JOURNAL OF SCIENCE OF HNUE DOI: 10.18173/2354-1075.2015-0062
Educational Sci., 2015, Vol. 60, No. 7A, pp. 145-156
This paper is available online at http://stdb.hnue.edu.vn

KHAI PHÁ HIỆU QUẢ TẬP MỤC THƯỜNG XUYÊN VỚI TRỌNG SỐ THÍCH NGHI TRÊN DÒNG DỮ LIỆU

Nguyễn Hưng Long, Nguyễn Thị Thu Thủy
Khoa Hệ thống Thông tin Kinh tế, Trường Đại học Thương mại

Tóm tắt. Bài báo đề xuất thuật toán SWFI-miner cho bài toán khai phá tập mục thường xuyên với trọng số thích nghi trên dòng dữ liệu. Trong bài báo này, chúng tôi xem xét lại mô hình khai phá tập mục thường xuyên với trọng số thích nghi trong cơ sở dữ liệu tĩnh và mô hình khai phá tập mục thường xuyên với trọng số trên dòng dữ liệu bằng cách sử dụng một độ đo mới để tỉa cây SAWFI-tree và các cây điều kiện hiệu quả hơn, và mở rộng việc khai phá TMTX với trọng số thích nghi hơn trên dòng dữ liệu. Qua phân tích và đánh giá cho thấy thuật toán SWFI-miner thật sự hiệu quả trong khai phá tập mục thường xuyên với trọng số thích nghi trên dòng dữ liệu.
Từ khóa: Khai phá dữ liệu, tập mục thường xuyên, trọng số, trọng số thích nghi, dòng dữ liệu.

1. Mở đầu

Trong những năm gần đây, khai phá dữ liệu ngày càng trở nên cấp thiết cùng với sự xuất hiện của các ứng dụng mới trong thực tiễn. Ở đó các dữ liệu được xử lí không còn là dữ liệu tĩnh, mà là các dữ liệu động, liên tục và có thể coi như là vô hạn (không bị chặn) [1,3,4,6,9-12,14,15]. Các dữ liệu đến như vậy tạo thành dòng dữ liệu *(data stream).* Một số ứng dụng trên thực tế sử dụng dòng dữ liệu như: phân tích lưu lượng mạng *(network traffic analysis)*, phát hiện xâm nhập mạng *(network intrusion detection),* hay phân tích giao dịch trực tuyến *(on-line transaction analysis),…* Có ba thách thức trong khai phá dòng dữ liệu: *Thứ nhất,* để phát hiện ra các tập mục thường xuyên (TMTX) cần phải tìm kiếm trên không gian là một hàm mũ. *Thứ hai,* dữ liệu được cập nhật liên tục và không bị chặn dẫn đến những hạn chế cho không gian nhớ để sử dụng. *Thứ ba,* cần phải có thuật toán khai phá hiệu quả để xử lí dữ liệu càng nhanh càng tốt, đồng thời thuật toán chỉ được phép quét dữ liệu một lần trên dòng dữ liệu.

Gần đây, trong [5], Chowdhury F. A. và cộng sự đã đề cập đến vấn đề trọng số thay đổi theo thời gian (trọng số thích nghi) trong cơ sở dữ liệu (CSDL) tĩnh. Các tác giả công trình đã đề xuất mô hình và thuật toán AWFPM *(Adaptive Weighted Frequent Patterns Mining)* khai phá TMTX với trọng số thích nghi trên CSDL, theo nghĩa trọng số của các mục có thể thay đổi theo thời gian, từ lô giao tác này sang lô giao tác khác của CSDL. Tập mục được gọi là thường xuyên với trọng

Ngày nhận bài: 7/7/2015. Ngày nhận đăng: 15/11/2015.
Liên hệ: Nguyễn Hưng Long, e-mail: ntthlong@gmail.com.

số thích nghi nếu có tổng độ hỗ trợ với trọng số trong các lô lớn hơn ngưỡng đã cho. AWFPM sử dụng cấu trúc cây FP-tree *(Frequency Pattern)* để lưu trữ các thông tin nén các giao tác của CSDL lên cây. Việc tỉa cây được thực hiện bằng cách sử dụng trọng số cực đại toàn cục và trọng số cực đại địa phương. Trong đó, trọng số cực đại toàn cục là trọng số lớn nhất của tất cả các mục trong CSDL khai phá, còn trọng số cực đại địa phương là trọng số lớn nhất của các mục trong một CSDL điều kiện. Tuy nhiên, việc sử dụng trọng số cực đại toàn cục và trọng số cực đại địa phương để tỉa cây chưa thật sự hiệu quả. Bởi vì, mỗi lần tính các trọng số cực đại này phải xét tới toàn bộ các giao tác của CSDL cần khai phá hay CSDL điều kiện.

Trong [13], Tsai P. S. M. đã đã đề xuất một quy trình mới cho việc khai phá dòng dữ liệu được gọi là mô hình cửa sổ trượt với trọng số *(Weighted sliding window model).* Mô hình này cho phép người sử dụng ấn định số cửa sổ cần khai phá và kích thước của chúng. Tuy nhiên, tất cả các mục trong lô lại được gán cho cùng một trọng số. Tsai P. S. M. đã đề xuất hai thuật toán WSW và WSW-Imp. Hạn chế của hai thuật toán WSW và WSW-Imp là xây dựng theo kiểu Apriori [2], ở đó sử dụng tính bao đóng xuống *(downward closure property)* của TMTX: "Nếu một tập mục là TMTX thì mọi tập con của nó cũng là TMTX"). Như vậy, các thuật toán phải quét CSDL rất nhiều lần để sinh ra và tỉa các tập mục ứng viên nếu nó chứa bất kì một tập con nào không phải là TMTX.

Trong bài báo này, chúng tôi xem xét lại mô hình khai phá TMTX với trọng số thích nghi trong CSDL tĩnh của Chowdhury F. A. và cộng sự [5]. Chúng tôi cũng xem xét và phát triển mô hình khai phá TMTX với trọng số trên dòng dữ liệu sử dụng cửa sổ trượt của Tsai P. S. M. [13] theo nghĩa các trọng số của các tập mục sẽ là thích nghi theo các lô trong dòng dữ liệu và đề xuất thuật toán cải tiến SWFI-miner. Thuật toán SWFI-miner có một số đóng góp sau: *Thứ nhất,* sử dụng một độ đo mới cho phép tỉa cây SAWFI-tree và các cây điều kiện hiệu quả hơn, ở đó chúng tôi chỉ tính trọng số lớn nhất của các mục theo từng lô được xét. *Thứ hai,* mở rộng việc khai phá TMTX với trọng số thích nghi (trọng số thay đổi theo thời gian) trên dòng dữ liệu hiệu quả hơn, nhằm đáp ứng các ứng dụng thực tiễn.

2. Nội dung nghiên cứu

2.1. Mô hình bài toán khai phá tập mục thường xuyên với trọng số thích nghi trên dòng dữ liệu

Sau đây chúng tôi dựa trên cách tiếp cận mô hình khai phá TMTX với trọng số thích nghi trên CSDL tĩnh của Chowdhury F. A. và cộng sự [5], và mô hình khai phá TMTX với trọng số trên dòng dữ liệu của Tsai P. S. M. [13] để phát triển, đề xuất bài toán khai phá TMTX với trọng số thích nghi trên dòng dữ liệu.

Cho I là tập các mục, $I = \{i_1, i_2, ..., i_k\}$. Một tập mục con $X \subseteq I$, gồm k mục phân biệt được gọi là một *k-tập mục* hay tập mục độ dài k. Để đơn giản, thay vì viết $\{i_1, i_2, \ldots, i_r\}$đôi khi ta viết $i_1 i_2 \ldots i_r$; chẳng hạn, tập mục $\{a, b, c\}$được viết ngắn gọn là abc. Mỗi giao tác là một bộ $t = (TID, X)$ trong đó TID là một định danh và X là một tập mục.

Một dòng dữ liệu giao tác (CSDL giao tác) DS là một dãy các giao tác, $DS = \{t_{i1}, t_{i2}, \ldots, t_{im}, \ldots\}$, trong đó $t_{ij}, i = 1, 2, \ldots; j = 1, 2, \ldots$ là giao tác đến tại thời điểm thứ i.

Một lô giao tác (hay một lô) là tập các giao tác nhằm phản ánh thực tế quản lí (tùy thuộc ngữ cảnh) theo một đơn vị thời gian (ngày, tháng, quý, năm, ...).

Một cửa sổ W trên dòng dữ liệu giao tác được xem là một tập các lô xét tại một thời điểm.

Giả sử tại thời điểm T_i $(i = 1, 2, ...)$, cửa sổ trượt W được chia thành K lô B_{ij} $(i = 1, 2...; j = 1, 2, ..., K)$ và mỗi mục trong mỗi lô được gán một trọng số riêng biệt, là số thực không âm.

Định nghĩa 1. Độ hỗ trợ với trọng số thích nghi của tập mục X là đại lượng $AWsupp(X)$, xác định bởi

$$AWsupp(X) = \sum_{j=1}^{K} \mathrm{W}(X, j) \times F(X, j) \tag{2.1}$$

Trong đó $W(X, j)$ là trọng số của X tại lô thứ j được tính bằng trọng số trung bình của các mục trong lô thuộc X, $F(X, j)$ là tần số xuất hiện của X trong lô thứ j tại thời điểm T_i.

Định nghĩa 2. Độ hỗ trợ với trọng số tối thiểu trên dòng dữ liệu DS, tại thời điểm T_i, xác định bởi:

$$\xi = minsupp \times \sum_{j=1}^{K} |B_{ij}| \times \overline{W_{ij}} \tag{2.2}$$

Trong đó $|B_{ij}|$ là số các giao tác và $\overline{W_{ij}}$ trọng số của lô thứ j tại thời điểm T_i, và $minsupp$ là độ hỗ trợ tối thiểu cho dòng dữ liệu DS.

Định nghĩa 3. Tập mục X được gọi là tập mục thường xuyên với trọng số thích nghi trên dòng dữ liệu DS nếu độ hỗ trợ với trọng số thích nghi của X không nhỏ hơn ngưỡng độ hỗ trợ với trọng số tối thiểu ξ, nghĩa là:

$$\mathrm{AWsupp}(X) \geq \xi \tag{2.3}$$

Khi đó ta cũng nói tập mục X thỏa ξ, trường hợp ngược lại tập mục X không thỏa ξ.

Định nghĩa 4. Khai phá TMTX với trọng số thích nghi trên dòng dữ liệu DS sử dụng mô hình cửa sổ trượt là tìm tập AWFI chứa tất cả các TMTX với trọng số, tức là tìm tập:

$$\mathrm{AWFI} = \{X/X \subseteq I, \mathrm{AWsupp}(X) \geq \xi\} \tag{2.4}$$

Giả sử T_1, có 12 giao tác, 3 lô B_{11}, B_{12}, B_{13} với trọng số của các mục tại các lô trong Bảng 2 và độ hỗ trợ tối thiểu $minsupp$ là 30%.

Tại thời điểm T_1 :

Độ hỗ trợ với trọng số tối thiểu trên dòng dữ liệu là:

$\overline{W}_{11} = 0.6$; $\overline{W}_{12} = 0.7$; $\overline{W}_{13} = 0.6$;

Nên ta được:

$$\xi = \mathrm{minsupp} \times \sum_{j=1}^{K} |B_{ij}| \times \overline{W}_{ij} = 30\% \left(0.6 \times 4 + 0.7 \times 3 + 0.6 \times 5\right) = 2.25 \ .$$

Độ hỗ trợ với trọng số thích nghi của tập mục "e" là:

Bảng 1. Dòng dữ liệu tại thời điểm T_1

Cửa sổ W (B_{13} - B_{11})

Lô	TID	Giao tác
B_{13}	T1	a e
	T2	b c d
	T3	a b c
	T4	c d
	T5	b c d e
B_{12}	T6	c d e
	T7	b c d
	T8	d e
B_{11}	T9	b c
	T10	a d e
	T11	b d e
	T12	a b c d

Thời điểm T_1

Bảng 2. Trọng số các mục theo lô tại thời điểm T_1

Lô	Các trọng số				
	a	b	c	d	e
B_{13}	0.5	0.8	0.7	0.7	0.3
B_{12}	0.9	0.7	0.7	0.8	0.4
B_{11}	0.8	0.7	0.6	0.5	0.4

$$AWsupp(e) = 0.3 \times 2 + 0.4 \times 2 + 0.4 \times 2 = 2.2;$$

Vì $AWsupp(e) = 2.2 < \xi = 2.25$, nên tập mục "e" không là TMTX với trọng số thích nghi trên dòng dữ liệu, hay nói cách khác "e" không thỏa ξ.

Độ hỗ trợ với trọng số thích nghi của tập mục "de" là:

$$AWsupp(de) = \frac{0.7 + 0.3}{2} \times 1 + \frac{0.8 + 0.4}{2} \times 2 + \frac{0.5 + 0.4}{2} \times 2 = 2.6;$$

Vì $AWsupp(de) = 2.6 > \xi = 2.25$, nên tập mục "de" là TMTX với trọng số thích nghi trên dòng dữ liệu, hay nói cách khác "de" thỏa ξ.

Nhận xét, qua ví dụ ta thấy TMTX với trọng số thích nghi trên dòng dữ liệu được định nghĩa như trên không thỏa mãn tính chất Apriori. Bởi lẽ, "e" không là TMTX với trọng số thích nghi trên dòng dữ liệu nhưng tập cha của nó là "de" lại là TMTX với trọng số thích nghi trên dòng dữ liệu.

Để có được tính chất Apriori, chúng tôi đưa ra khái niệm TMTX với trọng số thích nghi cực đại và sẽ chỉ ra nếu một tập mục là TMTX với trọng số thì trước hết chúng phải là TMTX với trọng số thích nghi cực đại.

Định nghĩa 5. Tại thời điểm T_i, cho dòng dữ liệu DS gồm K lô và X là một tập mục. Khi đó, số đo

$$\text{MAXAWsupp(X)} = \sum_{j=1}^{K} MAXW(j) \times F(X, j) \tag{2.5}$$

được gọi là độ hỗ trợ với trọng số thích nghi cực đại của X trên dòng dữ liệu DS. Với là giá trị trọng số lớn nhất của các mục trong X tại lô thứ j.

Ví dụ: Xét dòng dữ liệu tại thời điểm T_1 như Bảng 1 và trọng số của các mục theo lô Bảng 2. Ta có, $K = 3$,

$MAXW(1) = 0.8,\ MAXW(2) = 0.9,\ MAXW(3) = 0.8;$

tần số xuất hiện của "bd" trong lô 1, 2 và 3 lần lượt là 2, 1 và 2. Nên

$\text{MAXAWsupp}(bd) = 0.8 \times 2 + 0.9 \times 1 + 0.8 \times 2 = 4.1;$

Định nghĩa 6. Tại thời điểm T_i, cho dòng dữ liệu DS gồm K lô và X là một tập mục. Với ngưỡng ξ tính bởi (2), X được gọi là TMTX với trọng số thích nghi cực đại nếu

$$\text{MAXAWsupp}(X) \geq \xi \tag{2.6}$$

Mệnh đề 1. TMTX với trọng số thích nghi cực đại có tính chất Apriori, nghĩa là nếu X là một TMTX với trọng số thích nghi cực đại thì mọi tập con của nó cũng là TMTX với trọng số thích nghi cực đại trên dòng dữ liệu DS.

Chứng minh. Tại thời điểm T_i. $\forall Y \subseteq X$, ta có $F(Y, j) \geq F(X, j), j = 1, ..., K$.

Suy ra

$$\sum_{j=1}^{K} \text{MAXW}(j) \times F(Y, j) \geq \sum_{j=1}^{K} MAXW(j) \times F(X, j).$$

Hay $MAXAWsupp(Y) \geq MAXAWsupp(X)$.

Do đó, nếu $MAXAWsupp(X) \geq \xi$

Thì ta cũng có $MAXAWsupp(Y) \geq \xi$

Mệnh đề 2. Tại thời điểm T_i, cho dòng dữ liệu DS và X là một tập mục. Nếu X là TMTX với trọng số thích nghi thì X phải là TMTX với trọng số thích nghi cực đại trên dòng dữ liệu DS.

Chứng minh. Tại thời điểm T_i. $\forall X \subseteq I$, ta luôn có $MAXW(j) \geq W(X, j)\ \forall\ j = 1, ..., K$.

Do đó, nếu $\sum_{j=1}^{K} W(X, j) \times F(X, j) \geq \xi$ thì cũng có $\sum_{j=1}^{K} MAXW(j) \times F(X, j) \geq \xi$

Các Mệnh đề 1 và Mệnh đề 2 trên đây cho thấy các TMTX với trọng số thích nghi cực đại có tính chất Apriori và chúng là những ứng viên cho các TMTX với trọng số thích nghi trên dòng dữ liệu. Do đó, để khai phá các TMTX với trọng số thích nghi trên dòng dữ liệu, trong thuật toán SWFI-miner gồm hai công đoạn chính:

Thứ nhất, tìm tất cả các TMTX với trọng số thích nghi cực đại trên dòng dữ liệu.

Thứ hai, từ tập các TMTX với trọng số thích nghi cực đại, áp dụng (1) để xác định tập các TMTX với trọng số thích nghi trên dòng dữ liệu.

2.2. Khai phá tập mục thường xuyên với trọng số thích nghi trên dòng dữ liệu

2.2.1. Xây dựng cấu trúc cây SAWFI-tree

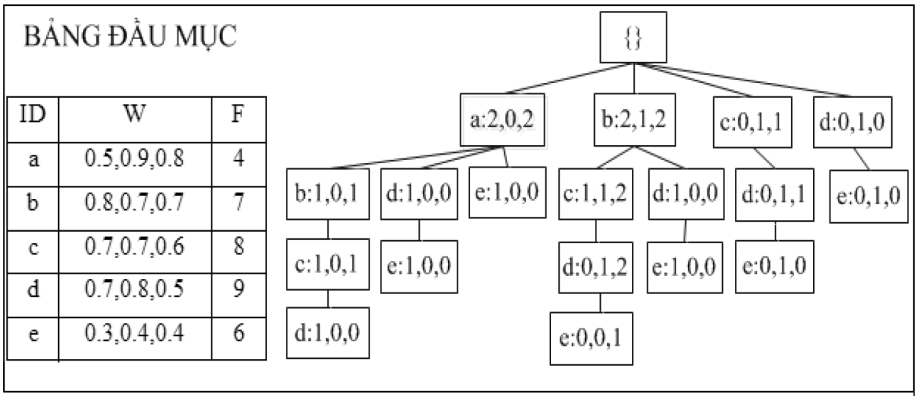
Sử dụng kiểu xây dựng cấu trúc cây FP-tree [7,8], SAWFI-tree bao gồm một cây và một bảng đầu mục. Để xây dựng cấu trúc cây SAWFI-tree thuật toán chỉ cần quét toàn bộ dòng dữ liệu một lần.

Cây SAWFI-tree

Gồm một nút gốc gọi là nút "null" (kí hiệu là) và một tập các cây tiền tố là các cây con của nút gốc. Các giao tác của mỗi lô trong CSDL sẽ lần lượt được chèn lên cây theo thứ tự từ điển của các mục. Ngoại trừ nút gốc, mỗi nút của SAWFI-tree ghi lại tên của mục mà nó đại diện, thông tin về tần số xuất hiện của nút trong mỗi lô trên đường đi từ gốc đến nó và các con trỏ trỏ đến nút cha, nút con, nút cùng tên tiếp theo trên cây. Khi một nút mới được tạo ra trên cây bởi việc chèn một giao tác từ lô thứ k của cửa sổ hiện tại gồm K lô, thì tại đó một danh sách gồm K giá trị tần số trong K lô sẽ được khởi tạo với giá trị bằng 1 tại vị trí thứ k, giá trị bằng 0 tại tất cả các vị trí còn lại. Ví dụ, nếu cửa sổ hiện tại gồm 3 lô và “b” là một nút xuất hiện lần đầu tiên trên cây do chèn một giao tác từ lô thứ hai, khi đó cấu trúc của nút “b” sẽ là b:0,1,0.

Bảng đầu mục

Bảng đầu mục lưu trữ các mục theo thứ tự từ điển, thông tin về trọng số, tần số của các mục và con trỏ trỏ đến nút cùng tên đầu tiên của SAWFI-tree. Hình 1 biểu diễn cây SAWFI-tree và bảng đầu mục (để đơn giản hình chúng tôi không vẽ các con trỏ). Ta có thể dễ dàng phát hiện ra các giao tác của mỗi lô và tần số xuất hiện của các mục trong các lô của dòng dữ liệu. Chẳng hạn, giao tác $\{b, c, d, e\}$ xuất hiện một lần ở lô thứ ba (B_{13}) và giao tác $\{b, c, d\}$ xuất hiện hai lần: một lần ở lô thứ hai (B_{12}) và một lần ở lô thứ ba (B_{13}) (nằm trên nhánh thứ tư từ phải sang). Ta cũng có số đếm hỗ trợ của các mục trong cửa sổ khai phá lần lượt là a:4, b:7, c:8, d:9 và e:6.



Hình 1 Cây SAWFI-tree(d), cây điều kiện của "e"

2.2.2. Thuật toán khai phá SWFI-miner

Dưới đây là một số tính chất quan trọng của SAWFI-tree được chúng tôi sử dụng trong quá trình khai phá TMTX với trọng số thích nghi trên dòng dữ liệu theo kiểu FP-growth [7,8].

Tính chất 1. Cấp cao nhất của cây SAWFP-tree bằng độ dài của giao tác dài nhất trên dòng dữ liệu.

Tính chất 2. Tổng các giá trị tần số trong các lô tại bất kì nút nào trên cây cũng lớn hơn hoặc bằng tổng các giá trị tần số tại các nút con của nó.

Tính chất 3. Tần số xuất hiện trong mỗi lô của một mục trên cây bằng tổng các tần số tương ứng của các nút cùng tên.

Tính chất 4. Phân bố tần số trong các lô của đường đi trên cây chính là phân bố tần số của nút hậu tố.

Tính chất 5. Cây điều kiện của mục cao nhất theo thứ tự từ điển là cây rỗng.

Sử dụng cách tiếp cận FP-growth [7,8], thủ tục SWFI-miner khai phá TMTX với trọng số thích nghi trên dòng dữ liệu từ cây SAWFP-tree như sau:

Algorithm SWFI-miner;
Input: (1) T_i là thời điểm cần khai phá.
(2) Cây SAWFI-tree.
(3) Bảng trọng số các mục theo các lô.
(4) minsupp - độ hỗ trợ tối thiểu.
Output: L - Tập các TMTX với trọng số thích nghi trên dòng dữ liệu;

Method:
1. Tại thời điểm T_i. Tính độ hỗ trợ với trọng số tối thiểu ξ theo (2);
2. $L = \emptyset$;
3. Từ Bảng đầu mục, xác định tập C_1 là tập các 1-tập mục ứng viên thỏa ξ;
4. $L = C_1$;
5. **For each** (mục σ trong Bảng đầu mục, theo thứ tự từ điển từ dưới lên) do
6. **Begin**
   6.1. Tạo cây có điều kiện cho mục σ tương ứng;
   6.2. Thiết lập các tập ứng viên;
   6.3. Loại bỏ các tập ứng viên có số đếm hỗ trợ không thỏa ξ;
   6.4. Nhập các tập ứng viên thỏa ξ vào L;
   6.5. Xóa tất cả các nút σ đã được xét trên cây điều kiện;
7. **End;**
8. Tính độ hỗ trợ thực tế của tập ứng viên theo (1). Theo (3), ta thu được L là tập các TMTX thỏa ξ trên dòng dữ liệu tại thời điểm T_i.
9. **Return** L.

Ví dụ:

Cho dòng dữ liệu trong Bảng 1, tại thời điểm T_1,có 12 giao tác, 3 lô B_{11}, B_{12}, B_{13} với trọng số của các mục tại các lô trong Bảng 2 và độ hỗ trợ tối thiểu minsupp là 30%.

1. Tại thời điểm T_1. Xây dựng cây SAWFI-tree, ta thu được cây như Hình 1.

2. Tính độ hỗ trợ với trọng số tối thiểu ξ:

3. $\xi = \text{minsupp} \times \sum_{j=1}^{K} |B_{ij}| \times \overline{W_{ij}} = 2.25;$

4. Từ bảng đầu mục, ta có:

$MAXW(1) = 0.8; MAXW(2) = 0.9; MAXW(3) = 0.8;$

$$\text{MAXAWsupp}(a) = 0.8 \times 2 + 0.9 \times 1 + 0.8 \times 2 = 4.1; \text{MAXAWsupp}(b) = 5.7;$$
$$\text{MAXAWsupp}(c) = 6.6; \text{MAXAWsupp}(d) = 7.5; \text{MAXAWsupp}(e) = 5.0;$$

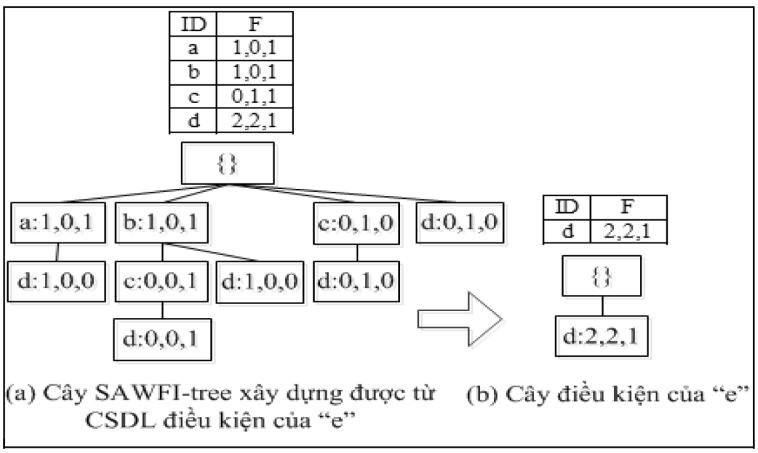
Tất cả các mục đơn đều có giá trị $MAXAWsupp$ lớn hơn $\xi = 2.25$, nên chúng không bị tỉa trên cây và đều là những ứng viên đơn. Vậy ta có $L = \{a, b, c, d, e\}$.

5. Xây dựng và khai phá các cây điều kiện của các mục theo thứ tự dưới lên trong bảng đầu mục.

a) Xây dựng và khai phá cây điều kiện của "e".

CSDL điều kiện của mục "e" gồm các nhánh tiền tố $\{ad : 1,0,0; a : 0,0,1; bcd : 0,0,1; bd : 1,0,0; cd : 0,1,0; d : 0,1,0\}$. Từ CSDL điều kiện này ta có cây SAWFI-tree(e) trong Hình 2(a).

Vì CSDL điều kiện của "e" có đầy đủ các mục của CSDL ban đầu nên $MAXW(1) = 0.8$, $MAXW(2) = 0.9$, $MAXW(3) = 0.8$. Từ bảng đầu mục ta có tần số xuất hiện cùng với "e" của các mục trong từng lô là $< a : 1,0,1; b : 1,0,1; c : 0,1,1; d : 2,2,1 >$.



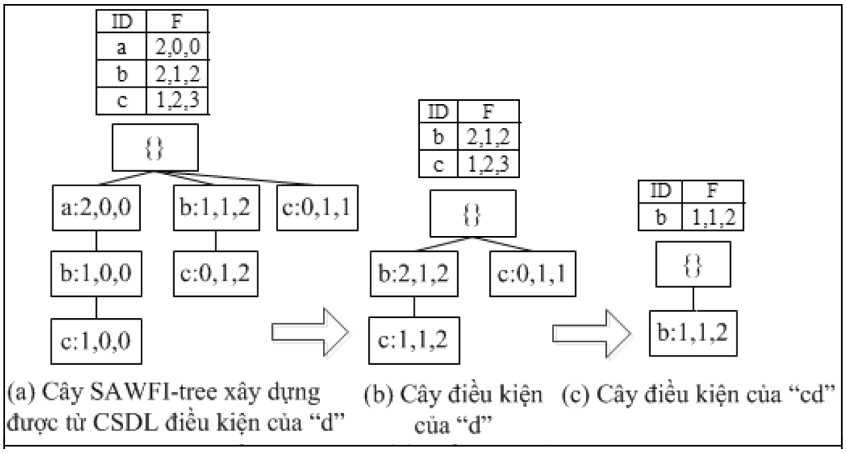
Hình 2. Cây SAWFI-tree, cây điều kiện của "e"

Sử dụng (5), ta tính độ hỗ trợ với trọng số thích nghi cực đại của các mục là $\langle a : 1.6; b : 1.6; c : 1.7; d : 4.2\rangle$. Với $\xi = 2.25$, chỉ có mục "d" không bị loại khỏi cây SAWFI-tree(e). Sau khi loại bỏ các mục không thỏa ξ, chỉ giữ lại mục "d" ta có cây điều kiện của mục "e" là cây Hình 2(b).

Từ cây điều kiện này cùng với bảng đầu mục, đồng thời sử dụng (5), ta thu được một 2-tập mục "de" cùng độ hỗ trợ với trọng số thích nghi cực đại $\langle de : 4.2\rangle$, thỏa ξ. Khai phá tiếp cây điều kiện của "de", thu được cây rỗng. Vậy ta có tập ứng viên $L = \{a, b, c, d, e, de\}$.

b) Xây dựng và khai phá cây điều kiện của "d".

CSDL điều kiện của mục "d" bao gồm các nhánh tiền tố $\{abc : 1,0,0; a : 1,0,0; bc : 0,1,2; b : 1,0,0; c : 0,1,1\}$. Từ CSDL điều kiện này ta có cây SAWFI-tree(d) trong Hình 3(a). Vì CSDL điều kiện của "d" có các mục "a", "b" và "c" của CSDL ban đầu nên $MAXW(1) = 0.8$, $MAXW(2) = 0.9$, $MAXW(3) = 0.8$. Từ bảng đầu mục ta có tần số xuất hiện cùng với "d" của các mục trong từng lô là $\langle a : 2,0,0; b : 2,1,2; c : 1,2,3\rangle$.

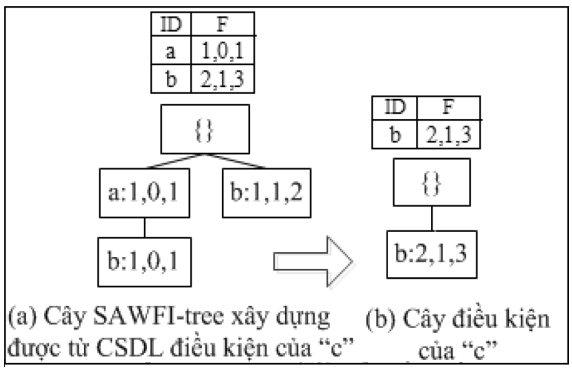


Hình 3. Cây SAWFI-tree(d), cây điều kiện của "d" và "cd"

Sử dụng (5), ta tính độ hỗ trợ với trọng số thích nghi cực đại của các mục là $\langle a:1.6; b:4.1; c:5.0\rangle$. Với $\xi = 2.25$, mục "a" bị loại khỏi cây SAWFI-tree(d), ta thu được cây điều kiện của mục "d" là cây Hình 3(b). Từ cây điều kiện này, đồng thời sử dụng (5), ta thu được hai 2-tập mục ứng viên là "bd" và "cd". Tần số xuất hiện của các 2-tập mục trong từng lô là $\langle bd:2,1,2; cd:1,2,3\rangle$ và độ hỗ trợ với trọng số thích nghi cực đại tương là $\langle bd:4.1; cd:5.0\rangle$. Các 2-tập mục này thỏa ξ. Vậy ta có, $L = \{a, b, c, d, e, de, bd, cd\}$. Tiếp tục khai phá cây điều kiện của "bd" là cây rỗng và khai phá cây điều kiện của "cd" ta thu được cây điều kiện là cây Hình 3(c), với một 3-tập mục "bcd" cùng tần số xuất hiện trong từng lô là $\langle bcd:1,1,2\rangle$ và độ hỗ trợ với trọng số thích nghi cực đại là và $\langle bcd:3.3\rangle$. Vậy ta có, $L = \{a, b, c, d, e, de, bd, cd, bcd\}$.

c) Xây dựng và khai phá cây điều kiện của "c".

CSDL điều kiện của mục "c" có các nhánh tiền tố $\{ab:1,0,1; b:1,1,2\}$. Từ CSDL điều kiện ta có cây SAWFI-tree(c) trong Hình 4(a).



Hình 4. Cây SAWFI-tree(c), cây điều kiện của "c"

Vì CSDL điều kiện của "c" có các mục "a", "b" của CSDL ban đầu nên $MAXW(1) = 0.8$, $MAXW(2) = 0.9$, $MAXW(3) = 0.8$; Từ bảng đầu mục ta có tần số xuất hiện cùng với "c" của các mục trong từng lô là $\langle a:1,0,1; b:2,1,3\rangle$. Sử dụng (5), ta tính độ hỗ trợ với trọng số thích nghi cực đại của các mục là $\langle a:1.6; b:4.9\rangle$. Với $\xi = 2.25$, mục "a" bị loại khỏi cây

SAWFI-tree(c), ta thu được cây điều kiện của mục "c" là cây Hình 4(b). Từ cây điều kiện này, đồng thời sử dụng (5), ta thu được một 2-tập mục ứng viên là "bc". Tần số xuất hiện của một 2-tập mục trong từng lô là $\langle bc : 2, 1, 3\rangle$ và độ hỗ trợ với trọng số thích nghi cực đại là $\langle bc : 4.9\rangle$, thỏa ξ. Nên $L = \{a, b, c, d, e, de, bd, cd, bcd, bc\}$. Tiếp tục khai phá cây điều kiện của "bc" thu được cây rỗng. Vậy ta có, $L = \{a, b, c, d, e, de, bd, cd, bcd, bc\}$.

d) Xây dựng và khai phá cây điều kiện của "b".

CSDL điều kiện của mục "b" có một nhánh tiền tố $\{a : 1, 0, 1\}$. Từ CSDL điều kiện này ta được cây SAWFI-tree(b) chỉ có một nút a:1,0,1 và từ bảng đầu mục ta có tần số xuất hiện cùng với "b" của các mục trong từng lô là $\langle a : 1, 0, 1\rangle$ và độ hỗ trợ với trọng số thích nghi cực đại của mục "a" là $\langle a : 1.6\rangle$, với $\xi = 2.25$, mục "a" bị loại khỏi cây. Vậy ta có, $L = \{a, b, c, d, e, de, bd, cd, bcd, bc\}$.

e) Xây dựng và khai phá cây điều kiện của "a".

Theo Tính chất 5, ta thu được cây rỗng.

6. Tính độ hỗ trợ thực tế của các tập ứng viên theo (1), loại bỏ các tập không thỏa ξ. Kết quả khai phá dòng dữ liệu tại thời điểm T_1 thu được tập các TMTX với trọng số thích nghi cùng với độ hỗ trợ:

7. $L = \left\{ \begin{array}{l} 8.a : 2.6, b : 5.2, c : 5.2, d : 4.0, de : 2.6, \\ 9.bd : 3.45, cd : 4.15, bcd : 2.8, bc : 4.2510. \end{array} \right\}$

2.2.3. Thủ tục cập nhật cây SAWFI-tree

Theo như đã trình bày trong mục 3.1, việc tổ chức lưu trữ dữ liệu dòng giao tác dưới dạng cấu trúc cây như SAWFI-tree cho phép ta có thể dễ dàng cập nhật thông tin (xóa các giao tác trong một lô cũ nhất, bổ sung các giao tác cho một lô mới nhất), đáp ứng sự biến đổi nhanh của dòng dữ liệu tại những thời điểm tiếp theo.

Để xóa thông tin của lô cũ nhất trên cây SAWFI-tree, ta cần thực hiện như sau:

Trong danh sách các giá trị tần số xuất hiện của mỗi nút, tại ví trí thứ j $(1 < j \leq K)$ bằng giá trị tần số của vị trí thứ $j - 1$ và thay giá trị tại vị trí thứ nhất bằng 0.

Tỉa tất cả nút mà tại đó mọi giá trị tần số đều bằng 0.

Các giao tác của lô mới được chèn lên cây như thường lệ sau khi đã xóa bỏ thông tin của lô cũ nhất.

2.3. Một số phân tích và đánh giá

Thuật toán đề xuất đã có những ưu điểm sau:

Bước xây dựng cây SAWFI-tree chỉ cần duyệt một lần trên toàn bộ dòng dữ liệu. Đặc biệt, bước cập nhật cây chỉ duyệt một lần trên lô mới nhất để chèn các giao tác lên cây.

Việc xây dựng cấu trúc cây chỉ cần duyệt dòng dữ liệu một lần. Cây SAWFI-tree có cấu trúc giống cây FP-tree, do vậy dễ dàng xây dựng và xử lí khai phá. Bản chất cấu trúc của cây SAWFI-tree(x) là kết quả của phép chiếu của cây SAWFI-tree cho từng mục dữ liệu x. Như vậy, với cách làm này đã "chia để trị" bài toán lớn thành nhiều bài toán nhỏ đơn giản hơn với các xử lí tương tự.

Dễ thấy, chi phí chèn một giao tác T lên cây là $\mathrm{O}(|T \cap C|)$, với C là tập mục có khả năng là TMTX với trọng số cực đại.

Không kể nút gốc, chiều cao của cây SAWFI-tree có cận trên là (Với $N = |I|$ là số các mục trong dòng dữ liệu). Vì thông thường mỗi giao tác được chèn lên cây tương ứng với một nhánh trên cây, các giao tác có phần tiền tố giống nhau có đường đi chung trên cây, do vậy chiều cao của cây bằng số mục có độ dài lớn nhất mà là TMTX với trọng số thích nghi cực đại, tức là $\underset{T \in DS}{Max}\{|T \cap C|\} \leq N$.

$\sum\limits_{T \in DS} |T \cap C| \leq N \times M$ Không kể nút gốc, kích thước (số nút) của cây có cận trên là $\sum\limits_{T \in DS} |T \cap C| \leq N \times M$ với $M = |DS|$ là số giao tác của dòng dữ liệu. Lí do là: (1) Trường hợp tốt nhất, tất cả $\underset{T \in DS}{Max}\{|T \cap C|\} \leq N$các giao tác đều có chung một mục (nghĩa là tất cả các giao tác trong dòng dữ liệu đều là tập con của giao tác có độ dài lớn nhất), khi đó cây SAWFI-tree chỉ có duy nhất một nhánh, số nút của cây bằng số nút của nhánh đó. (2) Trường hợp xấu nhất, mỗi giao tác không chứa chung một tập mục nào, khi đó số nút tối đa của của cây bằng tổng sô mục xuất hiện trong các giao tác.

Cũng vì các giao tác thường chia sẻ với nhau một số nút trên cây, nên kích thước cây SAWFI-tree thường nhỏ hơn kích thước của dòng dữ liệu. Dòng dữ liệu càng dày thì kích thước cây SAWFI-tree càng nhỏ. Đồng thời, các cây SAWFI-tree(x) cũng có kích thước không lớn hơn kích thước cây SAWFI-tree.

Cây SAWFI-tree được xây dựng có cấu trúc giống cây FP-tree [7,8], nên việc khai phá TMTX với trọng số thích nghi cực đại, trọng số thích nghi trên dòng dữ liệu khả thi và hiệu quả.

Thuật toán AWFI-miner được phát triển dựa trên phương pháp khai phá của thuật toán FP-growth [7,8] nên chắc chắn đảm bảo tính dừng và hiệu quả.

3. Kết luận

Bài báo đã đề xuất một độ đo độ mới (độ hỗ trợ với trọng số thích nghi cực đại) bởi (5) cho phép tỉa cây SAWFI-tree và các cây điều kiện hiệu quả hơn trong đề xuất bởi bởi Chowdhudy F. A. và cộng sự [5]. Bài báo cũng mở rộng việc khai phá TMTX với trọng số thích nghi cho dòng dữ liệu. Trong [13], Tsai P. S. M. đề xuất gán trọng số cho từng lô (có nghĩa tất cả các các tập mục trong các giao tác của lô đều gán trọng số như nhau), thì trong đề xuất của chúng tôi mỗi mục trong mỗi lô đều được gán một trọng số khác nhau, và trọng số của tập mục được tính là trung bình các trọng số tham gia trong tập mục đó.

Với những phân tích, đánh giá trên đây có thể nói rằng thuật toán SWFI-miner là một thuật toán hiệu quả để khai phá TMTX với trọng số thích nghi trên dòng dữ liệu.

TÀI LIỆU THAM KHẢO

[1] Aggarwal C. (Ed.), 2007. *Data Streams: Models and algorithms*. Springer.

[2] Agrawal R., Srikant, R., 1994. *Fast Algorithms for Mining Association Rules*. In: 20th Int. Conf. on Very Large Data Bases (VLDB), pp. 487-499.

[3] Aneri P., Chaudhari M. B., 2014. *Frequent pattern mining of continuous data over data streams*. Int. Jour. for Technology Research Engineering, Vol. 1, Issue 9, pp. 935-940.

[4] Chi Y., Wang H., Yu P. S., Muntz R. R., 2006. *Catch the moment: Maintaining closed frequent itemsets over a data stream sliding window*. Knowledge and Information Systems,

Vol. 10, No. 3, pp. 265-294.

[5] Chowdhury F. A., Syed K. T., Byeong-Soo J., Young-Koo L., 2008. *Mining Weighted Frequent Patterns Using Adaptive Weights*. In: Fyfe et al. (Eds.): IDEAL 2008, LNCS 5326, 2008, 258-265.

[6] Fan W., Huang Y., Wang H., Yu, P. S., 2004. *Active mining of data streams*. In: Proceedings of the Fourth SIAM Int. Conf. on Data Mining, pp. 457-461.

[7] Han J., Kamber M., 2000. *Data Mining: Concepts and Techniques*. Morgan Kanufmann.

[8] Han J., Pei J., Yin Y., Mao R., 2004. *Mining frequent patterns without candidate generation: a frequent-pattern tree approach*. Data Mining and Knowledge Discovery 8, pp. 53-87.

[9] Kuen-Fang J., Chao-Wei L., 2010. *A sliding-window based adaptive approximating method to discover recent frequent itemsets from data streams*. Proc. of the Int. Multiconference of Engineering and Computer Scientists (IMECS 2010), Vol. I, March 17-19, Hong Kong.

[10] Li Su, Hong-yan Liu, 2011. *A new classfication algorithm for data stream*. Int. Jour. Modern Education and Computer Science, Vol. 3, No. 4, pp, 32-39.

[11] Reshma Yusuf B., Chenna Reddy B., Mining data stream using option trees, Int. Jour. Network and Information Security, Vol. 4, No. 8, pp. 49-54, (2012)

[12] Shaik H., Murthy J. V. R., Anuradha Y., Chandra M., 2012. *Mining frequent patterns from data streams using dynamic DP-tree*. Int. Jour. of Computer Applications, Vol. 52, No. 19, pp. 23-27.

[13] Tsai P. S. M., 2009. *Mining frequent itemsets in data streams using the weighted sliding window model*. Expert Systems with Applications, pp. 11617-11625.

[14] Wang J., Zeng Y., 2012. *SWFP-Miner: An efficient algorithm for mining weight frequent pattern over data streams*. High Technology Letters, Vol. 3, No. 3, pp. 289-294.

[15] Younghee K., Wonyoung K., Ungmo K., 2010. *Mining frequent itemsets with normalized weight in continuous data streams*. Journal of Information Processing Systems, Vol. 6, No. 1, pp. 79-90.

ABSTRACT

Eficient Mining frequent itemsets with adaptive weights over data streams

The SWFI-miner algorithm has been proposed for mining the frequent index itemsets with adaptive weights over data streams. In this paper, we have proposed a new measurement unit to prune the SAWFI-tree and conditional trees. We also expand the algorithm from mining frequent itemsets with adaptive weights in a static database to the one over data streams. These are based on the models derived from Chowdhury F. A. et al. [5] and Tsai P. S. M. [13]. By analysis and evaluation of samples, the proposed algorithm of the SWFI-miner shows better performance in mining frequent itemsets with adaptive weights over data streams.

Keywords: Data Mining, frequent itemsets, weights, adaptive weights, data stream.