

ỨNG DỤNG DÀN TẬP ĐÓNG TRONG KHAI THÁC LUẬT THIẾT YẾU NHẤT

Lê Hoài Bắc⁽¹⁾, Võ Đình Bảy⁽²⁾

(1) Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM

(2) Trường Đại học Kỹ thuật Công nghệ Tp. HCM

Tóm tắt: Theo cách khai thác luật kết hợp truyền thống, việc tìm tất cả các luật kết hợp từ CSDL thỏa minSup và minConf gặp nhiều bất lợi khi số tập phổ biến lớn. Do đó cần có một phương pháp thích hợp để khai thác với số luật ít hơn nhưng vẫn bảo đảm tích hợp đầy đủ tất cả các luật của phương pháp khai thác truyền thống. Một trong những cách tiếp cận đó là khai thác luật thiết yếu nhất (Essential rules): chỉ lưu lại các luật có về trái tối thiểu và về phải tối đại (theo quan hệ cha – con). Trong bài báo này, chúng tôi đề nghị một thuật toán mới để khai thác luật thiết yếu nhất từ dàn tập phổ biến đóng với mong muốn làm giảm thời gian khai thác luật. Chúng tôi sử dụng quan hệ cha – con trên dàn để giảm chi phí xét quan hệ cha con và vì vậy làm giảm được thời gian khai thác luật.

Từ khóa: Tập phổ biến, tập phổ biến đóng, Minimal generator, luật truyền thống, luật thiết yếu nhất, dàn tập đóng, khai thác luật từ dàn.

1. GIỚI THIỆU

Trong hầu hết các thuật toán khai thác luật, các tác giả đặc biệt chú ý đến vấn đề làm thế nào để khai thác nhanh tập phổ biến. Chính vì vậy, có khá nhiều tác giả chỉ tập trung vào việc nghiên cứu nhằm tìm ra thuật toán hiệu quả nhất cho bài toán tìm tập phổ biến. Tuy nhiên, với các CSDL đặc (mật độ trùng lặp các item giữa các dòng dữ liệu cao) hoặc khi minSup nhỏ dẫn đến số lượng tập phổ biến khá lớn thì thời gian khai thác và khối lượng bộ nhớ yêu cầu để lưu trữ tập phổ biến và luật kết hợp khá lớn – Vì vậy, các tác giả M. Zaki [12] và Y. Bastide [6] đã đưa ra một cách tiếp cận mới nhằm làm giảm khối lượng lưu trữ và thời gian khai thác: đó chính là khai thác luật kết hợp không dư thừa dựa vào tập đóng. Cách tiếp cận này có ưu điểm là số luật kết hợp giảm đáng kể so với phương pháp truyền thống nhưng vẫn bảo đảm tích hợp đầy đủ các luật còn lại. Do muốn bảo toàn thông tin về độ phổ biến (support) và độ tin cậy (confidence) của luật nên cả hai đều chỉ rút gọn trên các tập luật có cùng độ phổ biến và độ tin cậy. Tuy nhiên, khi người dùng muốn khai thác tập các luật thỏa minSup và minConf (nhưng không cần biết thông tin về độ phổ biến và độ tin cậy của từng luật), làm thế nào để khai thác tập luật nhỏ nhất thỏa mãn yêu cầu người dùng? Gần đây, các tác giả T. Xia, Y. Du, J. Shan, D. Zhang trong [10] đề xuất phương pháp khai thác luật thiết yếu nhất dựa vào tập phổ biến tối đại nhằm giới hạn không gian lưu trữ và thời gian khai thác so với phương pháp của Aggarwal và Yu [3]. Nhưng do khai thác trực tiếp trên tập phổ biến tối đại nên việc tính độ phổ biến của các tập con mất nhiều thời gian do phải đọc nhiều lần CSDL.

Trong [5], chúng tôi đề xuất thuật toán khai thác luật thiết yếu nhất dựa vào tập phổ biến đóng. Cách tiếp cận này nhằm tránh việc đọc CSDL nhiều lần, chỉ cần dựa vào minimal generator của tập đóng X và tập đóng Y để sinh luật. Việc ứng dụng tập phổ biến đóng để khai thác luật thiết yếu nhất đã mang lại hiệu quả nhất định. Tuy nhiên, thuật toán khai thác luật thiết yếu nhất trong [5] vẫn tốn nhiều thời gian cho việc xét quan hệ cha – con giữa các tập đóng với nhau. Vì vậy, chúng tôi muốn thử nghiệm dùng dàn tập phổ biến đóng cho việc khai thác luật thiết yếu nhất nhằm làm giảm chi phí xét quan hệ cha con này.

2. KHAI THÁC TẬP PHỔ BIẾN ĐÓNG

Để khai thác tập phổ biến đóng, chúng tôi sử dụng thuật toán **CHARM** được trình bày trong [11]. **CHARM** có ưu điểm là không sinh ứng viên và dựa vào phương pháp chia để trị nhằm tìm kiếm các tập phổ biến đóng với chỉ một lần đọc CSDL.

3. XÂY DỰNG DÀN TẬP PHỔ BIẾN ĐÓNG

Xây dựng dàn tập phổ biến đóng là xây dựng quan hệ cha con (trực tiếp) giữa các tập phổ biến đóng với nhau. Do vậy sẽ tiết kiệm được thời gian khi duyệt dàn để sinh luật so với cách sinh luật trong [5]. Có nhiều thuật toán xây dựng dàn khái niệm được trình bày trong [7, 10]. Tuy nhiên, hầu hết chúng được xây dựng dưới dạng tăng cường nên không thích hợp cho việc xây dựng dàn tập đóng trong trường hợp CSDL cố định vì như thế sẽ mất nhiều thời gian. Trong [11], các tác giả Zaki và Hsiao trình bày thuật toán **CHARM-L** mở rộng trực tiếp từ thuật toán **CHARM** để sinh ra dàn tập đóng. Do sử dụng phương pháp tìm phần giao của hai Cidset (Closed Identification Set) để tìm quan hệ cha – con nên cách tiếp cận này có hạn chế là tốn bộ nhớ để lưu Cidset và tốn thời gian để tính phần giao của Cidset khi số lượng tập đóng lớn. Chính vì vậy, chúng tôi đề nghị thuật toán xây dựng dàn sau khi tìm tập đóng. Thuật toán đề nghị có thể chậm hơn so với **CHARM-L** nhưng bộ nhớ yêu cầu để lưu quan hệ trên dàn là không đáng kể so với **CHARM-L**.

3.1. Thuật toán xây dựng dàn tập đóng

Đầu vào: tập FCI thỏa ngưỡng phổ biến *minSup*.

Kết quả: dàn FCI chứa tất cả các tập phổ biến đóng của CSDL D.

Phương pháp thực hiện:

```

BUILD-LATTICE()
  SORT (FCI) // Sắp xếp FCI tăng theo k-itemset
  Ir = ∅
  for each Y ∈ FCI do
    INSERT_LATTICE(Y, Ir)

INSERT_LATTICE(Y, Ir)
  if Ir.daDuyet = false then
    Ir.daDuyet = true
    f = true
    for each {X} ∈ Ir.Children do
      if X ⊂ Y then
        f = false
        INSERT_LATTICE(Y, {X})
  if f = true then
    Ir.Children.Add({Y})
    
```

Hình 1. Thuật toán xây dựng dàn tập đóng

Quá trình xây dựng dàn chỉ đơn giản là chèn tất cả các tập phổ biến đóng Y thuộc FCI vào dàn (ban đầu được khởi tạo bằng rỗng). Như vậy, để xem xét thuật toán, ta chỉ cần xét hàm **INSERT_LATTICE**: hàm này nhận đầu vào là nút dàn cần chèn $\{Y\}$ và nút gốc dàn Ir , có 2 khả năng xảy ra như sau:

1. Nếu Y là tập cha của bất kì nút con lc nào của Ir thì chèn Y vào dàn với gốc là lc .

2. Nếu tất cả các nút con của Ir không là tập con của Y ($f = \text{true}$) thì $\{Y\}$ là nút con trực tiếp của $Ir \Rightarrow \{Y\}$ là nút đàn con của nút gốc Ir .

3.2. Minh họa

Xét CSDL sau:

Bảng 1. CSDL mẫu \Rightarrow Định dạng dữ liệu đọc

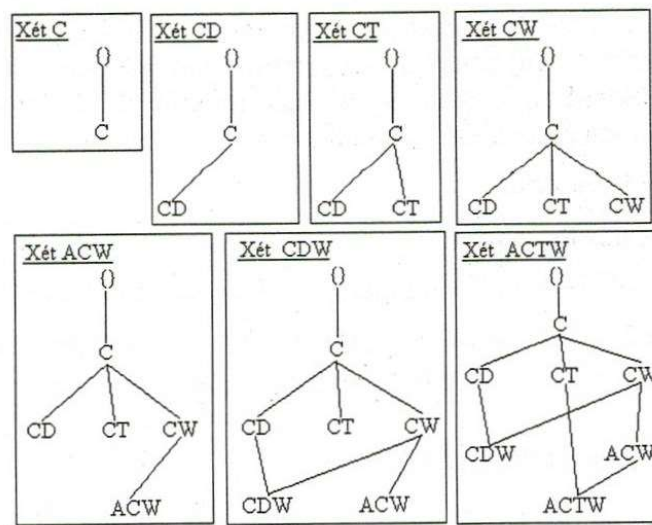
Mã giao dịch	Nội dung giao dịch	Mã danh mục	Các giao dịch có chứa danh mục
1	A, C, T, W	A	1, 3, 4, 5
2	C, D, W	C	1, 2, 3, 4, 5, 6
3	A, C, T, W	D	2, 4, 5, 6
4	A, C, D, W	T	1, 3, 5, 6
5	A, C, D, T, W	W	1, 2, 3, 4, 5
6	C, D, T		

Xét các tập đóng trong CSDL của Bảng 1 theo thuật toán **CHARM** với $minSup = 50\%$ sau khi được sắp xếp như sau:

Bảng 2. Danh sách các tập phổ biến đóng

FCI	C	CD	CT	CW	ACW	CDW	ACTW
sup	6	4	4	5	4	3	3

Quá trình xây dựng dàn được minh họa trong hình 2.



Hình 2. Minh họa quá trình chèn tập đóng vào dàn

Xét tập đóng CDW với các nút con của $\{ \}$: đầu tiên nó là tập cha của tập đóng C , vì vậy nút CDW được chèn vào con của nút C . Tiếp tục xét CDW với các nút con của C : do CDW là tập cha của CD và CW nên nó được chèn vào con của nút CD và nút CW . Do các nút con của nút CD, CW không là tập con của CDW nên CDW là nút con trực tiếp của các nút $CD, CW \Rightarrow$ dừng.

4. LUẬT KẾT HỢP THIẾT YẾU NHẤT

4.1. Luật kết hợp

4.1.1. Định nghĩa 1 [12]

Luật kết hợp là phép kéo theo có dạng $Y \xrightarrow{q,p} X - Y$ (X, Y là các tập phổ biến) trong đó $Y \subset X, Y \neq \emptyset$ và $p = \frac{Sup(X)}{Sup(Y)} \geq minConf$ gọi là độ tin cậy của luật còn $q = Sup(X)$ gọi là độ phổ biến của luật. Do X là phổ biến nên luật sinh ra là phổ biến.

4.1.2. Tính chất

1. Nếu $X \rightarrow Y$ là luật kết hợp thì $X \cup A \rightarrow Y \setminus A$ cũng là luật kết hợp $\forall A \subset Y$.
2. Nếu $X \rightarrow Y$ là luật kết hợp thì $X \rightarrow Y \setminus A$ cũng là luật kết hợp $\forall A \subset Y$.
3. Nếu $X \rightarrow Y$ không là luật kết hợp thì $X \setminus A \rightarrow Y \cup A$ cũng không là luật kết hợp $\forall A \subset X$.

4.2. Minimal Generator (mG)

Định nghĩa 2 [12]: Cho X là tập đóng. Ta nói itemset X' là một **generator** của X khi và chỉ khi:

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

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 $G(X)$. Đặt $G^{min}(X)$ là tập tất cả các **mG** của X . Theo định nghĩa, $G^{min}(X) \neq \emptyset$ 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 $G^{min}(ACTW) = \{AT, TW\} \Rightarrow mG(ACTW) = .$

Thuật toán tìm **mG** của tập đóng được trình bày trong [6, 12], các thuật toán này dựa vào nguyên lý Apriori (sinh ứng viên) nên làm cho quá trình tìm **mG** chậm. Vì vậy, chúng tôi dựa vào thuật toán **MG-CHARM** được chúng tôi đề xuất trong [4] để khai thác tập đóng cùng với **mG** của chúng bằng phương pháp không sinh ứng viên.

4.3. Khai thác luật thiết yếu nhất

4.3.1. Định nghĩa 3 – Luật tổng quát

Cho R_i chỉ luật $X_i \xrightarrow{q_i,p_i} Y_i$; Ta nói luật R_1 là tổng quát hơn luật R_2 , kí hiệu $R_1 \prec R_2$, nếu $X_1 \subseteq X_2, Y_1 \supseteq Y_2$ và $X_1 \cup Y_1 \subset X_2 \cup Y_2$, nghĩa là R_2 có thể được sinh ra từ R_1 bằng cách thêm các item vào về trái hoặc giảm bớt các item ở về phải của R_1 đồng thời nó phải là luật chứa trong R_1 .

4.3.2. Định nghĩa 4 – Tập luật thiết yếu nhất

Cho $R = \{R_1, \dots, R_n\} (R_i = X_i \xrightarrow{q_i,p_i} Y_i)$ là tập tất cả các luật kết hợp truyền thống. Đặt $R_E = \{R_i \in R: (\neg \exists R_j \in R: R_j \prec R_i)\}$, R_E được gọi là tập luật thiết yếu nhất của R .

4.3.3. Nhận xét

1. Các luật thiết yếu nhất được suy từ **Minimal Generator** của tập đóng X sang tập đóng Y trong đó $X \subseteq Y$.
2. Luật thiết yếu nhất có độ tin cậy 100% được suy từ **Minimal Generator** của tập đóng X sang chính X .
3. Nếu $X \subset Y$ và $X \rightarrow Y - X (R_1)$ là luật thiết yếu nhất thì

- (a) $X \rightarrow Z - X (R_2)$ và
- (b) $Z \rightarrow Y - Z (R_3)$ không là luật thiết yếu nhất $\forall Z: X \subset Z \subset Y$.

Chứng minh:

- (a) Do $X \subseteq X$ và $Y - X \supset Z - X$ (vì $Z \subset Y$) nên theo định nghĩa 3, $R_1 \prec R_2 \Rightarrow R_2$ không là luật thiết yếu nhất.
- (b) Do $X \subset Z$ và $Y - X \supset Y - Z$ (vì $X \subset Z$) nên theo định nghĩa 3, $R_1 \prec R_3 \Rightarrow R_3$ không là luật thiết yếu nhất.

4.3.4. Thuật toán sinh tập luật thiết yếu nhất dựa vào dàn tập phổ biến đóng

Như đã được đề cập ở phần 1, hiện nay có nhiều thuật toán nhằm khai thác luật thiết yếu nhất [3, 5, 8]. Trong [3], các tác giả dựa vào tập phổ biến để khai thác luật. Đến năm 2004, các tác giả trong [8] sử dụng tập phổ biến tối đại và đến đầu năm 2008, thuật toán khai thác luật thiết yếu nhất dựa vào tập phổ biến đóng được trình bày [5]. Mục tiêu của các cải tiến này là nhằm tăng tốc độ khai thác luật. Bài báo này cũng nhằm mục đích tăng tốc độ khai thác luật bằng cách dựa vào dàn tập phổ biến đóng.

Đầu vào: dàn chứa các tập phổ biến đóng thỏa ngưỡng phổ biến *minSup* với nút gốc L_r và ngưỡng tin cậy *minConf*.

Kết quả: tập AR gồm tất cả các luật thiết yếu nhất thỏa *minConf*.

Phương pháp thực hiện: được trình bày chi tiết trong Hình 3.

Thuật toán xét tất cả các nút con của nút gốc dàn và gọi hàm **ENUMERATE_RULE**(L_c), hàm này duyệt qua tất cả các nút con của nút hiện hành L_c để tìm luật. Dựa vào nhận xét 4.3.3, hàm tìm luật sẽ xét các nút con L (trực tiếp) của L_c để kiểm tra luật tạo ra giữa L_c và L có thỏa *minConf* hay không? Nếu có thì luật tạo ra giữa L_c và L_s không là luật thiết yếu nhất. Ngược lại thì tạo ra luật giữa $mG(L_c)$ và L_s nếu các luật này thỏa *minConf*.

```

MINING_ESSENTIAL_AR_LATTICE( $L_r$ )
  AR =  $\emptyset$ 
  for each  $L_c \in L_r.Children$  do
    EXTEND_AR_LATTICE( $L_c$ )
  return AR

AR_LATTICE( $L_c$ )
  if  $L_c.flag = False$  then
    ENUMERATE_RULE( $L_c, L_c$ )
     $L_c.flag = True$ 
  for each  $L_s \in L_c.Children$  do
    AR_LATTICE( $L_s$ )

ENUMERATE_RULE( $L_c$ )
  For all  $L_s \in ChildrenNode(L_c)$  do
    FIND_RULE( $L_c, L_s$ )

FIND_RULE( $L_c, L_s$ )
  For all  $L \in L_s.Children$  do
    if  $Sup(L) / Sup(L_c) \geq minConf$  then
      return
  conf =  $Sup(L_s) / Sup(L_c)$ 
  if conf  $\geq minConf$  then
    for all  $Z \in mG(L_c)$  do
      if ESSENTIAL_RULE( $Z, L_s.itemset \setminus Z$ ) then
         $AR = AR \cup \{Z \rightarrow L_s.itemset \setminus Z (Sup(L_s), conf)\}$ 
    
```

Hình 3. Thuật toán sinh luật thiết yếu nhất từ dàn tập phổ biến đóng

4.3.5. Minh họa thuật toán

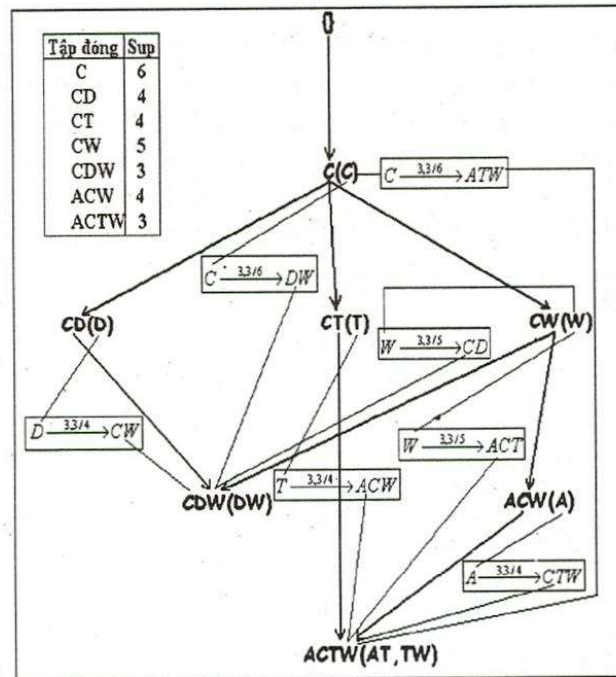
Xét CSDL trong bảng 1 với $minSup = 50\%$, $minConf = 50\%$: kết quả có 7 luật thiết yếu nhất trong khi số luật truyền thống là 60. Tỷ lệ tích hợp luật = $7/60 * 100\% = 11.67\%$.

5. KẾT QUẢ THỰC NGHIỆM

Các CSDL thực nghiệm chuẩn được lấy từ [9] có đặc điểm như sau:

Bảng 3. Đặc điểm của các CSDL chuẩn

Tên CSDL	Số giao dịch	Số danh mục	Độ dài trung bình	Độ dài tối đa
Chess	3196	75	37	37
Mushroom	8124	120	23	23
Pumsb*	49046	7117	50	62
Pumsb	49046	7117	73.6	74
Connect	67557	130	43	43
Retail	88162	16469	10.3	76
Accidents	340183	468	33.8	51



Hình 4. Kết quả khai thác luật thiết yếu nhất từ dàn tập phổ biến đóng

Kết quả từ bảng 4 cho thấy thời gian khai thác luật từ dàn thường nhanh hơn so với khai thác trực tiếp từ tập phổ biến đóng ((4) so với (2)). Nếu tính luôn cả việc khai thác tập phổ biến đóng và xây dựng dàn thì tổng thời gian khai thác từ dàn có chậm hơn đôi chút so với khai thác trực tiếp. Tuy nhiên, nếu chúng ta tiếp tục khai thác trên dàn với các $minConf$ khác thì thời gian khai thác từ dàn sẽ hiệu quả hơn.

6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Bài báo trình bày một cách khai thác luật thiết yếu nhất từ dàn tập đóng. Có thể thấy thời gian khai thác luật từ dàn thường nhỏ hơn so với khai thác trực tiếp từ tập phổ biến đóng. Do

đó, có thể nói dàn là một công cụ hữu hiệu để khai thác luật kết hợp. Vì vậy, cần có sự nghiên cứu sâu về khía cạnh toán học của dàn để áp dụng hiệu quả vào khai thác dữ liệu nói chung và luật kết hợp nói riêng. Bên cạnh đó, chất lượng tập luật cũng sẽ được quan tâm nhằm tìm kiếm các luật đáp ứng các yêu cầu khai thác của người sử dụng chúng.

Việc phát triển thuật toán hiệu quả để xây dựng dàn tập phổ biến đóng cũng được quan tâm nhằm làm giảm thời gian khai thác luật.

Bảng 4. Kết quả thực nghiệm trên các CSDL với $minConf = 50\%$

CSDL	$minSup$ (%)	#FCI	#Luật	Theo FCI (giây) [5]		Theo dàn (giây)		Tỉ lệ $\frac{(3)}{(1)} \times 100\%$	Tỉ lệ $\frac{(4)}{(2)} \times 100\%$
				FCI + Luật (1)	Luật (2)	FCI + Dàn + Luật (3)	Luật (4)		
Chess	80	5083	1678	7.78	7.61	8.12	5.03	104.4	66.1
	70	23892	7466	270.28	255.16	285.31	183.5	105.6	71.9
Mushroom	30	427	294	0.39	0.1	0.42	0.09	107.7	90
	20	688	879	1.13	0.52	1.36	0.5	120.4	96.2
Pumsb	95	110	119	0.34	0.1	0.33	0.02	97.1	20
	90	1465	1389	2.17	1.41	2.66	1.47	122.6	104.3
Pumsb*	50	248	107	1.23	0.016	1.22	0.021	99.2	131.3
	45	713	246	2.67	0.17	2.55	0.05	95.5	29.4
Accident	70	529	126	3.75	0.08	3.71	0.03	98.9	37.5
	60	2074	486	8.91	1.05	8.84	0.36	99.2	34.3
Retail	0.4	831	441	9.42	0.11	9.42	0.02	100	18.2
	0.2	2690	1368	42.81	1.53	42.75	0.27	99.9	17.6
connect	97	284	230	0.36	0.03	0.39	0.03	108.3	100
	92	2212	1443	4.36	2.94	4.38	2.88	100.5	98

MINING ESSENTIAL RULES USING FREQUENT CLOSED ITEMSETS LATTICE

Le Hoai Bac⁽¹⁾, Vo Dinh Bay⁽²⁾

(1) University of Science, VNU-HCM

(2) University of technology, VNU-HCM

ABSTRAC: In traditional mining of association rules, finding all association rules from databases that satisfy $minSup$ and $minConf$ faces with some problems in case of the number of frequent itemsets is large. Thus, it is necessary to have a suitable method for mining fewer rules but they still embrace all rules of traditional mining method. One of the approaches that is the mining method of essential rules: it only keeps the rule that its left hand side is minimal and its right side is maximal (follow in parent-child relationship). In this paper, we propose a new algorithm for mining the essential rules from the frequent closed itemsets lattice to reduce the time of mining rules. We use the parent-child relationship in lattice to reduce the cost of considering parent-child relationship and lead to reduce the time of mining rules.

Keywords: association rules, essential rules, minimal generator, the frequent closed itemsets lattice, mining rules.

TÀI LIỆU THAM KHẢO

- [1]. R. Agrawal, T. Imielinski, and A. Swami, Mining association rules between sets of items in large databases, Proceedings of the 1993 ACM SIGMOD Conference Washington DC, USA, May 1993, 207 – 216 (1993).
- [2]. R. Agrawal and R. Srikant, Fast algorithms for mining association rules, In VLDB'94 (1994).
- [3]. C. C. Aggarwal and P. S. Yu, Online Generation of Association Rules, In Proceedings of International Conference on Data Engineering (ICDE), 402 – 411 (1998).
- [4]. Lê Hoài Bắc, Võ Đình Bảy, Thuật toán tìm nhanh Minimal Generator của tập phổ biến đóng, Tạp chí Phát triển Khoa học và Công nghệ, Đại học Quốc gia Tp. HCM, tập 10, số 12, 11-19 (2007).
- [5]. Lê Hoài Bắc, Võ Đình Bảy, Khai thác luật thiết yếu nhất từ tập phổ biến đóng, Tạp chí Phát triển Khoa học và Công nghệ, Đại học Quốc gia Tp. HCM, tập 11, số 1, 40-50 (2008).
- [6]. Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, L. Lakhal, Mining Minimal Non-Redundant Association Rules using Closed Frequent Itemsets, In 1st International Conference on Computational Logic, 972 – 986 (2000).
- [7]. V. Choi, Faster Algorithms for Constructing a Concept (Galois) Lattice, arXiv:cs.DM/0602069 (2006).
- [8]. T. Xia, Y. Du, J. Shan and D. Zhang, ERMiner: A Novel Method to Mine Essential Rules, ACM SIGKDD'04 August 2225, 2004. Seattle, USA, (2004).
- [9]. <http://fimi.cs.helsinki.fi/data/> (download on april 2005).
- [10]. S. O. Kuznetsov, S. A. Obiedkov, Comparing performance of algorithms for generating concept lattices, J. Exp. Theor. Artif. Intell. 14(2-3), 189-216 (2002)
- [11]. 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, 462-478 (2005).
- [12]. M. J. Zaki, Mining Non-Redundant Association Rules, Data Mining and Knowledge Discovery, 9, Kluwer Academic Publishers. Manufactured in The Netherlands, 223–248 (2004).
- [13]. 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, 326 – 335 (2003).