KHAI THÁC CÁC LUẬT KẾT HỢP CÓ CHU KỲ TRONG CƠ SỞ DỮ LIỆU CÓ TÍNH THỜI GIAN

Nguyễn Đình Ngọc[1], Nguyễn Xuân Huy[2], Nguyễn Đình Thuân[3]
[1]Trường Đại học Thăng Long, [2]Viện Công nghệ Thông tin, [3]Trường Đại học Thuỷ sản Nha Trang
(Bài nhận ngày 5 tháng 1 năm 2004)

TÓM TẮT: *Vấn đề khai thác dữ liệu, tìm ra các dữ liệu thường xuyên xảy ra theo quy luật trong cơ sở dữ liệu có tính thời gian đã được nhiều tác giả nghiên cứu trong thời gian gần đây[3,8,9]. Trong bài báo này, chúng tôi đề nghị thuật toán khai thác các luật kết hợp có tính chu kỳ theo thời gian. Chẳng hạn, bánh mì và càphê thường được bán cùng nhau vào buổi sáng hoặc bánh trung thu, lồng đèn và nến thường được bán vào tết Trung thu. Thuật toán này là sự mở rộng của thuật toán Apriori trong việc khai thác các luật kết hợp có chu kỳ.*

1. Giới thiệu

Giả sử một siêu thị bày bán một số lượng lớn các mặt hàng. Những người quản lý siêu thị cần đưa ra các quyết định như: làm thế nào để tăng trưởng trong kinh doanh, bố trí các mặt hàng trong siêu thị theo thứ tự như thế nào để tăng lợi nhuận, các mặt hàng khi mua có liên hệ với nhau như thế nào, nếu mua mặt hàng A thì thường sẽ mua thêm mặt hàng nào nữa, ... Để đưa ra các quyết định này, người ta sẽ phân tích trên dữ liệu lưu trữ các hoá đơn bán hàng.

Với các kho dữ liệu, vấn đề đặt ra là phải khai thác dữ liệu để tìm ra các quy luật giữa các nhóm dữ liệu với một độ tin cậy nào đó. Chẳng hạn 80% khách hàng mua bàn chải răng thì sẽ mua kem đánh răng, hoặc 85% sinh viên đạt điểm tốt môn Toán Cao cấp và Lập trình Căn bản thì sẽ đạt điểm tốt môn Cấu trúc dữ liệu.

Thuật toán về khai thác dựa trên các luật liên hệ lần đầu tiên được giới thiệu trong [2], và bao gồm 2 bước chính như sau:

+ Tìm ra các tập đơn vị dữ liệu xảy ra thường xuyên(nghiã là tìm tất cả các tập đơn vị dữ liệu có số lần xuất hiện lớn hơn một ngưỡng Min_Supp cho trước)

+ Từ các tập đơn vị dữ liệu thường xảy ra, thiết lập các luật thể hiện mối liên hệ giữa các đơn vị dữ liệu.

Trong thuật toán trên chưa xét đến các yếu tố sau:

- Một số đơn vị dữ liệu có tính chất là sự xuất hiện của chúng có tính chu kỳ. Chẳng hạn, số lượng bán cá chép sẽ tăng trong dịp đưa Ông táo 23 tháng Chạp hoặc số lượng vịt sẽ được bán nhiều trong dịp Tết Đoan ngọ.

- Một số luật kết hợp không thoả mãn trong toàn bộ Cơ sở dữ liệu mà chỉ thoả mãn tại một số thời điểm nào đó. Chẳng hạn, vào mùa đông nếu khách hàng mua mũ len thì 75% sẽ mua tất tay.

Trong thuật toán được giới thiệu trong phần 3, chúng ta sẽ cải tiến thuật toán Apriori, với việc xét thêm lược đồ thời gian và xét đến yếu tố lặp lại theo chu kỳ của sự xuất hiện của các đơn vị dữ liệu.

Chẳng hạn, với lược đồ thời gian có dạng (ngày, tháng, năm) thì ký hiệu (*,5,2003) nghĩa là tất cả các ngày trong tháng 5 của năm 2003 hoặc (10,*,*) là ngày 10 của mỗi tháng.

Bằng cách đưa vào khái niệm lược đồ thời gian, tính có chu kỳ của việc xuất hiện của các đơn vị dữ liệu, chúng ta sẽ khai thác sự xuất hiện của các luật kết hợp theo chu kỳ của chúng.

2. Cách tiếp cận của thuật toán APRIORI

2.1. Các định nghĩa: [2]

Cho $I = \{I1, I2, \ldots, Im\}$ là tập các đơn vị dữ liệu. Ta qui ước mỗi giao tác làm phát sinh một tập các đơn vị dữ liệu. Cho D là tập các giao tác, mỗi giao tác T cho kết quả là tập con các đơn vị dữ liệu $T \subseteq I$.

Định nghĩa 1

Một luật kết hợp là một phép suy diễn có dạng $X \rightarrow Y$, trong đó $X \subset I$, $Y \subset I$, $X \neq \emptyset$, $Y \neq \emptyset$ và $X \cap Y = \emptyset$

Giả sử, ta khảo sát tập D gồm N giao tác. Với mỗi tập con đơn vị dữ liệu $X \subset I$, ta xét số lượng M các giao tác trong D làm xuất hiện các đơn vị dữ liệu có trong X.

Ký hiệu: $v(X) = M/N$ gọi là tần suất xuất hiện của các đơn vị dữ liệu trong X

Định nghĩa 2

Ký hiệu Supp(X → Y) = v(X∪Y) và gọi là mức xác nhận của luật X → Y

Định nghĩa 3

$$Conf(X \rightarrow Y) = \frac{Supp(X \cup Y)}{Supp(X)}$$

Kí hiệu:

Và gọi là độ tin cậy của luật X → Y

Như vậy, Supp(X) là tần suất xuất hiện của đơn vị dữ liệu X và Conf(X → Y) là tỷ lệ giữa tần suất xuất hiện giữa X∪Y và X.

Định nghĩa 4

Cho trước các giá trị Min_Supp = s0 và Min_Conf = c0

- Ta nói tập X⊂ I là xảy ra thường xuyên trong D nếu tỷ lệ giao tác trong D chứa X >= s0
- Ta nói luật X → Y là xảy ra thường xuyên nếu:

Supp(X → Y) > s0 và Conf(X → Y)>c0

Vớ duù: Xét CSDL gồm N=12 giao tác: t1, t2, ..., t12 như sau:

Giao tác	Đơn vị dữ liệu					
t_1		B		D		
t_2		B	C			
t_3	A	B		D		
t_4	A			D		
t_5		B			E	
t_6	A		C	D		
t_7	A			D	E	
t_8	A	B			E	
t_9		B	C			
t_{10}		B		D		
t_{11}	A			D	E	F
t_{12}				D		F

Ta có:

v(A) = 6/12, v(B) = 7/12, v(C) = 3/12, v(D) = 8/12

v(A∪B) = 2/12, v(A∪D) = 5/12

Supp(A→D) = 5/12= 41.66%, Conf(A→D) = 5/6 = 83.33%

Supp(B → D) = 25% và Conf(B → D)= 42.85%

Supp(D → F) = 16.7% và Conf(D → F) = 25%

Supp(F → D) = 16.7% và Conf(F → D) =100%

Hai bước chính của thuật toán Apriori là:[2]

1. Cho trước các giá trị s_0- ngưỡng xác nhận và c_o-ngưỡng tin cậy, hãy tìm tất cả các đơn vị dữ liệu có mức xác nhận lớn hơn ngưỡng s_o và gọi là các đơn vị dữ liệu là tập xảy ra thường xuyên.
2. Dựa trên các tập thường xuyên này để tìm ra các luật thoả mãn bài toán. Nếu X∪Y và X là các tập thường xuyên xảy ra, ta tính được độ tin cậy của luật này: Conf(X → Y) = Supp(X∪Y)/Supp(X)

   Nếu Conf(X → Y) ≥ c_o , thì X → Y là một luật cần tìm

2.2. Thuật toán Apriori

Input: Tập các đơn vị dữ liệu I
Tập các giao tác D gồm N giao tác
Các ngưỡng s_o và c_o

Output: Tất cả các luật liên hệ thoả so, co

Thuật toán: Cách tiếp cận này dựa trên Heuristic sau: Nếu bất kỳ tập k-đơn vị dữ liệu nào là không xảy ra thường xuyên thì bất kỳ tập (k+1)-đvdl chứa chúng cũng sẽ không xảy ra thường xuyên, nghĩa là:

Nếu ∃X⊂I : v(X) ≤ so thì ∀X'⊃ X : v(X') ≤ so

Thuật toán gồm các bước chính sau:

* Tìm các tập thường xuyên:

- Thuật toán khám phá ra các tập đơn vị dữ liệu thường xuyên bằng cách duyệt qua nhiều lần các dữ liệu. Trong lần duyệt đầu tiên, ta tính mức xác nhận của từng đơn vị dữ liệu riêng và xác định đơn vị dữ liệu nào là thường xuyên (tương ứng với s_o cho trước).

- Trong các lần sau đó, ta sẽ bắt đầu với tập thường xuyên đã xét lần trước đó, tạo ra các tập đơn vị dữ liệu ứng viên và từ các tập ứng viên này, ta tính được mức xác nhận của chúng.
- Cuối cuả phép duyệt ta xác định được tập ứng viên nào là tập thường xuyên thật sự và chúng trở thành nhân trong phép duyệt tiếp theo.
- Quá trình được tiếp tục cho đến khi không còn tập thường xuyên nào được tìm.

Ký hiệu:

- Ta gọi số đơn vị dữ liệu trong một tập là kích thước của chúng và tập có k phần tử là k-đơn vị dữ liệu.
- Gọi L_k: Tập hợp tập thường xuyên gồm có *k* đvdl. Thông tin về mỗi phần tử của L_k có dạng (X,m) trong đó X là tập con của I và m là số lần xuất hiện của X trong D.
- C_k : Tập hợp các tập ứng viên k- đơn vị dữ liệu. Thông tin về mỗi phần tử của C_k cũng có dạng (X,m) như trên.

Thuật toán Apriori dựa trên các thủ tục sau

Thủ tục 1: Tạo ra các tập thường xuyên

```
begin
   L1 = {tập thường xuyên 1-đvdl};
   for ( k = 2; Lk-1 ≠ ∅; k++ ) do
         begin
               Ck=Tạo ra tập ứng viên từ(Lk-1); //Ck là phép kết nối Lk-1*Lk-1
               for mỗi giao tác t ∈ D do
                     begin
                           Ct = Tập con của (Ck) chứa t
                           for mỗi c ∈ Ct do
                                 c.count++;
                     end
               Lk = {c ∈ Ck | c.count ≥ so*N}
         end
    Return (∪k Lk);
end
```

Thủ tục 2: Tìm ra tất cả các luật liên hệ

```
begin
         Result = ∅
         for mỗi tập xảy ra thường xuyên X∈L do
         begin
               for mỗi a ∈ L sao cho a ≠∅ do
               if (Supp(a)/Supp(L) >= co) then
                           Result = Result ∪{X → (X-a)}
               end
         Return Ressult
end
```

Ví dụ 2: Trong ví dụ 1, với c_0=60% và s_0=40%

Ta có tập L gồm các tập đơn vị dữ liệu xảy ra thường xuyên như sau:

L = {{A}, {B},{D}, {E}, {AD}}

Có các luật liên hệ như sau:

A→D với c=83.33% và s=41.66%

D→A với c=62.5% và s=41.66%

3. Thuật toán tìm các luật kết hợp có chu kỳ trong cơ sở dữ liệu có tính thời gian

3.1 Giới thiệu

Việc tìm ra các luật kết hợp có chu kỳ đã được một số tác giả nghiên cứu và được đề nghị như sau:

- Trong [10] các tác giả đã đưa ra thuật toán tìm ra các luật kết hợp một cách định kỳ và đã giải quyết trong trường hợp khoảng cách giữa các thời điểm được xét luôn luôn là 1 hằng số, chưa giải quyết trường hợp khoảng cách các thời điểm khác hằng số.

- Trong [12] là sự mở rộng thuật toán của [10] và tìm được các luật với khoảng thời gian do người sử dụng định nghĩa trước. Với cách này yêu cầu người sử dụng phải biết trước các khoảng thời gian cần tìm các luật và chu kỳ của chúng.

Trong thuật toán sau đây, chúng ta sẽ cải tiến thuật toán Apriori, bởi xét thêm yếu tố lược đồ thời gian. Bằng cách đưa vào khái niệm lược đồ thời gian, tính có chu kỳ của việc xuất hiện của các đơn vị dữ liệu, chúng ta sẽ khai thác sự xuất hiện của các luật kết hợp theo chu kỳ của chúng.

Định nghĩa 5: Ta gọi lược đồ thời gian là tập $R=\{d_i \in DT_i$ với $i=1,..,m\}$

Trong đó: d_i là yếu tố thời gian, chẳng hạn: ngày, tháng, năm, giờ, buổi, ...

và $DT_i = Dom(d_i) \cup \{*\}$: là miền giá trị cho trước của t_i và ký hiệu *

Ví dụ 3: Cho R_1 = {ngày∈ [1..31], tháng ∈[1..12], năm∈[1980..2003]}

Cho R_2 = {buổi∈[sáng, trưa, chiều, tối], thứ∈[Chủ nhật, .., Thứ bảy]}

Định nghĩa 6: Ta gọi thời gian cơ sở trên R là bộ t ∈ R và bậc của t là số ký hiệu '*' xuất hiện trong r. Ký hiệu t_* nếu bậc của t lớn hơn hoặc bằng 1

Ví dụ 4: Trong ví dụ 3 với R_1: t_1 = (25,2,2003) có bậc là 0

t_2 = (25,*,2003) có bậc là 1

Định nghĩa 7: Cho 2 thời gian cơ sở $t=(t_1, t_2, ..., t_m)$ và $t'=(t'_1, t'_2, ..., t'_m)$ trên lược đồ thời gian R.

Ta gọi t là *-bao gồm t' ký hiệu: $t >_* t'$ nếu hoặc $t_i=t'_i$ hoặc t_i='*' $\forall i=1,..,m$

Ví dụ 5: Ta có (25,*,*) $>_*$ (25,*,2003) và (25,*,2003) $>_*$ (25,2,2003)

Định nghĩa 8: Cho D là tập các giao tác và thời gian cơ sở $t=(t_1, t_2, ..., t_m) \in R$. Ta gọi phân hoạch D ứng với t là $D_{t*1}, D_{t*2}, ..., D_{t*n}$ với D_{t*i} là tập giao tác ứng với các giá trị của $t_i \neq$'*'.

Chẳng hạn, trong ví dụ 3, R = {ngày, tháng, năm}

+ Với t=(ngày,*,*) thì D sẽ được phân hoạch thành $D_{t*1}, D_{t*2}, ..., D_{t*31}$ trong đó D_{t*i} ứng với (i,*,*) với i=1,..31
+ Với t=(ngày, tháng,*) thì D sẽ được phân hoạch thành $\{D_{t*ij} \mid i=1,..,31, j=1,..,12\}$

Nhận xét: Theo thời gian, các giao tác sẽ được ghi vào CSDL, qua ví dụ trên nếu áp dụng phép khai thác các luật theo như trong [2] sẽ nảy sinh vấn đề sau:

Nếu các thành phần dữ liệu chỉ xuất hiện tại một vài thời điểm nào đó và có tính chất lặp lại, nếu xét theo cách thông thường thì tần suất xuất hiện của các thành phần dữ liệu này nếu xét trên toàn cục thì không thể thoả mãn Min_Supp, tuy nhiên nếu xét tại các thời điểm xảy ra theo chu kỳ thì sẽ thoả mãn.

Như vậy, với các phần dữ liệu khác nhau theo thời gian, sẽ được xét với các khoảng thời gian theo chu kỳ mà dữ liệu đó xuất hiện. Các khái niệm độ tin cậy và mức xác nhận sẽ được gắn thêm khái niệm ứng với khoảng thời gian theo chu kỳ mà đơn vị dữ liệu đó xuất hiện trong phân hoạch.

Tuỳ theo yếu tố thời gian đang xét mà CSDL được phân hoạch thành n phần, mỗi phần ứng với một phần tử của thời gian cơ sở t∈R trong việc lưu trữ dữ liệu.

Gọi $D = \bigcup_{k=1}^{n} D_{t*k}$

$|D_{t*k}|$ là số giao tác trong D_{t*k}

$N_{Dt*k}(X)$ là số giao tác trong D_{t*i} có chứa tập đơn vị dữ liệu X

Trong phần này, chúng ta sẽ xét các luật có dạng là 1 cặp $<(X \rightarrow_{t*} Y), t^*>$ với $Supp(X \rightarrow_{t*} Y) > Min\_Supp$ và $Conf(X \rightarrow_{t*} Y) > Min\_Conf$. Ngoài ra mỗi luật tương ứng với thời gian cơ sở t^* có bậc >=1.

Với cách phân hoạch như trên khái niệm mức xác nhận Supp(X) được thay bởi mức xác nhận $Supp_{t*}(X)$ như sau:

Định nghĩa 9: Mức xác nhận theo phân hoạch của X là giá trị:

$$Supp_{D_{t*k}}(X) = \frac{N_{D_{t*k}}(X)}{|D_{t*k}|}$$

Định nghĩa 10: Mức xác nhận theo phân hoạch của luật $(X \rightarrow_{t*} Y)$ là giá trị:

$$Supp(X \rightarrow_{t*} Y) = \frac{N_{D_{t*k}}(XY)}{|D_{t*k}|}$$

Định nghĩa 7: Ta gọi độ tin cậy theo phân hoạch của luật $(X \rightarrow_{t*} Y)$ là giá trị:

$$Conf(X \rightarrow_{t^*} Y) = \frac{Supp_{D_{t^*k}}(XY)}{Supp_{D_{t^*k}}(X)}$$

Trong thuật toán được giới thiệu sau, chúng ta sẽ tìm ra các tập thường xuyên theo khái niệm mức xác nhận và độ tin cậy theo phân hoạch và các giá trị này được tính ứng với khoảng thời gian theo chu kỳ mà đơn vị dữ liệu đó xuất hiện trong phân hoạch.

3.2 Thuật toán: Thuật toán gồm các thủ tục sau

Thủ tục 1: Tạo ra tập xảy ra thường xuyên có chu kỳ gồm 2 đvdl

```
Begin
    Ret = ∅
    L2*0={tập tất cả các 2-đvdl với thời gian cơ sở t có bậc 0}
    For Với mỗi giá trị k của thời gian cơ sở t* chứa t (nghĩa là t* >* t)
        For Với mỗi X2 ∈ L2*0
            if (X2 ∉ Ret) then
                X2.Count = 1
                X2.Start= k
                Ret = Ret ∪ X2
            end if
            if (X2 ∈ Ret) then
                X2.Count = X2.Count +1
        if (X2.Count + (|Dt*k|-k) < Min_Supp*|Dt*k| ) then Ret = Ret - X2
            end if
        end for
    end for
    Return <Ret, t*>
end
```

Thủ tục 2: Tìm tất cả các tập dữ liệu xảy ra thường xuyên có chu kỳ D

```
Begin
    L = ∅
    Tạo ra tập xảy ra thường xuyên có chu kỳ gồm 2-đvdl;
    m = 2
    while (Cm ≠∅) do
            Cm = Cm-1* Cm-1  //Thực hiện phép kết nối với tất cả đvdl
            m = m+1
    end
    For Với mỗi thời gian cơ sở t* chứa t (nghĩa là t* >* t)
            For với mỗi tập đvdl X ∈ Cm do
                X.Count = X.Count + 1
            end for
        For với mỗi tập đvdl X∈ Cm do
        if (X2.Count + (|Dt*k|-k) >= Min_Supp*|Dt*k| ) then L = L ∪ X;
    Return <L, t*>;
end
```

Ví dụ 4: Xét lược đồ thời gian R={thứ , tuần}, cho Min_Supp=50% và Min_Conf=70% và dữ liệu được cho bởi bảng sau:

	Thứ 2	Thứ 3	Thứ 4	Thứ 5	Thứ 6	Thứ 7	Chủ nhật
tuần 1	ABD	ABC		AD			BCF
tuần 2	BDF	BC	CD	D		EF	CF
tuần 3	AD	B	AF	BD	BC		
tuần 4	ACDF	BD		DE			ACEF
tuần 5	ADF	BE			DEF		DEF

Với thuật toán Apriori, ta có: Supp(A→ D) = 5/35= 14.38%

và Conf(A→ D) =5/8 = 62.5%

Như vậy, nếu xét trong toàn cục thì luật A→ D không thể thoả mãn.

Tuy nhiên, áp dụng thuật toán 3.2 tìm các luật kết hợp theo chu kỳ, với thời gian cơ sở t=(thứ 2,*), ta có các kết quả sau:

$L_2(1,*)$ với k=1		
X_2	Start	Count
AB	1	1
AD	1	1
BD	1	1

$L_2(1,*)$ với k=2		
X_2	Start	Count
AB	1	1
AD	1	1
BD	1	2
BF	2	1
DF	2	1

Sau lần duyệt thứ 2, tập các tập đơn vị dữ liệu ứng viên là:

{AB, AD, BD, BF, DF}

$L_2(1,*)$ với k=3		
X_2	Start	Count
AB	1	1
AD	1	2
BD	1	2
BF	2	1
DF	2	1

$L_2(1,*)$ với k=4		
X_2	Start	Count
AB	1	1*
AC	4	1*
AD	1	3
AF	4	1*
BD	1	2
BF	2	1*
CD	4	1*
CF	4	1*
DF	2	2

Với mỗi lần duyệt qua $L_2(1,*)$ với các hệ số khác nhau của k, tuỳ theo sự xuất hiện của các đơn vị dữ liệu thoả mãn điều kiện: $X_2.Count + (|D_{t*k}|-k) > Min\_Supp*|D_{t*k}|$ mà sẽ được xét trong lần duyệt tiếp theo (ký hiệu * các đơn vị dữ liệu không thoả), trong ví dụ trên $|D_{t*k}|=5$, Min_Supp=50%

$L_2(1,*)$ với k=5		
X_2	Start	Count
AD	1	4
AF	5	1*
BD	1	2*
DF	2	3

$L_2(1,*)$ với k=5		
X_2	Start	Count
AD	1	4
DF	2	3

Luật kết hợp ứng với $L_2(1,*)$	Supp	Conf
$(A\rightarrow_{t*} D)$	80%	100%
$(D\rightarrow_{t*} A)$	80%	80%
$(D\rightarrow_{t*} F)$	60%	60%
$(F\rightarrow_{t*} D)$	60%	100%

Với Min_Conf=70% , ta có tập các luật được tạo ra theo thuật toán trên như sau:

Luật kết hợp ứng với $L_2(1,*)$	Supp	Conf
$(A\rightarrow_{t*} D)$	80%	100%
$(D\rightarrow_{t*} A)$	80%	80%
$(F\rightarrow_{t*} D)$	60%	100%

Nhận xét: Nếu áp dụng cách tính như trong thuật toán [2] thì luật trên không thể thoả mãn các ngưỡng vì sự xuất hiện của các đơn vị dữ liệu trên là không thể thoả Min_Supp. Tuy nhiên, bằng cách xét sự xuất hiện theo chu kỳ, chúng ta có thể tìm ra các luật kết hợp tương ứng.

4. Chứng minh tính đúng đắn của thuật toán 3.2

Bổ đề 1: Điều kiện cần và đủ để 1 tập đơn vị dữ liệu qua phép xử lý của thuật toán trên là xảy ra thường xuyên theo chu kỳ ở bước phân hoạch thứ k là:

$$X.Count + (|D_{t^*k}|-k) >= Min\_Supp*|D_{t^*k}|$$

Chứng minh:

1. Điều kiện cần: Ta thấy trong thuật toán 3.2, các đơn vị dữ liệu nào không thoả điều kiện trên đều bị loại ra khỏi tập ứng viên.
2. Điều kiện đủ: Các đơn vị dữ liệu là thường xuyên theo chu kỳ phải thoả mãn: sau khi xử lý đến thời điểm sau cùng trong mỗi phân hoạch, phải có $X.Count >= Min\_Supp*|D_{t^*k}|$.

Như vậy, tại thời điểm thứ k, số lần còn lại có thể cập nhật X.Count tối đa là $(|D_{t^*k}|-k)$. Vậy để 1 tập đơn vị dữ liệu qua phép xử lý là xảy ra thường xuyên theo chu kỳ ở bước phân hoạch thứ k là: $X.Count+(|D_{t^*k}|-k)>= Min\_Supp*|D_{t^*k}|$. Ta có điều phải chứng minh.

Định lý 1: Gọi L_{k^*0} là tập k-đơn vị dữ liệu xảy ra thường xuyên ứng với các thời gian cơ sở bậc 0 là t^*_0 và được xử lý bởi thuật toán 3.2, khi đó tập kết quả <L, t^*> sẽ chứa tất cả và chỉ chứa những tập đơn vị xảy ra thường xuyên với các thời gian cơ sở $t^* >_* t$ và thoả mãn Min_Supp ứng với mỗi phân hoạch này.

Chứng minh:

1. $\forall X \in$ <L, t*> ta cần chứng minh X là xảy ra thường xuyên với các thời gian cơ sở $t^* >_* t$ và thoả mãn Min_Supp.

   Vì $X \in$ <L, t*> do đó $X.Count >= Min\_Supp *|D_{t^*k}|$ sau phép xử lý cuối cùng. Với mỗi lần cập nhật, theo Bổ đề 1 ta có $X.Count + (|D_{t^*k}|-k) >= Min\_Supp*|D_{t^*k}|$. Như vậy, khi tất cả các đơn vị dữ liệu xảy ra thường xuyên được duyệt ứng với thời gian cơ sở t* thì sẽ có ít nhất tỷ lệ số đơn vị dữ liệu được duyệt >=Min_Supp.
2. Giả sử ngược là $\exists Y$ là xảy ra thường xuyên ứng với thời gian cơ sở t^* mà $Y \notin$ <L, t*>. Vì $Y \notin$ <L, t*> do đó $\exists k'$ sau lần xử lý này Y bị loại ra khỏi tập <L, t*>. Theo Bổ đề 1, khi đó, $Y.Count + (|D_{t^*k}|-k') < Min\_Supp*|D_{t^*k}|$. Như vậy, tỷ số lần Y.Count không được cập nhật ứng với thời gian t^*_0 là < Min_Supp. Do đó, Y không thể là tập xảy ra thường xuyên ứng với thời gian cơ sở t^*.

5. Kết luận

Trong bài báo này chúng tôi đã giới thiệu thuật toán với khai thác các luật kết hợp theo thời gian có tính chu kỳ. Thuật toán sẽ hữu hiệu khi có các dữ liệu xuất hiện có tần suất thấp và sự xuất hiện của chúng có tính chu kỳ.

USING CYCLIC ASSOCIATION RULES IN TEMPORAL DATABASE

Nguyen Dinh Ngoc, Nguyen Xuan Huy, Nguyen Dinh Thuan

ABSTRACT: *Because the data being mined in the temporal database will evolve with time, many researchers have focused on the incremental mining of frequent sequences in temporal database [3,8,9]. In this paper, we study the problem of discovering association rules that display regular cyclic variation over time. For example, bread and coffee are frequently sold together in morning hours, or moon cake, lantern and candle are often sold before Mid-autumn Festival. This paper extends the Apriori algorithm and develop the optimization technique for discovering cyclic association rules.*

TÀI LIỆU THAM KHẢO

[1] R. Agarwal, C. Aggarwal, and V. Prasad. *A Tree Projection Algorithm for Generation of FrequentItemsets*. Jornal of Parallel and Distributed Computing 2000

[2] R. Agrawal, T. Imielinski, and A. Swami. *Mining Association Rules between Sets of Items in LargeDatabases*. Proc.of ACMSIGMOD, 1993

[3] J. Ale and G. Rossi. *An Approach to Discovering Temporal Association Rules.* ACM Symposium on Applied Computing, 2000

[4] D. Cheung, J. Han, V. Ng, and C. Wong. *Maintenance of Discovered Association Rules in LargeDatabases: An Incremental Updating Technique*. Proc. of 1 996 Int'l Conf. on Data Engineering, 1996

[5] J. Han and J. Pei. *Mining Frequent Patterns by Pattern-Growth: Methodology and Implications*.ACM SIGKDD Explorations (Special Issue on Scaleble Data Mining Algorithms), 2001

[6] J.Han, J.Pei, B.Mortazavi-Asl, Q.Chen,U.Dayal, and M.C.Hsu.FreeSpan: *Frequentpattern-projected sequential pattern mining*. Proc. of 2000 Int. Conf. on Knowledge Discovery and Data Mining, 2000

[7] Y. Li, P. Ning. *Discovering Calendar-baesd Temporal Association Rules*. 2001

[8] A. Savasere, E. Omiecinski, and S. Navathe. *An Ecient Algorithm for Mining Association Rules in Large Databases*. Proc.of the2 th International Conference on Very Large Data Bases,1996

[9] Nguyễn Đình Thuân. *Update Algorithm in Temporal Database*. Báo cáo tại International Conference on High Performance Scientific Computing. Hanoi, 3/2003

[10] Nguyễn Đình Ngọc, Nguyễn Xuân Huy, Nguyễn Đình Thuân. *Các dạng chuẩn trong cơ sở dữ liệu có tính thời gian*. Báo cáo tại Hội thảo Quốc gia "Một số vấn đề về chọn lọc CNTT". Thái Nguyên 8/2003

[11] Nguyễn Đình Thuân. *Khai thác các luật kết hợp với trọng số trong Cơ sở dữ liệu có tính thời gian*. Báo cáo tại Báo cáo tại Hội thảo Khoa học Quốc gia lần thứ 1 "Nghiên cứu cơ bản và ứng dụng Công nghệ Thông tin" Hà Nội 10/2003

[12] Juan M. Ale, Gustavo H. Rossi. *An Approach to Discovring temporal association rules*. ACM 2000

[13] B.Ozden, S. Ramaswamy, A. Silbetschatz. *Cyclic Association Rules*. In Proc. of the 14th Int'l Conf. on Data Engineering. 1998

[14] Fabrizio A., Giovambattista I,. *On the complexity of Categorical and Quantitative Association Rules*. 10/2003