THUẬT TOÁN TÌM NHANH MINIMAL GENERATOR CỦA TẬP PHỔ BIẾN ĐÓNG

Lê Hoài Bắc, Võ Đình Bảy
Trường Đại học Khoa học Tự nhiên, ĐHQG- HCM
(Bài nhận ngày 18 tháng 05 năm 2006, hoàn chỉnh sửa chữa ngày 16 tháng 09 năm 2007)

TÓM TẮT: *Số lượng tập phổ biến đóng (FCI) thường nhỏ hơn so với tập phổ biến. Tuy nhiên, để khai thác luật kết hợp từ chúng cần phải tìm Minimal Generator (mG) [3],[5]. Việc tiếp cận tìm mG dựa vào phương pháp sinh ứng viên như trong [3],[5] mất nhiều thời gian khi số lượng tập phổ biến lớn. Trong bài báo này chúng tôi đề xuất thuật toán MG-CHARM, một thuật toán hiệu quả để tìm tất cả các mG của các tập phổ biến đóng. Dựa vào nhận xét về các tính chất của mG ở phần 2.4, chúng tôi phát triển một thuật toán không sinh ứng viên bằng cách trực tiếp khai thác mG của một tập đóng cùng lúc với việc tạo ra nó. Vì vậy, thời gian để tìm mG của một tập đóng là không đáng kể. Thực nghiệm chứng tỏ thời gian khai thác theo MG-CHARM nhỏ hơn nhiều so với tìm mG sau khi tìm tất cả các tập đóng (CHARM) , nhất là khi số lượng tập phổ biến lớn.*

1. GIỚI THIỆU

Hiện nay, việc tìm **Minimal Generator** của tập phổ biến đóng đều dựa vào thuật toán **Apriori** như trong [3],[5].

Phương pháp thứ nhất được trình bày bởi Bastide và đồng sự trong [3], các tác giả đã mở rộng thuật toán Apriori để tìm ***mG***. Đầu tiên thuật toán tìm các ứng viên là các ***mG***, sau đó nó tính bao đóng (closure) của chúng để tìm tập đóng.

Phương pháp thứ hai được trình bày bởi Zaki trong [5]. Đầu tiên, tác giả dùng thuật toán **CHARM** [4] để tìm tất cả các tập đóng. Sau đó, ứng với mỗi tập đóng tác giả sử dụng phương pháp Apriori để tìm tất cả các ***mG*** của nó.

Cả hai phương pháp này đều gặp bất lợi khi kích thước của tập phổ biến lớn bởi vì số lượng tập ứng viên cần xét lớn.

Bài báo trình bày một phương pháp tìm nhanh ***mG*** dựa vào thuật toán **CHARM**. Nó khắc phục nhược điểm của hai phương pháp trên bằng cách dựa vào thuật toán không sinh ứng viên (**CHARM**) để sinh tập phổ biến đóng và đồng thời cũng tìm luôn ***mG*** của chúng dựa vào các nhận xét trong phần 2.4.

2. TẬP PHỔ BIẾN ĐÓNG VÀ MINIMAL GENERATOR

2.1 Một số định nghĩa [4], [5]:

2.1.1. Định nghĩa về dữ liệu giao dịch:

Cho $I = \{i_1, i_2, \ldots, i_n\}$ là tập tất cả các mục dữ liệu (ItemSet). $T = \{t_1, t_2, \ldots, t_m\}$ là tập tất cả các giao dịch (Transactions) trong CSDL giao dịch D. CSDL được cho là quan hệ hai ngôi $\delta \subseteq I \times T$. Nếu mục $i \in I$ xảy ra trong giao dịch $t \in T$ thì ta viết là: $(i, t) \in \delta$, kí hiệu: $i\delta t$.

Ví dụ: xét CSDL sau [4].

Bảng 1: CSDL mẫu ⇒ **Định dạng dữ liệu dọc**

Mã giao dịch	Nội dung giao dịch
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

Mã danh mục	Các giao dịch có chứa danh mục
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5

Giao dịch thứ hai có thể được biểu diễn là {Cδ2, Dδ2, Wδ2}.

2.1.2. Định nghĩa độ phổ biến:

Cho CSDL giao dịch D và tập dữ liệu $X \subseteq I$. Độ phổ biến của X trong D, kí hiệu $\sigma(X)$, được định nghĩa là số giao dịch mà X xuất hiện trong D.

2.1.3. Định nghĩa tập phổ biến:

$X \subseteq I$ được gọi là phổ biến nếu $\sigma(X) \geq minSup$ (với *minSup* là giá trị do người dùng chỉ định).

2.1.4. Kết nối Galois:

Cho quan hệ hai ngôi $\delta \subseteq I \times T$ chứa CSDL cần khai thác. Đặt: $X \subseteq I$ và $Y \subseteq T$. Với $P(S)$ gồm tất cả các tập con của S. Ta định nghĩa hai ánh xạ giữa $P(I)$ và $P(T)$ được gọi là *kết nối Galois* như sau:

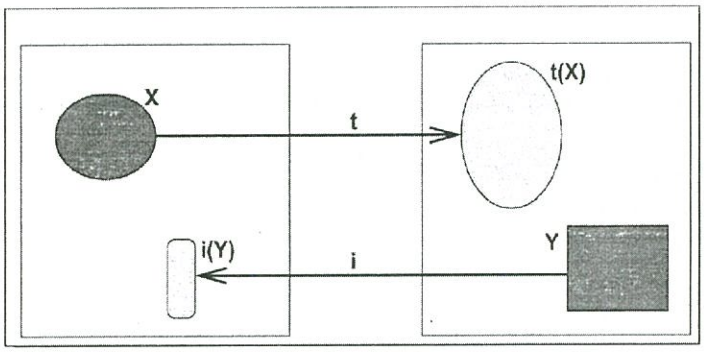
a. $t : P(I) \mapsto P(T),\ t(X) = \{y \in T \mid \forall x \in X, x\delta y\}$.

b. $i : P(T) \mapsto P(I),\ i(Y) = \{x \in I \mid \forall y \in Y, x\delta y\}$.

Hình 1 minh họa hai ánh xạ này trong đó ánh xạ $t(X)$ là tập tất cả các giao dịch có chứa itemset X (hay còn gọi là **Tidset** của **Itemset** X) và $i(Y)$ là tập tất cả các mục dữ liệu có trong tất cả các giao dịch trong Y. Kí hiệu itemset X và tập các giao dịch tương ứng với nó $t(X)$ là: $X \times T(X)$ và được gọi là **IT-pair**. Tương tự với tập giao dịch Y và $i(Y)$ là $i(Y) \times Y$.

Ví dụ: $t(ACW) = t(A) \cap t(C) \cap t(W) = 1345 \cap 123456 \cap 12345 = 1345$

$i(245) = i(2) \cap i(4) \cap i(5) = CDW \cap ACDW \cap ACDTW = CDW$



Hình 1.Kết nối Galois.

***2.1.5. Định nghĩa toán tử đóng*:**

Cho: $X \subseteq I$ và ánh xạ $c : P(I) \rightarrow P(I)$ với $c(X) = i(t(X))$. Ánh xạ c được gọi là toán tử đóng.

Ví dụ: xét CSDL được cho trong bảng 1 ta có

c(AW) = $i(t$(AW)) = i(1345) = ACW.

c(ACW) = $i(t$(ACW)) = i(1345) = ACW .

***2.1.6. Định nghĩa tập đóng*:**

Cho $X \subseteq I$. X được gọi là tập đóng khi và chỉ khi $c(X) = X$. X được gọi là tập phổ biến đóng nếu X phổ biến và X là tập đóng.

Ví dụ: xét CSDL được cho trong bảng 1 ta có:

Do: c(AW) = $i(t$(AW)) = i(1345) = ACW $\Rightarrow$ AW không phải là tập *đóng*.

Do: c(ACW) = $i(t$(ACW)) = i(1345) = ACW $\Rightarrow$ ACW là tập *đóng*.

2.2. Các tính chất của IT-pair [4]

Cho $X_i \times t(X_i)$ và $X_j \times t(X_j)$ là hai **IT-pair**. Ta có 4 tính chất sau:

1. Nếu $t(X_i) = t(X_j)$ thì $c(X_i) = c(X_j) = c(X_i \cup X_j)$.
2. Nếu $t(X_i) \subset t(X_j)$ thì $c(X_i) \neq c(X_j)$ nhưng $c(X_i) = c(X_i \cup X_j)$.
3. Nếu $t(X_i) \supset t(X_j)$ thì $c(X_i) \neq c(X_j)$ nhưng $c(X_j) = c(X_i \cup X_j)$.
4. Nếu $\begin{cases} t(X_i) \not\subseteq t(X_j) \\ t(X_i) \not\supseteq t(X_j) \end{cases}$ thì $c(X_i) \neq c(X_j) \neq c(X_i \cup X_j)$

2.3. Khái niệm Minimal Generator (mG) [5]

Cho X là tập đóng. Ta nói itemset X' là một ***generator*** của X khi và chỉ khi:

1. $X' \subseteq X$.
2. $\sigma(X) = \sigma(X')$.

Gọi $G(X)$ là tập các ***generator*** của X. Ta nói rằng $X' \in G(X)$ là một ***mG*** nếu nó không có tập con trong $G(X)$. Gọi $G^{min}(X)$ là tập tất cả các ***mG*** của X. Theo định nghĩa, $G^{min}(X) \neq \varnothing$ vì nếu nó không có ***generator*** hoàn toàn thì chính X là ***mG***.

Chẳng hạn: xét tập đóng *ACTW*, các ***generator*** là {*AT, TW, ACT, ATW, CTW*} và $G^{min}(ACTW)$={*AT, TW*}.

2.4. Một số nhận xét về *mG*

1. ***mG*** của 1-itemset là chính nó.
2. Trong quá trình áp dụng thuật toán **CHARM** để tìm tập phổ biến đóng, nếu thỏa tính chất 1 thì: $\boldsymbol{mG}(X_i \cup X_j) = \boldsymbol{mG}(X_i) + \boldsymbol{mG}(X_j)$.
3. Nếu thỏa tính chất 2 thì $\boldsymbol{mG}(X_i \cup X_j) = \boldsymbol{mG}(X_i)$.
4. Nếu thỏa tính chất 3 thì $\boldsymbol{mG}(X_i \cup X_j) = \boldsymbol{mG}(X_j)$.
5. Nếu thỏa tính chất 4 thì $\boldsymbol{mG}(X_i \cup X_j) = \cup[\boldsymbol{mG}(X_i), \boldsymbol{mG}(X_j)]$

Chứng minh:

1. Hiển nhiên vì không thể có tập con khác rỗng của tập 1-itemset.
2. Ta có $t(X_i) = t(X_j) = t(\boldsymbol{mG}(X_i)) = t(\boldsymbol{mG}(X_j)) = t(X_i \cup X_j)$
   $\Rightarrow \boldsymbol{mG}(X_i)$ và $\boldsymbol{mG}(X_j)$ là ***mG*** của $X_i \cup X_j$.
3. Ta có $t(X_i) = t(X_i \cup X_j) = t(\boldsymbol{mG}(X_i)) \neq t(\boldsymbol{mG}(X_j))$
   $\Rightarrow$ Chỉ có $\boldsymbol{mG}(X_i)$ là ***mG*** của $X_i \cup X_j$.
4. Tương tự 3.
5. Để chứng minh 5, ta cần chứng minh:
   5.1. $\cup[\boldsymbol{mG}(X_i), \boldsymbol{mG}(X_j)]$ là các **generator** của $X_i \cup X_j$.
   Thật vậy, xét $Z = \boldsymbol{mG_k}(X_i) \cup \boldsymbol{mG_l}(X_j)$ (trong đó $\boldsymbol{mG_k}(X_i) \in \boldsymbol{mG}(X_i)$, $\boldsymbol{mG_l}(X_j) \in \boldsymbol{mG}(X_j)$) ta có: do $t(\boldsymbol{mG_k}(X_i)) = t(X_i)$ và $t(\boldsymbol{mG_l}(X_j)) = t(X_j)$ nên $t(X_i \cup X_j) = t(Z) \Rightarrow Z$ là **generator** của $X_i \cup X_j$.
   5.2. $\cup[\boldsymbol{mG}(X_i), \boldsymbol{mG}(X_j)]$ là các ***mG*** của $X_i \cup X_j$.
   Thật vậy, giả sử ở mức m+1 trên cây **IT-tree**, xét $Z = \boldsymbol{mG_k}(X_i) \cup \boldsymbol{mG_l}(X_j)$ (trong đó $\boldsymbol{mG_k}(X_i) \in \boldsymbol{mG}(X_i)$, $\boldsymbol{mG_l}(X_j) \in \boldsymbol{mG}(X_j)$) ta có: $\boldsymbol{mG_k}(X_i)$ và $\boldsymbol{mG_l}(X_j)$ cùng ở mức m và chia sẻ chung (m-1)-itemset. Rõ ràng, $\boldsymbol{mG}(X_i)$ và $\boldsymbol{mG}(X_j)$ là các tập con trực tiếp của Z nên sẽ không tồn tại tập con Y của $\boldsymbol{mG_k}(X_i) \cup \boldsymbol{mG_l}(X_j)$ sao cho $t(Y) = t(Z)$ hay Z là ***mG*** của $X_i \cup X_j$.

3. THUẬT TOÁN MG-CHARM

3.1. Thuật toán

Đầu vào: CSDL D và ngưỡng phổ biến *minSup*
Kết quả: tất cả các tập phổ biến đóng thỏa *minSup* cùng với ***mG*** của chúng
Phương pháp thực hiện:

MG-CHARM(D, *minSup*)
1. $[\varnothing] = \{l_i \times t(l_i), (l_i): l_i \in I \wedge \sigma(l_i) \geq minSup\}$
2. **MG-CHARM-EXTEND**($[\varnothing]$, $\boldsymbol{C} = \varnothing$)
3. Return ***C***

MG-CHARM-EXTEND($[\boldsymbol{P}]$, ***C***)
4. for each $l_i \times t(l_i), \boldsymbol{mG}(l_i)$ in $[\boldsymbol{P}]$ do
5. $P_i = P_i \cup l_i$ and $[\boldsymbol{P_i}] = \varnothing$
6. for each $l_j \times t(l_j), \boldsymbol{mG}(l_j)$ in $[\boldsymbol{P}]$, with j > i do
7. $X = l_j$ and $Y = t(l_i) \cap t(l_j)$
8. **MG-CHARM-PROPERTY**($X \times Y, l_i, l_j, \boldsymbol{P_i}, [\boldsymbol{P_i}], [\boldsymbol{P}]$)
9. **SUBSUMPTION-CHECK**(***C***, $\boldsymbol{P_i}$)
10. **MG-CHARM-EXTEND**($[\boldsymbol{P_i}]$, ***C***)
11. delete $[\boldsymbol{P_i}]$

MG-CHARM-PROPERTY(X×Y,l_i,l_j,P_i, [P_i],[P])

12. if σ(X) ≥ *minSup* then
13. if $t(l_i) = t(l_j)$ then // tính chất 1
14. Remove l_j from P
15. $P_i = P_i \cup l_j$
16. $mG(P_i) = mG(P_i) + mG(P_j)$
17. else if $t(l_i) \subset t(l_j)$ then // tính chất 2
18. $Pi = Pi \cup lj$
19. else if $t(l_i) \supset t(l_j)$ then // tính chất 3
20. Remove l_j from [P]
21. Add X×Y, $mG(l_j)$ to [P_i]
22. else if $t(l_i) \neq t(l_j)$ then // tính chất 4
23. Add X×Y, $\cup[mG(l_i), mG(l_j)]$ to [P_i]

Trong đó hàm **SUBSUMPTION-CHECK** kiểm tra xem tập phổ biến P_i có là tập đóng hay không? Nếu đúng thì bổ sung vào C, nếu không thì loại bỏ. Nó sử dụng bảng băm để lưu C, chính vì vậy việc kiểm tra chiếm thời gian không đáng kể[4].

3.2. Nhận xét

Việc tìm ***mG*** của một tập đóng **X** theo thuật toán trên là đầy đủ.

Chứng minh:

Giả sử tồn tại một tập phổ biến **X'** sao cho:

i) **X'** ⊂ **X**

ii) σ(**X'**) = σ(**X**)

iii) **Z** ⊄ **X'**, ∀**Z** ∈ ***mG***(**X**)

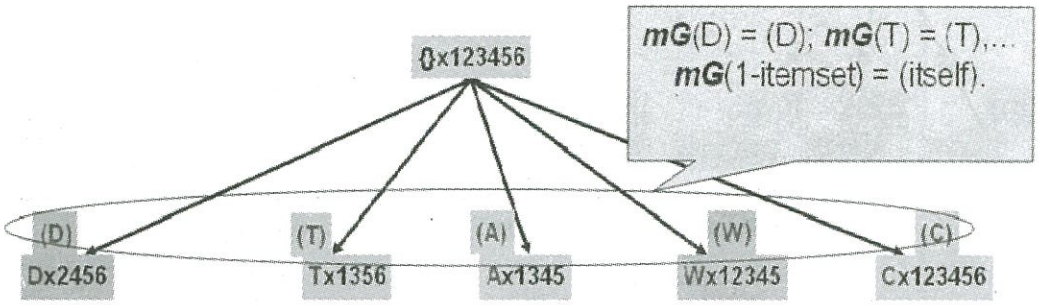
Nghĩa là **X'** là một ***minimal generator*** của **X** nhưng không thuộc về tập ***mG***(X).

Do **X'** ⊂ **X** nên **X'** không thể là tập đóng, vì vậy **X'** phải là ***generator*** của một tập đóng **Y** nào đó. Do t(**X'**) = t(**X**) và t(**X'**) = t(**Y**) nên t(**X**) = t(**Y**) và vì vậy, theo tính chất 1 của IT-pair, **X**, **Y** không là tập đóng vì c(**X**) = c(**Y**) = c(**X**∪**Y**), nghĩa là **Y** không tồn tại hay **X'** không tồn tại theo giả thuyết (đpcm).

3.3. Minh họa

Xét CSDL trong bảng 1 với *minSup* = 50%, ta có kết quả như sau:

Đầu tiên, lớp tương đương {} chứa các tập phổ biến 1-itemset. Theo nhận xét 1: ***mG*** của tập 1-itemset là chính nó như minh họa trong **hình 2**.



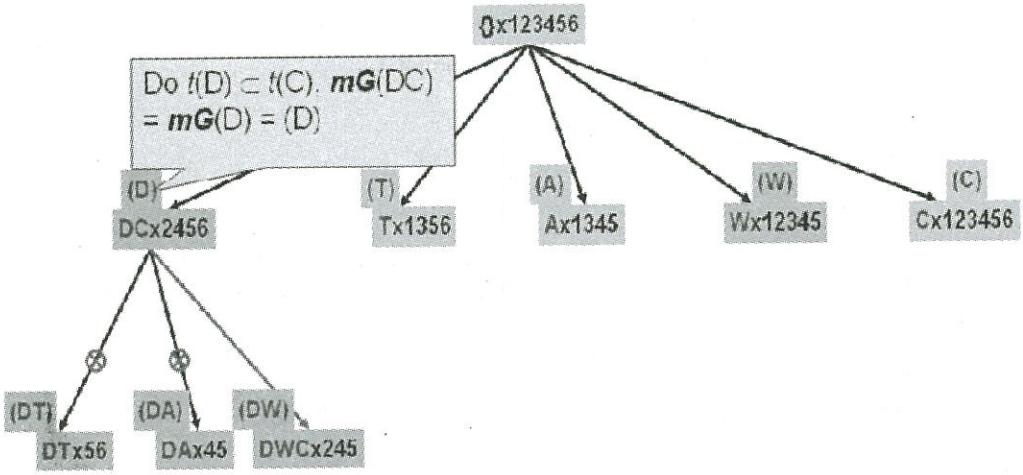
Hình 2: *mG* của tập 1-itemset

Kế tiếp, khi kết hợp hai **IT-pair** D×2456 và T×1356 thành DT×56 ta có σ(DT) = |56| = 2 < *minSup* ⇒ loại. Tương tự cho DA.

Kế tiếp, khi kết hợp hai **IT-pair** D×2456 và T×1356 thành DT×56 ta có σ(DT) = |56| = 2 < *minSup* ⇒ loại. Tương tự cho DA.

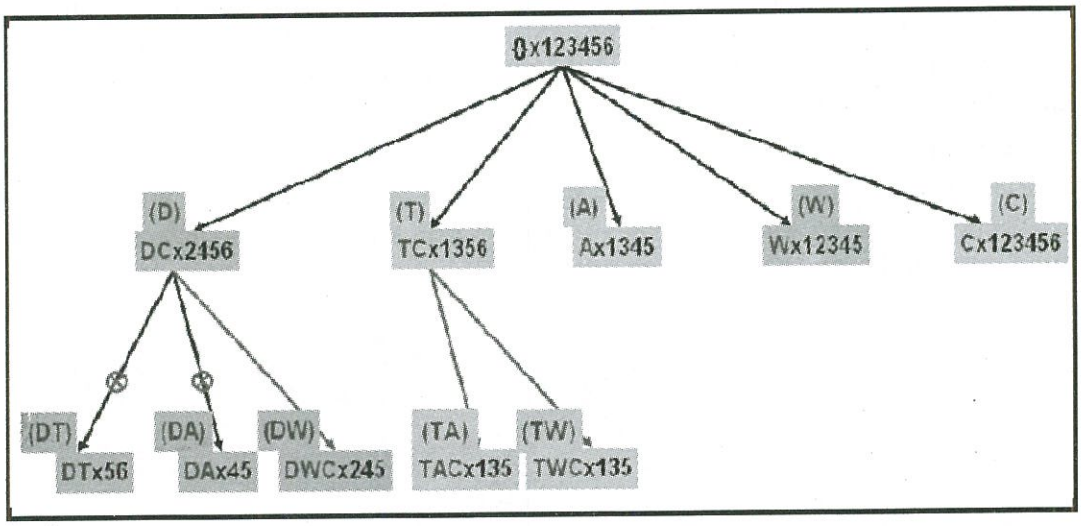
Xét DW, do $t(D) \neq t(W)$ nên theo nhận xét 5 ta có $\boldsymbol{mG}(DW) = \boldsymbol{mG}(D) \cup \boldsymbol{mG}(W) = DW$.

Xét DC, do $t(D) \subset t(C)$ nên theo nhận xét 3 ta có $\boldsymbol{mG}(DC) = \boldsymbol{mG}(D)$.



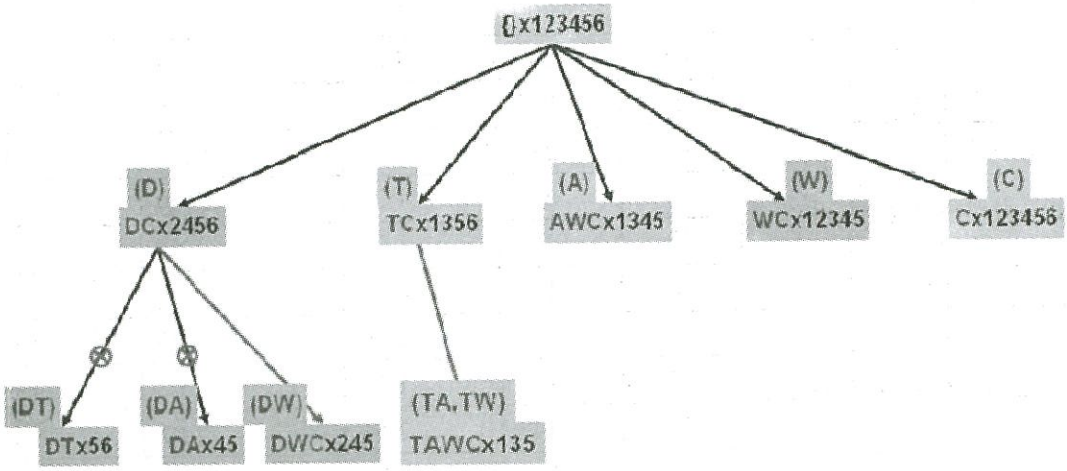
Hình 3: Minh họa việc cập nhật mG khi thỏa nhận xét 3

Để minh họa cho nhận xét 4, ta xét kết quả sau khi thực hiện việc kết hợp T với các **IT-pair** sau nó như hình 4.



Hình 4: Kết quả sau khi kết hợp T với các IT-pair đứng sau nó

Xét việc kết hợp TAC với TWC ta có: do $t(TAC) = t(TWC)$ nên $\boldsymbol{mG}(TAWC) = \boldsymbol{mG}(TAC) + \boldsymbol{mG}(TWC) = \{TA, TW\}$. Ta có cây biểu diễn kết quả cuối cùng trong hình 5.



Hình 5: Cây tìm kiếm IT-tree sau khi thực hiện MG-CHARM
(trong ngoặc là mG của tập đóng tương ứng)

4. KẾT QUẢ THỰC NGHIỆM

Kết quả thực nghiệm được thực hiện với nhiều CSDL có đặc điểm khác nhau được lấy từ: http://fimi.cs.helsinki.fi/data. Chương trình được cài đặt bằng ngôn ngữ VC7, hệ điều hành Windows XP, CPU: 1GHz, RAM: 384MB.

Bảng 2: Đặc điểm của các CSDL thử nghiệm

Tên CSDL	Số giao dịch	Số danh mục	Độ dài trung bình	Độ dài tối đa
Chess	3196	76	37	37
Mushroom	8124	120	23	23
Pumsb*	49046	7117	50.48	63
Pumsb	49046	7117	74	74
Connect	67557	130	43	43
Accidents	340183	468	33.8	51

Bảng 3: Kết quả thực nghiệm so sánh CHARM và MG-CHARM

Tên CSDL	Số Items	Số Trans	*minSup* (%)	Số lượng **FCI**	Thời gian thực hiện		tỉ lệ (1)/(2)
					CHARM (1)	MG-CHARM (2)	
Chess	76	3196	80	5083	5.52	0.24	23
			75	11525	40.11	0.43	93.28
			70	23892	120.8	1.05	115.05
			65	49034	651.12	2.44	266.85
Mushroom	120	8124	40	140	0.32	0.25	1.28
			30	427	0.45	0.4	1.13

			20	1200	1.29	1.03	1.25
			10	4095	7.82	2.54	3.08
			5	12940	56.03	5.55	10.1
Pumsb	7117	49046	94	216	0.69	0.66	1.05
			92	610	1.21	1.09	1.11
			90	1465	1.77	1.52	1.16
			88	3160	4.05	1.88	2.15
Pumsb*	7117	49046	60	68	0.69	0.65	1.06
			55	116	1.11	1.02	1.09
			50	248	2.51	2.01	1.25
			45	713	4.72	3.62	1.3
Connect	130	67557	95	811	1.32	1.16	1.14
			90	3486	5.91	2.33	2.54
			85	8252	28.24	4.56	6.19
			80	15107	100.77	5.86	17.2
Accidents	468	340183	80	149	2.55	2.5	1.02
			70	529	6.39	5.7	1.12
			60	2074	16.21	15.18	1.07
			50	8057	49.34	31	1.59

(1) Thời gian tìm **FCI** theo **CHARM** + thời gian tìm ***mG*** của các **FCI**
(2) Thời gian thực hiện **MG-CHARM**

Kết quả ở bảng 3 cho thấy thời gian thực hiện của **MG-CHARM** nhỏ hơn so với **CHARM** kết hợp với tìm ***mG*** theo phương pháp của M. J. Zaki. Chẳng hạn, xét CSDL chess với minSup = 65%, **MG-CHARM** nhanh hơn gấp 266 lần. Số lượng tập phổ biến đóng càng lớn, tỉ lệ này càng cao.

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Bài báo đưa ra một phương pháp mới để tìm ***mG*** của tập phổ biến đóng mà không cần phát sinh ứng viên. Thực nghiệm chứng tỏ thời gian để cập nhật ***mG*** của các tập đóng là không đáng kể. Đặc biệt là ứng với các trường hợp kích thước của tập đóng lớn, thời gian cập nhật này nhỏ hơn nhiều so với phương pháp sinh ứng viên trong [3],[5].

Trong tương lai, chúng tôi sẽ sử dụng kết quả này cho bài toán khai thác tập luật không dư thừa, truy vấn luật không dư thừa,....

FAST ALGORITHM TO FIND MINIMAL GENERATOR

Le Hoai Bac, Vo Dinh Bay
University of Natural Sciences, VNU- HCM

ABSTRACT**: Number of frequent closed itemsets* (**FCI**) *is usually fewer than frequent itemsets. However, it is necessary to find Minimal Generator* (mG***) *for mining association rule from them [3],[5]. Finding **mG** approach based on the method of generating candidate is very time-consuming when number of frequent itemsets is large. In this paper, we present* **MG-CHARM**, *an efficient algorithm to find all **mG** of frequent closed itemsets. Based on the considering of **mG** features mentioned in 2.4 section we develop an algorithm which do not generate candidate by mining directly **mG** of closed itemsets at the same time of generating. Thus, the time for finding **mG** of closed itemsets is insignificant. Experiment shows that the time of* **MG-CHARM** *mining method is significant fewer than the time of finding **mG** after finding all closed itemsets* (**CHARM**), *especially in case the frequent itemsets number is large.*

TÀI LIỆU THAM KHẢO

[1]. R. Agrawal and R. Srikant, *Fast algorithms for mining association rules*, In VLDB'94, (1994).

[2]. R. Agrawal, T. Imielinski, and A. Swami, *Mining association rules between sets of items in large databases*, In SIGMOD'93, (1993).

[3]. Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, L. Lakhal, *Mining Minimal Non-Redundant Association Rules using Closed Frequent Itemssets*, In 1st International Conference on Computational Logic, (2000).

[4]. M. J. Zaki, C.J. Hsiao, Efficient Algorithms for Mining Closed Itemsets and Their Lattice Structure, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No 4, April (2005).

[5]. M. J. Zaki, *Mining Non-Redundant Association Rules*, Data Mining and Knowledge Discovery, 9, 223–248, 2004 Kluwer Academic Publishers. Manufactured in The Netherlands, (2004).

[6]. M. J. Zaki and K. Gouda, *Fast Vertical Mining Using Diffsets*, Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, Aug. (2003).

[7]. M. J. Zaki and B. Phoophakdee, *MIRAGE: A Framework for Mining, Exploring, and Visualizing Minimal Association Rules*, Technical Report 03-4, Computer Science Dept., Rensselaer Polytechnic Inst., July (2003).

[8]. M. J. Zaki, *Generating Non-Redundant Association Rules*, In 6th ACM SIGKDD Intl Conf. Knowledge Discovery and Data Mining. (2001).

[9]. M. J. Zaki, M. Ogihana, *Theoretical Foundations of Association Rules*, In 3rd ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, (1998).