Minería de Reglas de Asociación sobre Datos Mezclados

Ansel Yoan Rodríguez González, José Francisco Martínez Trinidad,
Jesús Ariel Carrasco Ochoa, José Ruiz Shulcloper

Reporte Técnico No. CCC-09-001
31 de Marzo de 2009

Luis Enrique Erro 1
Sta. Ma. Tonantzintla,
72840, Puebla, México.

Minería de Reglas de Asociación sobre Datos Mezclados

Ansel Yoan Rodríguez González [1,2], José Francisco Martínez Trinidad [1], Jesús Ariel Carrasco Ochoa [1], José Ruiz Shulcloper [2]

[1]Coordinación de Ciencias Computacionales
Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro # 1, Santa María Tonantzintla, Puebla, 72840, México
{ansel,fmartine,Ariel}@ccc.inaoep.mx

[2]Centro de Aplicaciones de Tecnologías de Avanzada.
Ave. 7ma, # 21812 %218 y 222, Rpto. Siboney, Municipio Playa,
Ciudad de la Habana, Cuba
{arodriguez,jshulcloper}@cenatav.co.cu

Resumen. La mayoría de los algoritmos existentes para el minado de reglas de asociación asumen que dos subdescripciones de objetos son similares si y solo si ellas son iguales, sin embargo en problemas reales son usadas otras medidas de semejanzas. Esta propuesta de tesis doctoral aborda el problema de minería de reglas de asociación usando funciones de semejanza en colecciones de datos que contienen Datos Mezclados, es decir, que combinan diferentes tipos de datos (numéricos y no numéricos) en las descripciones de los objetos que la forman. Como resultado preliminar se proponen dos algoritmos para el minado de patrones similares frecuentes para funciones de semejanza binarias. Los resultados obtenidos muestran que el comportamiento de los algoritmos propuestos es superior al del único algoritmo existente para el minado reglas de asociación usando funciones de semejanza diferentes de la igualdad.

Palabras clave. Minería de datos, patrón frecuente, regla de asociación, datos mezclados, funciones de semejanza.

1 Introducción

La Minería de Reglas de Asociación es una técnica importante en la Minería de Datos y consiste en encontrar las asociaciones interesantes en forma de relaciones de implicación entre los valores de los atributos de los objetos de un conjunto de datos. Numerosos y recientes estudios [1-7] avalan su actualidad e importancia y su aplicación en áreas como mercadeo, bioinformática, medicina y seguridad de redes entre otras.

Esta técnica emergió en la década de los 90 con una aplicación práctica, el análisis de información de ventas para el mercadeo [8, 9]. Mediante ella se descubrían las relaciones entre los datos recopilados a gran escala por los sistemas de terminales de punto de venta de supermercados. Los datos consistían en colecciones de transacciones, también conocidas como bases de datos transaccionales, donde cada transacción expresa qué productos compró un cliente. Un ejemplo de este tipo de colecciones se muestra en la Tabla 1.

Tabla 1. Ejemplo de colección de transacciones.

Transacciones	
No.	**Productos comprados**
1	Leche, Pan
2	Pan, Mantequilla
3	Cerveza
4	Leche, Pan, Mantequilla
5	Pan
6	Leche, Pan, Mantequilla

En este contexto una regla de asociación podría ser "Si un cliente compra pan y leche, entonces también compra mantequilla", formalmente

$$(pan \wedge leche) \Rightarrow (mantequilla)$$

El interés de una regla de asociación está dado por su soporte y su confianza, entendiéndose por soporte la frecuencia de aparición en la colección de la combinación de productos involucrados en la regla. Por ejemplo para la colección mostrada en la Tabla 1 se tiene que:

$$supp\left((pan \wedge leche) \Rightarrow (mantequilla)\right) = supp(pan \wedge leche \wedge mantequilla) = \frac{2}{6}$$

Por confianza de una regla entendemos cuánto representa el soporte de la regla, del soporte del antecedente de la regla. Por ejemplo para la colección mostrada en la Tabla 1 se tiene que:

$$conf\left((pan \wedge leche) \Rightarrow (mantequilla)\right) = \frac{supp(pan \wedge leche \wedge mantequilla)}{sup(pan \wedge leche)} = \frac{2}{3}$$

Se considera que una regla es interesante si su soporte y su confianza son mayores o iguales que ciertos umbrales de mínimo soporte y mínima confianza especificados.

Este tipo de reglas fue denominado Reglas de Asociación Binarias. Varios han sido los algoritmos desarrollados para su minado [8-14]. Sin embargo, existen colecciones de datos que contienen Datos Mezclados, es decir, combinan diferentes tipos de datos (numéricos y no numéricos) en las descripciones de los objetos que la forman. Ejemplos de estas colecciones son los censos de población, encuestas, datos geológicos, datos forestales, datos de clientes, bitácoras de servidores de red e historiales clínicos, entre otros.

El problema de descubrir reglas de asociación sobre datos mezclados, introducido en [15] como minado de reglas de asociación cuantitativas, ha sido abordado siguiendo dos esquemas fundamentales: I) discretizar el dominio de los atributos cuantitativos, y transformar el problema al minado de Reglas de Asociación Binarias [15-23]; II) usar los conceptos de la teoría de los conjuntos difusos para manipular los valores de los atributos tanto categóricos como cuantitativos (el grado de pertenencia de los valores de los atributos a los conjuntos difusos es utilizado en el cálculo del soporte y la confianza) [5, 24-32].

El proceso de discretización comúnmente consiste en particionar en intervalos el dominio de los atributos. Esto conlleva a una pérdida de información pues si los intervalos son muy pequeños las ocurrencias de los intervalos pueden ser pequeñas también, y no sobrepasar el umbral para ser frecuentes. Así algunos posibles patrones frecuentes podrían perderse y por consiguiente, reglas interesantes también podrían quedar ocultas. Por otro lado, el proceso de discretización puede resultar en intervalos, que podrían no ser semánticamente significativos e incluso no tener sentido para los expertos humanos. Además, en la práctica hay atributos numéricos que no pueden ser discretizados, por ejemplo, en geociencias la Anomalía de Bouguer y su gradiente [33]. En estos ejemplos prácticos,

los especialistas del área, consideran que dos valores son equivalentes si el valor absoluto de su diferencia es menor que un umbral diferente de cero. Por tanto, para todo particionamiento del dominio de dichos atributos que se realice, siempre existen dos valores equivalentes que quedan separados por la frontera que define a dicho particionamiento.

El esquema basado en conjuntos difusos, a diferencia de las fronteras duras utilizadas en el esquema de discretización para definir los intervalos, permite definir fronteras difusas, lo cual aumenta la posibilidad de modelar las relaciones entre los valores de los atributos. Además, incorpora información adicional específica del problema mediante la definición de los conjuntos difusos para cada atributo.

Por otra parte, no necesariamente las descripciones de los objetos tienen que ser las mismas para ser consideradas como iguales. El concepto de semejanza (o similaridad) o su contrario, el concepto de diferencia (o disimilaridad), no necesariamente distancia, entre descripciones o subdescripciones de objetos, es una herramienta metodológica comúnmente utilizada en las ciencias poco formalizadas como la geología, medicina, sociología, etc., para tomar decisiones. Además varios algoritmos de clasificación supervisada [34-36] y no supervisada [37, 38] hacen uso del mismo.

Por ejemplo, se pudiera considerar que dos personas son semejantes en términos de sus edades, si ellas son de la misma generación, lo cual podría ser equivalente a considerar que dos edades son semejantes si el valor absoluto de sus diferencias es a lo sumo 5 años. Observe que este criterio de semejanza es diferente al de intervalos de edades (grupos etarios). En este ejemplo la semejanza fue usada para comparar valores de un atributo en los objetos de estudio. No obstante, la semejanza puede ser usada para comparar objetos completos o partes de ellos. Por ejemplo, se puede considerar que dos objetos son semejantes si ellos son semejantes en todos los atributos o si ellos son semejantes en al menos 90% de los atributos.

Los primeros avances en el uso de funciones de semejanza entre objetos completos o partes de ellos en la Minería de Reglas de Asociación se reportan en [39]. Aunque el algoritmo propuesto fue diseñado sólo para una familia restringida de funciones de semejanza binaria y la eficiencia del mismo no es evaluada, dicho trabajo muestra cómo mediante la incorporación del concepto de semejanza en el conteo de ocurrencias, pueden ser descubiertas reglas de asociación ocultas para los anteriores dos enfoques; abriéndose así un nuevo enfoque de minado de Reglas de Asociación en colecciones de datos mezclados.

En este trabajo se aborda el Minado de Reglas de Asociación en colecciones de datos mezclados, es decir, colecciones donde los objetos están descritos simultáneamente por atributos numéricos y no numéricos. No se utilizará el enfoque de discretización, ni el enfoque difuso, sino que, se explorará el nuevo enfoque, el cual hace uso de funciones de semejanza entre objetos completos o partes de ellos para minar de Reglas de Asociación en colecciones de datos mezclados. La investigación se centrará en el desarrollo de algoritmos de minado de reglas de asociación para este tipo de colecciones de datos, que permitan el uso de funciones de semejanza menos restrictivas que las permitidas actualmente.

2 Problema a resolver

Minar reglas de asociación en colecciones de datos mezclados, considerando la semejanza entre objetos y partes de ellos al contar las ocurrencias de los mismos, que permitan el uso de funciones de semejanza menos restrictivas que las permitidas por el algoritmo existente.

3 Trabajos relacionados

Como se comentó en la introducción, la minería de reglas de asociación se restringió inicialmente a colecciones de datos binarios. Posteriormente esta técnica se desarrolló también para colecciones de datos mezclados. A continuación se presentan los trabajos relacionados con la minería de reglas de asociación para ambos tipos de colecciones.

3.1 Reglas de Asociación en colecciones de datos binarios

Este tipo de reglas fue introducido en [8]. Formalmente se conceptualiza de la siguiente manera: sea $I = \{i_1, \ldots, i_m\}$ un conjunto de ítems, $T = \{t_1, \ldots, t_n\}$ un conjunto de transacciones, cada una contiene ítems del conjunto I, es decir, cada transacción t_i es un conjunto de ítems tal que $t_i \subseteq I$. Una regla de asociación es una implicación de la forma $X \Rightarrow Y$, donde $X \subseteq I$, $Y \subseteq I$ y $X \cap Y = \emptyset$. El problema de minería de Reglas de Asociación Binarias consiste en encontrar todas las reglas interesantes a partir de un conjunto de transacciones.

Comúnmente se considera que una regla es interesante si su soporte es mayor que un umbral de mínimo soporte (minSupp) y su confianza es mayor que un umbral de mínima confianza (minConf). Ambas medidas están basadas en el soporte de un conjunto de ítems.

Definición 1. (Soporte de un conjunto de ítems): El soporte de un conjunto de ítems X en un conjunto de transacciones T, es la fracción de transacciones de T que contienen los ítems de X.

$$supp(X) = \frac{|\{t \in T \mid X \subseteq t\}|}{|T|} \quad (1)$$

Si el soporte de X es mayor o igual que un umbral de mínimo soporte minSupp, se dice que el conjunto de ítems X es frecuente.

Definición 2. (Soporte de una regla): El soporte de una regla $\boldsymbol{X \Rightarrow Y}$ en un conjunto de transacciones $\boldsymbol{T}$, es la fracción de transacciones de $\boldsymbol{T}$ que contienen los ítems de $\boldsymbol{X \cup Y}$.

$$supp(X \Rightarrow Y) = supp(X \cup Y) \quad (2)$$

Definición 3. (Confianza de una regla): La confianza de una regla $\boldsymbol{X \Rightarrow Y}$ es la fracción de transacciones de $\boldsymbol{T}$ que conteniendo a $\boldsymbol{X}$, también contienen a $\boldsymbol{Y}$.

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} \quad (3)$$

El proceso de minado de reglas de asociación consta de dos pasos fundamentales. Primero se buscan en el conjunto de transacciones, los conjuntos frecuentes de ítems [4, 8, 9, 40-44] . Nótese que para que una regla $X \Rightarrow Y$ sea interesante $X \cup Y$ debe ser un conjunto frecuente de ítems. Luego a partir de los conjuntos frecuentes de ítems son generadas las reglas de asociación interesantes [9-11, 45].

Generación de conjuntos frecuentes de ítems

La generación de los conjuntos frecuentes de ítems es el paso más costoso del proceso de minado de reglas de asociación y en el cual se enfocan la mayoría de los trabajos. Esto se debe a que el tamaño del espacio de los posibles conjuntos frecuentes de ítems depende exponencialmente del tamaño del conjunto de ítems I ($|\{X|X \subseteq I, X \neq \emptyset\}| = 2^{|I|} - 1$). Sin embargo, no es necesario recorrer todo el espacio de búsqueda para encontrar los conjuntos frecuentes de ítems, debido a la siguiente propiedad.

Propiedad 1 (Clausura Descendente del soporte): Todo subconjunto de un conjunto frecuente de ítems es frecuente, mientras todo superconjunto de un conjunto no frecuente de ítems tampoco es frecuente.

Como consecuencia de esta propiedad, el espacio de estados asociado con el retículo[1] que forman los posibles conjuntos de ítems, es dividido por una frontera en dos subespacios: el subespacio que sólo contiene conjuntos frecuentes de ítems y el subespacio que sólo contiene conjuntos no frecuentes de ítems.

La Figura 1 muestra el retículo formado por un conjunto de ítems $I = \{i_1, i_2, i_3, i_4\}$, así como la frontera entre el subespacio que sólo contiene conjuntos frecuentes de ítems y el subespacio que sólo contiene conjuntos no frecuentes de ítems, para un conjunto de transacciones $T = \{\{i_1, i_2, i_3\}, \{i_2, i_4\}, \{i_3, i_4\}, \{i_1, i_2, i_3, i_4\}\}$ y un umbral de mínimo soporte $minSupp = 0.5$.

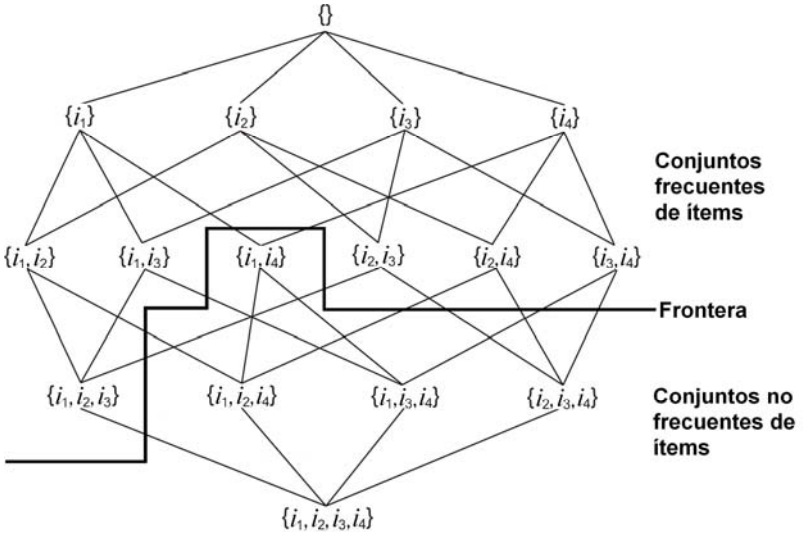

Figura 1. Retículo formado por el conjunto de ítems $\boldsymbol{I = \{i_1, i_2, i_3, i_4\}}$

Existen varias estrategias para recorrer el retículo que forman los posibles conjuntos de ítems. Las mismas pueden clasificarse atendiendo a la dirección de los recorridos en:

- Descendentes: El recorrido se realiza desde el conjunto vacío hasta la frontera.

- Ascendentes: El recorrido se realiza en el sentido opuesto, desde I hasta la frontera.

Estas estrategias a su vez pueden generar los conjuntos de ítems en dos formas:

- En amplitud: Se generan todos los conjuntos frecuentes de ítems de tamaño k antes de generar los conjuntos de ítems de tamaño $k + 1$ [9-11, 13, 14, 45].

- En profundidad: Recursivamente se generan los conjuntos ítems por cada rama de la estructura arbórea que se deriva del retículo [12, 41, 44, 46-49].

[1] En matemática, un retículo, red o lattice es un conjunto parcialmente ordenado en el cual todo subconjunto finito no vacío tiene un supremo y un ínfimo.

El primer algoritmo que hace uso de la propiedad *Clausura Descendente del Soporte* para podar el espacio de búsqueda de los conjuntos de ítems fue propuesto en [9] y se conoce como *Apriori* (Algoritmo 1). A partir de éste se han derivado toda una clase de algoritmos denominados tipo *Apriori.*

Algoritmo 1: Apriori

Input D: Set of transactions
$minSupp$: minimum support threshold
Output F: Frequent itemsets
$L_1 \leftarrow \{i | |\{t \in D | i \in t\}| \geq minSupp\}$
for $(k = 2; L_{k-1} \neq \emptyset; k++)$ **do**
  $C_k \leftarrow \{c | Join(c, L_{k-1}) \wedge Prune(c, L_{k-1})\}$
  forall $transaction \quad t \in D$ **do**
    $C_t \leftarrow \{c \in C_k | c \in t\}$
      forall $candidates \quad c \in C_t$ **do**
        $c.Support \leftarrow c.Support + 1$
      $L_k \leftarrow \{c \in c_k | c.Support \geq minSupp\}$
$F \leftarrow \cup_k L_k$

En este algoritmo, L_k contiene los conjuntos frecuentes de items de tamaño k y C_k los conjuntos de items candidatos a frecuentes de tamaño k. Primeramente son obtenidos y almacenados en L_1 los conjuntos frecuentes de ítems de tamaño 1. Posteriormente en cada iteración del algoritmo se generan los conjuntos de ítems de tamaño k, candidatos a frecuentes, combinando los conjuntos frecuentes de ítems de tamaño $k-1$; y son seleccionados los conjuntos frecuentes de ítems de tamaño k a partir de los conjuntos de ítems de tamaño k candidatos a frecuentes. Este proceso se realiza hasta que, como consecuencia de la propiedad de *Clausura Descendente del Soporte* (Propiedad 1), al comienzo de una iteración el conjunto de conjuntos frecuentes de ítems de tamaño $k-1$ sea vacío. Para generar los conjuntos de ítems candidatos a frecuentes se utilizan las operaciones *Join* y *Prune*:

$$Join(\{i_1, \ldots, i_k\}, L_{k-1}) \equiv \langle \{i_1, \ldots, i_{k-2}, i_{k-1}\} \in L_{k-1} \wedge \{i_1, \ldots, i_{k-2}, i_k\} \in L_{k-1} \rangle \quad (4)$$

$$Prune(c, L_{k-1}) \equiv \langle \forall s[s \subset c \wedge |s| = k-1 \rightarrow s \in L_{k-1}] \rangle \quad (5)$$

La operación $Join$ (ecuación 4) consiste en tomar todos los pares de conjuntos de ítems de tamaño $k-1$ que coincidan en sus $k-2$ primeros ítems y generar conjuntos de ítems de tamaño k manteniendo los $k-2$ ítems comunes y adicionando, en orden lexicográfico, los $(k-1)$-ésimos ítems de los dos conjuntos que se unen. La operación $Prune$ (ecuación 5) consiste en aplicar la propiedad Clausura Descendente del soporte para podar los conjuntos de ítems de tamaño k que tengan al menos un subconjunto no frecuente de ítems de tamaño $k-1$.

Otros algoritmos que hacen uso de la propiedad de clausura descendente son los basados en árboles [41, 44, 47, 50, 51] y los derivados del algoritmo *ECLAT* [4, 12, 49, 52].

Los algoritmos basados en árboles utilizan estructuras de datos arbóreas para almacenar de forma compacta la colección de datos y contar eficientemente las repeticiones de los conjuntos de ítems. Entre los algoritmos más significativos pertenecientes a esta clase se encuentran *FP-Growth* [44], *Patricia Trie-Mine* [41] y *CT-ITL* [47], *CT-PRO* [50], *Apriori-TFP* [51].

El algoritmo *FP-Growth* [44], se basa en el crecimiento o extensión de los conjuntos frecuentes de ítems. En un primer recorrido de la colección de datos, se obtienen los conjuntos frecuentes de ítems

de tamaño 1. En un segundo recorrido de la colección de datos, se inserta cada transacción, con los ítems ordenados descendentemente de acuerdo a su soporte, en una estructura de datos compacta denominada *FP-tree*, también conocida como árbol de prefijos. De esta forma, prefijos iguales, de transacciones diferentes, comparten la misma rama del árbol. Luego, a partir de esta estructura se generan los conjuntos de ítems frecuentes recorriendo recursivamente las ramas del árbol.

El algoritmo *Patricia Trie-Mine* [41], utiliza una estructura de datos denominada *PatriciaTrie* más compacta que la estructura *FP-tree*. *PatriciaTrie*, a diferencia de la estructura *FP-tree*, agrupa en cada nodo del árbol todos los nodos consecutivos que tienen igual valor de soporte. Los conjuntos de ítems frecuentes son generados análogamente al algoritmo *FP-Growth*.

Los algoritmos *CT-ITL* y *CT-PRO* se basan en las mismas ideas que los anteriores algoritmos. El algoritmo *CT-ITL* [47] utiliza una estructura de datos denominada *CT-tree*, la cual modifica la estructura *FP-tree* para almacenar grupos de transacciones, mientras, el algoritmo *CT-PRO* [50], del mismo autor, utiliza una estructura denominada *CFP-tree*, que puede reducir a la mitad el número de nodos de la estructura *FP-tree*.

Otro algoritmo que utiliza estructuras de datos arbóreas es *Apriori-TFP* [51]. En un primer recorrido de la colección de datos se construye una estructura de datos denominada *P-tree*, que almacena los soportes parciales de todos los conjuntos de ítems. A partir de la estructura anterior se construye una segunda estructura de datos denominada *T-tree*. Al finalizar la construcción de la segunda estructura, los conjuntos frecuentes de ítems y sus soportes quedan almacenados en la misma.

Los algoritmos derivados del algoritmo *ECLAT* [12] definen subárboles de búsqueda, mediante las clases de equivalencia de los conjuntos de ítems.

Una clase de equivalencia I'' de un conjunto $I' \subset I$ es el conjunto resultante de la unión de I' con un ítem $X \in I$ lexicográficamente mayor que todo ítem en I'. En la Figura 2 se muestran para el conjunto de ítems $I = \{i_1, i_2, i_3, i_4\}$, las clases de equivalencia de los conjuntos de ítems $\{i_1\}$, $\{i_2\}$, $\{i_3\}$, $\{i_4\}$ con línea continua y el siguiente nivel de clases de equivalencia con línea punteada.

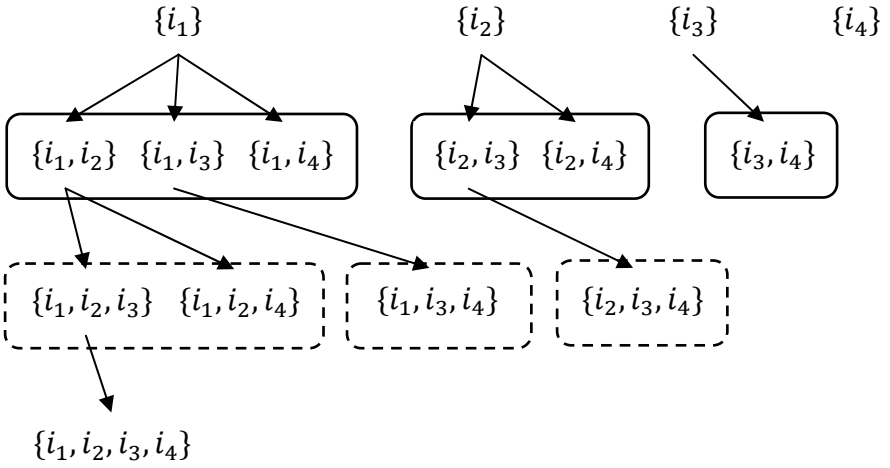

Figura 2. Clases de equivalencia.

El algoritmo *ECLAT* [12] se basa en el recorrido de retículo transformando recusivamente cada clase de equivalencia de tamaño k en clases de equivalencia de tamaño $k+1$. El soporte de cada nueva clase de equivalencia $I' \cup X$ se calcula a la vez que se obtiene la misma, mediante la intersección de las listas de los identificadores de las transacciones que contienen a I' y a X respectivamente. Una debilidad del este algoritmo es que procesa un gran número de conjuntos de ítems. Una implementación eficiente del mismo es propuesta en [49], mientras en [52] un nuevo algoritmo basado en el concepto *unión virtual* evita las operaciones de unión relativamente costosas usadas en el algoritmo original.

Siguiendo una estrategia similar de recorrido del retículo haciendo uso de las clases de equivalencia en [53] se propone el algoritmo *CA* en el cual las listas de identificadores de las transacciones que contienen a las clases de equivalencias son almacenadas de forma compacta en bloques de bits. La intersección de las mismas se realiza usando operadores lógicos sobre los bloques y sobre los índices de los bloques.

Otros algoritmos que poseen estrategias similares a *ECLAT* son *HybridMiner I* e *HybridMiner II* [4].

Generación de reglas de asociación

El método más común de generación de reglas de asociación interesantes fue propuesto en [9]. El mismo consiste en, por cada conjunto frecuente de ítems generar todas las reglas posibles separando el conjunto de ítems en dos subconjuntos disjuntos (Algoritmo 2).

Algoritmo 2: GenRules

Input F: Set of frequent itemsets

Output RA: Set of association rules

$L_1 \leftarrow \{i || \{t \in D | i \in t\}| \geq minSupp\}$

forall $itemset \quad Z \in F$ **do**

  forall $itemset \quad X \subset Z \quad and \quad X \neq \emptyset$ **do**

    if $(Z.support/X.support) \geq minConf$

      $RA = RA \cup \{X \Rightarrow (Z/X)\}$

El número de reglas generadas para un conjunto de transacciones y umbrales de mínimo soporte y mínima confianza dados, puede ser extremadamente grande. Para reducir este problema, en [54, 55] se propone generar sólo un subconjunto de las reglas interesantes (conjunto de *Reglas Representativas*), a partir del cual, mediante un mecanismo de inferencia, pueden ser generadas el resto de las reglas interesantes. Como mecanismo de inferencia es usado el cubrimiento de una regla. El cubrimiento de una regla de asociación $X \Rightarrow Y$ contiene todas las reglas obtenidas a partir de $X \Rightarrow Y$ mediante la eliminación de un subconjunto de los ítems de Y o el movimiento de un subconjunto de ítems de Y a X.

A continuación se definen formalmente el *cubrimiento de una regla* y el conjunto de *Reglas Representativas.* Además tres propiedades asociadas al *cubrimiento de una regla* son enunciadas [54, 55].

Definición 4: El cubrimiento $\boldsymbol{C}$ de una regla de asociación $\boldsymbol{X \Rightarrow Y}$ se define como:

$$C(X \Rightarrow Y) = \{X \cup Z \Rightarrow V | Z, V \subseteq Y \wedge Z \cap V = \emptyset \wedge V \neq \emptyset\}$$

Por ejemplo:

$$C(\{a\} \Rightarrow \{b,c\}) = \{\{a\} \Rightarrow \{b,c\}, \{a\} \Rightarrow \{b\}, \{a\} \Rightarrow \{c\}, \{a,b\} \Rightarrow \{c\}, \{a,c\} \Rightarrow \{b\}\}$$

Definición 5: El conjunto de Reglas Representativas $\boldsymbol{RR}$ se define como el conjunto de reglas de asociación interesantes[2] $(\boldsymbol{AR})$ que no están cubiertas por otras reglas de asociación interesantes:

$$RR = \{r \in AR | \neg \exists r' \in AR, r' \neq r \ \wedge r \in C(r')\}$$

Propiedad 2: Sea $\boldsymbol{r}$ una regla de asociación y $\boldsymbol{AR}$ el conjunto de reglas interesantes generadas para un conjunto de transacciones y umbrales de mínimo soporte y mínima confianza:

a) Si $r' \in C(r)$ entonces $supp(r') \geq supp\ (r)\ y\ conf(r') \geq conf(r)$,
b) Si $r \in AR$ y $r' \in C(r)$ entonces $r' \in AR$,
c) Si $r \in AR$ y $C(r) \subseteq AR$.

En la Tabla 3 se muestran el conjunto de reglas de asociación interesantes (AR) y el conjunto de Reglas Representativas (RR) obtenidas a partir de la colección de transacciones mostrada en la Tabla 2, para $minSupp = 0.6$ y $minConf = 0.70$.

Tabla 2. Ejemplo de colección de transacciones.

No.	Transacciones
1	$\{a,b,c,e\}$
2	$\{a,b,c,d,e\}$
3	$\{a,b,c,e\}$
4	$\{b,c\}$
5	$\{d,e\}$

Tabla 3. Reglas de asociación interesantes y Reglas Representativas obtenidas a partir de la Tabla 2, para $\boldsymbol{minSupp = 0.6}$ y $\boldsymbol{minConf = 0.70}$.

$\boldsymbol{AR}$			$\boldsymbol{RR}$
$\{a\} \Rightarrow \{bc\}$	$\{a\} \Rightarrow \{b\}$	$\{c\} \Rightarrow \{b\}$	$\{a\} \Rightarrow \{bc\}$
$\{b\} \Rightarrow \{ac\}$	$\{b\} \Rightarrow \{a\}$	$\emptyset \Rightarrow \{a\}$	$\{b\} \Rightarrow \{ac\}$
$\{c\} \Rightarrow \{ab\}$	$\{a\} \Rightarrow \{c\}$	$\emptyset \Rightarrow \{b\}$	$\{c\} \Rightarrow \{ab\}$
$\{ab\} \Rightarrow \{c\}$	$\{c\} \Rightarrow \{a\}$	$\emptyset \Rightarrow \{c\}$	$\emptyset \Rightarrow \{bc\}$
$\{ac\} \Rightarrow \{b\}$	$\emptyset \Rightarrow \{bc\}$	$\emptyset \Rightarrow \{d\}$	$\emptyset \Rightarrow \{d\}$
$\{bc\} \Rightarrow \{a\}$	$\{b\} \Rightarrow \{c\}$	$\emptyset \Rightarrow \{e\}$	$\emptyset \Rightarrow \{e\}$

Las Reglas Representativas también fueron estudiadas en [56] bajo el nombre *Base Representativa* del conjunto de reglas interesantes. Otros pares subconjuntos de reglas interesantes con sus propios mecanismos de inferencia han sido definidos [57-59].

[2] En esta definición se consideran también reglas de asociación interesantes a las reglas con soporte y confianza mayores que los umbrales de mínimo soporte y mínima confianza, cuyo antecedente es vacio. Ejemplo: $\emptyset \Rightarrow Y$, tal que, $Y \neq \emptyset$, $sup(\emptyset \Rightarrow Y) \geq minSupp$ y $conf(\emptyset \Rightarrow Y) \geq minConf$.

3.2 Reglas de Asociación en colecciones de datos mezclados

Se dice que $\Omega = \{O_1, \ldots, O_n\}$ es una colección de datos mezclados, si cada objeto de Ω está descrito por un conjunto $R = \{r_1, \ldots, r_m\}$ de atributos numéricos y no numéricos. Cada objeto de Ω se representa por una tupla $(v_{r_1}, \ldots, v_{r_m})$ donde $v_{r_j} \in D_{r_j}$ es el valor asociado al atributo r_j $(1 \leq j < m)$ y D_{r_j} es el dominio del atributo r_j. $O[r]$ denota el valor del atributo r del objeto O.

Un ejemplo de este tipo de colecciones se muestra en la Tabla 4.

Tabla 4. Ejemplo de colección de datos mezclados.

Ω	***Edad***	***Casado***	***Auto***
O_1	23	No	Chevrolet Chevy
O_2	25	Si	Chevrolet Chevy
O_3	29	No	Volkswagen Bora
O_4	34	Si	Ford Fiesta
O_5	38	Si	Ford Fiesta

Tres han sido los enfoques desarrollados para la minería de reglas de asociación en colecciones de datos mezclados, inicialmente el enfoque basado en discretización y el enfoque basado en conjuntos difusos; y más recientemente el enfoque basado en semejanza entre subdescripciones. A continuación se presentan particularidades de cada uno de ellos.

Enfoque basado en discretización

Srikant y Agrawal reportan en [15] el primer algoritmo para el minado de reglas de asociación en colecciones de datos cuyos objetos son descritos por atributos numéricos y no numéricos y las denominaron reglas de asociación cuantitativas.

La solución de estos autores consiste en realizar un particionamiento del dominio de los atributos numéricos y combinar intervalos adyacentes para disminuir la pérdida de información inherente al particionamiento. Luego el problema de minado de reglas de asociación en datos mezclados es transformado al problema de minado de reglas de asociación en datos binarios, haciendo corresponder en el problema binario un atributo por cada valor de cada atributo no numérico, así como por cada intervalo de cada atributo numérico del problema cuantitativo. Finalmente, una variación del algoritmo *Apriori* (Algoritmo 1) es usado para el minado de los conjuntos frecuentes de ítems, a partir de los cuales son generadas las reglas de asociación interesantes mediante el algoritmo *GenRules* (Algoritmo 2).

En este contexto un ejemplo de regla de asociación obtenida a partir de la colección de datos mezclados que se muestra en la Tabla 4 sería:

$$(Edad \in [30{,}39] \wedge Casado = Si) \Rightarrow (Auto = Ford\ Fiesta),$$

con soporte 0.4 y confianza 1.0.

Formalmente este tipo de reglas se conceptualizan de la siguiente manera: Sea Ω una colección de datos mezclados, una regla de asociación es una implicación de la forma $X \Rightarrow Y$, donde X es un conjunto de pares (r, c_r) con $r \in R$ y $c_r \subseteq D_r$, tal que, cada atributo sólo aparece a lo más una vez ya sea en X o en Y.

Análogamente a las reglas de asociación en datos binarios, en este enfoque se dice que una regla de asociación es interesante si su soporte y su confianza son mayores o iguales que los umbrales de soporte y confianza mínimos.

Sea X un conjunto de pares (r, c_r) con $r \in R$ y $c_r \subseteq D_r$ tal que, $\forall (r, c_r) \in X$ y $\forall (r', c_{r'}) \in X$, si $r = r'$ entonces $c_r = c_{r'}$. Se dice que un objeto $O \in \Omega$ soporta a X si $\forall (r, c_r) \in X$, $O[r] \in c_r$.

El soporte de X para una colección de datos mezclados Ω es la fracción de objetos de Ω que soportan a X.

$$supp\ (X) = \frac{|\{O \in \Omega | \forall (r, c_r) \in X, O[r] \in c_r\}|}{|\Omega|} \tag{6}$$

El soporte de una regla $X \Rightarrow Y$ para una colección de datos mezclados Ω es la fracción de objetos de Ω que soportan a $X \cup Y$.

$$supp(X \Rightarrow Y) = supp(X \cup Y) \tag{7}$$

La confianza de una regla $\boldsymbol{X \Rightarrow Y}$ es la fracción de objetos en $\boldsymbol{T}$ que soportando a $\boldsymbol{X}$, también soportan a $\boldsymbol{Y}$.

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} \tag{8}$$

A partir de Srikant y Agrawal varios autores [16, 17, 19-23, 60-62] se han centrado en cómo discretizar los atributos cuantitativos y en cómo reducir el número de reglas interesantes generadas, siempre teniendo en cuenta la relación entre el grado de discretización, el número de reglas, el tiempo de ejecución, y la pérdida de información inherente a la discretización.

En [22], usando técnicas provenientes de la geometría computacional, dos tipos de discretizaciones óptimas del dominio de los atributos numéricos son obtenidos: uno que maximiza el soporte de las reglas manteniendo la confianza de las mismas mayor que un umbral y otro que maximiza la confianza de las reglas manteniendo el soporte de las mismas mayor que un umbral. Otros autores [21], se centraron en obtener una discretización óptima de los atributos numéricos basada en la teoría de la información para minimizar la pérdida de información.

En [16] un enfoque basado en la teoría de la información es usado en el proceso de minado de patrones frecuentes y para esto se construye un grafo que indica las relaciones informativas fuertes, a partir de la información mutua entre los atributos. Los cliques en el grafo son utilizados para podar los conjuntos de atributos no promisorios.

Las técnicas de agrupamiento también han sido usadas para discretizar el dominio de los atributos numéricos. En [20] el algoritmo adaptativo de agrupamiento *BIRCH* [63], es aplicado para identificar intervalos. En [23] se propone un algoritmo de agrupamiento para mejorar el particionamiento. Otro algoritmo, basado en densidad y tipo *Apriori*, es propuesto en [60], para agrupar valores del dominio de los atributos y minar patrones frecuentes.

En [17], otra solución es reportada, en la cual los atributos numéricos no son inmediatamente transformados en atributos binarios y el concepto *Área de Interés* juega un importante papel en la obtención de los patrones frecuentes y las reglas de asociación.

Los algoritmos evolutivos también han sido usados para discretizar implícitamente o explícitamente el dominio de los atributos numéricos [1, 19, 61, 62].

Enfoque basado en conjuntos difusos

Los conjuntos difusos son una alternativa para discretizar el dominio de los atributos numéricos y también el dominio de los atributos no numéricos.

Los conjuntos difusos [64] pueden ser vistos como una generalización de los conjuntos duros (clásicos). A diferencia de los conjuntos duros, a los cuales los elementos pertenecen o no pertenecen, los conjuntos difusos permiten una gradación de la pertenencia de los elementos.

La pertenencia de un elemento del conjunto universo a un conjunto duro se define mediante una función binaria. Si el valor de la función evaluada en el elemento es 1 (*verdadero*), entonces el elemento pertenece al conjunto; si el valor de la función evaluada en el elemento es 0 (*falso*), entonces el elemento no pertenece al conjunto.

La teoría de conjuntos difusos generaliza las funciones de pertenencia, ampliando la imagen de las mismas a un intervalo especificado, típicamente $[0,1]$. En la medida en que el valor de la función evaluada en el elemento esté más cercano a 1, el elemento pertenecerá más al conjunto, mientras que, en la medida en que el valor de la función evaluada en el elemento esté más cercano a 0, el elemento pertenecerá menos al conjunto.

Formalmente, sea U el conjunto universo, un conjunto difuso A se caracteriza por una función de pertenencia $\mu_A: U \rightarrow [0,1]$, tal que, $\forall a \in U$, $\mu_A(a)$ representa el grado de pertenencia del elemento a al conjunto difuso A.

Lee y Kwang reportan en [65] el primer algoritmo para el minado de reglas de asociación usando conjuntos difusos. Se asume que por cada atributo se tienen conjuntos difusos asociados. A partir ellos son obtenidas reglas de asociación como la siguiente: $(Hamburguesa, precio\ medio) \Rightarrow (coka, precio\ bajo)$ donde $precio\ medio$ es uno de los tres conjuntos difusos asociados al precio de la *Hamburguesa* ($precio\ bajo$, $precio\ medio$, $precio\ alto$) y $precio\ bajo$ es uno de los tres conjuntos difusos asociados al precio de la *Coka.* Sin embargo los autores usan un umbral de pertenencia para transformar los conjuntos difusos en conjuntos duros y luego resuelven el problema de minado de reglas de asociación usando un algoritmo de minado de reglas de asociación binarias.

En [30, 66, 67] por cada atributo se tienen conjuntos difusos asociados. Sin embargo los conjuntos difusos no son transformados en conjuntos duros. Las reglas de asociación son obtenidas a partir de las combinaciones de conjuntos difusos asociados a atributos diferentes. El interés de las reglas es el calculado por medio de una medida llamada *diferencia ajustada*.

En [68] los autores se centran en descubrir reglas más comprensibles por los especialistas del área de aplicación y definen el *factor de significación* y el *factor de certeza* (definidos como una extensión del *soporte* y la *confianza*), los cuales son calculados considerando el grado de pertenencia de cada atributo a su correspondiente conjunto difuso.

Sea Ω una colección de datos mezclados. En este contexto una regla de asociación es una implicación de la forma $X \Rightarrow Y$, donde $X = \{A_1, \dots, A_k\}$ es un conjunto de conjuntos difusos, tal que, todo A_i está asociado con un atributo $r_i \in R$, y $\forall A_i, A_j \in X$, si $A_i \neq A_j$ entonces $r_i \neq r_j$; y $Y = \{B_1, \dots, B_l\}$ es un conjunto de conjuntos difusos, tal que, todo B_i está asociado con un atributo $r_i \in R$, y $\forall B_i, B_j \in X$, si $B_i \neq B_j$ entonces $r_i \neq r_j$.

El *factor de significación* de X en Ω se define como:

$$significance(X) = \frac{\sum_{O \in \Omega} T^*(X,O)}{|\Omega|}, \quad (9)$$

$$T^*(\{A_1, \dots, A_k\}, O) = T\left(\mu_{A_1}(O[r_1]), T\left(\mu_{A_2}(O[r_2]), \dots T\left(\mu_{A_{k-1}}(O[r_{k-1}]), \mu_{A_k}(O[r_k])\right) \dots\right)\right) \quad (10)$$

donde $T^*(X, O)$ es el grado de pertenencia de O a la intersección de los conjuntos difusos $A_1, \dots, A_k$ y T es una t-norma[3].

El *factor de significación* de $X \Rightarrow Y$ en Ω se define como:

$$significance(X \Rightarrow Y) = significance(X \cup Y) \quad (11)$$

El *factor de certeza* de $X \Rightarrow Y$ en Ω se define como:

$$certain(X \Rightarrow Y) = \frac{significance(X \cup Y)}{significance(X)} \quad (12)$$

Se dice que una regla de asociación $X \Rightarrow Y$ es interesante si su *factor de significación* y su *factor de certeza* son mayores que umbrales especificados apriori.

Esta misma definición de regla de asociación fue utilizada en [27], pero se denominó *soporte difuso* al *factor de significación* y *confianza difusa* al *factor de certeza.*

En [24] se extiende el enfoque de minado de conjuntos frecuentes ítems, basado en árboles, al contexto de los conjuntos difusos y se presenta una estructura denominada árbol de patrones frecuentes difusos que mantiene la eficiencia de las estructuras para el caso de conjuntos frecuentes de ítems.

Un algoritmo de aprendizaje de reglas de asociación basado en programación lógica inductiva fue presentado en [32]. El algoritmo maximiza la confianza de las reglas tratando de que éstas sean tan informativas como sea posible.

Otros trabajos proponen algoritmos para el minado de reglas de asociación usando taxonomías de conjuntos difusos [28, 31, 69-71].

El enfoque basado en conjuntos difusos, a diferencia de las fronteras duras utilizadas en el esquema de discretización para definir los intervalos, permite definir fronteras difusas, lo cual aumenta la posibilidad de modelar las relaciones entre los valores de los atributos, respecto al enfoque basado en discretización. Sin embargo al igual que en el enfoque en discretización, dos valores cercanos o semejantes pueden pertenecer a conjuntos difusos diferentes.

Enfoque basado en semejanza entre subdescripciones

El concepto de semejanza es comúnmente usado en disciplinas como Medicina, Geología, Sociología, etc., como herramienta para la toma de decisiones. En estos contextos las descripciones de los objetos no tienen que ser las mismas para ser consideradas como iguales. Por ejemplo:

[3] Una $T - norma$ es una función con dominio $[0,1]^2$ e imagen $[0,1]$, la cual es simétrica, asociativa, no decreciente en cada argumento y cumple que $T(x, 1) = x$ para todo $x \in [0,1]$. Ejemplos t-normas son la función mínimo y la función producto.

a) Considerar que dos personas son semejantes en términos de sus edades, si ellas son de la misma generación, lo cual podría ser equivalente a considerar que dos edades son semejantes si el módulo de sus diferencias es a lo sumo 5 años.

b) Considerar que dos personas son semejantes en términos su auto. Los autos compactos podrían ser considerados semejantes a los autos medianos; los autos medianos semejantes a los autos compactos y a los autos grandes; los autos grandes semejantes a los autos medianos y a los autos lujosos; y los autos lujosos semejantes a los medianos.

c) Considerar que dos objetos (partes de objetos) son semejantes, si ellos (sus partes) son semejantes en todos los atributos.

d) Considerar que dos objetos (partes de objetos) son semejantes, si ellos (sus partes) son semejantes en al menos 90% de los atributos.

En un problema real si un especialista de estas áreas emplea en su trabajo diario, una función de semejanza diferente de la igualdad coordenada a coordenada para comparar los objetos de estudio, entonces los algoritmos de minado de reglas de asociación que asumen la igualdad coordenada a coordenada, en el conteo de las ocurrencias de los objetos o partes de ellos, pierden información valiosa, y como consecuencia patrones frecuentes y reglas interesantes.
Por ejemplo para la colección de datos mezclados mostrada en la Tabla 5, si se considera la igualdad coordenada a coordenada, el umbral de mínimo soporte $minSupp = 0.66$ y el umbral de mínima confianza $minConf = 0.9$, entonces se tiene un solo patrón frecuente ($Casado = No$) y no se tiene ninguna regla interesante. Sin embargo si se considera como función de semejanza para el atributo *Edad* la mostrada en el ejemplo anterior inciso a), como función de semejanza para el atributo *Auto* la mostrada en el ejemplo anterior inciso b), como función de semejanza para el atributo *Casado* la igualdad y como función de semejanza entre objetos o partes de ellos la mostrada en el ejemplo anterior inciso c); se tienen los patrones frecuentes y reglas de asociación interesantes mostrados en la Tabla 6. Como puede apreciarse, al usar las funciones de semejanza anteriores, se producen patrones frecuentes y reglas de asociación ocultas para los algoritmos basados en la igualdad coordenada a coordenada.

Tabla 5. Ejemplo de colección de datos mezclados.

Ω	***Edad***	***Auto***	***Casado***
O_1	23	*Compacto*	No
O_2	25	*Grande*	No
O_3	25	*Mediano*	No
O_4	29	*Mediano*	No
O_5	34	*Grande*	Si
O_6	38	*Lujoso*	Si

Tabla 6. Patrones frecuentes y reglas de asociación interesantes obtenidas a partir de la colección mostrada en la Tabla 5, usando función de semejanza diferente de la igualdad coordenada a coordenada.

Patrones frecuentes	***Reglas de asociación interesantes***
$(Edad = 25)$	$(Edad = 25) \Rightarrow (Auto = Mediano)$
$(Edad = 29)$	$(Edad = 29) \Rightarrow (Auto = Mediano)$
$(Auto = Mediano)$	$(Edad = 25) \Rightarrow (Casado = No)$
$(Auto = Grande)$	$(Casado = No) \Rightarrow (Edad = 25)$
$(Casado = No)$	$(Auto = Mediano) \Rightarrow (Casado = No)$
$(Edad = 25, Auto = Mediano)$	$(Edad = 25) \Rightarrow (Auto = Mediano, Casado = No)$
$(Edad = 29, Auto = Mediano)$	$(Edad = 25, Auto = Mediano) \Rightarrow (Casado = No)$
$(Edad = 25, Casado = No)$	$(Edad = 25, Casado = No) \Rightarrow (Auto = Mediano)$
$(Auto = Mediano, Casado = No)$	$(Auto = Mediano, Casado = No) \Rightarrow (Edad = 25)$
$(Edad = 25, Auto = Mediano, Casado = No)$	$(Casado = No) \Rightarrow (Edad = 25, Auto = Mediano)$

En [39], se reportaron los primeros avances en el uso de funciones de semejanza entre objetos completos o partes de ellos en la Minería de Reglas de Asociación. Los autores proponen un algoritmo (*ObjectMiner*), que utiliza las funciones de semejanza en el conteo de las ocurrencias de partes de objetos, extendiendo para ello, el concepto de soporte y confianza. El algoritmo propuesto se centra en la primera fase de la minería de reglas de asociación (obtener los patrones frecuentes) y es efectivo para funciones de semejanza que cumplen que si dos objetos no son semejantes respecto a un conjunto de atributos S_1 entonces tampoco lo son respecto a cualquier conjunto S_2, tal que $S_1 \subset S_2$. Además sólo se consideran funciones de semejanza binaria; la misma para todos los subconjuntos de atributos.

Sea Ω una colección de datos mezclados. Una subdescripción de un objeto $O \in \Omega$ para un subconjunto de atributos $S \subseteq R$ denotada $I_S(O)$, es la proyección de los valores de O en términos de los atributos en S. Nótese que $I_{\{r\}}(O) = O[r]$, $r \in R$. En este contexto una regla de asociación es una implicación de la forma $X \Rightarrow Y$, donde $X = I_{s_1}(O)$ y $Y = I_{s_2}(O)$, tal que, $O \in \Omega$, $S_1 \subseteq R$, $S_2 \subseteq R$, $S_1 \neq \emptyset$, $S_2 \neq \emptyset$y $S_1 \cap S_2 = \emptyset$.
Se dice que una regla de asociación es interesante si su soporte y su confianza son mayores o iguales que los umbrales de soporte y confianza mínimos.

El *soporte* de $I_s(O)$ en Ω se define como:

$$sup(I_s(O)) = \frac{|\{O' \in \Omega | sim(I_S(O), I_S(O'))\}=1|}{|\Omega|}, \tag{13}$$

donde sim es una función de semejanza binaria entre subdescripciones de objetos, la cual está basada en el criterio de comparación binario $C_r: D_r \times D_r \rightarrow [0,1]$ de cada atributo. Un ejemplo de este tipo funciones es la función de semejanza coordenada a coordenada:

$$sim(\mathrm{I}_s(O), \mathrm{I}_s(O')) = \begin{cases} 1 & si\ \forall r \in S, C_r(O[r], O'[r]) = 1 \\ 0 & en\ otro\ caso \end{cases}, \tag{14}$$

donde C_r es criterio de comparación del atributo r. Dos ejemplo de criterios de comparación son:

$$C_{Edad}(x, y) = \begin{cases} 1 & si\ |x - y| \leq 5 \\ 0 & en\ otro\ caso \end{cases} \tag{15}$$

$$C_{Auto}(x, y) = \begin{cases} 1 & si\ (x = y) \vee \\ & \quad (x = Compacto \wedge y = Mediano) \vee \\ & \quad (x = Mediano \wedge (y = Compacto\ \vee\ y = Grande)) \vee \\ & \quad (x = Grande \wedge (y = Mediano\ \vee\ y = Lujoso)) \vee \\ & \quad (x = Lujoso \wedge y = Grande) \\ 0 & en\ otro\ caso \end{cases} \tag{16}$$

Se dice que una subdescripción es frecuente si su soporte es mayor que el umbral de mínimo soporte $minSupp$.

El *soporte* de $X \Rightarrow Y$ en Ω se define como:

$$supp(X \Rightarrow Y) = supp(X \cup Y) \tag{17}$$

La *confianza* de $X \Rightarrow Y$ en Ω se define como:

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} \tag{18}$$

El algoritmo ObjectMiner (inspirado en el algoritmo *A priori)* trabaja como sigue: Primero son determinadas todas las subdescripciones frecuentes para cada atributo (subdescripciones de tamaño 1). Después, en cada iteración k (inicialmente $k = 2$) son combinadas dos a dos las subdescripciones frecuentes de tamaño $k - 1$, para obtener subdescripciones de tamaño k candidatas a frecuentes, mientras hayan sido obtenidas al menos dos subdescripciones frecuentes de tamaño $k - 1$. En este paso por cada par de subdescripciones, los conjuntos de índices de los objetos de la colección que contienen subdescripciones semejantes a una u otra subdescripción del par, son intersecados para crear un conjunto con los índices de los objetos candidatos a ser semejantes a la subdescripción resultante de la combinación de ambas subdescripciones (subdescripción candidata a frecuente). A partir de este conjunto y usando la función de semejanza son determinadas cuáles subdescripciones son realmente semejantes a la candidata a frecuente, así como si la subdescripción es o no frecuente. Una vez obtenidas todas las subdescripciones frecuentes, a partir de ellas son generadas las reglas de asociación interesantes.

La repetición de las subdescripciones es un hecho común en las colecciones. Sin embargo este algoritmo no utiliza el conocimiento que encuentra en cada paso para disminuir el número de operaciones a realizar pasos posteriores. Por esta razón la semejanza entre repeticiones de una misma subdescripción es calculada sin necesidad. Adicionalmente, por cada subdescripción, se almacena el conjunto de índices de los objetos que contienen subdescripciones semejantes. Esto es un problema cuando se tienen colecciones muy grandes de datos.

Otra debilidad de este algoritmo es que no permite funciones de semejanza no binaria, o que no cumplan que si dos objetos no son semejantes respecto a un conjunto de atributos S_1 entonces tampoco lo son respecto a cualquier conjunto S_2, tal que $S_1 \subset S_2$, lo cual restringe su ámbito de aplicación. Un ejemplo simple de función de semejanza que no cumple las condiciones planteadas es considerar que dos objetos (partes de objetos) son semejantes, si ellos (sus partes) son semejantes en al menos 90% de los atributos.

A pesar de las debilidades mencionadas, en [39] se muestra cómo, mediante la incorporación del concepto de semejanza entre subdescripciones de objetos en el conteo de ocurrencias, pueden ser descubiertas reglas de asociación ocultas para los anteriores dos enfoques; abriéndose así un nuevo enfoque de minado de Reglas de Asociación en colecciones de datos mezclados. Sin embargo, a la vez, estas debilidades muestran que queda mucho por hacer en el desarrollo de este enfoque, lo cual motiva la propuesta de investigación que se da a continuación.

4 Propuesta

4.1 Pregunta de Investigación

¿Cómo extraer eficientemente reglas de asociación, en colecciones de datos mezclados, incorporando el concepto de semejanza entre descripciones y subdescripciones de objetos?

4.2 Objetivo general

Diseñar un método para extraer reglas de asociación, en colecciones de datos mezclados, incorporando el concepto de semejanza entre descripciones y subdescripciones de objetos, que permita el uso de funciones de semejanza menos restrictivas que las permitidas por el algoritmo existente.

4.3 Objetivos particulares

- Extender los conceptos de frecuencia y confianza, patrón frecuente y regla de asociación, incorporando el concepto de semejanza entre descripciones y subdescripciones de objetos con datos mezclados.

- Desarrollar algoritmos de búsqueda de patrones similares frecuentes[4] en colecciones de datos mezclados para funciones de semejanza binaria y no binaria, a partir de las extensiones de los conceptos frecuencia y patrón frecuente.

- Definir propiedades de poda del espacio de los posibles patrones similares frecuentes.

- Diseñar algoritmos eficientes de búsqueda de patrones similares frecuentes en colecciones de datos mezclados para funciones de semejanza binaria y no binaria, que incorporen en su estrategia de búsqueda las propiedades de poda encontradas.

- Proponer un algoritmo eficiente de búsqueda de reglas de asociación, a partir de los patrones similares frecuentes encontrados por los algoritmos anteriores.

4.4 Metodología propuesta

Para cumplir con los objetivos particulares propuestos en la sección anterior, se propone la siguiente metodología:

1. Extender los conceptos de frecuencia, confianza, patrón frecuente y regla de asociación, considerando la semejanza entre subdescripciones de objetos con datos mezclados.

    a. Extender el concepto de frecuencia considerando la semejanza entre subdescripciones de objetos al contar las ocurrencias de las subdescripciones.

    b. Extender el concepto de patrón frecuente a partir de la extensión del concepto de frecuencia.

    c. Extender el concepto de confianza a partir de la extensión del concepto de frecuencia.

    d. Extender el concepto de regla de asociación a partir de la extensión de los tres conceptos anteriores.

2. Proponer un algoritmo de búsqueda de patrones similares frecuentes en colecciones de datos mezclados, a partir de las extensiones de los conceptos frecuencia y patrón frecuente, para funciones de semejanza binaria:

    a. Diseño de nuevas estructuras de datos para el almacenamiento y recuperación eficiente de las subdescripciones y sus relaciones de semejanza, que aprovechen las repeticiones de las subdescripciones.

[4] La definición de patrón similar frecuente se muestra a continuación en la sección Resultados Preliminares.

b. Proponer un algoritmo de recorrido exhaustivo sobre el espacio de combinaciones de atributos que haga uso efectivo de la estructura propuesta en la búsqueda de subdescripciones frecuentes.

c. Implementación de la estructura de datos y el algoritmo de recorrido exhaustivo sobre el espacio de combinaciones de atributos.

d. Realización de pruebas y comparación de resultados con algoritmo *ObjectMiner*.

e. Análisis de los resultados obtenidos y retroalimentación.

3. Proponer un algoritmo de búsqueda de patrones similares frecuentes en colecciones de datos mezclados, para funciones de semejanza no binaria.

   Para esto se seguirán pasos análogos al los mostrados en el punto 2.

4. Definir propiedades de poda del espacio de posibles patrones similares frecuentes. Estas propiedades de poda serán definidas de mayor a menor restricción.

   a. Diseño de una propiedad de Clausura Descendente análoga a la propiedad Clausura Descendente del soporte usada en el minado de conjuntos frecuentes de ítems.

   b. Diseño de al menos 2 relajaciones de la propiedad Clausura Descendente.

5. Proponer para cada propiedad de poda, comenzando por las más restrictiva, algoritmos de búsqueda de patrones similares frecuentes en colecciones de datos mezclados, para funciones de semejanza binaria y no binaria que la satisfagan. Para cada algoritmo se seguirán pasos análogos al los mostrados en el punto 2.

6. Proponer un algoritmo de búsqueda de reglas de asociación a partir de los patrones similares frecuentes descubiertos por los algoritmos anteriores.

   a. Extender el algoritmo de búsqueda de reglas de asociación propuesto en [3] o diseñar un nuevo algoritmo, considerando las extensiones de los conceptos de frecuencia, confianza, patrón frecuente y regla de asociación.

   b. Implementación del algoritmo propuesto.

   c. Realización de pruebas y comparación de resultados.

   d. Análisis de los resultados obtenidos y retroalimentación.

4.5 Contribuciones

Las contribuciones esperadas en este trabajo de investigación son:

- Extensiones de los conceptos de frecuencia y confianza, patrón frecuente y regla de asociación, incorporando el concepto de semejanza entre descripciones y subdescripciones de objetos con datos mezclados.

- Definiciones de al menos 3 propiedades de poda del espacio de posibles patrones similares frecuentes.

- Algoritmos de minado patrones similares frecuentes para funciones de semejanza binaria y no binaria, cuando:

  - No cumplen la propiedad de clausura descendente.

  - Cumplen la propiedad de clausura descendente.

  - Cumplen con una relajación de la propiedad Clausura Descendente.

- Un algoritmo para la construcción de reglas de asociación que incorpora el concepto de semejanza entre descripciones y subdescripciones de objetos.

4.6 Cronograma

El calendario de actividades se muestra en la Figura 3.

	Actividad	Trimestres 2008				Trimestres 2009			
		1	2	3	4	1	2	3	4
1	Estudio del estado del arte								
2	Extensión de los conceptos de frecuencia, confianza, patrón frecuente y regla de asociación usados en el minado de patrones frecuentes, incorporando el concepto de semejanza entre descripciones y subdescripciones de objetos.								
3	Definición de propiedades de poda del espacio de posibles patrones similares frecuentes.								
	a Clausura Descendente								
	b Relajación 1 de la propiedad Clausura Descendente								
	c Relajación 2 de la propiedad Clausura Descendente								
	d Búsqueda de otras relajaciones de la propiedad Clausura Descendente								
4	Desarrollo de algoritmo de minado de patrones similares frecuentes con funciones de semejanza binaria:								
	a Que cumplen la clausura descendente								
	b Que no cumplen la clausura descendente								
6	Desarrollo algoritmo de minado de patrones similares frecuentes con funciones de semejanza no binaria:								
	a Que cumplen la clausura descendente								
	b Que no cumplen la clausura descendente								
7	Desarrollo de algoritmos de minado de patrones similares frecuentes con funciones de semejanza binaria que incorporen en su estrategia de búsqueda las propiedades de poda encontradas (relajaciones de la propiedad Clausura Descendente).								
8	Desarrollo de algoritmo de minado de patrones similares frecuentes con funciones de semejanza no binaria que incorporen en su estrategia de búsqueda las propiedades de poda encontradas (relajaciones de la propiedad Clausura Descendente).								
9	Desarrollo de algoritmo de búsqueda de reglas de asociación a partir de los patrones similares frecuentes.								
10	Evaluación experimental de los algoritmos desarrollados								
11	Escritura de artículos.								
12	Redacción de la propuesta de Tesis Doctoral.								
13	Defensa de la propuesta de Tesis Doctoral.								
14	Redacción del documento de Tesis Doctoral.								
15	Defensa de la Tesis Doctoral.								

Figura 3. Calendario de actividades.

5 Resultados Preliminares

El proceso de minado de reglas de asociación consta de dos pasos fundamentales: buscar los patrones frecuentes y a partir de éstos encontrar las reglas de asociación interesantes. Nuestros resultados preliminares se concentran en el primero de estos pasos.

En esta sección extendemos los conceptos de frecuencia, patrón frecuente, confianza, regla de asociación y la propiedad Clausura Descendente del soporte, considerando la semejanza no necesariamente simétrica, ni binaria, entre subdescripciones de objetos. Además se proponen dos algoritmos de minado de patrones similares frecuentes, considerando funciones de semejanza binaria.

5.1 Frecuencia, patrón similar frecuente, regla de asociación y confianza

Sea $\Omega = \{O_1, O_2, \ldots, O_n\}$ una colección de datos mezclados, en la cual cada objeto está descrito por un conjunto de atributos $R = \{r_1, r_2, \ldots, r_m\}$. Cada objeto de Ω se representa por una tupla $(v_1, v_2, \ldots, v_m)$ donde $v_i \in D_i$ $(1 \leq i \leq m)$ y D_i es el dominio del r_i. Una subdescripción de un objeto O para un conjunto de atributos $S \subseteq R$ denotada $I_S(O)$, es la proyección de los valores de O en términos de los atributos en S. $O[r]$ denota el valor de atributo r del objeto O $(r \in R)$.

Cada subconjunto de atributos $S \subseteq R$, $S \neq \emptyset$ tiene asociado una función de semejanza [72] f_S entre las subdescripciones de los objetos de Ω con imagen en $[0,1]$, no necesariamente binaria, ni simétrica. Dadas dos subdescripciones $I_S(O)$, $I_S(O')$, tal que $O, O' \in \Omega$; $f_S(O, O') = 1$ significa que O' es lo más semejante posible a O respecto al conjunto de atributos S; y $f_S(O, O') = 0$ significa que O' es lo menos semejante posible a O respecto al conjunto de atributos S. Algunos ejemplos de funciones de semejanza son:

$$f_S(O, O') = \begin{cases} 1 & si\ \ \forall r \in S, O[r] = O'[r] \\ 0 & en\ otro\ caso \end{cases} \tag{19}$$

$$f_S(O, O') = \begin{cases} 1 & if \quad \prod_{r \in S} C_r(O[r], O'[r]) \geq k \\ 0 & en\ otro\ caso \end{cases} \tag{20}$$

donde $C_r: D_r \times D_r \rightarrow [0,1]$ es un criterio de comparación entre dos valores del atributo r.

$$f_S(O, O') = \frac{\sum_{k=1}^{n} O[k]O'[k]}{\sum_{k=1}^{n} O[k]^2 + \sum_{k=1}^{n} O'[k]^2 - \sum_{k=1}^{n} O[k]O'[k]} \tag{21}$$

Aunque f_S pudiera ser diferente para distintos conjuntos de atributos S, por simplicidad en esta propuesta tomaremos la misma f_S para todo $S \subseteq R$.

Definición 6 (frecuencia de una subdescripción para una función de semejanza): Sea $S \subseteq R$, $S \neq \emptyset$, $O \in \Omega$, y una función de semejanza f_S; definimos la frecuencia de una subdescripción $I_S(O)$ en Ω para f_S como:

$$freq_{f_S,\Omega}(O) = \frac{\sum_{O' \in \Omega} f_S(O, O')}{|\Omega|} \tag{22}$$

Nótese que si f_S fuera la función de semejanza de igualdad coordenada a coordenada, entonces la frecuencia de $I_S(O)$ en Ω para f_S sería la fracción de objetos en Ω que contienen a la subdescripción $I_S(O)$.

Definición 7 (patrón similar frecuente): Decimos que una subdescripción $I_S(O)$ es una subdescripción *f_S-frecuente* o un patrón similar frecuente en Ω si $freq_{f_S,\Omega}(O) \geq minFreq$, donde f_S es una función de semejanza y $minFreq$ es el umbral de mínima frecuencia.

Definición 8 (regla de asociación): Una regla de asociación es una expresión de la forma $X \Rightarrow Y$, donde $X = I_{S_1}(O)$ y $Y = I_{S_2}(O)$, tal que, $O \in \Omega$, $S_1, S_2 \subseteq R$, $S_1 \neq \emptyset$, $S_2 \neq \emptyset$ y $S_1 \cap S_2 = \emptyset$.

Definición 9 (confianza de una regla de asociación para una función de semejanza): Sea f_S una función de semejanza; definimos la confianza de una regla de asociación $X \Rightarrow Y$, donde $X = I_{S_1}(O)$ y $Y = I_{S_2}(O)$, en Ω para f_S como:

$$conf_{f_S,\Omega}(I_{S_1}(O) \Rightarrow I_{S_2}(O)) = \frac{freq_{f_{(S_1 \cup S_2)},\Omega}(O)}{freq_{f_{S_1},\Omega}(O)} \quad (23)$$

Se dice que una regla de asociación $X \Rightarrow Y$, donde $X = I_{S_1}(O)$, $Y = I_{S_2}(O)$ y $S = S_1 \cup S_2$, es interesante si su confianza es mayor o igual que el umbral confianza mínimo y si $I_S(O)$ es una subdescripción f_S-*frecuente*.

5.2 Clausura descendente

La propiedad Clausura Descendente del soporte (Propiedad 1) es usada en la generación de conjuntos frecuentes de ítems para poda del espacio de búsqueda. Esta propiedad asegura que todos los subconjuntos de un conjunto frecuente de ítems también sean conjuntos frecuentes de ítems, y que todos los superconjuntos de un conjunto no frecuente de ítems sean también conjuntos no frecuentes de ítems. El análogo de esta propiedad en nuestro caso es que toda subdescripción de una subdescripción f_S-*frecuente* es una subdescripción f_S-*frecuente*, y que toda supra subdescripción de una subdescripción $no - f_S$-*frecuente* es una subdescripción $no - f_S$-*frecuente.* Sin embargo esta propiedad a diferencia de la anterior, no siempre se cumple y su cumplimento depende de la función de semejanza f_S. A continuación, se presenta formalmente esta propiedad.

Propiedad 3: ***(Clausura Descendente de la semejanza entre subdescripciones):*** Decimos que una función de semejanza f_S cumple la propiedad Clausura Descendente de la semejanza entre subdescripciones si y sólo si para todo $S_1 \subseteq S_2 \subseteq R$; $S_1 \neq \emptyset$; $O, O' \in \Omega$:

$$f_{S_1}(O, O') \geq f_{S_2}(O, O').$$

Propiedad 4: ***(Clausura Descendente de la frecuencia de las subdescripciones):*** Decimos que una función de semejanza f_S cumple la propiedad Clausura Descendente de la frecuencia de las subdescripciones si y sólo si para todo $S_1 \subseteq S_2 \subseteq R$; $S_1 \neq \emptyset$; $O \in \Omega$:

$$\boldsymbol{freq_{f_{S_1},\Omega}(O) \geq freq_{f_{S_2},\Omega}(O)}.$$

Proposición 1: Si f_S cumple la Propiedad 3 entonces f_S cumple también la Propiedad 4.

Demostracion: Si $\boldsymbol{f_S}$ cumple la Propiedad 3 entonces para todo $\boldsymbol{S_1 \subseteq S_2 \subseteq R}$; $\boldsymbol{S_1 \neq \emptyset}$; $\boldsymbol{O \in \Omega}$:

$$\sum_{O' \in \Omega} f_{S_1}(O, O') \geq \sum_{O' \in \Omega} f_{S_2}(O, O')$$
$$\frac{\sum_{O' \in \Omega} f_{S_1}(O,O')}{|\Omega|} \geq \frac{\sum_{O' \in \Omega} f_{S_2}(O,O')}{|\Omega|}$$
$$freq_{f_{S_1},\Omega}(O) \geq freq_{f_{S_2},\Omega}(O)$$

Por tanto, f_S cumple también la Propiedad 4 ■

Propiedad 5: ***(Clausura Descendente):*** Decimos que una función de semejanza f_S cumple la propiedad Clausura Descendente si y sólo si f_S cumple la Propiedad 3, consecuentemente se cumple la Propiedad 4 y por tanto para todo $S_1 \subseteq S_2 \subseteq R$; $S_1 \neq \emptyset$; $O \in \Omega$:

$$(freq_{f_{S_1},\Omega}(O) < minFreq) \Rightarrow (freq_{f_{S_2},\Omega}(O) < minFreq).$$

Un ejemplo de f_S que cumple la propiedad de clausura descendente es la igualdad coordenada a coordenada (ecuación 19). Este caso particular de función de semejanza es el usado en el minado de conjuntos frecuentes de ítems, el cual asegura que todos los subconjuntos de un conjunto frecuente de ítems también sean conjuntos frecuentes de ítems, y que todos los superconjuntos de un conjunto no frecuente de ítems sean también conjuntos no frecuentes de ítems.

Un ejemplo de función de semejanza que no necesariamente cumple la propiedad Clausura Descendente es la función f_S mostrada en la figura 4. Si fijamos $minFreq$ en 0.5 podemos ver en la misma figura que $\exists O = (0,1,1) \in \Omega$, $S_1 = \{r_1, r_2\} \subseteq S_2 = \{r_1, r_2, r_3\}$, tal que, $freq_{f_{S_1},\Omega}(O) = 0.25 < minFreq = 0.5$, y que $freq_{f_{S_2},\Omega}(O) = 0.75 > minFreq = 0.5$.

Ω	r_1	r_2	r_3
O_1	0	0	0
O_2	0	0	1
O_3	0	1	1
O_4	1	1	1

$$f_S(O,O') = \begin{cases} 1 \text{ si } \frac{|\{r \in S \mid O[r]=O'[r]\}|}{|S|} \geq 0.6 \\ 0 \text{ en otro caso} \end{cases}$$

$$freq_{f_{\{r_1,r_2\}},\Omega}((0,1,1)) = 0.25$$

$$freq_{f_{\{r_1,r_2,r_3\}},\Omega}((0,1,1)) = 0.75$$

Figura 4. Ejemplo de función de semejanza f_S que no cumple la propiedad Clausura Descendente.

5.3 Nuevos algoritmos de minado de patrones similares frecuentes

En las definiciones y propiedades presentadas en las secciones 5.1 y 5.2 las funciones de semejanza tienen como imagen el intervalo $[0,1]$. Como resultado preliminar en esta sección presentaremos dos algoritmos de minado de patrones similares frecuentes para funciones de semejanza binaria (con imagen en $\{0,1\}$).

El primer algoritmo que se propone está diseñado para funciones de semejanza que cumplan la propiedad Clausura Descendente y emplea esta propiedad para podar el espacio de posibles patrones similares frecuentes mediante una estrategia de recorrido descendente. También se propone una estructura de datos especial para almacenar las subdescripciones, sus valores de semejanza y otros datos para hacer eficiente el proceso de minado.

El segundo algoritmo que se propone está diseñado para funciones de semejanza que no cumplan la propiedad Clausura Descendente y emplea una variación del algoritmo de construcción de la estructura de datos anterior y una estrategia de recorrido ascendente para aprovechar las repeticiones de las subdescripciones y así reducir el esfuerzo computacional dedicado a contar las ocurrencias de las subdescripciones.

Algoritmo STree-DC Miner

La propiedad Clausura Descendente asegura que no existe subdescripción $no - f_S$*-frecuente* que pueda ser expandida (mediante la adición de un nuevo atributo) a una subdescripción f_S*-frecuente*. Así, dado un orden $\prec$ sobre los atributos en R, cada subconjunto de atributos $S \subseteq R$, puede ser expandido como $\hat{S} = S \cup \{r\}$, tal que $r \in R$ y $\forall r' \in S$, $r' \prec r$, si el número de subdescripciones f_S*-frecuentes* respecto a S es mayor que cero o $S = \emptyset$. Como resultado de todas las posibles

expansiones sucesivas a partir del conjunto vacío obtenemos todas las subdescripciones f_S*-frecuentes.* El algoritmo propuesto llamado *STreeDC-Miner* sigue esta idea.

Para facilitar la búsqueda de todas las subdescripciones f_S*-frecuentes* respecto a cada expansión $\hat{S}$ de un conjunto de atributos S, el algoritmo *STreeDC-Miner* construye una estructura de datos llamada $STree_{\hat{S}}$. Cada estructura $STree_{\hat{S}}$ es un árbol donde cada camino desde la raíz hasta una hoja representa una subdescripción P. En cada hoja es almacenado:

- $P.c_{=}$: Número de ocurrencias de la subdescripción P en la colección.
- $P.c_{\approx}$: Número de ocurrencias de subdescripciones semejantes a P, pero no iguales a P.
- $P.objs$*:* Lista de objetos que contienen a la subdescripción P.
- $P.similars$*:* Lista de subdescripciones de las cuales P es semejante pero no igual.

En la Figura 5(c) se muestra un ejemplo de estructura $STree$ para colección mostrada en la Figura 5(a).

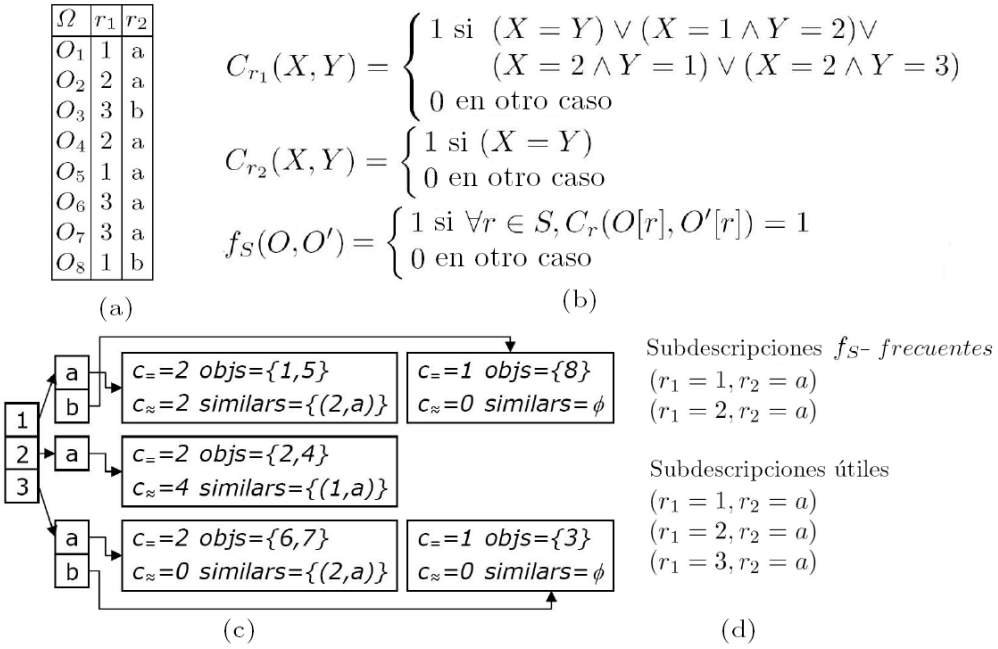

Figura 5. Ejemplo de $\boldsymbol{STree_{\{r_1,r_2\}}}$. (a) Colección $\boldsymbol{\Omega}$. (b) Función de semejanza $\boldsymbol{f_S}$. (c) Estructura de datos $\boldsymbol{STree_{\{r_1,r_2\}}}$. (d) Patrones similares frecuentes y patrones útiles respecto al conjunto de atributos $\{\boldsymbol{r_1}, \boldsymbol{r_2}\}$ para $\boldsymbol{minfreq = 0.5}$.

Nosotros consideramos que una subdescripción $I_S(O)$ es una subdescripción útil si es una subdescripción f_S*-frecuente* o es semejante a una subdescripción $I_S(O')$ f_S*-frecuente.* Si una subdescripción $I_S(O)$ es $no - f_S$ *-frecuente* pero es semejante a una subdescripción $I_S(O')$ f_S*-frecuente* entonces el número de ocurrencias de $I_S(O)$ $(P.c_{=})$ afecta a la frecuencia de $I_S(O')$ $(freq_{f_s,\Omega}(O))$. Si una subdescripción $I_S(O)$ no es una subdescripción útil entonces ninguna expansión de la misma es f_S*-frecuente*, ni afecta la frecuencia de una subdescripción f_S*-frecuente* y

por tanto no es necesario tenerla en cuenta en la construcción de ninguna estructura de datos $STree_{\hat{S}}$, donde $\hat{S}$ es una expansión del conjunto de atributos S.

Para construir cada $STree_{\hat{S}}$ proponemos el Algoritmo 3, el cual consiste en tres pasos:

I) Adicionar a la estructura $STree_{\hat{S}}$ toda subdescripción $I_{\hat{S}}(O)$ contenida en $STree_S$ tal que $I_S(O)$ es una subdescripción útil.

II) Calcular la semejanza entre todas las subdescripciones contenidas en $STree_{\hat{S}}$. Como consecuencia de la propiedad Clausura Descendente y de que las funciones de semejanza permitidas son binarias, la semejanza entre dos subdescripciones $I_{\hat{S}}(O)$ y $I_{\hat{S}}(O')$ tales que $f_S(O,O') = 0$ no es calculada pues $(f_S(O,O') = 0) \Rightarrow (f_{\hat{S}}(O,O') = 0)$.

III) Calcular el número de ocurrencias de subdescripciones semejantes a cada subdescripción contenida $STree_{\hat{S}}$.

Después de construir la estructura $STree_{\hat{S}}$, las subdescripciones f_S-frecuentes son obtenidas verificando la frecuencia de cada subdescripción contenida en la misma.

Algoritmo 3: Build *STree*.

Input $STree_S, \hat{S}, \Gamma, minFreq,$
Output $STree_{\hat{S}}$
$STree_{\hat{S}} \leftarrow$empty STree structure
foreach Subdescription $P \in STree_S \mid P$ is an useful patterns **do**
  foreach Object $O^* \in P.objs$ **do**
    if $\neg STree_{\hat{S}}.contain(I_{\hat{S}}(O^*))$ **then**
      $STree_{\hat{S}}.add(O^*)$
    $STree_{\hat{S}}.I_{\hat{S}}(O^*).c_{=} \leftarrow STree_{\hat{S}}.I_{\hat{S}}(O^*).c_{=} + 1$
foreach $P, P' \in STree_{\hat{S}} \mid I_S(P) \in I_S(P').similars$ **do**
  if $f_{\hat{S}}(P,P') = 1$ **then**
    $P'.similars \leftarrow P'.similars \cup \{P\}$
foreach $P \in STree_{\hat{S}}$ **do**
  foreach $P' \in P.similars \mid P' \neq P$ **do**
    $P'.c_{\approx} \leftarrow P'.c_{\approx} + P.c_{=}$
return $STree_{\hat{S}}$

Algoritmo STree-NDC Miner

Si la función de semejanza f_S no cumple la propiedad Clausura Descendente entonces la poda del espacio de posibles patrones similares frecuentes no es posible. Por tanto es necesario buscar las subdescripciones f_S-frecuentes para todo $S \subseteq R$, $S \neq \emptyset$. Sin embargo, el esfuerzo computacional dedicado a la búsqueda de las subdescripciones frecuentes puede ser reducido usando una estrategia de recorrido ascendente y una variación del algoritmo de construcción de la estructura de datos presentada previamente.
Si una estructura de datos $STree_S$ con $S \neq \emptyset$ es construida adicionando toda subdescripción $I_S(O)$, tal que $O \in \Omega$, calculando la semejanza entre todas las subdescripciones contenidas en $STree_S$, así como el número de ocurrencias de subdescripciones semejantes a cada subdescripción contenida en

$STree_S$, entonces esta estructura contiene toda la información necesaria para construir cualquier estructura $STree_{\check{S}}$, tal que , $\check{S} \subset S,\ S \neq \emptyset$. Por ejemplo a partir de la estructura $STree_{\{r_1,r_2\}}$ mostrada en la Figura 5, la estructura $STree_{\{r_1\}}$ puede ser construida sin necesidad de adicionar las 8 subdescripciones correspondientes a los objetos de la colección, sino sólo las 5 subdescripciones contenidas en $STree_{\{r_1,r_2\}}$. Así, el número de subdescripciones adicionadas se reduce en la medida en que el número de repeticiones de subdescripciones aumenta. Nótese que las listas con los objetos que contienen a cada subdescripción en $STree_{\{r_1,r_2\}}$ no son necesarias para construir $STree_{\{r_1\}}$ y por tanto son eliminadas de las estructuras.

En este caso el algoritmo propuesto es denominado *STreeNDC-Miner*. El mismo, consiste en construir la estructura $STree_R$ adicionando todos los objetos de la colección, y a partir de ésta construir todas sus posibles reducciones. Por cada reducción de $STree_R$ se obtienen sus subdescripciones f_S*-frecuentes.*

Decimos que $\check{S} = S - \{r\}$ es una reducción de S, $S \subseteq R$, $r \in R$, si $\check{S} \neq \emptyset$ y $\forall r' \in (R - S)$, $r' \prec r$. $STree_{\check{S}}$ es una reducción directa de $STree_S$, si $\check{S}$ es una reducción de S, y $STree_{\check{S}}$ es construido a partir de $STree_S$. Finalmente decimos que $STree_{\check{\check{S}}}$ es una reducción de $STree_S$, si $STree_{\check{\check{S}}}$ es un reducción directa de $STree_S$, o existe $STree_{\check{S}}$, tal que, $STree_{\check{S}}$ es una reducción directa de $STree_S$ y $STree_{\check{\check{S}}}$ es una reducción de $STree_{\check{S}}$.

El algoritmo *STreeNDC-Miner* es una solución factible para minar patrones similares frecuentes para colecciones de objetos descritos por un número pequeño de atributos.

En la Tabla 7 se muestran los patrones similares frecuentes encontrados, a partir de la colección mostrada en la Tabla 5, por los algoritmos *STreeNDC-Miner, STreeDC-Miner y ObjectMiner* para el umbral de frecuencia $minFreq = 0.66$; considerando como función de semejanza que dos subdescripciones son semejantes, si éstas son semejantes en al menos 60% de los atributos y como criterios de comparación los mismos utilizados en el ejemplo de la Sección 3.2.3.

Dado que la función de semejanza no cumple la propiedad Clausura descendente los algoritmos *STreeDC-Miner* y *ObjectMiner*, los cuales utilizan esta propiedad para podar el espacio de búsqueda, no encuentran algunos patrones similares frecuentes, a diferencia del algoritmo *STreeNDC-Miner* que sí los encuentra todos. Sin embargo el algoritmo *STreeDC-Miner* encuentra un superconjunto del conjunto de patrones similares frecuentes encontrados por el algoritmo *ObjectMiner.* Esto se debe a que en el algoritmo *STreeDC-Miner* los patrones $no - f_S$*-frecuentes* que son considerados patrones útiles no son podados y sus expansiones pueden resultar en patrones f_S*-frecuentes*.

Tabla 7. Patrones similares frecuentes obtenidos a partir de la colección mostrada en la Tabla 5, usando función de semejanza que no cumple la propiedad Clausura Descendente

STreeNDC-Miner	***STreeDC-Miner***	***ObjectMiner***
$(Edad = 25)$	$(Edad = 25)$	$(Edad = 25)$
$(Edad = 29)$	$(Edad = 29)$	$(Edad = 29)$
$(Auto = Mediano)$	$(Auto = Mediano)$	$(Auto = Mediano)$
$(Auto = Grande)$	$(Auto = Grande)$	$(Auto = Grande)$
$(Casado = No)$	$(Casado = No)$	$(Casado = No)$
$(Edad = 25, Auto = Mediano)$	$(Edad = 25, Auto = Mediano)$	$(Edad = 25, Auto = Mediano)$
$(Edad = 29, Auto = Mediano)$	$(Edad = 29, Auto = Mediano)$	$(Edad = 29, Auto = Mediano)$
$(Edad = 25, Casado = No)$	$(Edad = 25, Casado = No)$	$(Edad = 25, Casado = No)$
$(Auto = Mediano, Casado = No)$	$(Auto = Mediano, Casado = No)$	$(Auto = Mediano, Casado = No)$
$(Edad = 25, Auto = Mediano, Casado = No)$	$(Edad = 25, Auto = Mediano, Casado = No)$	$(Edad = 25, Auto = Mediano, Casado = No)$
$(Edad = 29, Auto = Mediano, Casado = No)$	$(Edad = 29, Auto = Mediano, Casado = No)$	
$(Edad = 23, Auto = Compacto, Casado = No)$		
$(Edad = 25, Auto = Grande, Casado = No)$		

5.4 Resultados experimentales

En esta sección reportamos nuestros resultados experimentales y comparamos los algoritmos *STreeDC-Miner* y *STreeNDC-Miner* contra el algoritmo *ObjectMiner,* el cual es el único algoritmo anterior a los nuestros que permite el uso de funciones diferentes de la igualdad para comparar objetos o partes de objetos. La comparación es hecha en término del tiempo de ejecución y el número de patrones similares frecuentes obtenidos para varios valores de umbrales de mínima frecuencia. La Tabla 8 muestra una descripción de las colecciones de datos[5] usadas en los experimentos.

Tabla 8. Descripción de las colecciones de datos.

Colección	Objetos	Atributos Numéricos	Atributos No Numéricos
Car Evaluation	1728	2	5
Contraceptive Method Choice	1473	2	8
Census	32561	6	9
Poker Hand	1000000	5	6

Como criterios de comparación para los atributos *Edad, Número de Puertas, Número de Personas, Ganancias* y *Pérdidas* es usada la ecuación 24, con $\epsilon = 5, 2, 2, 1000, 1000$ respectivamente, y para el resto de los atributos es usado el criterio de comparación por igualdad.

$$C(x, y) = \begin{cases} 1 & si\ |x - y| \leq \epsilon \\ 0 & en\ otro\ caso \end{cases} \tag{24}$$

El primer experimento (Figura 6) fue realizado usando la función de semejanza f_S mostrada en la ecuación 25, la cual cumple la propiedad Clausura Descendente. Como los 3 algoritmos encuentran los mismos patrones similares frecuentes, sólo se tienen en cuenta sus tiempos de ejecución.

$$f_S(O, O') = \begin{cases} 1 & si\ \ \forall r \in S, C_r(O[r], O'[r]) = 1 \\ 0 & en\ otro\ caso \end{cases} \tag{25}$$

Nótese, que los resultados del algoritmo *STreeNDC-Miner* no fueron mostrados en las figuras 6(c) y 6(d). Esto es debido al alto tiempo de ejecución del algoritmo *STreeNDC-Miner* dado el número de atributos y objetos en las colecciones.

El algoritmo *STreeDC-Miner* logró un mejor desempeño que los otros dos algoritmos (*STreeNDC-Miner y ObjectMiner*) para todas las colecciones. Este desempeño es aun mejor para los umbrales más pequeños de mínima frecuencia. El tiempo de ejecución del algoritmo *STreeDC-Miner* fue menor (7.1, 5.9, 3.9 y 4.2 veces respectivamente) que el tiempo de ejecución del algoritmo *ObjectMiner* para las colecciones de datos usadas.

En el segundo experimento (Figura 7) como función de semejanza f_S fue usada la ecuación 26, con $k = 0.7$. Esta de función no cumple la propiedad Clausura Descendente.

$$f_S(O, O') = \begin{cases} 1 & si\ \ \dfrac{|\{r \in R |\ C_r(O[r], O'[r]) = 1\}|}{|S|} \geq k \\ 0 & en\ otro\ caso \end{cases} \tag{26}$$

[5] http://archive.ics.uci.edu/ml/datasets.html

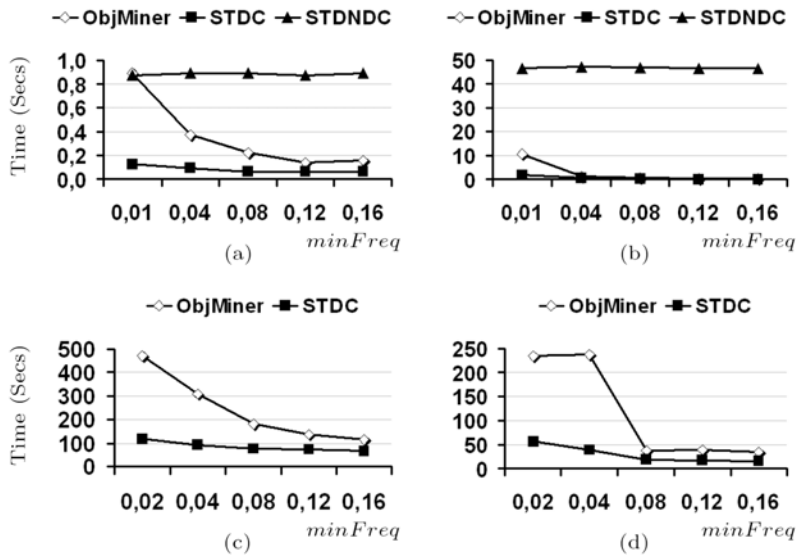

Figura 6. Resultados experimentales usando f_S que cumple la propiedad Clausura Descendente. (a) *Car Evaluation*. (b) *Contraceptive Method Choice.* (c) Census. (d) Poker Hand.

Como en el primer experimento los resultados del algoritmo *STreeNDC-Miner* no fueron mostrados en las figuras 7(c), 7(d), 7(g), 7(k) y 7(l). Sin embargo los tiempos de ejecución de este algoritmo para las colecciones *Car Evaluation* y *Contraceptive Method Choice*, ambas con pocos atributos y objetos, son aceptables.

Es importante destacar que los algoritmos que asumen que f_S cumple la clausura descendente (*ObjectMiner y STreeDC-Miner*) pueden no encontrar todos los patrones similares frecuentes, debido a la poda, a diferencia del algoritmo *STreeNDC-Miner* que encuentra todos los patrones. En nuestros experimentos el conjunto de patrones similares frecuentes encontrados por el algoritmo *ObjectMiner* es un subconjunto de los patrones similares frecuentes encontrados por el algoritmo *STreeDC-Miner.* Esto se debe a que nuestro algoritmo no poda las subdescripciones $no - f_S$*-frecuentes* que son consideradas subdescripciones útiles. Así algunas expansiones de subdescripciones consideradas subdescripciones útiles y $no - f_S$*-frecuentes, pueden ser* subdescripciones f_S*-frecuentes.*

Nótese que el algoritmo *ObjectMiner* pierde hasta 414 708 (80%) patrones similares frecuentes respecto a todos los patrones similares frecuentes existentes y 210 306 (67,1%) patrones similares frecuentes respecto a los patrones similares frecuentes encontrados por *STreeDC-Miner* para la colección de datos *Contraceptive Method Choice* (Figura 7(f)), y pierde hasta 4 023 600 (98,9%) patrones similares frecuentes respecto a los patrones similares frecuentes encontrados por *STreeDC-Miner* para la colección de datos *Census* (Figura 7(g)).

Otro punto relevante es que en la mayoría de los casos el algoritmo *STreeDC-Miner* tiene el mejor rendimiento, en términos de la fracción entre el número de patrones similares frecuentes encontrados y el tiempo de ejecución ($f_S - freq/msecs$).

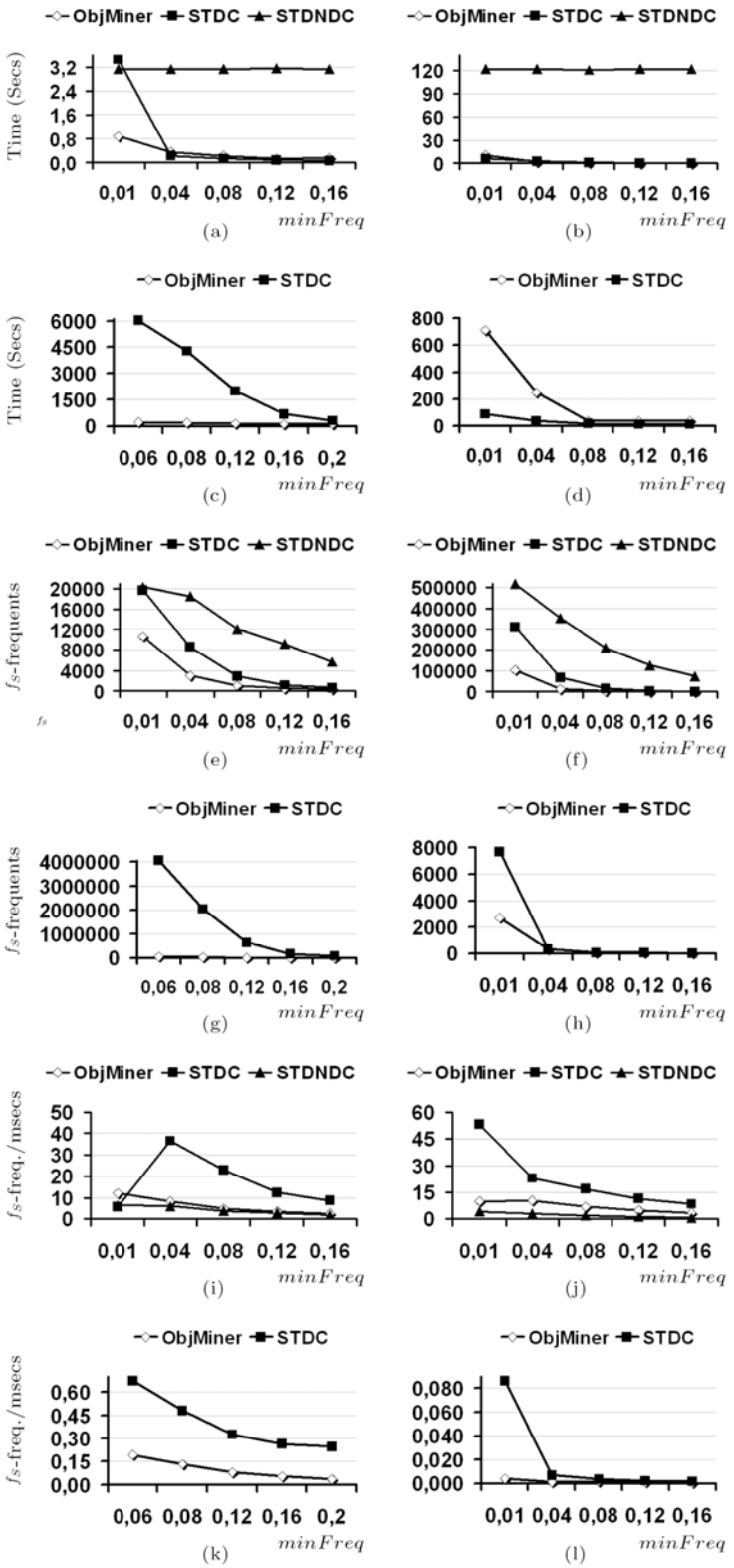

Figura 7. Resultados experimentales usando f_S que no cumple la propiedad Clausura Descendente. (a)(e)(i) *Car Evaluation*. (b)(f)(j) *Contraceptive Method Choice.* (c)(g)(k) Census. (d)(h)(l) Poker Hand.

5.5 Conclusiones

El concepto de semejanza es comúnmente usado en disciplinas como Medicina, Geología, Sociología, etc., como herramienta para la toma de decisiones. El enfoque de minería de reglas de asociación en colecciones de datos mezclados basado en semejanzas, incorpora este conocimiento al proceso de minado de reglas de asociación y permite modelar las relaciones de semejanza entre los objetos y partes de ellos. Con este enfoque pueden ser obtenidos patrones frecuentes y por consiguiente reglas de asociación que no son obtenidas ni por el enfoque basado en discretización, ni por el enfoque basado en conjuntos difusos.

Antes de esta propuesta en el enfoque basado en semejanzas sólo se permitían funciones de semejanza binaria basadas en los criterios de comparación de los atributos (también binarios) para modelar relaciones entre los objetos y partes de ellos. Además la misma función de semejanza debía ser usada para comparar tanto los objetos como sus partes y debía cumplir que si dos objetos no son semejantes respecto a un conjunto de atributos S_1 entonces tampoco lo son respecto a cualquier conjunto S_2, tal que $S_1 \subset S_2$. Estos elementos restringen la posibilidad de modelar otras relaciones de semejanzas y como consecuencia algunos patrones frecuentes y reglas de asociación pueden perderse; por lo cual se hace necesario desarrollar métodos para extraer reglas de asociación, que permitan el uso de funciones de semejanza menos restrictivas que las permitidas actualmente.

Como resultados preliminares de la investigación se extendieron los conceptos frecuencia, patrón frecuente, confianza, regla de asociación y la propiedad Clausura Descendente del soporte, considerando la semejanza no necesariamente simétrica, ni binaria, entre subdescripciones de objetos.

Adicionalmente se propusieron dos algoritmos (*STreeDC-Miner, STreeNDC-Miner*) que se centran en el minado de patrones frecuentes. Ambos algoritmos permiten el uso de funciones de semejanza binaria. El primer algoritmo sólo permite conjuntos de funciones de semejanza que cumplan la propiedad Clausura Descendente, mientras el segundo permite funciones de semejanza que no cumplan dicha propiedad.

Los resultados experimentales muestran que el comportamiento del primer algoritmo en cuanto al tiempo de ejecución es superior al del algoritmo existente (*ObjectMiner*), al emplear funciones que cumplan la propiedad Clausura Descendente; y que aunque este algoritmo no está diseñado explícitamente para funciones de semejanza que no cumplen esta propiedad, al emplear este tipo de funciones, obtiene patrones similares frecuentes imposibles de encontrar por el algoritmo *ObjectMiner*.

En el caso del segundo algoritmo, el cual encuentra todos los patrones similares frecuentes, al emplear funciones de semejanza que no cumplen la propiedad Clausura Descendente los resultados experimentales muestran que es factible para minar patrones similares frecuentes en colecciones de objetos descritos por un número pequeño de atributos.

Ambos algoritmos fueron presentados en el XIII Congreso Iberoamericano de Reconocimiento de Patrones, celebrado en la Habana, Cuba y publicados en las memorias de este evento [73].

Con base en los resultados preliminares podemos concluir que los objetivos planteados son alcanzables siguiendo la metodología propuesta.

Referencias

1. Alatas, B., E. Akin, and A. Karci, *MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules.* Applied Soft Computing 2008. **8**(1): p. 646–656.

2. LaRosa, C., L. Xiong, and K. Mandelberg. *Frequent pattern mining for kernel trace data*. in *2008 ACM Symposium on Applied Computing*. 2008. Fortaleza, Ceara, Brazil: ACM.
3. Han, J., et al., *Frequent pattern mining: current status and future directions.* Data Mining and Knowledge Discovery, 2007. **15**(1): p. 55-86.
4. Kalpana, B. and R. Nadarajan, *Incorporating heuristics for efficient search space pruning in frequent itemset mining strategies.* Current science, 2008. **94**(1): p. 97-101.
5. Lopez, F.J., et al., *Fuzzy association rules for biological data analysis: A case study on yeast.* BMC Bioinformatics, 2008. **9**(107).
6. Zhang, M., et al., *Mining periodic patterns with gap requirement from sequences.* ACM Transactions on Knowledge Discovery from Data, 2007. **1**(2): p. 7.
7. Hu, T., et al., *Discovery of maximum length frequent itemsets.* Information Sciences, 2008. **178**(1): p. 69-87.
8. Agrawal, R., T. Imieliski, and A. Swami. *Mining association rules between sets of items in large databases*. in *1993 ACM SIGMOD International Conference on Management of Data*. 1993. Washington, D.C., USA: ACM.
9. Agrawal, R. and R. Srikant. *Fast Algorithms for Mining Association Rules in Large Databases*. in *Proceedings of the 20th International Conference on Very Large Data Bases*. 1994. Santiago de Chile, Chile: Morgan Kaufmann.
10. Holt, J.D. and S.M. Chung, *Multipass algorithms for mining association rules in text databases.* Knowledge and Information Systems, 2001. **3**(2): p. 168-183.
11. Holt, J.D. and S.M. Chung. *Efficient mining of association rules in text databases*. in *eighth International Conference on Information and Knowledge Management*. 1999. Kansas City, Missouri, USA: ACM.
12. Zaki, M.J., et al., *New Algorithms for Fast Discovery of Association Rules*. 1997, University of Rochester.
13. Park, J.S., M.-S. Chen, and P.S. Yu, *Using a Hash-Based Method with Transaction Trimming for Mining Association Rules.* IEEE Transaction on Knowledge and Data Engineer, 1997. **9**(5): p. 813-825.
14. Savasere, A., E. Omiecinski, and S.B. Navathe. *An Efficient Algorithm for Mining Association Rules in Large Databases*. in *21th International Conference on Very Large Data Bases (VLDB'95)*. 1995. Zurich, Switzerland.: Morgan Kaufmann Publishers Inc.
15. Srikant, R. and R. Agrawal, *Mining quantitative association rules in large relational tables.* SIGMOD Rec., 1996. **25**(2): p. 1-12.
16. K. Ke, J.C., W. Ng, *MIC framework: an information-theoretic approach to quantitative association rule mining* in *ICDE '06*. 2006. p. 112-114.
17. Karel, F. *Quantitative and Ordinal Association Rules Mining (QAR Mining)*. in *10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES 2006)*. 2006. South Coast, UK: Springer, Heidelberg.
18. Lent, B., A. Swami, and J. Widom. *Clustering association rules*. in *IEEE Thirteenth International Conference on Data Engineering*. 1997. Birmingham, UK: IEEE Computer Society.
19. Salleb-Aouissi, A., C. Vrain, and C. Nortet. *QuantMiner: A Genetic Algorithm for Mining Quantitative Association Rules*. in *Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007)*. 2007. Hyderabad, India.
20. Miller, R.J. and Y. Yang. *Association rules over interval data*. in *1997 ACM SIGMOD international conference on Management of Data*. 1997. Tucson, Arizona, USA: ACM.
21. Born, S. and L. Schmidt-Thieme. *Optimal Discretization of Quantitative Attributes for Association Rules*. in *Meeting of the International Federation of Classification Societies (IFCS)*. 2004. Chicago, USA.
22. Fukuda, T., et al. *Mining optimized association rules for numeric attributes*. in *Fifteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of Database Systems*. 1996. Montreal, Quebec, Canada: ACM.
23. Zhang, Z., Y. Lu, and B. Zhang, *An Effective Partitioning-Combining Algorithm for Discovering Quantitative Association Rules*, in *First Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-97)*. 1997.

24. Papadimitriou, S. and S. Mavroudi. *The fuzzy frequent pattern Tree*. in *9th WSEAS International Conference on Computers*. 2005. Athens, Greece: World Scientific and Engineering Academy and Society.
25. Wang, X., C. Borgelt, and R. Kruse. *Fuzzy Frequent Pattern Discovering Based on Recursive Elimination*. in *Fourth International Conference on Machine Learning and Applications (ICMLA '05)*. 2005. Los Angeles, California, USA: IEEE Computer Society.
26. Fu, A.W.-C., et al. *Finding fuzzy sets for the mining of fuzzy association rules for numerical attributes*. in *First International Symposium on Intelligent Data Engineering and Learning (IDEAL'98)*. 1998. Hong Kong, China.
27. Gyenesei, A., *A Fuzzy Approach for Mining Quantitative Association Rules*. 2000, Turku Centre for Computer Science.
28. Tzung-Pei, H., L. Kuei-Ying, and W. Shyue-Liang, *Fuzzy data mining for interesting generalized association rules*. Fuzzy Sets and System, 2003. **138**(2): p. 255-269.
29. Guoqing, C. and W. Qiang, *Fuzzy association rules and the extended mining algorithms.* Inf. Sci. Inf. Comput. Sci., 2002. **147**(1-4): p. 201-228.
30. Au, W.-h. and K.C.C. Chan. *FARM: A Data Mining System for Discovering Fuzzy Association Rules*. in *8th IEEE International Conference on Fuzzy Systems*. 1999. Seoul, Korea.
31. Shitong, W., K.F.L. Chung, and S. Hongbin, *Fuzzy taxonomy, quantitative database and mining generalized association rules.* Intelligent Data Analysis, 2005. **9**(2): p. 207-217.
32. Serrurier, M., et al., *Learning fuzzy rules with their implication operators.* Data and Knowledge Engineer, 2007. **60**(1): p. 71-89.
33. Gómez, J., et al., *Prognostic of Gas-oil deposits in the Cuban ophiological association, applying mathematical modeling.* Geophysics International Journal, 1994. **33**(3): p. 447-467.
34. Cost, S. and S. Salzberg, *A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features.* Machine Learning, 1993. **10**(1): p. 57-78.
35. Rodriguez-Colin, R., J.A. Carrasco-Ochoa, and J.F. Martinez-Trinidad. *Reward-Punishment Editing for Mixed Data*. in *X Iberoamerican Congress on Pattern Recognition (CIARP 2005)*. 2005. Havana, Cuba: Springer, Heidelberg.
36. Hernández-Rodríguez, S., J.F. Martínez-Trinidad, and J.A. Carrasco-Ochoa. *Fast k Most Similar Neighbor Classifier for Mixed Data Based on a Tree Structure*. in *XII Iberoamerican Congress on Pattern Recognition (CIARP 2007)*. 2007. Viña del Mar-Valparaiso, Chile: Springer Heidelberg.
37. Chung-Chian, H., C. Chin-Long, and S. Yu-Wei, *Hierarchical clustering of mixed data based on distance hierarchy.* Information Sciences, 2007. **177**(20): p. 4474-4492.
38. Sánchez-Díaz, G. and J. Ruiz-Shulcloper. *MID Mining: a Logical Combinatorial Pattern. Recognition approach to Clustering in Large Data Sets*. in *V Simposio Iberoamericano de Reconocimiento de Patrones (SIARP 2000)*. 2000. Lisboa, Portugal.
39. Dánger, R., J. Ruiz-Shulcloper, and R. Berlanga. *Objectminer: A New Approach for Mining Complex Objects*. in *6th International Conference on Enterprise Information Systems (ICEIS 2004)*. 2004. Oporto, Portugal.
40. Song, M. and S. Rajasekaran, *A Transaction Mapping Algorithm for Frequent Itemsets Mining.* IEEE Transactions on Knowledge and Data Engineering, 2006. **18**(4): p. 472-481.
41. Pietracaprina, A. and D. Zandolin. *Mining Frequent Itemsets Using Patricia Tries*. in *IEEE ICDM Workshop on Frequent Itemset Mining Implementation (FIMI'03)*. 2003. Melbourne, Florida, USA.
42. Grahne, G. and J. Zhu, *Fast Algorithms for Frequent Itemset Mining Using FP-Trees.* IEEE Trans. on Knowl. and Data Eng., 2005. **17**(10): p. 1347-1362.
43. Gopalan, R.P. and Y.G. Sucahyo. *High Performance Frequent Patterns Extraction using Compressed FP-Tree*. in *SIAM International Workshop on High Performance and Distributed Mining* 2004. Orlando, USA.
44. Han, J., J. Pei, and Y. Yin. *Mining frequent patterns without candidate generation*. in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. 2000. Dallas, Texas, USA: ACM.
45. Holt, J.D. and S.M. Chung, *Mining association rules using inverted hashing and pruning.* Information Processing Letters, 2002. **83**(4): p. 211-220.
46. Mohammed Javeed, Z., P. Srinivasan, and L. Wei, *A localized algorithm for parallel association mining*, in *Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures*. 1997, ACM: Newport, Rhode Island, United States.

47. Sucahyo, Y.G. and R.P. Gopalan. *CT-ITL: efficient frequent item set mining using a compressed prefix tree with pattern growth*. in *14th Australasian Database Conference (ADC 2003)*. 2003. Adelaide, Australia: Australian Computer Society, Inc.
48. Erwin, A., R.P. Gopalan, and N.R. Achuthan. *A bottom-up projection based algorithm for mining high utility itemsets*. in *2nd International Workshop on Integrating Artificial Intelligence and Data Mining (AIDM '07)*. 2007. Gold Coast, Australia: Australian Computer Society.
49. Borgelt, C. *Efficient implementations of apriori and eclat*. in *IEEE ICDM Workshop on Frequent Itemset Mining Implementation (FIMI'03)*. 2003. Melbourne, Florida, USA.
50. Sucahyo, Y.G. and R.P. Gopalan. *CT-PRO: A Bottom-Up Non Recursive Frequent Itemset Mining Algorithm Using Compressed FP-Tree Data Structure*. in *IEEE ICDM Workshop on Frequent Itemset Mining Implementation (FIMI'04)*. 2004. Brighton, UK.
51. Ahmed, S., F. Coenen, and P. Leng, *Tree-based partitioning of data for association rule mining.* Knowledge and Information Systems Journal, 2006. **10**(3): p. 315-331.
52. Kim, M., G.H. Kim, and R.S. Ramakrishna. *A Virtual Join Algorithm for Fast Association Rule Mining*. in *4th International Conference on Intelligent Data Engineering and Learning (IDEAL 2003)*. 2003. Hong Kong, China: Springer, Heidelberg.
53. Hernández-León, R., *Descubrimiento de Conjuntos Frecuentes de Ítems en Datos Estaticos y Dinámicos*. 2008, Instituto Nacional de Astrofísica, Óptica y Electrónica.
54. Kryszkiewicz, M. *Representative Association Rules*. in *Second Pacific-Asia Conference on Research and Development in Knowledge Discovery and Data Mining (PAKDD'98)*. 1998. Melbourne, Australia: Springer, Heidelberg.
55. Kryszkiewicz, M. *Fast Discovery of Representative Association Rules*. in *First International Conference on Rough Sets and Current Trends in Computing*. 1998. Warsaw, Poland: Springer, Heidelberg.
56. Luong, V.P. *The Representative Basis for Association Rules*. in *2001 IEEE International Conference on Data Mining (ICDM'2001)*. 2001. San Jose, California, USA: IEEE Computer Society.
57. Bastide, Y., et al. *Mining Minimal Non-redundant Association Rules Using Frequent Closed Itemsets*. in *Proceedings of the First International Conference on Computational Logic*. 2000. London, UK: Springer, Heidelberg.
58. Stumme, G., et al. *Intelligent Structuring and Reducing of Association Rules with Formal Concept Analysis*. in *Joint German/Austrian Conference on AI: Advances in Artificial Intelligence (KI 2001)*. 2001. Vienna, Austria: Springer Heidelberg.
59. Zaki, M.J. *Generating non-redundant association rules*. in *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2000)*. 2000. Boston, Massachusetts, USA: ACM.
60. Takashi, W., M. Yuki, and M. Hiroshi. *Mining Quantitative Frequent Itemsets Using Adaptive Density-Based Subspace Clustering*. in *Fifth IEEE International Conference on Data Mining (ICDM'05)*. 2005. Houston, Texas, USA: IEEE Computer Society.
61. Mata, J., J.-L.A. Macías, and J.-C.R. Santos. *An evolutionary algorithm to discover numeric association rules*. in *2002 ACM Symposium on Applied Computing (SAC'2002)*. 2002. Madrid, Spain: ACM.
62. Mata, J., J.-L.A. Macías, and J.-C.R. Santos. *Discovering Numeric Association Rules via Evolutionary Algorithm*. in *6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2002)*. 2002. Taipei, Taiwan: Springer, Heidelberg.
63. Zhang, T., R. Ramakrishnan, and M. Livny, *BIRCH: an efficient data clustering method for very large databases.* SIGMOD Rec., 1996. **25**(2): p. 103-114.
64. Zadeh, L.A., *Fuzzy Sets.* Information and Control, 1965. **8**(3): p. 338-353.
65. Lee, J.-H. and H. Lee-Kwang. *An extension of association rules using fuzzy sets*. in *Seventh IFSA World Congress (IFSA'97)*. 1997. Prague, Czech Republic.
66. Chan, K.C.C. and W.-H. Au. *Mining fuzzy association rules*. in *Sixth International Conference on Information and Knowledge Management*. 1997. Las Vegas, Nevada, USA: ACM.
67. Chan, K.C.C. and W.-h. Au. *An Effective Algorithm for Mining Interesting Quantitative Association Rules*. in *12th ACM Symposium on Applied Computing (SAC'97)*. 1997. San Jose, California, USA.
68. Kuok, C.M., A. Fu, and M.H. Wong, *Mining fuzzy association rules in databases.* SIGMOD Rec., 1998. **27**(1): p. 41-46.

69. Chen, G. and Q. Wei, *Fuzzy association rules and the extended mining algorithms.* Information Sciences, 2002. **147**(1-4): p. 201-228.
70. De-Graaf, J.M., W.A. Kosters, and J.J.W. Witteman. *Interesting Fuzzy Association Rules in Quantitative Databases*. in *5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2001)*. 2001. Freiburg, Germany: Springer, Heidelberg.
71. Farzanyar, Z., M. Kangavari, and S. Hashemi. *A New Algorithm for Mining Fuzzy Association Rules in the Large Databases Based on Ontology*. in *Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06)*. 2006. Hong Kong, China: IEEE Computer Society.
72. Martínez-Trinidad, J.F., J. Ruiz-Shulcloper, and M.S. Lazo-Cortés, *Structuralization of universes.* Fuzzy Sets System, 2000. **112**(3): p. 485-500.
73. Rodríguez-González, A.Y., et al. *Mining Frequent Similar Patterns on Mixed Data*. in *13th Iberoamerican Congress on Pattern Recognition (CIARP 2008)*. 2008. La Havana, Cuba: Springer, Heidelberg.