Apriori-like-1.*

Las últimas dos entradas definidas para este algoritmo son los valores mínimos de soporte y confianza respectivamente. Estos valores oscilaran ente 0 y 100 de acuerdo al porciento que determine el usuario para que una regla sea relevante.

El algoritmo *Apriori-like-1* como es básicamente una versión del algoritmo *Apriori-Like* propuesto en la tesis (Garabote, 2012), además de las entradas clásicas, mantiene los mismos subprocesos básicos de

generación de ítemsets frecuentes y de generación de las reglas de asociación relevantes. Para el funcionamiento del mismo se diseñaron dos clases:

- La clase "**Rule**" contiene las propiedades básicas de las reglas de asociación definidas como el antecedente, el consecuente y la confianza.
- La clase "**Apriori**" contiene una lista de todas las transacciones así como de los ítems, contiene además un listado de los ítemsets que resulten frecuentes y una serie de funciones que responden a la solución propuesta.

A continuación se muestra la representación en UML (Lenguaje Unificado de Modelado por sus siglas en ingles) de las clases anteriormente descritas:

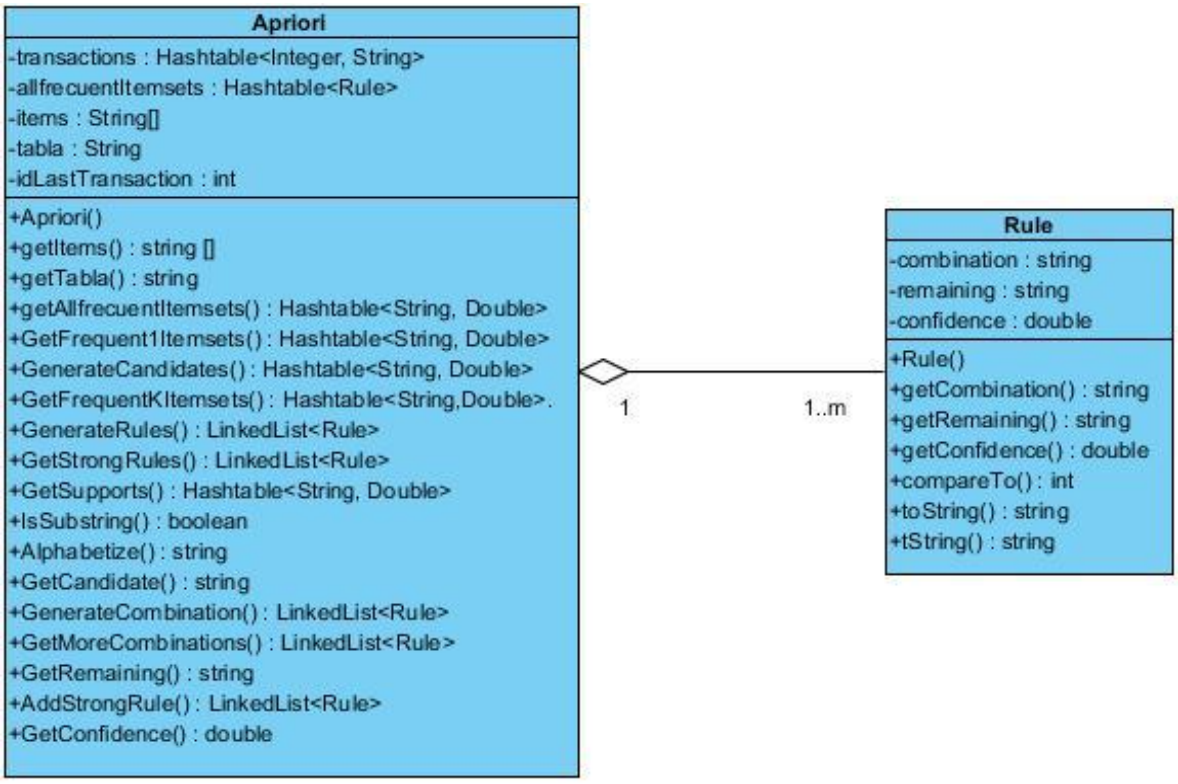


Figura 7. Diagrama UML

Las funciones definidas son las mismas que las propuestas en la tesis de (Garabote, 2012), haciendo modificaciones a la función "*GetSupports*", puesto que esta es la encargada de realizar el mecanismo de conteo de soporte. A continuación se muestra el código de dicha función una vez modificado:

```
private Hashtable<String, Double> GetSupports(Object[] candidates) {
    double[] supp = new double[candidates.length];
    Hashtable<String, Double> candReturn = new Hashtable<>();
    ArrayList<ArrayList<Double>> alfaCut = new ArrayList<>();

    a++;
    b += b + candidates.length;

    for (int j = 0; j < candidates.length; j++) {
        alfaCut.add(new ArrayList<Double>());
    }

    for (int i = 0; i < transactions.size(); i++) {
        for (int j = 0; j < candidates.length; j++) {
            String transAlp = (String) candidates[j];

            if (IsSubstring(transAlp, transactions.get(i + 1), alfaCut.get(j))) {
                supp[j]++;
            }
        }
    }


    for (int i = 0; i < candidates.length; i++) {
        Object[] auxAlfa = alfaCut.get(i).toArray();
        Arrays.sort(auxAlfa);
        double operacion = 0;

        if (auxAlfa.length == 1) {
            operacion = (double) auxAlfa[0] * supp[i];
        } else {
            operacion = ((double) auxAlfa[auxAlfa.length - 1] - (double) auxAlfa[auxAlfa.length - 2]) * supp[i];

            for (int j = auxAlfa.length - 2; j > 0; j--) {
                operacion += ((double) auxAlfa[j] - (double) auxAlfa[j - 1]) * supp[i];
            }
        }
```

Figura 8. Algoritmo encargado de contar todos los soportes de los ítemsets candidatos.

Esta función se encarga de contar cada uno de los candidatos mediante la fórmula antes propuesta. Para ello utiliza una variable *alfaCorte* que es la encargada de guardar los distintos valores de pertenencia de los ítems. A partir de ella se efectúa la sumatoria de la resta entre los alfa cortes de mayor a menor y luego se multiplica ese resultado por la cantidad de transacciones que contienen al Itemset entre el total de transacciones.

Otras de las funciones modificadas es la encargada de verificar si un Itemset es parte de una transacción. Esta función además se encarga de llenar la variable *alfaCorte* utilizada en el método anterior. El código de la función IsSubstring se muestra a continuación:

```
private boolean IsSubstring(String subStr, String str, ArrayList<Double> alfaCut) {
    String[] subString = subStr.split(",");

    for (int i = 0; i < subString.length; i++) {
        if (!str.contains(subString[i] + "(")) {
            return false;
        } else {
            int pos = str.indexOf(subString[i] + "(") + subString[i].length();
            String alfa = "";

            while (str.charAt(pos + 1) != ')') {
                alfa += str.charAt(pos + 1);
                pos++;
            }

            double auxAlfa = Double.parseDouble(alfa);

            if (alfaCut.isEmpty()) {
                alfaCut.add(auxAlfa);
            } else if (!alfaCut.contains(auxAlfa)) {
                alfaCut.add(auxAlfa);
            }
        }
    }
    return true;
}
```

Figura 9. Algoritmo encargado de verificar la existencia de un Itemset en una transacción.

La función Solution es la función principal del algoritmo la misma se encarga de buscar todos los ítemset frecuentes de valor 1, después genera todos los candidatos frecuentes y finalmente genera las reglas que sean fuertes ya que descarta las que no sean fuertes. A continuación se muestra el código del mismo:

```
public LinkedList<Rule> Solution(int minSup, int minConf) {
    double minSupport = (double) minSup / 100;
    double minConfidence = (double) minConf / 100;

    Hashtable<String, Double> frequent1Itemsets = GetFrequent1Itemsets(minSupport);

    Hashtable<String, Double> frequentKItemsets = frequent1Itemsets;
    Hashtable<String, Double> candidates = new Hashtable<>();

    do {
        candidates = GenerateCandidates(frequentKItemsets);
        frequentKItemsets = GetFrequentKItemsets(candidates, minSupport);
    } while (!candidates.isEmpty());

    LinkedList<Rule> rules = GenerateRules();
    LinkedList<Rule> strongRules = GetStrongRules(minConfidence, rules);

    return strongRules;
}
```

Figura 10. Algoritmo encargado de retornar las reglas fuertes.

Una vez mencionados los principales cambios realizados al *Apriori-Like* para establecer un mecanismo de conteo de soporte que permite la extracción de reglas de asociación difusas. Se puede establecer el algoritmo para dicho mecanismo como se muestra a continuación:

Mecanismo de Conteo del Soporte para *Apriori-Like-1*	
1.	**Buscar todos los α del Itemset para todas las transacciones.**
2.	**Ordenar los α de mayor a menor.**
3.	**Restar $\alpha_i - \alpha_{i+1}$**
4.	**Sumar $\alpha_{i+1} - \alpha_{i+2}$**
5.	**Repetir el paso 3 mientras existan α en la lista.**
6.	**Encontrar la cantidad de transacciones que contienen al Itemset.**
7.	**Multiplicar el resultado de la resta por la cantidad encontrada en el paso 6.**
8.	**Dividir entre la cantidad total de transacciones.**

Las salidas del algoritmo son Itemsets candidatos con su respectivo soporte. Estos Itemset luego son evaluados para determinar cuáles de ellos son frecuentes. A partir de los frecuentes entonces se determinan las reglas de asociación fuertes, que son aquellas que cumplen con los umbrales de soporte y la confianza mínimos definidos por el usuario.

2.3 Pruebas funcionales

El funcionamiento de Apriori-like-1 se valida a partir de pruebas de caja negra que verifican si la solución desarrollada ofrece resultados positivos. Estos resultados deben obtenerse a partir del conjunto de datos escritos en el formato establecido anteriormente. La muestra con la que se prueba el algoritmo fue extraída de las bases de datos del censo de los Ángeles. Una vez procesada y escrita de forma correcta, de acuerdo a los estándares de entrada del algoritmo, se puede contabilizar un total de 2675 ítems y 10000 transacciones. Con este volumen de datos, para valores de soporte y confianza del 50% y 80% respectivamente, Apriori-like-1 devuelve resultados positivos en 5 minutos y 51 segundos. La prueba fue desarrollada en una computadora con 1 Gb de memoria RAM y un procesador intelCore2Duo a 2.8 GHz. Los resultados obtenidos se muestran a continuación:

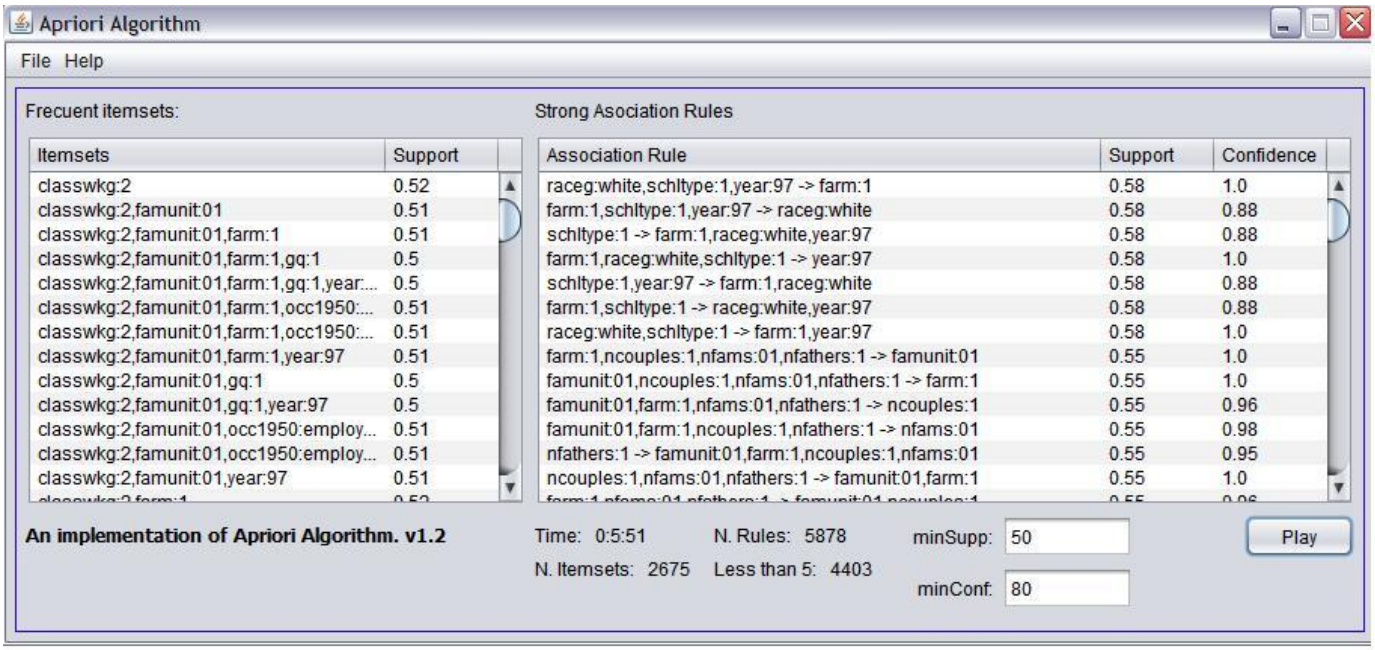


Figura 11. Resultados obtenidos al ejecutar *Apriori-like-1*.

2.4 Conclusiones

En el presente capítulo se muestran varios ejemplos con el que se pretende aclarar el problema que representa la extracción de reglas de asociación difusas. Se presenta una propuesta de solución que pretende modificar el algoritmo de extracción de reglas de asociación propuesto en (Garabote, 2012) bajo

el nombre de Apriori-Like. El Apriori-Like, entre sus limitaciones, está diseñado para descubrir reglas a partir de datos categóricos, las bases de datos del censo de Los Ángeles contienen datos tanto categóricos como difusos. Debido a esto, los principales ajustes están enmarcados a su mecanismo de conteo de soporte. Se presenta la solución del mismo, exponiendo como es que funciona el mecanismo de conteo de soporte. Se valida dicho algoritmo a partir de pruebas de caja negra mostrando resultados satisfactorios en un tiempo aceptable.

Capítulo 3. Validación de la solución

3.1 Introducción

Existen una serie de métodos o patrones que permiten evaluar y validar los resultados alcanzados en una investigación. Entre los cuales se encuentran la demostración, experimentación, simulación, uso de métricas, evaluación comparativa, razonamiento lógico y los modelos matemáticos (Vaishnavi, 2008). A continuación se describen algunos de estos métodos debido a que son los de mayor relevancia en las ciencias de la computación según los desarrolladores de esta investigación:

- **Demostración**

  Intenta demostrar que la solución es factible y válida para una o varias situaciones predefinidas. Es especialmente relevante cuando la demostración de una solución en sí misma se considera una contribución. Consta de dos momentos importantes, construir la solución o prototipo de solución que demuestre que esta es factible y demostrar que la solución construida es razonable para un conjunto predefinido de situaciones. Estas situaciones deben estar predefinidas y no se ha creado para adaptarse a la solución. También puede mostrar las deficiencias de la solución. Por otra parte, puede mostrar que la solución es viable y aceptable (Vaishnavi, 2008). Ha sido utilizado en innumerables artículos científicos de la especialidad hasta el punto de ser el tipo de validación que más se utiliza desde el año 1999 (Shaw, 2002).
- **Experimentación**

  Intenta validar o rechazar un conjunto de hipótesis asociada a las afirmaciones acerca de la solución. Según (H. Sampieri, 1991) un experimento es un estudio de investigación en el que se manipulan deliberadamente una o más variables independientes para analizar las consecuencias de esa manipulación sobre una o más variables dependientes, dentro de una situación de control para el investigador.

- **Simulación**

  Intenta validar la solución propuesta para el problema de investigación a través de un software de simulación. Consta de cinco momentos importantes, el primero de ellos es desarrollar el modelo conceptual del problema y su solución para que sea simulado en una computadora. Luego se desarrolla el conjunto inicial de datos de prueba, se selecciona la simulación diseñada

específicamente para el dominio del problema. Se ejecuta dicha simulación para el conjunto de prueba elaborado con anterioridad y por último se demuestra la validez de la solución argumentando que las pruebas realizadas representan situaciones de la vida real (Vaishnavi, 2008).

- **Razonamiento Lógico**
  Utiliza la argumentación como forma de validación de la solución. Es una forma más débil de validación que el modelo matemático o el uso de experimentos. Consta de tres momentos importantes, identificación de las suposiciones (axiomas) relacionadas con el problema, identificación de las reglas (reglas de deducción) relacionadas con el problema o la solución y construcción de un camino lógico de las hipótesis (axiomas) a los planteamientos de la solución, utilizando las reglas de deducción que se identifiquen. Esta técnica constituye un modelo matemático para validar cualquier investigación siempre que no exista vaguedad en demostrar que las afirmaciones son consecuencia lógica de los axiomas (Vaishnavi, 2008)

- **Modelos Matemáticos**
  Intenta demostrar matemáticamente las afirmaciones acerca de la solución desarrollada. Consta de cuatro momentos importantes, expresar las afirmaciones acerca de la hipótesis de forma cuantitativa y precisa, convertir dichas afirmaciones para ser probadas como un teorema bien definido, demostrar los resultados auxiliares (lemas) que pueden ayudar a demostrar el teorema y por último demostrar el teorema de las afirmaciones, que pueden utilizar los lemas ya probados. Este modelo ofrece la forma más segura de validación de la solución. Esta validación es incluso más certera que la validación experimental (Vaishnavi, 2008).

Estos patrones varían en términos de su idoneidad y la fuerza con la que pueden establecer la validez de una solución. El patrón de demostración proporciona la forma más débil de validación. Se puede, sin embargo, ser apropiado si la solución es novedosa y resuelve un problema para el que no existe ninguna solución. En el otro extremo, el patrón de pruebas matemáticas proporciona la forma más fuerte de validación. La fuerza del patrón de razonamiento lógico depende de la fuerza y la precisión de sus argumentos y suposiciones. En general, es una alternativa o suplemento a la utilización de patrones de experimentación y simulación. Patrones de experimentación y simulación son útiles cuando el problema es

complejo y no susceptible de una demostración matemática. El uso del patrón de métricas es valioso en la experimentación, simulación, y los patrones de las pruebas matemáticas. Le ayuda en la cuantificación de las afirmaciones acerca de la solución (Vaishnavi, 2008).

El uso de uno o varios métodos de validación depende en gran medida de las características del problema estudiado. La presente investigación utiliza el método de “Demostración” como mecanismo de validación. Este patrón se adapta correctamente a las características del trabajo siendo ideal para corroborar la validez del mismo pues no existen soluciones anteriores para este problema. La demostración en sí misma tiene valor práctico al estar compuesta por un prototipo funcional que permite construir modelos descriptivos aun cuando es necesario perfeccionar algunos detalles de usabilidad en dicho prototipo. Por último, la “Demostración” constituye el patrón de validación más utilizado en las publicaciones científicas para el área de las ciencias de la computación de acuerdo a los trabajos de (Shaw, 2002), (Cañete, 2002), (Shaw, 2003).

3.2 Recursos computacionales utilizados

Los recursos computacionales en esta investigación refieren las características de hardware y software de la computadora utilizada en el proceso de validación de la solución. Para efectuar el proceso de “Demostración” se utilizó una computadora ACPI Multiprocessor PC. Esta máquina cuenta con una motherboard Intel Rogers City DG965RY que incorpora un procesador DualCore Intel Core 2 Duo E4500 a 2.20GHz y una memoria Ram con una capacidad de 1024 MB. Los algoritmos propuestos fueron probados en una plataforma Microsoft Windows XP Professional publicada en el año 2002, a la que se le incorpora el paquete de corrección de errores Service Pack 3.

3.3 Conjunto de datos utilizados

El conjunto de datos utilizados para realizar la demostración fue recolectado de la base de datos del censo de población de Los Ángeles que se encuentra público en internet, del mismo solamente se trabaja con los años desde 1970 hasta 1990. Específicamente, fueron utilizados los datos de mayor interés para el presente trabajo. Los datos fueron transformados en transacciones escritas en el formato definido para la entrada del algoritmo *Apriori-like-1*. A partir de estos se evaluó la cantidad de reglas obtenidas definiendo un máximo de 250 reglas como valor adecuado para la utilización por parte de los usuarios finales. Se

definieron como reglas de fácil comprensión aquellas que contengan como máximo 5 elementos o ítems debido a que es la cantidad de elementos que se considera que una persona puede interpretar fácilmente.

La muestra de datos obtenida a partir de las bases de datos del censo consta de 10000 tuplas y 36 columnas. Se determinó que fuera 10000 el número de las transacciones puesto que muchas de las soluciones publicadas para extraer reglas de asociación (Li, 2009), (Agrawal, 1993), (Yu-Lu, 2009), (Yueqin, 2009) utilizan valores semejantes para probar su validez. Además, con este volumen de datos se puede determinar el buen funcionamiento del algoritmo y obtener reglas de asociación difusas. En el presente ejemplo los valores serán variables, de manera que permitan obtener diferentes conjuntos de reglas.

Ítems	Transacciones	Soporte (%)	Confianza (%)
2675	10000	60%	70%
2675	10000	86%	94%
2675	10000	90%	98%
2675	10000	87%	74%

Tabla 7. Especificación de los datos utilizados.

3.4 Discusión de los resultados

Al aplicar el algoritmo Apriori-like-1 utilizando los datos antes descritos se obtuvo un grupo variable de reglas de asociación difusas para cada combinación de soporte y confianza. A partir del total de reglas generadas para cada caso, se puede determinar con cuáles combinaciones de soporte y confianza el algoritmo devuelve resultados adecuados para el trabajo de los usuarios finales, atendiendo al indicador mencionado anteriormente. A continuación se muestra la cantidad de reglas generadas para cada caso así como la clasificación del algoritmo de acuerdo a esta cantidad:

Soporte	Confianza	Total	Tiempo (h:m:s)	Clasificación
60%	70%	1943	0:5:8	No adecuado
86%	94%	89	0:4:28	Adecuado
90%	98%	77	0:4:23	Adecuado

87%	74%	84	0:4:38	Adecuado

Tabla 8. Clasificación del algoritmo de acuerdo a la cantidad de reglas generadas.

A partir de los datos mostrados en la tabla anterior se puede determinar que el algoritmo desarrollado devuelve un volumen de reglas adecuado en la mayoría de los casos. Para valores de soporte y confianza por debajo de los 60 y 70 % respectivamente la cantidad de reglas se hace demasiado grande. Se puede concluir que la implementación desarrollada cumple con las expectativas esperadas para umbrales de soporte y confianza por encima del 85%. Con respecto al tiempo, se evidencia una mejora considerable puesto que para la misma cantidad de transacciones e ítems el *Apriori-Like* devuelve resultados en 5 minutos y 23 segundos para un soporte del 86% y una confianza del 94%.

A continuación se presentan algunas de las reglas de asociación generadas por *Apriori-Like-1*:

Reglas de asociación	Sop.	Conf.
Unidad Familiar: 01, Grupo Sanguíneo: 1, No. Familiares: 01, Año:97 -> Estado Vivienda: 1	88%	98%
Estado Vivienda:1, Grupo Sanguíneo:1,Raza:Negra -> Unidad Familiar:01, Año:97	92%	100%
Grupo Sanguíneo: 1, No. Familiares: 01, Año:97 -> Unidad Familiar: 01, Estado Vivienda:1	88%	96%
Raza:Negra -> Unidad Familiar: 01, Estado Vivienda:1, Año:97	92%	96%

Tabla 9. Muestra de reglas de asociación.

Atendiendo a la cantidad de ítems que las componen, las reglas pueden clasificarse en fáciles o difíciles de comprender, como se muestra en la tabla siguiente. Aquellas reglas cuya cantidad de elementos que la componen es menor o igual que 5, se consideran de fácil comprensión para los usuarios finales. A continuación se muestran las reglas obtenidas clasificadas de acuerdo al criterio anteriormente expuesto:

Entrada		Salida		
Soporte	**Confianza**	**Difícil Comprensión**	**Fácil Comprensión**	**Total**

60%	70%	13	1930	1943
86%	94%	0	89	89
90%	98%	0	77	77
87%	74%	0	84	84

Tabla 10. Clasificación de las reglas de asociación según la cantidad de ítems.

Al observar los resultados se puede apreciar que en la mayoría de los casos todas las reglas descubiertas fueron de fácil comprensión, por tanto los ítemsets frecuentes generados no tuvieron un tamaño mayor a 5 elementos. Lo que denota una mejora en el mecanismo de extracción puesto que extrae reglas fáciles de analizar por los especialistas favoreciendo el proceso de toma de decisiones.

En la siguiente tabla se muestran los resultados obtenidos en cuanto a cantidad de reglas generadas por los algoritmos Apriori-Like (Garabote, 2012), Apriori (Agrawal, 1994) y el Apriori-Like-1 desarrollado en el presente trabajo. Para efectuar dicha comparación se tomaron las mismas entradas definidas para la demostración, variando los umbrales de soporte y confianza mínimos.

Entrada		**Salida**		
Sop.	**Conf.**	**Cantidad de Reglas Generadas por:**		
		Apriori-Like	**Apriori-Like-1**	**Apriori**
60%	70%	14447	1943	0
86%	94%	465	89	0
90%	98%	245	77	0
87%	74%	377	84	0

Tabla 11. Comparación de la cantidad de reglas generadas.

A partir de los resultados mostrados en la tabla anterior se puede concluir que para soporte y confianza mínimos por debajo del 70% los algoritmos generan gran cantidad de reglas a excepción del Apriori (Agrawal, 1994) que no genera reglas puesto que este no está diseñado para operar sobre datos categóricos ni difusos. Para soporte y confianza mínimos por encima del 70% los algoritmos obtienen

menor cantidad de reglas, evidenciándose en todos los casos que los cambios implementados por Apriori-Like-1 contribuyen a la reducción del número de reglas minadas por Apriori-Like (Garabote, 2012).

3.5 Conclusiones parciales

El presente capítulo presenta una serie de patrones o métodos utilizados para validar los resultados de una investigación científica. Se determinó que el patrón “Demostración” es el adecuado para realizar las pruebas de validación para la solución propuesta. Se definieron los elementos fundamentales de la demostración propuesta así como los recursos de hardware y software con que se cuenta para efectuar las pruebas.

Para demostrar el correcto funcionamiento del algoritmo se variaron los valores de soporte y confianza sobre un conjunto de datos extraído de las bases de datos del censo de población de los Ángeles. Los resultados obtenidos demuestran, primeramente, que Apriori-like-1 obtiene una cantidad de reglas de asociación mucho menor y con un tiempo más eficiente que el algoritmo expuesto en la tesis de (Garabote, 2012). En la mayoría de los casos de prueba, el algoritmo devolvió resultados adecuados en cuanto a la cantidad de reglas que genera, de acuerdo a los valores de soporte y confianza mínimos. Por último se clasificaron las reglas de acuerdo a la cantidad de elementos que la conforman, la evaluación devolvió que en la mayoría de los casos se obtiene que todas las reglas son de fácil comprensión o que están formadas por 5 elementos o menos.

Conclusiones

En el presente trabajo se implementó un algoritmo para la extracción de reglas de asociación difusas en conjuntos de datos pertenecientes al censo de población de los Ángeles. El referido algoritmo sigue los supuestos teóricos establecidos en el diseño del algoritmo Apriori-Like y realiza un grupo de mejoras enfocadas a la tipología específica de los conjuntos de datos bajo estudio, dentro de ellas se destacan las siguientes:

- Se ajustó el algoritmo para que pueda trabajar con datos difusos favoreciendo el acercamiento de los resultados a la forma de razonamiento del cerebro humano.
- Se modificó la expresión para el conteo de soporte de forma tal que se reduce significativamente la cantidad de reglas de asociación generadas.

Los resultados alcanzados permiten concluir que:

1. Es posible obtener un conjunto de reglas de asociación difusas a partir de la aplicación del algoritmo diseñado, utilizando los formatos de entrada definidos en el presente trabajo.
2. Las reglas de asociación difusas permiten obtener modelos significativos que pueden ser empleados en diferentes sectores de la sociedad favoreciendo el acierto en la toma de decisiones.
3. La cantidad de reglas y efectividad del algoritmo dependen en gran medida de los valores de soporte y confianza determinados por los usuarios.
4. Aunque se logra una disminución evidente en el volumen de reglas generadas aún es numeroso y debe ser tratado para reducirlo en función de su relevancia.

Recomendaciones

Para el mejoramiento de este trabajo es importante optimizar un grupo de elementos que no fueron tomados en cuenta por cuestiones de tiempo, dentro de ellos señalamos los siguientes, debido a su relevancia y a que forman parte del trabajo futuro que se desarrollará en la temática:

1. Establecer un mecanismo de ponderación de las reglas.
2. Determinar un volumen de reglas manejables por los especialistas humanos.

Referencias Bibliográficas

A. Savasere, E. Omiecinski, and S.Navathe. 1995. *An eficient algorithm for mining association rules in large.* Atlanta : Institute of Technology, 1995.

Agrawal, Imielinski,Swami. 1993. *Mining associations between sets of items in massive databases.* s.l. : In ACM-SIGMOD, 1993.

Agrawal, R y Srikant, R. 1994. *Fast algorithms for mining association rules.* s.l. : Proceedings of the 20th,, 1994.

Agrawal, R, Imielinski, T y Swami, A. 1993. *Mining Associations between sets of items in massive databases.* s.l. : ACM-SIGMOD International Conference on Data, 1993.

Angélica Urrutia, Marcela Varas, José Galindo. 2001. *Diseño de una Base de Datos Difusa Modelada con UML.* Chile, : s.n., 2001.

Brin, S, y otros. 1997. *Dynamic itemset counting and implication rules for market basket data.* s.l. : SIGMOD, 1997.

Castro, Dr. Rogelio Silverio. 2008. *Minería de datos.* Santa Clara : Dpto. de Ciencias de la Computación, 2008.

Chung., J. D. Holt and S. M. 2001. *Multipass algorithms for mining association rules in text databases.* s.l. : Springer-Verlag, 2001.

Delgado, M., Sánchez, D. y Vila, M. A. 2000. *Fuzzy cardinality based evaluation of quantified sentences.* s.l. : International Journal of Approximate Reasoning, 2000.

Delgado, M., y otros. 2005. *Mining fuzzy association rules: an overview.* s.l. : Soft Computing for Information Processing and Analysis, 2005.

Delgado, Miguel, Ruiz, M. Dolores y Sánchez, Danel. 2008. *Reglas de asociación difusas:Nuevos Retos.* Granada : ESTYLF08, Cuencas Mineras, 2008.

Delgado, Miguel, y otros. 2003. *Fuzzy Association Rules: General Model.* s.l. : IEEE TRANSACTIONS ON FUZZY SYSTEMS, 2003.

Diaz. 2010. *FSI CONJUNTOS DIFUSOS Y LÓGICA DIFUSA.* 2010.

Fayyad, U M, y otros. 1996. *Advances in Knowledge Discovery and Data Mining.* Cambridge : MA : AAAI Press and MIT Press, 1996.

Frank, Witten &. 2000. *Data mining.* 2000.

FSI CONJUNTOS DIFUSOS Y LÓGICA DIFUSA.

Garabote, Andy Fernandez. 2012. *Implementación de un algoritmo Apriori-like para el minado de reglas de asociación en las declaraciones aduanales de mercancías en Cuba.* 2012.

Gyenesei, A. 2001. *A fuzzy approach for mining quantitative.* s.l. : Acta Cybern, 2001.

H. Sampieri, C. Roberto, Fernández Collado, Carlos y Baptista Lucio, Pilar. 1991. *Metodología de la investigación.* MÉXICO, : s.n., 1991.

1999. IPUMS Census Database. *IPUMS Census Database.* [En línea] Irvine, 9 de noviembre de 1999. [Citado el: 13 de marzo de 2013.] dd.ics.uci.edu/databases/ipums/ipums.html.

J. Hernández. Montes, D. Martinetti, S. Montes. 2010. *ESTUDIO DE LA PROBABILIDAD DE ZADEH PARA T-NORMAS ARQUIMEDIANAS.* s.l. : Massachusetts Institute of Technology, 2010.

Jimenez Ruiz, Maria Dolores. 2010. *Modelado formal para la representación y evaluación de reglas de asociación.* Granada : Departamento de ciencias de la computación e inteligencia artificial, 2010.

Jimenez, Maria Dolores Ruiz. 2010. *Modelado Formal para Representación y Evaluación de Reglas de Asociación.* Granada : Universidad de Granada, 2010.

Kuok, C.M, Fu, A.W. y Wong, M.H. 1998. *Mining fuzzy association rules in databases.* s.l. : SIGMOD Record, 1998.

Li, Jiye y Cercone, Nick. 2009. *Introducing A Rule Importance Measure.* Canada : s.n., 2009.

M.Sc., Ing. Oscar G. Duarte V. 2009. *Sistemas de Lógica Difusa - Fundamentos.* Colombia : Departamento de Ingeniería, 2009.

Medina Pagola, José E., y otros. 2007. *Generación de conjuntos de items y reglas de asociación.* La Habana : Dpto. Minería de Datos, Centro de Aplicaciones de Tecnología de Avanzada (CENATAV), 2007.

Mesa Rodriguez, Alejandro, y otros. 2009. *Obtencion d conjuntos frecuentes usando computo reconfigurable.* La Habana : CENATAV, 2009.

Motoda, Hiroshi y Ohara, Kouzou. 2009. *Apriori.* s.l. : Taylor & Francis Group, 2009.

Orallo Hernández, José, Quintana Ramírez, Ma José y Ferri Ramírez, Cesar. 2004. *Introducción a la Minería de Datos.* Madrid : Pearson Educación S.A, 2004.

Puente, Marcelo de la. 2010. *Gestión del conocimiento y minería de datos.* 2010.

Ruiz, Maria Dolores. 2010. *Modelado Formal Para Representaciòn y Evaluaciòn de Reglas de Asociaciòn.* Granada : Universidad de Granada, 2010.

Shaw. 2002. *What makes good research in software engineering.* BERLIN : FOR TECHNOLOGY TRANSFER (STTT), 2002.

Silverstein, C, Brin, S y Motwani, R. 1998. *Beyond market baskets: Generalizing association rules to dependence rules.* s.l. : Data Mining Knowl. Disc, 1998.

2011. Teoría de conjuntos difusos y lógica difusa. *Teoría de conjuntos difusos y lógica difusa.* [En línea] 15 de mayo de 2011. [Citado el: 10 de diciembre de 2012.] http://www.lcc.uma.es/~eva/aic/apuntes/fuzzy.pdf.

Triantaphyllou, Evangelos. 2010. *Data Mining and Knowledge Discovery via Logic-Based Methods.* Louisiana : Springer, 2010.

Triantaphyllou, Evangelos. 2010. *Data Mining and Knowledge Discovery via Logic-Based Methods.* Louisiana : Springer, 2010.

Universidad de Granada. 2011. Modo. *Modo.* [En línea] Granada, 19 de abril de 2011. [Citado el: 28 de noviembre de 2012.] http://modo.ugr.es/es/soft_computing.

Vaishnavi, Vijay K. y Kuechler Jr., William. 2008. *Design Science Research Methods and Patterns.Innovating Information and Communication Technology.* NewYork : Auerbach Publications is an imprint of the , an informa business ,, 2008.

Vidal, Tomás Arredondo. 2012. *Introducción a la Lógica Difusa.* 2012.

Yuliana Corzo. 2009. La Lógica Difusa. [En línea] 2009. casanchi.com/casanchi_2001/difusa01.htm.