Descubrimiento Incremental de las Reglas de Asociación Temporales

Ji Hae (Sara) Kim
sarasofiakim@hotmail.com
Facultad de Ingeniería - U.B.A.

Juan M. Ale
ale@acm.org
Facultad de Ingeniería - U.B.A.

Resumen

El descubrimiento de las Reglas de Asociación es una de las técnicas de Data Mining. Tiene como objetivo buscar patrones interesantes (reglas) para el soporte en los procesos de tomas de decisiones. Las Reglas de Asociación Temporales son aquellas que tienen asociadas un conjunto de intervalos de tiempo que corresponden a los períodos considerados frecuentes de dicha regla. El manejo del concepto de la "temporalidad" permite considerar no sólo aquellos itemsets frecuentes en todo el período almacenado en la Base de Datos, sino también aquellos itemsets que cumpliendo ciertas condiciones son considerados frecuentes en algunos subintervalos. Así también, permite depurar reglas que contienen items que son obsoletos a un tiempo definido por el usuario.

Puesto que suponemos una Base de Datos que sufre actualizaciones en el tiempo, ya sea por la incorporación de nuevas transacciones, como la eliminación de transacciones obsoletas, es necesario también contar con un mecanismo de actualización de las Reglas de Asociación.

En este trabajo se presenta una técnica para la actualización de las Reglas de Asociación Temporales, que permita optimizar el uso de los recursos y reducir el tiempo de procesamiento de grandes volúmenes de datos, como los usados normalmente para los propósitos de datamining.

Palabras claves: Data Mining, Data Mining Temporal, Reglas de Asociación, Reglas de Asociación Temporales, Mantenimiento de las Reglas de Asociación.

1. INTRODUCCIÓN

En este trabajo se presenta una técnica para el descubrimiento “incremental” y “actualización” de las Reglas de Asociación Temporales. Es una extensión del “Modelo de Reglas de Asociación Temporales basado en Lifespans” publicado en [4].

El problema del descubrimiento de las reglas de asociación temporales surge de la necesidad de descubrir patrones interesantes en los datos transaccionales. Ya que los datos transacciones son temporales, se espera encontrar patrones que dependan del tiempo.

En grandes volúmenes de datos, se pueden encontrar información relacionada a productos que no necesariamente existen a lo largo del período completo de la recolección de datos. Podemos encontrar un producto nuevo que fue introducido luego del comienzo de la recolección, como así también productos que han sido discontinuados antes del final de la misma recolección de datos. Puede ser posible que un nuevo producto no sea incluido en ninguna regla debido a la restricción de soporte, pero que tienen soporte relativo a su lifespan (o period of life) mayor al soporte mínimo propuesto. Por tanto, estos nuevos productos podrían aparecer en las reglas de asociación interesantes y potencialmente útiles. Además deberíamos considerar los casos de productos que pueden ser frecuentes sólo en algunos subintervalos contenidos estrictamente en sus períodos de vida, pero no en todo el intervalo entero de su lifespan.

Podemos solucionar este problema incorporando el tiempo en el modelo de descubrimiento de las reglas de asociación. Y llamamos a estas nuevas reglas como las “Reglas Generales de Asociación Temporales”. Un derivado de esta idea es la posibilidad de la eliminación de reglas desactualizadas, de acuerdo a los criterios del usuario. Además es posible la eliminación de

conjuntos de items obsoletos como una función de sus lifestime, reduciendo la cantidad de trabajo que debe ser realizado en la determinación de los items frecuentes y por lo tanto, en la determinación de las reglas.

Las **reglas de asociación temporales** introducidas en **[3]** y **[4]** son una extensión del modelo no temporal. La idea básica es limitar la búsqueda de conjuntos frecuentes de items o itemsets al lifetime de los miembros que integran el itemset. Por otro lado, para evitar considerar frecuente un itemset con un período de vida (period of life) muy corto, por ejemplo un item que se vende solo una vez, se incorpora el concepto de soporte temporal.

De este modo, cada regla tiene un **time frame** asociado, correspondiente al lifetime de los items participantes en la regla. *Si la extensión del lifetime de la regla excede al mínimo estipulado por el usuario, analizamos si la regla es frecuente en ese período.* Este concepto permite encontrar reglas que, con el punto de vista tradicional de frecuencia, no hubiera sido posible descubrir.

Este trabajo tiene el mismo objetivo que los trabajos como [5], [12], [13] ya que todos tienen proponen alguna técnica de actualización de las reglas. En [5] y [12] los autores proponen una técnica de actualización de las reglas de asociación en el caso en que se agregan nuevas transacciones a la base de datos. En [13] se amplía el trabajo último ([12]) contemplando los casos de eliminación y modificación de transacciones de la base de datos. Nuestra propuesta difiere de las anteriores ya que contempla el factor "tiempo". Además a diferencia de [5], no se vuelve a reprocesar la base de datos de la cual ya se extrajeron las reglas anteriormente.

La contribución de este trabajo para la actualización de las Reglas de Asociación es que propone procesar solo las nuevas transacciones incorporadas a la Base de Datos. De esta manera se evita el tiempo de procesar transacciones ya existentes y anteriormente procesadas.

Las nuevas transacciones incorporadas a la Base de Datos, pueden generar eventualmente nuevas Reglas de Asociación. A fin de cumplir con la propiedad de maximalidad, es necesario evaluar si el lifespan de las nuevas Reglas de Asociación se puede extender sobre la Base de Datos existente. Del mismo modo, se deberá evaluar si el lifespan de las Reglas de Asociación preexistentes, pueden extenderse sobre la Base de Datos nueva.

En el caso de la depuración de la Base de Datos, es necesario también depurar las Reglas asociadas.

El siguiente trabajo está organizado de la siguiente forma: en la sección 2 se describe el Modelo General Temporal, sus definiciones y el proceso para el descubrimiento de las Reglas Generales Temporales. En las seccion 3 y 4 se presenta el modelo y el algoritmo para la actualización de las reglas. Finalmente la sección 5 se presenta la conclusión y el trabajo futuro.

2. MODELO GENERAL TEMPORAL

A efectos de que este trabajo sea autocontenido se reproducen algunos conceptos del "Modelo de Reglas de Asociación Temporales basado en Lifespans" publicado en [4].

Definiciones

Llamamos **d** a la base de datos de transacciones temporalmente ordenado. **d** es una colección de subconjuntos de **R**, siendo $\mathbf{R = \{A_1, \ldots, A_p\}}$, donde $\mathbf{A_i}$ son llamados items.
Cada transacción **s** en **d** es un set de items tal que $\mathbf{s \subseteq R}$.
Asociado a **s** tenemos el timestamp $\mathbf{t_s}$ el cual representa el tiempo válido de la transacción **s**.

El *lifespan (o period of life) de* A_i ($\mathbf{l_{Ai}}$) es el tiempo sobre el que éste aparece en la base de datos de transacciones **d**. Todos los items tienen el lifespan en la base de datos, el cual explicitamente representa la duración temporal de la información de un item, esto es, el tiempo en el cual el item es

relevante al usuario. El $\mathbf{l_{Ai}}$ es dado por el intervalo $\mathbf{[A_i.t_1,A_i.t_2]}$ o $[t_1,t_2]$, con $t_1 \leq t_2$, donde t_1 es el timestamp de la primera transacción en d que contiene A_i, y t_2 es el timestamp de la última transacción en d que contiene A_i.
Llamamos **X** al conjunto de items (itemset), siendo que $\mathbf{X \subseteq R}$.
El conjunto de transacciones en **d** que contiene **X** es indicado por: $\mathbf{V(X) = \{s / s \in d \wedge X \subseteq s\}}$. Si la cardinalidad de **X** es **k**, **X** es llamado **k-itemset**.
Podemos estimar el lifespan de un **k-itemset X** ($\mathbf{l_X}$), con $\mathbf{k} > 1$, por **[t,t']** donde:
$t = \max \{t_1 / [t_1,t_2]$ es el lifespan de un item A_i en X$\}$ y
$t' = \min \{t_2 / [t_1,t_2]$ es el lifespan de un item A_i en X$\}$

Entonces, $\mathbf{d_{lx}}$ es el subconjunto de transacciones de **d** con sus timestamps $\mathbf{t_i \in l_X}$.
Con $\mathbf{|d_{lx}|}$ indicamos el número de transacciones de $\mathbf{d_{lx}}$.
La incorporación del tiempo permite determinar si un itemset es frecuente, hallando la razón entre el número de transacciones que contiene el itemset y el número de transacciones de la base de datos.

El *soporte temporal de X* es la amplitud del lifespan de ***X***, llamado $\mathbf{|l_X|}$.
También definimos el *umbral (threshold) para el soporte temporal* (τ): si $\mathbf{l_d}$ es el lifespan de la base de datos y $\mathbf{|l_d|}$ es su duración, entonces el umbral del soporte temporal τ es una fracción de $\mathbf{|l_d|}$.

Por otro lado, el usuario podría especificar el instante temporal $\mathbf{t_0}$, tal que cualquier item cuyo lifespan es $[t_1,t_2]$ y $\mathbf{t_2<t_0}$ es considerado obsoleto.

En ciertos caso, un itemset puede no ser frecuente en el intervalo correspondiente a su período de vida entero, pero sí en uno o más subintervalos de su período. Entonces, este podría participar en interesantes reglas en esas porciones de su lifespan. Estos subintervalos no dependen estrictamente de los datos, pero en los parámetros, es decir, el umbral para el *soporte temporal* τ y el *soporte mínimo* σ, provistos por el usuario.
No estamos interesados en todos los posibles subintervalos, sino solo en aquellos que son maximales con respecto al soporte temporal y son frecuentes en su frame de tiempo (time frame). ($\mathbf{s(X,[t,t'])} \geq \sigma$ y $\mathbf{|[t,t']|} \geq \tau$, y **[t,t']** es maximal en $\mathbf{l_X}$)

El *lifespan frecuente de X* ($\mathbf{fl_X}$) es el conjunto de subintervalos, contenidos en su lifespan, en los cuales el itemset es frecuente ($\mathbf{|[t,t']|} \geq \tau$).

El lifespan $\mathbf{l_X}$ de k-itemset **X**, donde **X** es la unión de (k-1)-itemsets ***V*** y ***W*** con lifespans $\mathbf{l_V}$ y $\mathbf{l_W}$, respectivamente, es dado por $\mathbf{l_X = l_V \cap l_W}$. Lo mismo no siempre es verdadero para lifespans frecuentes, debido a las restricciones de frecuencia mínima.

El *soporte de **X** en **d** sobre su lifespan* $\boldsymbol{l_X}$ ($\mathbf{s(X,l_X)}$), es el conjunto de fracciones de transacciones en **d** que contiene **X** en todo intervalo maximal correspondiente a $\mathbf{l_X}$. Para cada subintervalo **[t,t']** registramos el soporte como $\mathbf{|V(X,[t,t'])| / |d_{[t,t']}|}$. Dado un umbral de soporte $\sigma \in [0,1]$ en $\mathbf{l_X}$ tal que $\mathbf{s(X,[t,t'])} \geq \sigma$, en este caso, se dice que X tiene soporte mínimo en $\mathbf{l_X}$ (minimum support).

En algunos casos $\mathbf{fl_X}$ contiene un solo subintervalo, y esto puede ser igual o más pequeño que $\mathbf{l_X}$.

Teorema: *Dado un itemset **X** no frecuente en su lifespan $\boldsymbol{l_X}$ entonces no existe ningún itemset **Y** frecuente en su lifespan $\boldsymbol{l_Y}$, con $\boldsymbol{X \subset Y}$.*

La *Regla General de Asociación Temporal* para **d** es una expresión de la forma $\mathbf{X \Rightarrow Y: fl_{X \cup Y}}$, donde $\mathbf{X \subseteq R}$, $\mathbf{Y \subseteq R \setminus X}$, y $\mathbf{fl_{X \cup Y}}$ es el lifespan frecuente de $\mathbf{X \cup Y}$, en la granularidad determinada por el usuario.

La *confianza de una regla* $\mathbf{X \Rightarrow Y: fl_{X \cup Y}}$, denotado por $\boldsymbol{conf}\mathbf{(X \Rightarrow Y: fl_{X \cup Y})}$, es la probabilidad condicional que una transacción de **d**, elegida al azar en el lifespan frecuente $\mathbf{X \Rightarrow Y: fl_{X \cup Y}}$, que contiene **X** contenga también a **Y**.
La probabilidad condicional varía de acuerdo al subintervalo considerado dentro de $\mathbf{X \Rightarrow Y: fl_{X \cup Y}}$. Entonces esto se expresa como: $\boldsymbol{conf}\mathbf{(X \Rightarrow Y: fl_{X \cup Y}) = \{s(X \cup Y,[t,t']) / s(X,[t,t']) \mid [t,t']) \subseteq fl_{X \cup Y}\}}$
donde
$\mathbf{fl_{X \cup Y} = \{[t_1,t_2]) / \mid [t_1,t_2] \mid \geq \tau}$ y $\mathbf{s(X \cup Y,[t_1,t_2]) \geq \sigma}$ y $\mathbf{\neg \exists\, [t_j,t_k]([t_1,t_2] \subset [t_j,t_k]}$ y $\mathbf{s(X \cup Y,[t_j,t_k]) \geq \sigma)\}}$

La *regla general de asociación temporal* $\mathbf{X \Rightarrow Y: fl_{X \cup Y}}$ se sostiene en **d** con soporte $\mathbf{s_1, \ldots, s_p}$, soporte temporal $\mathbf{| fl_{X \cup Y} |}$ y la confianza $\mathbf{c_1, \ldots c_p}$, si $\mathbf{s_1\%, \ldots, s_p\%}$ de las transacciones de **d** en $\mathbf{fl_{X \cup Y} = \{subintervalo_1, \ldots, subintervalo_p\}}$ contiene $\mathbf{X \cup Y}$ y $\mathbf{c\%_1, \ldots, c\%_p}$ de transacciones de **d** que contiene **X** también contiene **Y**, en el conjunto de frames de tiempo **[t,t']** tal que $\mathbf{[t,t'] \subseteq fl_{X \cup Y}}$.

Descubrimiento de las Reglas Generales Temporales

El descubrimiento de todas las reglas de asociación en un conjunto de transacciones **d** puede ser realizado en 2 fases:

Fase 1T: Descubrimiento de los Itemsets frecuentes
Halla cada itemset $\mathbf{X \subseteq R}$ tal que **X** es frecuente en su lifespan $\mathbf{l_x}$, esto es, $\mathbf{s(X,[t,t']) \geq \sigma}$, $\mathbf{| [t,t'] | \geq \tau}$, y **[t,t']** maximal para **[t,t']** en $\mathbf{l_X}$, con $\mathbf{fl_X \neq \varnothing}$.

Fase 2T: Generación de Reglas Generales Temporales
Utiliza los itemsets frecuentes **X** para hallar las reglas. En el caso general, es necesario encontrar todos los subconjuntos para cada itemset frecuente. Verifica para cada $\mathbf{Y \subset X}$, con $\mathbf{Y \neq \varnothing}$, si la regla $\mathbf{Y \Rightarrow (X-Y): fl_{X \cup Y}}$, tal que $\mathbf{conf(Y \Rightarrow (X-Y)) \geq \theta}$ en el intervalo **[t,t]** para todos los **[t,t']** en $\mathbf{fl_{X \cup Y}}$.

Algoritmo Temporal para descubrir Itemsets frecuentes
Input: **d** (con transacciones con timestamp y ordenadas ascendentemente según el tiempo), σ, τ.
Output: conjunto de todos los itemsets frecuentes en sus lifespan frecuentes.
Como en la notación original, $\mathbf{L_k}$, representa el conjunto de k-itemsets frecuentes. Cada miembro de este conjunto tendrá asociado los siguientes campos:

- identificación del itemset.
- límite inferior y superior del tiempo de vida (lifetime) del item ($\mathbf{[t_1, t_2]}$).
- cuenta de soporte (**Fr**) del itemset en $\mathbf{[t_1, t_2]}$.
- número total de transacciones (**FTr**) encontrado en el intervalo $\mathbf{[t_1, t_2]}$.
- vector de contadores. Cada contador está asignado a un intervalo $\mathbf{[t_1+j.\Delta t, t_1+(j-1).\Delta t]}$, con **j**=0,1,... Este array va a mantener un histograma (*) con el número de transacciones conteniendo el itemset en cada intervalo. Δ**t** es definido por el usuario y puede ser expresado en diferentes granularidades.

$\mathbf{C_k}$ es el conjunto de k-itemsets candidatos (itemsets potencialmente frecuentes) que tienen asociados la misma información que los miembros de $\mathbf{L_k}$.

Podría existir algunos itemset no frecuentes en todo el lifespan entero pero en algunos intervalos. Para estos casos usamos el algoritmo descripto en la sección siguiente para hallar los intervalos maximales contenidos en sus lifespan, y generar nuevos candidatos de ellos.
Algunos items podrían ser eliminados de $\mathbf{L_1}$ porque se vuelven obsoletos, es decir, tienen lifespan de intervalos $\mathbf{[t_1,t_2]}$ y $\mathbf{t_2<t_0}$. Luego de la eliminación de items obsoletos, el siguiente lema nos asegura que no es necesario chequear los k-itemsets obsoletos con k>1.

Lema: Un k-itemset con k>1, es obsoleto si y sólo si contienen un item obsoleto.

(*) Se asocia a cada itemset un **histograma** el cual va a ser construido durante el proceso de búsqueda de itemsets frecuentes. En este histograma se acumula el número de transacciones en los cuales el itemset aparece en su período de vida entero. La amplitud del intervalo para los histogramas, llamado unidad de tiempo (time unit), es provista por el usuario. Se utiliza un algoritmo para descubrir todos los itemsets frecuentes, excepto aquellos no frecuentes en su período de tiempo entero.
La tarea de acumulación del número de transacciones conteniendo el itemset deseado en los diferentes puntos de tiempo produce una secuencia que puede ser definida por un conjunto de pares: $\mathbf{\{(v_1,f_1),...,(v_p,f_p)\}}$ donde $\mathbf{v_1=t_1}$, $\mathbf{v_2=v_1+\Delta t,...,}$ $\mathbf{v_p+\Delta t=t_2,}$ y $\mathbf{f_i}$ es el número de transacciones conteniendo a X en el período entre $\mathbf{v_i}$ y $\mathbf{v_i+\Delta t}$. Entonces, consideramos $\mathbf{F_x=f_1, f_2, ..., f_p}$ la serie de tiempo asociada con el itemset **X**.

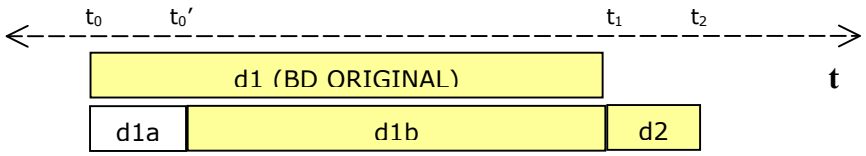
Para los itemsets no frecuentes, se emplea otro algoritmo para hallar los intervalos frecuentes maximales que están estrictamente contenidas en sus períodos de vida.

Algoritmo para hallar los intervalos maximales contenidos en $\mathbf{l_X}$ para el itemset **X** ($\mathbf{lf_X}$):
Entrada: **Vector S** con la información acumulada en el histograma, **vector Q** con el total de transacciones para cada unidad de tiempo, σ y τ.
Salida: Conjunto de intervalos frecuentes maximales ($\mathbf{lf_X}$). Este conjunto podría estar vacío en el caso en que dicho itemset no es frecuente para nada.
Método: La búsqueda sobre **S** se realiza en dirección hacia atrás. Si encontramos un intervalo satisfactorio, marcamos el fin del intervalo, y buscamos otra vez desde el final hacia la marca; sino, avanzamos la marca una unidad y repetimos el proceso.

El resultado de este modelo es la obtención de todos los itemsets frecuentes más sus series de tiempo.

3. DESCUBRIMIENTO INCREMENTAL

Supongamos que tenemos inicialmente la Base de Datos transaccional d_1 con su lifespan $l_{d1} = [t_0,t_1]$ y hemos hallado las reglas de asociación temporales obtenidas a través de los algoritmos vistos en la sección anterior.
Luego de un tiempo, se desea obtener las reglas de la Base de Datos actualizada. Siendo la Base de Datos actualizada: $d_{1b} \cup d_2$, dicha actualización de la Base de Transacciones cuenta con nuevo conjunto de transacciones d_2 y con la eliminación de las transacciones anteriores a t_0' definido por el usuario (d_{1a}).
Dicho conjunto nuevo de transacciones debiera ser temporalmente contiguo en t_1 al conjunto de transacciones existentes d_1 ($l_{d1} = [t_0,t_1]$ y $l_{d1b \cup d2} = [t_1,t_2]$).



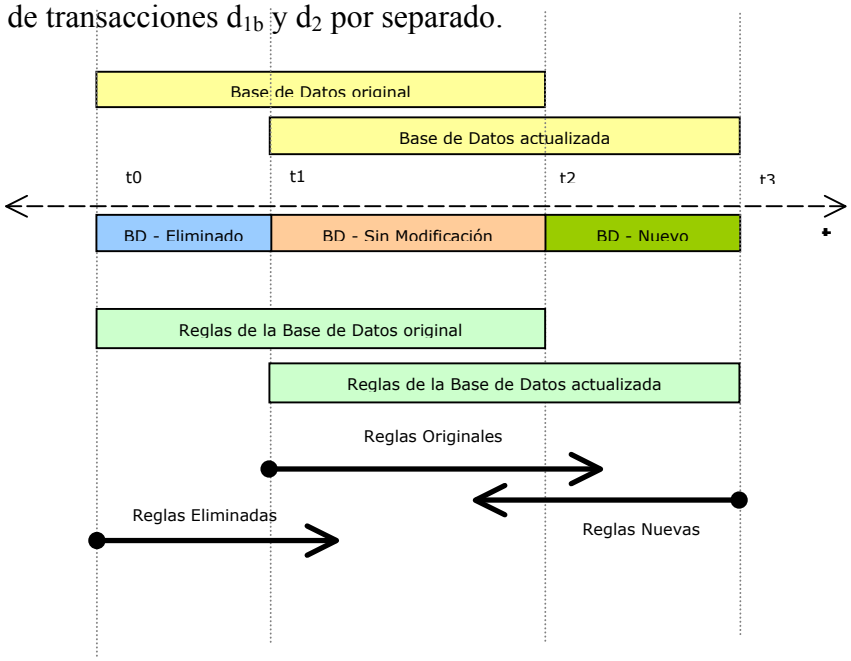
Para la actualización de las Reglas de Asociación, se propone procesar solo el nuevo conjunto de transacciones incorporadas a la Base de Datos (d_2) con los Algoritmos propuestos en el Modelo General Temporal.
A partir de la aplicación de la técnica que se detallará más adelante, se espera obtener el mismo conjunto de reglas de asociación como las que hubiéramos obtenido al procesar nuevamente todas las transacciones de la Base de Datos actualizada $d_{1b} \cup d_2$ habiendo sólo procesado la base de datos d_2. De esta manera se busca ahorrar el tiempo en procesar transacciones procesadas anteriormente y optimizar el uso de los recursos.
Las nuevas transacciones incorporadas a la Base de Datos, pueden generar eventualmente nuevas Reglas de Asociación. A fin de cumplir con la propiedad de maximalidad, es necesario evaluar si el lifespan de las nuevas Reglas de Asociación se puede extender sobre la Base de Datos existente.
Del mismo modo, se deberá evaluar si el lifespan de las Reglas de Asociación preexistentes, pueden extenderse sobre la Base de Datos nueva.
En el caso de la depuración de la Base de Datos, es necesario también depurar las Reglas asociadas.

Casos que se pueden presentar al eliminar el conjunto de transacciones de la Base de Datos del intervalo [t_0,t_0']:

- Las reglas que tienen algún subintervalo frecuente [t_a, t_b] con $t_b < t_0$', dicho subintervalo se elimina del conjunto de Reglas.
- Las reglas que tienen algún subintervalo [t_a, t_b], tal que $t_a < t_0' < t_b$, puede ser que dicho intervalo permanezca o no en el conjunto de Reglas.

Casos que se pueden presentar al agregar un conjunto de transacciones a la Base de Datos [t_1,t_2]:

- Surgen nuevas reglas en d_2.
- Algunas reglas de d_{1b}, pueden extender algún subintervalo en **[t_1,t_2]**.
- Algunas reglas de d_2, pueden extender algún subintervalo en **[t_0',t_1]**.
- Surgen nuevas reglas de la unión de d_{1b} y d_2, que no se hallaron en el proceso de análisis del conjunto de transacciones d_{1b} y d_2 por separado.



A partir del procesamiento de la Base de Datos original (d_1) y el nuevo conjunto de datos transaccionales (d_2) tenemos almacenadas las siguientes informaciones disponibles para el análisis. Tanto para d_1 como para d_2 se cuenta con:

- Conjunto de todos los itemsets frecuentes con sus subintervalos frecuentes (L_k).
- Cada miembro del conjunto (itemset) tendrá asociado los siguientes campos:
    - Identificación del itemset
    - Límite inferior y superior del lifespan del item (**$[t_1, t_2]$**).
    - cuenta de soporte (**Fr**) del itemset en **$[t_1, t_2]$.**
    - número total de transacciones (**FTr**) encontrado en el intervalo **$[t_1, t_2]$.**
    - vector de contadores (vector S). Cada contador está asignado a un intervalo **$[t_1+j.\Delta t, t_1+(j-1).\Delta t]$**, con **j**=0,1,... Este array va a mantener un histograma con el número de transacciones conteniendo el itemset en cada intervalo. **Δt** es definido por el usuario y puede ser expresado en diferentes granularidades.
- Conjunto de conjunto de k-itemsets candidatos $\mathbf{C_k}$.
- Vector Q: contiene el total de transacciones para cada unidad de tiempo.

Con el cociente de **Fr** y **FTr** obtenemos el valor de la frecuencia: s(X, [t, t']).

4. ALGORITMO PARA EL DESCUBRIMIENTO INCREMENTAL

Paso 1: Eliminar los subintervalos frecuentes $[t_a, t_b]$ de las reglas tal que $t_b < t_0'$.
Paso 2: Procesar todo d_2 y obtener las reglas de d_2.
Paso 3: Análisis de la validez de las reglas que tienen algún subintervalo frecuente $[t_a, t_b]$, tal que $t_a < t_0' < t_b$ (en la Base de Datos resultante). (Sufrió corte por izquierda).
Paso 4: Análisis de la validez de las reglas de d_{1b}, en d_2.
Paso 5: Análisis de la validez de las reglas obtenidas en d_2, en d_{1b}.
Paso 6: Análisis de la existencia de alguna regla, que no haya podido identificarse ni en d_{1b}, tampoco en d_2; pero que una vez unificado dichos intervalos, pueda obtenerse alguna regla.

Casos que se presentan en los pasos de ejecución:

Paso 3
Análisis de la validez de las reglas que tienen algún subintervalo de su lifespan $[t_a, t_b]$, tal que $t_a < t_0' < t_b$

Algoritmo:
Si $t_a = t_0'$, entonces, no pasa nada. Dicho subintervalo del k-itemset frecuente permanece.
Si $t_b = t_0'$, entonces, dicho subintervalo del k-itemset frecuente se elimina de la Base.
Si $t_a < t_0' < t_b$, entonces, analizamos el $[t_0', t_b]$.

$[t_0', t_b]$ puede entrar en uno de los siguientes grupos:
Grupo 1) $s(X, [t_0',t_b]) \geq \sigma \wedge |[t_0',t_b]| < \tau$
Grupo 2) $s(X, [t_0',t_b]) \geq \sigma \wedge |[t_0',t_b]| \geq \tau$
Grupo 3) $s(X, [t_0',t_b]) < \sigma \wedge |[t_0',t_b]| < \tau$
Grupo 4) $s(X, [t_0',t_b]) < \sigma \wedge |[t_0',t_b]| \geq \tau$

Procedimientos para el Grupo 1:
Decir que $[t_0', t_b]$ pertenece al Grupo 1, significa que $s(X, [t_0',t_b]) \geq \sigma \wedge |[t_0',t_b]| < \tau$.

Este grupo presenta los siguientes casos:
1) Existe un subintervalo vecino del k-itemset frecuente. (Se puede expandir.) Entonces, se une.
➔ Seguir con Análisis de Frecuencia <Tipo 2>.
2) No existe un subintervalo vecino del k-itemset frecuente. Se puede expandir. Si logra tener $\geq \tau$
➔ Seguir con Análisis de expansión.
3) No existe un subintervalo vecino del k-itemset frecuente. Se puede expandir. Pero finalmente, $< \tau$. ➔ Se elimina. Fin.
4) No existe un subintervalo vecino del k-itemset frecuente. No se puede expandir (porque no existe transacciones a incluir o bien, que al incluir transacciones inevitablemente deja de ser frecuente)
➔ Se elimina. Fin.

Procedimientos para el Grupo 2:
Decir que $[t_0', t_b]$ pertenece al Grupo 2, significa que $s(X, [t_0',t_b]) \geq \sigma \wedge |[t_0',t_b]| \geq \tau$.
Este grupo presenta los siguientes casos:
1) Existe un subintervalo vecino del k-itemset frecuente. (Se puede expandir.) Entonces, se une.
➔ Seguir con Análisis de Frecuencia <Tipo 1>.
2) No existe un subintervalo vecino del k-itemset frecuente. Se puede expandir.
➔ Seguir con Análisis de expansión.
3) No existe un subintervalo vecino del k-itemset frecuente. No se puede expandir (porque no existe transacciones a incluir o bien, que al incluir transacciones inevitablemente deja de ser frecuente)
➔ Se mantiene la regla con el subintervalo $[t_0', t_b]$. Fin.

Procedimientos para el Grupo 3:
Decir que $[t_0', t_b]$ pertenece al Grupo 3, significa que $s(X, [t_0',t_b]) < \sigma \wedge |[t_0',t_b]| < \tau$.
Este grupo presenta los siguientes casos:
1) Existe un subintervalo vecino del k-itemset frecuente. (Se puede expandir.) Entonces, se une.
➔ Seguir con Análisis de Frecuencia <Tipo 2>.
2) No existe un subintervalo vecino del k-itemset frecuente. Se puede expandir. Si logra tener $\geq \sigma$ y $\geq \tau$. ➔ Seguir con Análisis de expansión.
(sería el caso de que exista un subintervalo $< \tau$ muy frecuente que por alguna razón no pudo ser incluida en algún subintervalo de k-itemset frecuente)
3) No existe un subintervalo vecino del k-itemset frecuente. Se puede expandir. Pero finalmente, $< \sigma$ o $< \tau$. ➔ Se elimina. Fin.
4) No existe un subintervalo vecino del k-itemset frecuente. No se puede expandir (porque no existe transacciones a incluir o bien, que al incluir transacciones no hay manera de hacerlo $\geq \sigma$)
➔ Se elimina. Fin.

Procedimientos para el Grupo 4:
Decir que $[t_0', t_b]$ pertenece al Grupo 4, significa que $s(X, [t_0',t_b]) < \sigma \wedge |[t_0',t_b]| \geq \tau$-
Este grupo presenta los siguientes casos:
1) Existe un subintervalo vecino del k-itemset frecuente. (Se puede expandir.) Entonces, se une.
➔ Seguir con Análisis de Frecuencia <Tipo 2>
2) No existe un subintervalo vecino del k-itemset frecuente. Se puede expandir. Si logra tener $\geq \sigma$.
➔ Seguir con Análisis de expansión.
(sería el caso de que exista un subintervalo $< \tau$ muy frecuente que por alguna razón no pudo ser incluida en algún subintervalo de k-itemset frecuente)
3) No existe un subintervalo vecino del k-itemset frecuente. Se puede expandir. Pero finalmente, $< \sigma$. ➔ Se elimina. Fin.

4) No existe un subintervalo vecino del k-itemset frecuente. No se puede expandir (porque no existe transacciones a incluir o bien, que al incluir transacciones no hay manera de hacerlo ≥ σ)
➔ Se elimina. Fin.

Paso 4
Análisis de la validez de las reglas de d_{1b}, en d_2. En caso de que una regla de d_{1b} tenga varios subintervalos del k-itemset frecuente, entonces, comienzo el análisis con el último subintervalo. (Análisis similar al paso 5)

Paso 5
Análisis de la validez de las reglas obtenidas en d_2, en d_{1b}. En caso de que una regla de d_2 tenga varios subintervalos del k-itemset frecuente, entonces, comienzo el análisis con el primer subintervalo. Los casos posibles son:
1) No existe en d_{1b} un subintervalo del k-itemset frecuente. No se puede expandir.
(porque no existe transacciones a incluir o bien, que al incluir transacciones inevitablemente deja de ser frecuente) ➔ Se mantiene el subintervalo. Fin.
2) No existe en d_{1b} un subintervalo del k-itemset frecuente. Se puede expandir (hacia la izquierda), respetando la propiedad de maximal. ➔ Seguir con Análisis de expansión.
3) Existe en d_{1b} un subintervalo del k-itemset frecuente. Entonces, se une. ➔ Seguir con Análisis de frecuencia <Tipo 1>

Luego del Análisis de frecuencia <Tipo 1> se pueden observar los siguientes casos:
Si el subintervalo vecino del k-itemset frecuente (del d_{1b}) es adyacente...
3.1) No se puede expandir (ni por derecha y ni por izquierda)
3.2) Se puede expandir (por derecha y/o izquierda)
Si el subintervalo vecino del k-itemset frecuente (del d_{1b}) no es adyacente...
3.3) No se puede expandir (ni por derecha y ni por izquierda).
3.4) Se puede expandir (por derecha y/o izquierda).
3.5) Finalmente, luego del análisis queda dividido otra vez (como si no se hubiera unido). Tampoco se puede expandir (por izquierda, ni por derecha). (El análisis por derecha queda descartado, ya que dicha posibilidad se habría analizado en el procesamiento de d_2).
3.6) Finalmente, luego del análisis queda dividido otra vez (como si no se hubiera unido). Pero se puede expandir (por izquierda)

Nota:
* 2) son nuevas reglas para d_{1b} (surgidas de la extensión de las reglas de d_2)
* Para 2) y 3.6): Puede ocurrir que al extender por izquierda se encuentre con un intervalo con $< \tau$ y frecuencia $> \sigma$ (o sea, no es un subintervalo del k-itemset frencuente), pero que hace al intervalo resultando $\geq \sigma$. Entonces, mientras no sea $< \sigma$; se podría analizar la posibilidad de expandir hacia la derecha.

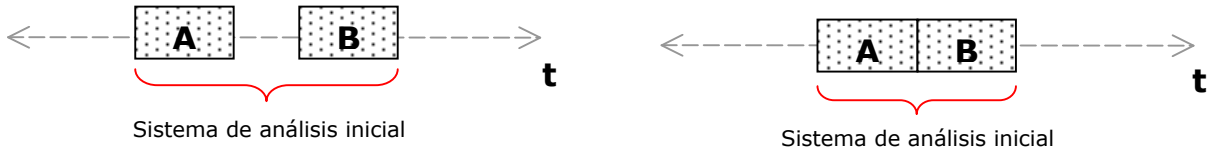
Paso 6:
Análisis de la existencia de alguna regla, que no haya podido identificarse ni en d_{1b}, tampoco en d_2; pero que unificado dichos intervalos, pueda obtenerse alguna regla.
Precondición: No existe la regla en los intervalos d_{1b}, tampoco en d_2.
Sería el caso en que existe en d_{1b} un intervalo que no alcanza las condiciones de ser un subintervalo de k-itemset frecuente, y que en d_2 también existe un intervalo que por sí solo no alcanza las condiciones de formar una un subintervalo de k-itemset frecuente. Pero, que al unirse los d_{1b} y d_2, puede llegar a un subintervalo de k-itemset frecuente. Ya sea por σ y τ.

O bien, el caso de que al eliminar el d_{1a}, el intevalo resultante en d_{1b} de la regla (llamémosle "X"), no llegue a las condiciones de τ, y que en d_{1b} y d_2 tampoco haya un subintervalo frecuente de dicha regla. Pero que al extender el intervalo resultante "X" sobre el d_{1b}, pueda llegar a cumplir con las condiciones de reglas con σ y τ.

<u>Análisis de frecuencia (inicial)</u>:

Tipo 1: Unión de dos subintervalos de k-itemset frecuente (o sea, $\geq \sigma$)
Se hace en primer lugar, una agrupación de los subintervalos de tiempo con $\geq \sigma$ (respetando la propiedad de maximal). Si los intervalos intervinientes son [t1, t2] y [t3, t4], siendo $t1 < t2 < t3 < t4$, entoces el intervalo inicial es: [t1, t4]. Se analiza si la frecuencia del intervalo resultante es $> \sigma$.
Si es $= \sigma$, se mantiene el último intervalo resultante.
Si es $< \sigma$, se va achicando el intervalo (mientras no sea menor al temporal support τ), hasta que la frecuencia supere la mínima establecida. Puede darse el caso de que se haga una división generando más de un subintervalo frecuente.
Si es $> \sigma$, entonces se hace el **análisis de expansión**(*).

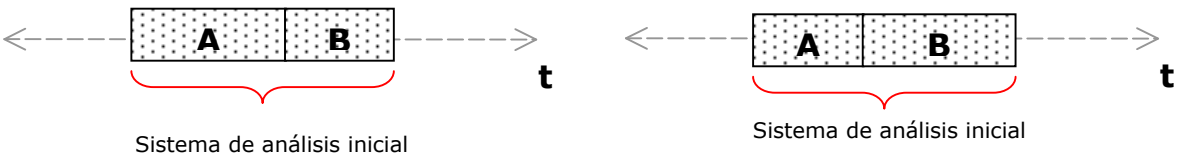
Ejemplos del Tipo 1:



A = [t1, t2]
B = [t3, t4]
A y B: subintervalos con $\geq \sigma$
Puede ser que uno de los subintervalos tenga soporte temporal $< \tau$. (Caso de la eliminación).
En el caso de que $t2 \neq t3$, y los dos subintervalos tengan soporte $= \sigma$, entonces, no es posible la unión, ni la expansión.

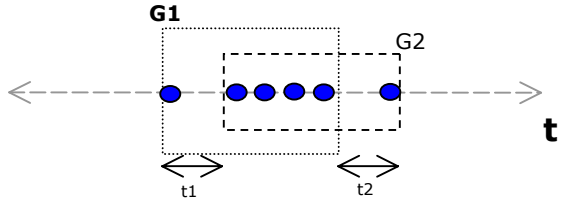
Tipo 2: Unión de dos subintervalos, sólo uno de ellos con $\geq \sigma$
Se hace en primer lugar, una agrupación de los subintervalos de tiempo, de los cuales uno es $\geq \sigma$ (respetando la propiedad de maximal). Si los intervalos intervinientes son [t1, t2] y [t3, t4], siendo $t1 < t2 < t3 < t4$, entoces el intervalo inicial es: [t1, t4]. Se analiza si la frecuencia del intervalo resultante es $> \sigma$.
Si es $= \sigma$, se mantiene el último intervalo resultante.
Si es $< \sigma$, se va achicando el intervalo (mientras no sea menor al temporal support τ), hasta que la frecuencia supere la mínima establecida. Puede darse el caso de que se haga una división generando más de un subintervalo frecuente.
Si es $> \sigma$, entonces se hace el **análisis de expansión**(*).

Ejemplos del Tipo 2:



A = [t1, t2] ; B = [t3, t4] ; A ó B (sólo uno de los 2) con subintervalo con ≥ σ.
Puede ser que uno de los subintervalos tenga < τ. (Caso de la eliminación)
En el caso de que el subintervalo frecuente tenga soporte = σ, entonces, no es posible la unión, ni la expansión.

Análisis de expansión (*): Para el análisis de maximalidad, existen casos en los cuales nos resulta lo mismo extender el subintervalo hacia la izquierda o hacia la derecha (Δt1 = Δt2); pero que no podemos incluir a ambas transacciones debido a que dejamos de cumplir la condición de s(X, [t,t']) ≥ σ. En nuestro caso priorizamos la dirección hacia la izquierda. En nuestro ejemplo, el G1.



Pre-condición: La frecuencia del intervalo a analizar es mayor o igual al soporte mínimo (≥ σ).
Esquema de Análisis: Si los próximos subintervalos vecinos de k-itemset frecuentes son = σ, o bien, no existen subintervalos vecinos de k-itemset frecuentes, entonces, sólo se sigue expandiendo hasta que la frecuencia del subintervalo no sea < σ. (Validar que el subintervalo | [t,t'] | ≥ τ).
Si los próximos subintervalos vecinos de k-itemset frecuentes son ≥ σ, entonces, se las agrupa también para el Análisis de frecuencia.

Notas:
Se utilizan las siguientes expresiones:
"se puede expandir": cuando la frecuencia del intervalo s(X, [t,t']) ≥ σ y existe transaccion/es consecutivas (en el tiempo) que aunque se las incluya la frecuencia se mantiene ≥ σ.
"no se puede expandir": cuando no existe transacción por más que la frecuencia s(X, [t,t']) ≥ σ. O bien, cuando no hay posibilidades de incluir una o más transacciones consecutivas (en el tiempo) porque hace que la frecuencia sea < σ.

5. CONCLUSIÓN

Sobre el modelo de Reglas de Asociación Temporales, propuesto por los autores Ale y Rossi en [4], se propuso un método de actualización de dichas reglas con el objetivo de optimizar el uso de los recursos físicos y visualizar una notable disminución del tiempo del procesamiento total, al no re-procesar gran parte de la Base de Datos. En el cual se procesan sólo las nuevas transacciones incorporadas a la Base de Datos, aprovechando la información almacenada de un procesamiento previo. Vimos los diferentes casos que se pueden presentar en el proceso de actualización de las reglas y su procesamiento:

- Generación de nuevas reglas o generación de subintervalos en reglas existentes.
- Ampliación o reducción de los subintervalos o intervalos de las reglas existentes.
- Eliminación de reglas o subintervalos de reglas. (Caso de depuración de reglas)

A través de la ejecución de los pasos propuestos en este trabajo para la actualización de las reglas sobre una Base de Datos transaccional actualizada se buscó obtener el mismo conjunto de Reglas de Asociación como si hubiéramos procesado toda la nueva Base de Datos nuevamente.

Este trabajo es parte de un proyecto en desarrollo. Se realizará la correspondiente implementación como una extensión a WEKA.

6. REFERENCIAS

[1] Agrawal, R. – Imielinski, T. – Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. Proc. ACM SIGMOD: 207-216. 1993.
[2] Agrawal, R. - Srikant, R.: Fast Algorithms for Mining Association Rules in Large databases. Proc. of the 20th International Conference on Very Large Data Bases, pages 478-499, September 1994.
[3] Ale, J. – Rossi, G.: An Approach to Discovering Temporal Association Rules. Proc. ACM 15th Symposium on Applied Computing, pages 294-300. March 2000.
[4] Ale, J. - Rossi, G.: The Itemset's Lifespan Approach to Discovering General Temporal Association Rules. Proceedings of ACM the Second Workshop on Temporal Data Mining held - KDD - 2002. July 2002.
[5] Ayan, N. F. – Tansel, A. U. – Arkun, E.: An Efficient Algorithm To Update Large Itemsets With Early Pruning.
[6] Bettini, C. – Wang, X. – Jajodia, S. - Lin, J.: Discovering Frequent Event Patterns With Multiple Granularities In Time Sequences. IEEE TOKDE Vol.10 N°2: 222-237. April 1998.
[7] Brin, S. – Motwani, R. – Ullman, J. – Tsur, S.: Dynamic Itemset Counting and Implication Rules for Market Basket Data. Proc. ACM SIGMOD: 255-264. 1997.
[8] Chakrabarti, S. – Sarawagi, S. – Dom, B.: Mining surprising patterns Using Temporal Description Length. Proc. 24th VLDB Conf. 1998.
[9] Chan, K. – Fu, A.: Efficient Time Series Matching by Wavelets. Proc. IEEE 15th Intl. Conf. On Data Engineering, 1999.
[10] Chen, X. – Petrounias, I. – Heathfield, H.: Discovering Temporal Association Rules in Temporal Databases. Proc. Int'l Workshop IADT '98. July 1998.
[11] Chen, X. - Petrounias, I.: Discovering Temporal Association Rules: Algorithms, Language and System. Proc. of 2000 Int. Conf. on Data Engineering, 2000.
[12] Cheung, D. W. – Han, J. – Ng, V. T. – Wong, C. Y.: Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique. Proc. Of 1996 Int'l Conf. On Data Engineering, pages 106-114 Feb. 1996.
[13] Cheung, D. W – Lee, S. D. – Kao, B.: A General Incremental Technique for Maintaining Discovered Association Rules.
[14] Cheung, W. – Zaïane, O. R.: Incremental Mining of Frequent Patterns Without Candidate Generation or Support Constraint.
[15] Cheung, D. W. – Ng, V. T. – Tam, B. W.: Maintenance of Discovered Knowledge: A Case in Multi-level Association Rules
[16] Lee, C.H. – Lin, C.R. – Chen, M.S.: On Mining General Temporal Association Rules in a Publication Database. Proc. Of the IEEE Int'l conf. On Data Mining (ICDM-01). November 2001.
[17] Lee, S. D. – Cheung, D. W.: Maintenance of Discovered Association Rules: When to update?
[18] Mannila, H. – Toivonen, H. – Verkamo, I.: Discovering frequent Episodes in Sequences. KDD'95. AAAI: 210-215. August 1995.
[19] Mannila, H. - Toivonen, H. - Verkamo, I.: Efficient algorithms for discovering association rules. in Usama M. Fayyad and Ramasamy Uthurusamy, editors, Knowledge Discovery in Databases (KDD'94), pages 181-192, Seattle, Washington, July 1994. AAAI Press.
[20] Ozden, B. – Ramaswamy, S. – Silberschatz, A.: Cyclic Association Rules. ICDE 1998.
[21] Valtchev, P. – Missaoui, R. – Hacene, M. R. – Godin, R.: Incremental Maintenance of Association Rules.