SỬ DỤNG DIFFSET ĐỂ KHAI THÁC TẬP ĐÓNG ĐƯỢC GÁN TRỌNG PHỔ BIẾN TRÊN CƠ SỞ DỮ LIỆU SỐ LƯỢNG

Trần Như Ý*, Nguyễn Văn Tùng, Ngô Dương Hà

Trường Đại học Công nghiệp Thực phẩm TP.HCM

*Email: *ytn@cntp.edu.vn*

Ngày nhận bài: 09/11/2016 ; Ngày chấp nhận đăng: 12/04/2017

TÓM TẮT

Khai thác tập phổ biến đóng vai trò quan trọng trong khai thác luật kết hợp. Đối với cơ sở dữ liệu số lượng, khai thác tập đóng được gán trọng phổ biến (FWCIs) là một trong những phương pháp khai thác tập phổ biến đã được tác giả đề xuất. Tuy nhiên đối với cơ sở dữ liệu dày đặc, thời gian khai thác tập phổ biến (FWCIs) vẫn còn cao. Trong bài báo này, tác giả đề xuất thuật toán sử dụng diffset để khai thác tập đóng được gán trọng phổ biến (FWCIs-DIFF). Dựa trên cơ sở các định lý và tính chất, tác giả đề xuất thuật toán (FWCIs-DIFF). Kết quả thực nghiệm cho thấy, với cơ sở dữ liệu dày đặc thời gian khai thác của (FWCIs-DIFF) là nhanh hơn so với (FWCIs).

Từ khóa: khai thác tập phổ biến, khai thác tập đóng được gán trọng phổ biến, diffset.

1. GIỚI THIỆU

Điều kiện chặt hơn của tập đóng phổ biến so với tập phổ biến làm giảm đáng kể số lượng tập được sinh ra, và vì vậy khai thác luật từ tập đóng phổ biến sẽ hiệu quả hơn. Khái niệm tập đóng phổ biến được đưa ra lần đầu tiên vào năm 1999 [1] bởi Pasquier và đồng sự.

Về sau này, thuật toán được sử dụng nhiều nhất là CHARM [2]. Vào năm 2013, Võ Đình Bảy, Frans Coenen, Lê Hoài Bắc đã đưa ra thuật toán khai thác tập được gán trọng phổ biến (FWIs) [3]. Cuối năm 2013, Võ Đình Bảy, Ngô Dương Hà, Trần Như Ý đã đưa ra thuật toán (FWCIs) [4].

Dựa trên WIT-tree [1], FWCIs [4], tính chất của IT-pair trên cơ sở Diffset [2], Diffset là một phần nhỏ của kích thước Tidset nên thao tác tính phần khác nhau được thực thi khá hiệu quả. Bên cạnh đó Diffset còn làm giảm kích thước bộ nhớ yêu cầu để lưu trữ Tidset. Trong cùng một lớp tương đương, Diffset được tính dựa trên sự khác biệt giữa hai Tidset. Vì vậy, đối với CSDL dày đặc, kích thước của Diffset là nhỏ hơn Tidset. Tác giả đề xuất thuật toán cải tiến (FWCIs-DIFF) nhằm rút ngắn thời gian khai thác tập đóng được gán trọng phổ biến đối với những cơ sở dữ liệu dày đặc, từ đó giúp cho việc khai thác luật kết hợp được nhanh hơn.

Phần còn lại của bài báo được tổ chức như sau: Phần 2 chúng tôi trình bày những tính chất và định lý liên quan, phần 3 chúng tôi trình bày thuật toán cải tiến FWCIs-DIFF, phần 4 chúng tôi sẽ trình bày kết quả thực nghiệm, đánh giá và cuối cùng là kết luận lại vấn đề.

2. MỘT SỐ ĐỊNH LÝ VÀ TÍNH CHẤT

2.1. Cơ sở dữ liệu số lượng giao dịch

Cho CSDL D với tập giao dịch T = $\{t_1, t_2, ..., t_m\}$, tập các items $I = \{i_1, i_2, ..., i_n\}$ và tập trọng số dương $W = \{w_1, w_2, w_3, w_n\}$ tương ứng với mỗi item trong tập I [3].

Trong Bảng 2.1 có 6 giao dịch $T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$, 5 items $I = \{A, B, C, D, E\}$. Trọng số của những items này lần lượt là $W = \{0.6, 0.1, 0.3, 0.9, 0.2\}$.

Bảng 2.1. CSDL số lượng giao dịch: a. CSDL giao dịch, b. Trọng số (lợi ích) của các item.

Mã giao dịch	Nội dung giao dịch
1	*A, B, D, E*
2	*B, C, E*
3	*A, B, D, E*
4	*A, B, C, E*
5	*A, B, C, D, E*
6	*B, C, D*

(a)

item	Trọng số
A	0.6
B	0.1
C	0.3
D	0.9
E	0.2

(b)

Kết nối Galois [6]:

Cho quan hệ hai ngôi $\delta \subseteq I \times T$ chứa CSDL cần khai thác. Đặt $X \subseteq I$ và $Y \subseteq T$. Ta định nghĩa hai ánh xạ giữa $P(I)$ (Tập tất cả các tập con $\neq \emptyset$ của I) và $P(T)$. Ta có:

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

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

Tính chất của kết nối Galois:

Cho $X, X_1, X_2 \in P(I)$ và $Y, Y_1, Y_2 \in P(T)$

i) $X_1 \subset X_2 \Rightarrow t(X_1) \supseteq t(X_2)$

ii) $Y_1 \subset Y_2 \Rightarrow i(Y_1) \supseteq i(Y_2)$

iii) $X \subseteq i(t(X))$ và $Y \subseteq t(i(Y))$

Định nghĩa[2]:

Đặt $X \subseteq I$ là tập được gán trọng phổ biến. X được gọi là tập đóng được gán trọng phổ biến nếu và chỉ nếu không tồn tại bất kỳ tập được gán trọng phổ biến Y, sao cho $X \subset Y$ và $ws(X) = ws(Y)$.

Định lý 1: Cho 2 tập item X, Y với $X \subseteq Y$, $ws(X) = ws(Y) \Leftrightarrow t(X) = t(Y)$.

Định lý 2: Cho 2 node $\begin{matrix} X \times t(X) \\ ws(X) \end{matrix}$ và $\begin{matrix} Y \times t(Y) \\ ws(Y) \end{matrix}$ trong lớp tương đương $[P]$. Có các mệnh đề sau:

i. Nếu $t(X) = t(Y)$ thì X, Y không là tập đóng.

ii. Nếu $t(X) \subset t(Y)$ thì X không là tập đóng.

iii. Nếu $t(X) \supset t(Y)$ thì Y không là tập đóng.

2.2. Sử dụng Diffset để làm giảm không gian lưu trữ và cho phép tính nhanh ws (trọng số phổ biến) [3].

Xét một lớp tiền tố là P, PX và PY là hai thành viên bất kì của lớp tương đương P. Gọi d(PX), d(PY) là Diffset của PX và PY, ta có các công thức [5]:

$$d(PXY) = t(PX) \backslash t(PY)$$

$$d(PXY) = d(PY) \backslash d(PX)$$

Và các công thức tính trọng số [3]:

$$ws(PXY) = ws(PX) - \frac{\sum_{t \in d(PXY)} tw(t)}{\sum_{t \in T} tw(t)}$$

Nếu $d(PXY) = \emptyset$ thì $ws(PXY) = ws(PX)$

2.3. Sử dụng tính chất IT-Pair trên cơ sở Diffset

Do CSDL ban đầu được lưu dưới dạng Tidset và việc tính toán được áp dụng trên 4 tính chất của IT-pair, vậy nếu muốn áp dụng Diffset và việc tính toán cũng áp dụng được hiệu quả trên 4 tính chất của IT-pair ta cần phải xem xét mối quan hệ giữa Tidset và Diffset [2].

Gọi $m(X_i)$ và $m(X_j)$ là số phần tử khác nhau của $d(X_i)$ và $d(X_j)$. Gồm bốn tính chất:

Tính chất 1: Nếu $m(X_i) = 0$ và $m(X_j) = 0$ thì $d(X_i) = d(X_j)$ hay $t(X_i) = t(X_j)$.

Tính chất 2: Nếu $m(X_i) > 0$ và $m(X_j) = 0$ thì $d(X_i) \supset d(X_j)$ hay $t(X_i) \subset t(X_j)$.

Tính chất 3: Nếu $m(X_i) = 0$ và $m(X_j) > 0$ thì $d(X_i) \subset d(X_j)$ hay $t(X_i) \supset t(X_j)$.

Tính chất 4: Nếu $m(X_i) > 0$ và $m(X_j) > 0$ thì $d(X_i) \neq d(X_j)$ hay $t(X_i) \neq t(X_j)$.

Vậy có thể xử lý trên Diffset tương tự như trên Tidset.

Từ 2.2 và 2.3 có thể ứng dụng hoàn toàn Diffset để cải tiến thuật toán WIT-FWCIs.

3. THUẬT TOÁN WIT-FWCIS-DIFF

Các bước thực hiện thuật toán:

Bước 1: Khai báo và khởi tạo 1 số biến sau:

Đặt I là tập các item.

Đặt $[P] = \{i \in I \,/\, ws(i) \geq minws\}$

Đặt $ws(i) = \frac{\sum_{t_k \in t(i)} tw(t_k)}{\sum_{t_k \in T} tw(t_k)}, tw(t_k) = \frac{\sum_{i_j \in t_k} w_j}{|t_k|}$; Trong đó, t_k là giao dịch thứ k trong CSDL và $k = \overline{1..n}$

$FWCIs\_DIFF = \{\emptyset\}$

Bước 2: Sắp xếp tăng các node của $[P_{00}]$ theo số lượng tidset:

$[P_{00}] = \{l_{00k} \in [P_{00}] \,/\, |t(l_{00k})| \leq |t(l_{00m})|; l_{00m} \in [P_{00}]$

$, \forall m = \overline{k..(|[P_{00}]| - 1)}, \forall k = \overline{0..(|[P_{00}]| - 1)}\}$

Bước 3: Tìm các lớp tương đương con của $[P_{00}]$:

Bước 3.1:

$l_{00k} = \{l_{00k} \cup l_{00m} \,/\, t(l_{00k}) \subseteq t(l_{00m}); l_{00m} \in [P_{00}]$

$, \forall m = \overline{(k+1)..(|[P_{00}]|-1)\}}, l_{00k} \in [P_{00}], \forall k = \overline{0..(|[P_{00}]|-1)\}}$

Bước 3.2:

$t(l_{00k}) = t(l_{00m}) \Longrightarrow remove\ l_{00m}$ với $l_{00k}, l_{00m} \in [P_{00}]$

$, \forall m = \overline{(k+1)..(|[P_{00}]|-1)}, \forall k = \overline{0..(|[P_{00}]|-1)}$

Bước 3.3:

$[P_{1,k}] = \{l_{00k} \cup l_{00m} \,/\, \mathrm{t}(l_{00k}) \supset t(l_{00m})\ or\ \mathrm{t}(l_{00k}) \neq t(l_{00m})$

$and\ ws(l_{00k} \cup l_{00m}) \geq minws;\ l_{00k}, l_{00m} \in [P_{00}]$

$, \forall m = \overline{(k+1)..(|[P_{00}]|-1)\}}, \forall k = \overline{0..(|[P_{00}]|-1)}$

Trong đó, tính $\mathrm{d}(l_{00k} \cup l_{00m})$ và $\mathrm{ws}(l_{00k} \cup l_{00m})$ cụ thể:

$\mathrm{d}(l_{00k} \cup l_{00m}) = t(l_{00k}) \backslash \mathrm{t}(l_{00m})$

$\mathrm{ws}(l_{00k} \cup l_{00m}) = ws(l_{00k}) - \frac{\sum_{t \in \mathrm{d}(l_{00k} \cup l_{00m})} tw(t)}{\sum_{t \in T} tw(t)}$

Bước 3.4:

$t(l_{00k}) \supset t(l_{00m}) \Longrightarrow remove\ l_{00m}$ với $l_{00k}, l_{00m} \in [P_{00}]$

$, \forall m = \overline{(k+1)..(|[P_{00}]|-1)}, \forall k = \overline{0..(|[P_{00}]|-1)}$

Bước 4: Tìm các lớp tương đương con của $[P_{ij}]$:

Bước 4.1:

$l_{ijk} = \{l_{ijk} \cup l_{ijm}\ /\ d(l_{ijk}) \supseteq d(l_{ijm}); l_{ijm} \in [P_{ij}]$

$, \forall m = \overline{(k+1)..(|[P_{ij}]|-1)\}}, l_{ijk} \in [P_{ij}], \forall k = \overline{0..(|[P_{ij}]|-1)\}}$

Trong đó, với $d(l_{ijk} \cup l_{ijm}) = d(l_{ijk})$ và $ws(l_{ijk} \cup l_{ijm}) = ws(l_{ijk})$

Bước 4.2:

$d(l_{ijk}) = d(l_{ijm}) \Longrightarrow remove\ l_{ijm}$ với $l_{ijk}, l_{ijm} \in [P_{ij}]$

$, \forall m = \overline{(k+1)..(|[P_{ij}]|-1)}, \forall k = \overline{0..(|[P_{ij}]|-1)\}}$

Bước 4.3:

$[P_{(i+1),\left(\sum_{e=0}^{j-1}(|[P_{i,e}]|)\right)+k}] = \{l_{ijk} \cup l_{ijm} \,/\, \mathrm{d}(l_{ijk}) \subset d(l_{ijm})$

$or\ \mathrm{d}(l_{ijk}) \neq d(l_{ijm})\ and\ ws(l_{ijk} \cup l_{ijm}) \geq minws; l_{ijk}, l_{ijm} \in [P_{ij}]$

$, \forall m = \overline{(k+1)..(|[P_{ij}]|-1)\}}, \forall k = \overline{0..(|[P_{ij}]|-1)}$

Trong đó, tính $\mathrm{d}(l_{ijk} \cup l_{ijm})$ và $\mathrm{ws}(l_{ijk} \cup l_{ijm})$ cụ thể:

$\mathrm{d}(l_{ijk} \cup l_{ijm}) = d(l_{ijm}) \backslash \mathrm{d}(l_{ijk})$

$\mathrm{ws}(l_{ijk} \cup l_{ijm}) = ws(l_{ijk}) - \frac{\sum_{t \in \mathrm{d}(l_{ijm})} tw(t)}{\sum_{t \in T} tw(t)}$

Bước 4.4:

$d(l_{ijk}) \subset d(l_{ijm}) \Longrightarrow remove\ l_{ijm}$ với $l_{ijk}, l_{ijm} \in [P_{ij}]$

$, \forall m = \overline{(k+1)..(\left|[P_{ij}]\right|-1)}, \forall k = \overline{0..(\left|[P_{ij}]\right|-1)}$

Bước 5: Lặp lại bước 4 với các lớp tương đương khác $[P_{ij}]$, j= $\overline{0..(n-1)}$

Bước 6: Lặp lại bước 5 với các mức khác của cây $[P_{ij}]$, $i = \overline{0..(n-1)}$

Bước 7: $FWCIs\_DIFF = \{l_{ijk}: l_{ijk} \in [P_{ij}], \forall i,j,k = \overline{0..(n-1)}\}$

Thuật toán WIT-FWCIs-DIFF:

WIT-FWCIs-DIFF()

1. $FWCIs\_DIFF = \emptyset$
2. $[\emptyset] = \{i \in I: ws(i) \geq minws\}$
3. SORT([∅]) //Sắp xếp những node trong [∅] tăng theo tidset và ws
4. WIT_FWCIs_DIFF_EXTEND([∅],$FWCIs\_DIFF = \emptyset$)
5. return $FWCIs\_DIFF$ //Itemset phổ biến thỏa ngưỡng *minws*

WIT_FWCIs_DIFF_EXTEND([*P*], $FWCIs\_DIFF$)

6. for each $l_i \in [P]$ do
7. $P_i = P \cup l_i$ and $[P_i] = \emptyset$
8. for each $l_j \in [P]$, with $j > i$ do
9. if $t(l_i) = t(l_j)$ then //Theo tính chất 1 của mục 2.3
10. $P_i = l_i \cup l_j$
11. remove l_j from $[P]$
12. for each $l_k \in [P_i]$ do //Hội thêm l_j cho các node thuộc lớp tương đương $[P_i]$
13. $l_k = l_k \cup l_j$
14. else if $t(l_i) \subset t(l_j)$ then //Theo tính chất 2 của mục 2.3
15. $P_i = l_i \cup l_j$
16. for each $l_k \in [P_i]$ do // Hội thêm l_j cho các node thuộc lớp tương đương $[P_i]$
17. $l_k = l_k \cup l_j$
18. else
19. $X = l_i \cup l_j$
20. $Y = d(l_i \cup l_j) = t(l_i) \backslash t(l_j)$
21. $ws(X) = ws(l_i) - \frac{\sum_{t \in Y} tw(t)}{\sum_{t \in T} tw(t)}$
22. if $t(l_i) \supset t(l_j)$ then //Theo tính chất 3 của mục 2.3
23. remove l_j from $[P]$
24. add $X \times_{ws(X)} Y$ to$[P_i]$
25. else//Theo tính chất 4 của mục 2.3

26. if $ws(X) \geq minws$ then
27. add $X \times_{ws(X)} Y$ to $[P_i]$
28. WIT_FWCIs_DIFF_EXTEND_level2($[P_i]$, $FWCIs\_DIFF$)
29. if (SUBSUMPTION_CHECK(C, P_i)==*TRUE*) then //Kiểm tra tập đóng
30. Add P_i to $FWCIs\_DIFF$

WIT_FWCIs_DIFF_EXTEND_level2($[P]$, $FWCIs\_DIFF$)

31. for each $l_i \in [P]$ do
32. $P_i = P \cup l_i$ and $[P_i] = \emptyset$
33. for each $l_j \in [P]$, with $j > i$ do
34. $X = l_i \cup l_j$
35. $Y = d(l_i \cup l_j) = d(l_j) \backslash d(l_j)$
36. if $Y = \emptyset$ then $ws(X) = ws(l_i)$//Sử dụng công thức của mục 2.2
37.
38. else $ws(X) = ws(l_i) - \frac{\sum_{t \in Y} tw(t)}{\sum_{t \in T} tw(t)}$
39. if ($ws(X) \geq minws$) then
40. if $d(l_i) = d(l_j)$ then // Theo tính chất 1 của mục 2.3
41. $P_i = l_i \cup l_j$
42. remove l_j from $[P]$
43. for each $l_k \in [P_i]$ do// Hội thêm l_j cho các node thuộc lớp tương đương $[P_i]$
44. $l_k = l_k \cup l_j$
45. else if $d(l_i) \supset d(l_j)$then // Theo tính chất 2 của mục 2.3
46. $P_i = l_i \cup l_j$
47. for each $l_k \in [P_i]$ do // Hội thêm l_j cho các node thuộc lớp tương đương $[P_i]$
48. $l_k = l_k \cup l_j$
49. else
50. if $t(l_i) \subset t(l_j)$ then // Theo tính chất 3 của mục 2.3
51. remove l_j from $[P]$
52. add $X \times_{ws(X)} Y$to$[P_i]$
53. else // Theo tính chất 4 của mục 2.3
54. if $ws(X) \geq minws$ then
55. add $X \times_{ws(X)} Y$to$[P_i]$
56. WIT_FWCIs_DIFF_EXTEND_level2($[P_i]$, $FWCIs\_DIFF$)
57. if (SUBSUMPTION_CHECK(C, P_i)==*TRUE*) then // Kiểm tra tập đóng

58. Add P_i to $FWCIs\_DIFF$

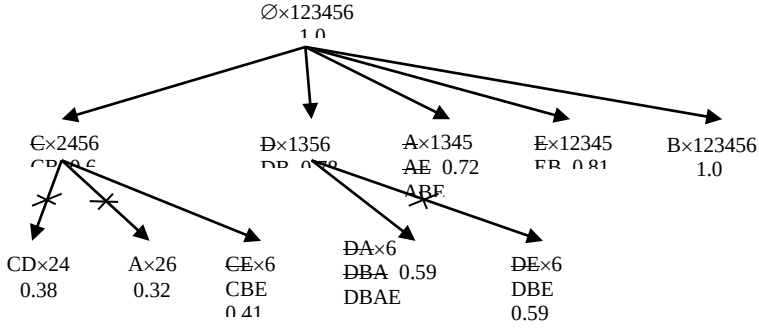
SUBSUMPTION_CHECK(*C, P)*

59. for each $Y \in$ HASHTABLE $[|t(P)|]$ do
60. if $\sigma(Y) = \sigma(P)\ and\ P \subset Y$ then
61. return *FALSE*
62. $C = C \cup P$
63. return *TRUE*

Vậy FWCIs_DIFF chính là tập đóng được gán trọng phổ biến trong CSDL số lượng với minws cho trước.

Ví dụ minh họa thuật toán WIT-FWCIs-DIFF:

Từ bảng dữ liệu 2.1 với *minws* = 0.4.



Hình 3.1. Hình minh họa ví dụ thuật toán WIT-FWCIs-DIFF.

Khởi tạo [∅] = {A, B, C, D, E}

Khởi tạo FWCIs-DIFF = ∅

Sau khi sắp xếp và gom nhóm ta được [∅] = {*C, D, A, E, B}*

Ta lần lượt tìm các lớp tương đương của [∅]

* Với $l_i = C$

Kết hợp với các $l_j \in \{D, A, E, B\}$

+ Do $d(CD) = t(C) \backslash t(D) = 24 \Rightarrow ws(CD) = 0.6 - \frac{0.2+0.3}{2.25} = 0.38 < minws$. Vậy không thêm node $\begin{pmatrix} CD \times 24 \\ 0.38 \end{pmatrix}$ vào cây.

+ Do $d(CA) = t(C) \backslash t(A) = 26 \Rightarrow ws(CA) = 0.6 - \frac{0.2+0.43}{2.25} = 0.32 < minws$. Vậy không thêm node $\begin{pmatrix} CA \times 26 \\ 0.32 \end{pmatrix}$ vào cây.

+ Do $d(CE) = t(C) \backslash t(E) = 6 \Rightarrow ws(CE) = 0.6 - \frac{0.43}{2.25} = 0.41 > minw$. Vậy ta thêm node $\begin{pmatrix} CE \times 6 \\ 0.41 \end{pmatrix}$ vào cây.

+ Do $t(C) \subset t(B)$ nên theo tính chất 2 của mục 2.3 thì *C* không thể là tập đóng. Vậy thay $\begin{pmatrix} C \times 2456 \\ 0.6 \end{pmatrix}$ bằng $\begin{pmatrix} CB \times 2456 \\ 0.6 \end{pmatrix}$ và thay $\begin{pmatrix} CE \times 6 \\ 0.41 \end{pmatrix}$ bằng $\begin{pmatrix} CBE \times 6 \\ 0.41 \end{pmatrix}$.

Kiểm tra CBE trong bảng băm thấy rằng CBE là tập đóng nên thêm CBE vào *FWCIs-DIFF* $\Rightarrow$ *FWCIs-DIFF* = {*CBE*}.

Kiểm tra CB trong bảng băm thấy rằng CB là tập đóng nên thêm CB vào *FWCIs-DIFF* $\Rightarrow$ *FWCIs-DIFF* = {*CBE, CB*}.

* Với $l_i = D$

Kết hợp với các $l_j \in$ {A, E, B}

+ Do$(DA) = t(D) \backslash t(A) = 6 \Rightarrow ws(DA) = 0.78 - \frac{0.43}{2.25} = 0.59 > minws$. Vậy ta thêm node $\begin{pmatrix} DA \times 6 \\ 0.59 \end{pmatrix}$ vào cây.

+ Do$t(DE) = t(D) \backslash t(E) = 6 \Rightarrow ws(DE) = 0.78 - \frac{0.43}{2.25} = 0.59 > minws$. Vậy ta thêm node $\begin{pmatrix} DE \times 6 \\ 0.59 \end{pmatrix}$ vào cây.

+ Do $t(D) \subset t(B)$ nên theo tính chất 2 của mục 2.3 thì *D* không thể là tập đóng. Vậy thay $\begin{pmatrix} D \times 1356 \\ 0.78 \end{pmatrix}$ bằng $\begin{pmatrix} DB \times 1356 \\ 0.78 \end{pmatrix}$, thay $\begin{pmatrix} DA \times 6 \\ 0.59 \end{pmatrix}$bằng $\begin{pmatrix} DBA \times 6 \\ 0.59 \end{pmatrix}\begin{pmatrix} DE \times 6 \\ 0.59 \end{pmatrix}$ bằng $\begin{pmatrix} DBE \times 6 \\ 0.59 \end{pmatrix}$ .Do $t(DBA) = t(DBE)$ nên theo tính chất 1 của mục 2.3 thì *DBA* không thể là tập đóng. Vậy ta thay $\begin{pmatrix} DBA \times 6 \\ 0.59 \end{pmatrix}$ thành $\begin{pmatrix} DBAE \times 6 \\ 0.59 \end{pmatrix}$ và xóa $\begin{pmatrix} DBE \times 6 \\ 0.59 \end{pmatrix}$.

Kiểm tra $DBAE$ trong bảng băm thấy rằng $DBAE$ là tập đóng nên thêm $DBAE$ vào *FWCIs-DIFF* $\Rightarrow$ *FWCIs-DIFF* = {*CBE, CB, DBAE*}.

Kiểm tra DB trong bảng băm thấy rằng DB là tập đóng nên thêm DB vào *FWCIs-DIFF* $\Rightarrow$ *FWCIs-DIFF* = {*CBE, CB, DBAE, DB*}.

* Với $l_i = A$

Kết hợp với các $l_j \in \{E, B\}$

+ Do $t(A) \subset t(E)$ nên theo tính chất 2 của mục 2.3 thì *A* không thể là tập đóng. Vậy thay $\begin{pmatrix} A \times 1345 \\ 0.72 \end{pmatrix}$ bằng $\begin{pmatrix} AE \times 1345 \\ 0.72 \end{pmatrix}$.

* Do $t(AE) \subset t(B)$ nên theo tính chất 2 của mục 2.3 thì *AE* không thể là tập đóng. Vậy thay $\begin{pmatrix} AE \times 1345 \\ 0.72 \end{pmatrix}$ bằng $\begin{pmatrix} AEB \times 1345 \\ 0.72 \end{pmatrix}$.

Kiểm tra AEB trong bảng băm thấy rằng AEB là tập đóng nên thêm AEB vào *FWCIs-DIFF* $\Rightarrow$ *FWCIs-DIFF* = {*CBE, CB, DBAE, DB, AEB*}.

* Với $l_i = E$

Kết hợp với các $l_j \in \{B\}$

+ Do $t(E) \subset t(B)$nên nên theo tính chất 2 của mục 2.3 thì E không thể là tập đóng. Vậy thay $\left(\frac{E \times 12345}{0.81}\right)$ bằng $\left(\frac{EB \times 12345}{0.81}\right)$.

Kiểm tra EB trong bảng băm thấy rằng EB là tập đóng nên thêm EB vào *FWCIs-DIFF* $\Rightarrow$ *FWCIs-DIFF* = {*CBE, CB, DBAE, DB, AEB, EB*}.

* Với $l_i = B$

Kết hợp với các $l_j \in \{\}$

Kiểm tra B trong bảng băm thấy rằng B là tập đóng nên thêm B vào *FWCIs-DIFF* $\Rightarrow$ *FWCIs-DIFF* = {*CBE, CB, DBAE, DB, AEB, EB, B*}.

Kết quả: Tập đóng được gán trọng phổ biến *FWCIs-DIFF* = {*CBE, CB, DBAE, DB, ABE, EB, B*} thỏa ngưỡng *minws* = 0.4.

4. KẾT QUẢ THỰC NGHIỆM

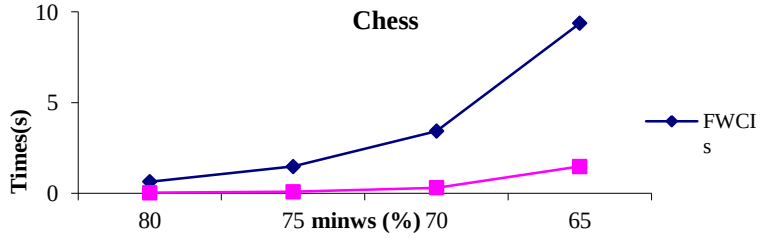
Giải thuật trình bày trong bài được hiện thực bằng ngôn ngữ C# 2010, dữ liệu được lưu trữ file text; và chạy trên máy Intel Core i5 2.5GHz 8GB RAM, Windows 8. Bài báo sử dụng tập CSDL chuẩn được lấy từ http://www.fimi.ua.ac.be/data/ với đặc điểm được mô tả trong Bảng 4.1.

Bảng 4.1. CSDL chuẩn dùng để thử nghiệm.

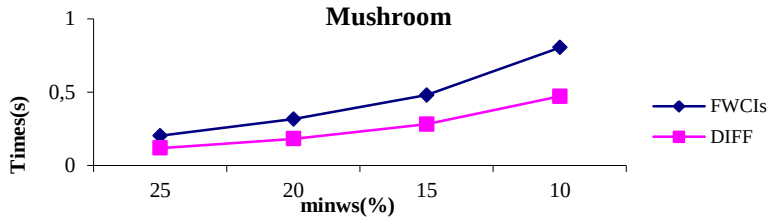
Tên CSDL	Số giao dịch	Số mục	Độ dài trung bình	Độ dài tối đa
Chess	3196	75	37	37
Mushroom	8124	120	23	23
Connect	67557	130	43	43
Accident	340183	467	33.8	51

Đặc điểm cho biết CSDL là thưa (mật độ trùng lắp của các item trên các giao dịch thấp) hay đặc (mật độ trùng lắp cao). Tính chất của các CSDL cũng khác nhau. CSDL Chess với số lượng giao dịch và số item ít nhưng rất đặc. Mushroom với số giao dịch, số item ít và là CSDL thưa. Connect với số giao dịch trung bình và số item ít.

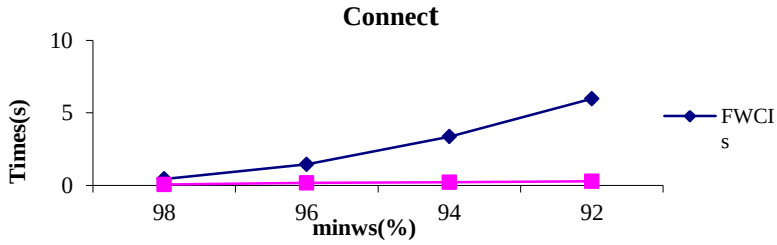
Kết quả đánh giá thời gian giữa thuật toán WIT-FWCIs và thuật toán WIT-FWCIs-DIFF:



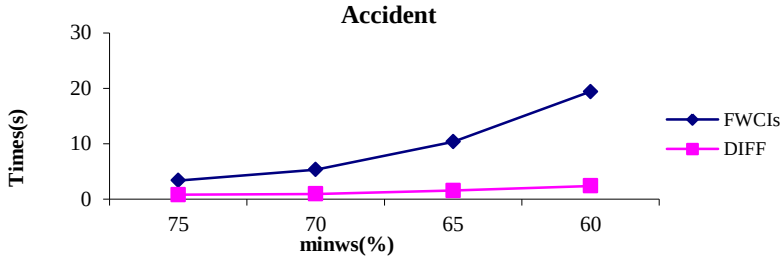
Hình 4.1. Biểu đồ thời gian giữa WIT-FWCIs và WIT-FWCIs-DIFF với CSDL Chess.



Hình 4.2. Biểu đồ thời gian giữa WIT-FWCIs và WIT-FWCIs-DIFF với CSDL Mushroom.



Hình 4.3. Biểu đồ thời gian giữa WIT-FWCIs và WIT-FWCIs-DIFF với CSDL Connect.



Hình 4.4. Biểu đồ thời gian giữa WIT-FWCIs và WIT-FWCIs-DIFF với CSDL Accidents.

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Bài báo đã đưa ra thuật toán cải tiến trong khai thác tập đóng được gán trọng phổ biến trên cơ sở dữ liệu số lượng bằng cách sử dụng diffset nhằm rút ngắn được thời gian khai thác tập phổ biến. Diffset làm giảm kích thước bộ nhớ yêu cầu để lưu trữ Tidset. Trong cùng một lớp tương đương, Diffset được tính đựa trên sự khác biệt giữa hai Tidset. Vì vậy đối với CSDL dày đặc, kích thước của Diffset là nhỏ hơn Tidset. Vì vậy thời gian khai thác tập đóng phổ biến của WIT-FWCIs-DIFF là tối ưu hơn WIT-FWCIs.

Trong tương lai, nhóm tác giả sẽ nghiên cứu khai thác luật kết hợp dựa vào tập đóng được gán trọng phổ biến.

TÀI LIỆU THAM KHẢO

1. Nicolas Pasquier, Yves Bastide, Rafik Taouil, Lotfi Lakhal. Discovering frequent closed itemsets for association rules. Proceecings of the 5th International Conference on Database Theory, LNCS, Springer-Verlag, Jerusalem, Israel, pp.398–416. 1999.
2. Mohammed Javeed Zaki, Ching-Jui Hsiao. Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Transactions on Knowledge and Data Engineering 17 (4), pp.462-478. 2005.

3. Bay Vo, Frans Coenen, Bac Le. A new method for mining Frequent Weighted Itemsets based on WIT-trees. Expert Systems with Applications 40(4), pp.1256-1264. 2013.
4. Bay Vo, Nhu Y Tran, Duong Ha Ngo. Mining frequent weighted closed itemsets. ICCSAMA 2013, SCI 497, pp.379-390, Springer International Publishing Switzeland. 2013.
5. Mohammed Javeed Zaki. Mining non-redundant association rules. Data Mining and Knowledge Discovery 9 (3), pp.223-248. 2004.

ABSTRACT

USING DIFFSET FOR MINING FREQUENT WEIGHTED CLOSED ITEMSETS ON WEIGHTED ITEMS TRANSACTION DATABASES

Tran Nhu Y*, Nguyen Van Tung, Ngo Duong Ha

Ho Chi Minh city University of Food Industry

*Email: ytn@*cntp.edu.vn*

Mining frequent itemsets plays an important role in mining association rules. For weighted items transaction databases, mining frequent weighted closed itemsets (FWCIs) is one of the method proposed by author. However, for dense databases, the mining time of FWCIs is still high. In this paper, an algorithm for mining frequent weighted closed itemsets using diffset (FWCIs-DIFF) is proposed. Some theorems are presented first, base on them, an algorithm for mining FWCIs-DIFF is proposed. For dense databases, experimental results show that the mining time of FWCIs-DIFF is always smaller than that of FWCIs.

Keyword: mining frequent itemsets, mining frequent weighted closed itemset, diffset.