

MỘT THUẬT TOÁN HIỆU QUẢ KHAI THÁC MẪU ĐÓNG TRÊN CSDL ĐỊNH LƯỢNG ĐỘNG

An Efficient Algorithm for Mining Closed Patterns on Dynamic Quantitative Databases

Nguyễn Duy Hàm, Lê Thị Thúy Hằng*, Tô Thị Ngọc Châu

Trường Đại học Sài Gòn

Tóm tắt

Trong những năm gần đây, bài toán khai thác các mẫu hữu ích có trọng số phổ biến (FWUP), một biến thể của các mẫu phổ biến, đã thu hút được sự quan tâm nghiên cứu. Một số thuật toán đã được đề xuất để giải quyết vấn đề này bằng cách sử dụng cơ sở dữ liệu định lượng với trọng số của các mục hàng cố định trong toàn bộ Cơ sở dữ liệu (CSDL). Tuy nhiên, trong thực tế, nhiều yếu tố ảnh hưởng đến trọng số của các mục hàng, và chúng có thể thay đổi theo thời gian để phản ánh các điều kiện thực tế. Hơn nữa, việc khai thác FWUP truyền thống thường tốn nhiều tài nguyên do số lượng FWUP được tạo ra phụ thuộc nhiều vào ngưỡng khai thác. Trong bài báo này, chúng tôi giới thiệu bài toán khai thác các mẫu đóng hữu ích có trọng số phổ biến (FWUCP), đây là một dạng cô đọng mà từ đó có thể suy ra tất cả các FWUP, từ các cơ sở dữ liệu định lượng động – CSDL mà trọng số của các mục hàng có thể điều chỉnh được. Thuật toán đề xuất nFWUCP sử dụng cấu trúc WUNList, một phần mở rộng của cấu trúc N-list, để khai thác hiệu quả các FWUCP. Các kết quả thực nghiệm cho thấy phương pháp đề xuất hiệu quả hơn về thời gian thực thi so với thuật toán tốt nhất hiện tại.

Từ khóa: khai thác mẫu phổ biến; mẫu phổ biến đóng; CSDL định lượng; mẫu phổ biến trọng số hữu ích.

Abstract

In recent years, the problem of mining frequent weighted utility patterns (FWUPs)-a variant of frequent patterns-has attracted significant attention from the research community. Several algorithms have been proposed to address this problem using quantitative databases in which item weights remain fixed across the entire database. However, in real-world scenarios, various factors can affect item weights, and these weights may change over time to better reflect actual conditions. Moreover, traditional FWUP mining methods are often resource-intensive, as the number of generated patterns is highly sensitive to the chosen mining threshold.

In this paper, we propose the problem of mining frequent weighted utility closed patterns (FWUCPs), a condensed representation from which all FWUPs can be derived, from dynamic quantitative databases, where item weights can be adjusted. The proposed algorithm, nFWUCP, leverages the WUNList structure, an extension of the N-list structure, to efficiently discover FWUCPs. Experimental results demonstrate that our approach achieves superior execution performance compared to the current state-of-the-art algorithm.

Keywords: data mining; frequent weighted utility patterns; closed patterns; dynamic quantitative databases.

*Tác giả liên hệ: thuyhang@sgu.edu.vn

1. Giới thiệu tổng quan

Bài toán khai thác mẫu phổ biến (FP-frequent pattern) là một bài toán nền tảng, quan trọng của lĩnh vực khai thác dữ liệu, các kết quả của nó có thể là input cho các hệ thống thông minh. Do đó nó luôn được nhận được sự quan tâm nghiên cứu. Agrawal và các cộng sự [1] đã lần đầu tiên giới thiệu thuật toán Apriori để khai thác FP. Kể từ đó, nhiều thuật toán đã được đề xuất để giải quyết vấn đề này, bao gồm FP-Growth [2], Eclat [3], PrePost [4], và NSFI [5], cùng nhiều thuật toán khác.

Mặc dù khai thác FP có tính ứng dụng cao, nhưng nó phải đối mặt với thực tế tạo ra một số lượng lớn FP trong các ứng dụng thực tế. Điều này dẫn đến thời gian khai thác cao và bộ nhớ sử dụng lớn. Để giải quyết vấn đề này, Pasquier và các cộng sự [6] đã giới thiệu bài toán khai thác các mẫu đóng phổ biến (FCP- Frequent Close Pattern). FCP là các mẫu phổ biến mà không có tập cha nào có cùng độ hỗ trợ. Điều này dẫn đến số lượng mẫu giảm đáng kể, và điều quan trọng là tất cả các FP có thể được suy ra từ FCP. Do đó, khai thác FCP làm giảm số lượng mẫu được khai thác. Nhiều thuật toán hiệu quả đã được đề xuất để giải quyết bài toán này, như: CHARM [7], DCI_Closed [8], NAFCP [9], và NEclatClosed [10].

Khai thác mẫu hữu ích có trọng số phổ biến (FWUP) là một biến thể quan trọng trong cơ sở dữ liệu định lượng. Không giống như cơ sở dữ liệu nhị phân, mỗi giao dịch trong cơ sở dữ liệu định lượng lưu trữ số lượng của mỗi mặt hàng được mua, với mỗi mặt hàng được gán một trọng số. Các ràng buộc về tần suất và trọng số ảnh hưởng đáng kể đến độ chính xác của việc trích xuất thông tin. Do đó, khai thác FWUP tập trung vào việc xác định các mẫu vừa phổ biến vừa có giá trị. Đầu tiên, năm 2008 Khan và các cộng sự [11] đã giới thiệu bài toán này với thuật toán Apriori-FWUI. Sau đó, một số thuật toán khác hiệu quả hơn đã được đề xuất, chẳng hạn như MBIS-FWUI [12], WUN-Miner [13], và FWUP-SWUNL [14]. Gần đây, Nguyen và các cộng sự [15] đã mở rộng khái niệm này bằng cách đề xuất khai thác FWUP trên các cơ sở dữ liệu định lượng động, là CSDL mà trọng số của các mục có thể thay đổi. Đề xuất này dựa trên quan sát rằng trọng số của các mục thường thay đổi do các mục đích thực tế khác nhau. Ví dụ, trên các nền tảng thương mại điện tử, giá sản phẩm thường giảm trong những ngày khuyến mãi hoặc các sự kiện đặc biệt và có thể tăng khi các mặt hàng đang là xu hướng hoặc có nguồn cung hạn chế. Để giải quyết vấn đề này, các tác giả đã định nghĩa lại công thức trọng số hữu ích và đề xuất hai thuật toán, dFWUT và dFWUNL, sử dụng các cấu trúc dữ liệu tidset và WUNList.

Trong bài báo này, chúng tôi giới thiệu bài toán “Khai thác mẫu đóng trên CSDL định lượng động”, được gọi là mẫu đóng hữu ích có trọng số động (FWUCP – Frequent Weighted Utility Close Pattern). FWUCP là các mẫu đại diện cho FWUP mà không có tập cha nào có cùng độ hỗ trợ hữu ích với nó. Từ tập hợp cô đọng các FWUCP này, người dùng có thể suy ra tập hợp đầy đủ các FWUP cùng với các giá trị hỗ trợ hữu ích có trọng số tương ứng của chúng.

Để giải quyết hiệu quả bài toán này, chúng tôi sử dụng các chiến lược cắt tia tối ưu không gian tìm kiếm trong quá trình khai thác, đồng thời đề xuất một thuật toán mới nFWUCP dựa trên cấu trúc dữ liệu WUNlist [13][15] là một biến thể của cấu trúc N-List đã

được Deng giới thiệu tại [4], đây là cấu trúc hiệu quả nhất cho các bài toán khai thác mẫu hiện nay. Sau đó chúng tôi tiến hành đánh giá trên các bộ CSDL để xác định tính hiệu quả của phương pháp đề xuất.

Chúng tôi tổ chức phần còn lại của bài báo này như sau. Phần 2 đánh giá các công trình liên quan trong lĩnh vực. Phần 3 giới thiệu bài toán khai thác FWUCP trên cơ sở dữ liệu định lượng động và cấu trúc dữ liệu WUNList. Phần 4 mô tả ba chiến lược cắt tia được đề xuất và thuật toán nFWUCP, kèm theo các ví dụ minh họa. Phần 5 thảo luận về kết quả thực nghiệm, và Phần 6 kết luận bài báo với những hiểu biết chính và đề xuất các hướng nghiên cứu trong tương lai.

2. Nghiên cứu liên quan

2.1. Mẫu phổ biến đóng

Trong khai thác mẫu tổng quát, số lượng mẫu được tạo ra thường rất lớn, dẫn đến một tập hợp đầu ra đáng kể có thể chứa nhiều mẫu dư thừa, vốn có thể không có giá trị đối với người dùng. Nhận thấy vấn đề này, Pasquier et al. [6] đã giới thiệu bài toán khai thác mẫu đóng phổ biến. Một mẫu được coi là đóng nếu không tồn tại siêu tập hợp nào có cùng tần suất, điều này làm giảm đáng kể số lượng mẫu so với các FP. Các mẫu đóng cung cấp cho người dùng kết quả có ý nghĩa hơn, không dư thừa, và tất cả các FP có thể được suy ra từ FCP. Để giải quyết bài toán này, các tác giả đã đề xuất thuật toán A-Close, dựa trên phương pháp Apriori. Tuy nhiên, tiếp cận này không hiệu quả vì nó quét cơ sở dữ liệu nhiều lần và tạo ra vô số ứng cử viên. Để cải thiện hiệu quả, năm 2000 Pei và các cộng sự [18] đã đề xuất các thuật toán CLOSET, năm 2003, Wang và các cộng sự [19] đề xuất CLOSET+ là một cải tiến của CLOSET [18]. Sau đó, các thuật toán CHARM [7] đã chuyển từ biểu diễn cơ sở dữ liệu ngang sang dọc, tạo cấu trúc tidset để khai thác FCP. CHARM sử dụng cấu trúc IT-tree theo tiếp cận Eclat[3] cùng với bảng băm để lưu trữ và lọc hiệu quả các mẫu không đóng. Sau đó, Le và Vo [9] đã đề xuất thuật toán NAFCP, sử dụng cách tiếp cận lai để khai thác FCP. NAFCP sử dụng cấu trúc N-list [4] kết hợp với các định lý để nhanh chóng kiểm tra xem N-list của một mẫu có phải là tập con của mẫu khác hay không.

Bên cạnh đó, việc khai thác mẫu phổ biến trên CSDL định lượng (FWUI) cũng là một hướng được nhận được nhiều quan tâm, từ những nghiên cứu đầu tiên đặt nền móng cho bài toán này như Khan và các cộng sự [11], sau đó Nguyen và các cộng sự đề xuất cấu trúc Bit vecto [12] để giảm không gian lưu trữ tidset cho các tập mục theo hướng tiếp cận Eclat [3]. Các nghiên cứu tiếp theo [13-16] theo hướng tiếp cận N-list [4]. Gần đây Nguyen và các cộng sự [17] đề xuất một cấu trúc bit EIWS mở rộng tối ưu để khai thác hiệu quả FWUI trên CSDL phân cấp.

Cấu trúc N-list có ưu điểm nhờ khả năng nén dữ liệu, đặc biệt trên các tập dữ liệu dày đặc. Gần đây hơn, năm 2021 Aryabarzan và các cộng sự đã đề xuất thuật toán NeclatClosed [10], dựa trên thuật toán Eclat, để khai thác FCP. Kết quả thực nghiệm cho thấy NEclatClosed hiệu quả hơn các thuật toán trước đây về cả thời gian khai thác và mức sử dụng bộ nhớ.

3. Thuật toán khai thác mẫu đóng trọng số hữu ích trên CSDL động – nFWUCP

3.1. Phát biểu Bài toán

Định nghĩa 1. (Cơ sở dữ liệu định lượng động) Cơ sở dữ liệu định lượng động (dQDB) là một dạng (T, I, DW) , trong đó $T = \{t_1, t_2, \dots, t_m\}$ là một tập hợp các giao dịch định lượng (m là số lượng giao dịch), $I = \{i_1, i_2, \dots, i_n\}$ là một tập hợp các mục trong mỗi giao dịch (n là số lượng mục), và $DW = \{dw_1, dw_2, \dots, dw_p\}$ là một tập hợp các trọng số (p là số lượng lô). Mỗi giao dịch $t_k = \{x_{k1}, x_{k2}, \dots, x_{kn}\}$, trong đó x_{ki} cho biết số lượng mục i được mua trong giao dịch t_k và tương ứng với $dw_j \in DW$. Mỗi lô $dw_j = \{w_1, w_2, \dots, w_n\}$ là một tập hợp các trọng số của các mục. Mỗi giao dịch có một định danh duy nhất được gọi là Tid.

Trong bài toán này, “động” đề cập đến trọng số của một mục không được coi là một thuộc tính tĩnh, mà là một thuộc tính có thể thay đổi ngay từ đầu trên các giao dịch khác nhau. Do đó, các công thức tính toán cho các số đo liên quan đến tiện ích có trọng số trong bài toán được điều chỉnh một cách linh hoạt để phù hợp với sự không đồng nhất này trong trọng số của các mục trong một quy trình khai thác duy nhất.

Bảng 1. Một ví dụ về CSDL định lượng động (dQDB_E).

A						
CSDL giao dịch						
Giao dịch	A	B	C	D	E	Trọng số
t_1	1	0	2	3	1	dw_1
t_2	0	2	1	0	2	
t_3	2	0	1	0	1	
t_4	1	0	3	2	0	dw_2
t_5	3	1	1	2	1	
t_6	1	2	2	0	3	

B					
Trọng số	A	B	C	D	E
dw_1	0.3	0.5	0.2	0.4	0.7
dw_2	0.4	0.3	0.4	0.2	0.3

Ví dụ 1. Bảng 1 trình bày một ví dụ về CSDL định lượng động (dQDB_E). Có sáu giao dịch $\{t_1, t_2, t_3, t_4, t_5, t_6\}$ và năm mặt hàng (mục) $\{A, B, C, D, E\}$, với định lượng giao dịch trong Bảng 1A. Bảng 1B là tập trọng số của các mục hàng cho mỗi lô. Giao dịch t_1, t_2, t_3 có tập trọng số $dw_1 = \{0.3, 0.5, 0.2, 0.4, 0.7\}$, và tập trọng số cho các mục của của giao dịch t_4, t_5, t_6 là $dw_2 = \{0.4, 0.3, 0.4, 0.2, 0.3\}$.

Định nghĩa 2. (Trọng số hữu ích của giao dịch) [11] Trọng số hữu ích (twu) của một giao dịch t_k , với tập trọng số dw_p , được định nghĩa như sau:

$$twu(t_k) = \frac{\sum_{i_j \in t_k} w_{pi_j} \times x_{ki_j}}{|t_k|} \quad (1)$$

Với $w_{pi_j} \in dw_p$ là trọng số của mục i_j , x_{ki_j} là số lượng của mục i_j trong t_k , và $|t_k|$ là tổng số mục trong t_k .

Định nghĩa 3. (Độ hỗ trợ trọng số hữu ích) [11] Đặt $sumtwu = \sum_{t_k \in dQDB} twu(t_k)$ là tổng của twu trong CSDL. Độ hỗ trợ trọng số hữu ích (wus) của mỗi mẫu P được định nghĩa như sau:

$$wus(P) = \frac{\sum_{t_k \in t(P)} twu(t_k)}{sumtwu} \quad (2)$$

Trong đó, $t(P)$ là tập các giao dịch chứa P . $t(P)$ còn được gọi là tidset của mẫu P .

Bảng 2. Giá trị twu và wus của $dQDB_E$.

A		B	
Giao dịch	twu	Mục	wus
t_1	0.65	A	0.78
t_2	0.87	B	0.53
t_3	0.50	C	1.0
t_4	0.67	D	0.47
t_5	0.52	E	0.83
t_6	0.68		
Sum of twu values ($sumtwu$)	3.89		

Bảng 2 chỉ ra giá trị twu của các giao dịch và wus của các mục trong $dQDB_E$. Ví dụ, giao dịch t_1 có $twu(t_1) = (0.3 \times 1 + 0.2 \times 2 + 0.4 \times 3 + 0.7 \times 1)/4 = 2.6/4 = 0.65$ (theo công thức 1). Mục B xuất hiện trong ba giao dịch t_2, t_5, t_6 , nên $wus(B) = [twu(t_2) + twu(t_5) + twu(t_6)]/sumtwu = (0.87 + 0.52 + 0.68)/3.89 \approx 0.53$ (Theo công thức 2).

Định nghĩa 4. (Mẫu phổ biến trọng số hữu ích) [11] Cho trước một giá trị $minwus$ và một mẫu $P \subseteq I$, nếu $wus(P) \geq minwus$, thì P được gọi là mẫu phổ biến trọng số hữu ích theo ngưỡng $minwus$.

Định nghĩa 5. (Bao đóng) Bao đóng của mẫu $P \subseteq I$, định nghĩa là $c(P)$, Là một mẫu $Q \subseteq I$ lớn nhất sao cho $P \subseteq Q$ and $wus(P) = wus(Q)$. $c(P)$ cũng bằng giao của các giao dịch chứa P : $c(P) = \bigcap_{t_k \in t(P)} t_k$.

Ví dụ 2. Trong $dQDB_E$, ta có BCE là mẫu lớn nhất chứa mục B , nên $wus(BCE) = wus(B)$, do đó $c(B) = BCE$. Nói cách khác, $c(B)$ có thể được tính $= t_2 \cap t_5 \cap t_6 = BCE$.

Định nghĩa 6. (Mẫu đóng) Cho một mẫu $P \subseteq I$, P được gọi là mẫu đóng nếu và chỉ nếu $c(P) = P$.

Ví dụ 3. Trong $dQDB_E$, mẫu B không là mẫu đóng do $c(B) = BCE \neq B$, Còn BCE là một mẫu đóng do $c(BCE) = BCE$.

Định nghĩa 7. (Mẫu phổ biến trọng số hữu ích đóng) Cho một mẫu $P \subseteq I$ là một mẫu phổ biến trọng số hữu ích (FWUP), P được gọi là mẫu phổ biến trọng số hữu ích đóng (FWUCP) khi và chỉ khi không tồn tại một mẫu Q sao cho $P \subseteq Q$ và $wus(P) = wus(Q)$.

Ví dụ 4. Trong $dQBD_E$ với $minwus = 0.5$, BCE là một FWUP do $wus(BCE) = 0.53 > minwus$. Trong ví dụ 3 ta đã chỉ ra $(BCE) = BCE$ nên BCE là một FWUCP.

Với ngưỡng hỗ trợ trọng số hữu ích tối thiểu do người dùng đưa ra ($minwus$). Bài toán khai thác các mẫu phổ biến trọng số hữu ích đóng từ cơ sở dữ liệu định lượng động là tìm tất cả các FWUCP có giá trị hỗ trợ tiện ích có trọng số (wus) không nhỏ hơn $minwus$.

3.2. Thuật toán nFWUCP (N-List Frequent Weighted Utility Close Pattern)

a. Cấu trúc WUNList

Nguyen và cộng sự [15] đã đề xuất cấu trúc WUNList cho khai thác FWUPs từ CSDL định lượng động. Qua đó tập thể tác giả cũng đã đề xuất WUN-tree, là một cấu trúc cây bao gồm một nút gốc và một tập hợp các nút con. Mỗi nút duy trì năm thuộc tính: $\{name, weighted, pre, post, \text{ và } child\}$ tương ứng với tên của mục được biểu diễn, tổng số twu của các giao dịch đi qua nút, trình tự duyệt trước và sau của nút, và các nút con của nút đó.

Thuật toán xây dựng cây WUN-tree được Nguyen và cộng sự [15] đề xuất, gồm ba bước chính. *Bước 1*: quét toàn bộ CSDL định lượng động $dQDB$ một lần duy nhất để tính giá trị wus của từng mục, sau đó chọn các mục có $wus \geq minwus$ đưa vào tập I_1 và sắp xếp giảm dần theo wus .

Bước 2: với mỗi giao dịch $t \in dQDB$, loại bỏ các mục không thuộc I_1 , sắp xếp các mục còn lại theo thứ tự của I_1 , rồi chèn tuần tự vào cây bắt đầu từ nút gốc. Khi chèn một giao dịch, nếu nút con tương ứng đã tồn tại thì cộng dồn giá trị twu của giao dịch vào trọng số ($weight$) của nút đó; ngược lại, tạo nút con mới với $weight$ được khởi tạo bằng twu của giao dịch.

Bước 3: duyệt cây theo thứ tự *pre-order* và *post-order* để gán giá trị *pre* và *post* cho mỗi nút, phục vụ việc xác định quan hệ tổ tiên – con cháu (Định lý 1).

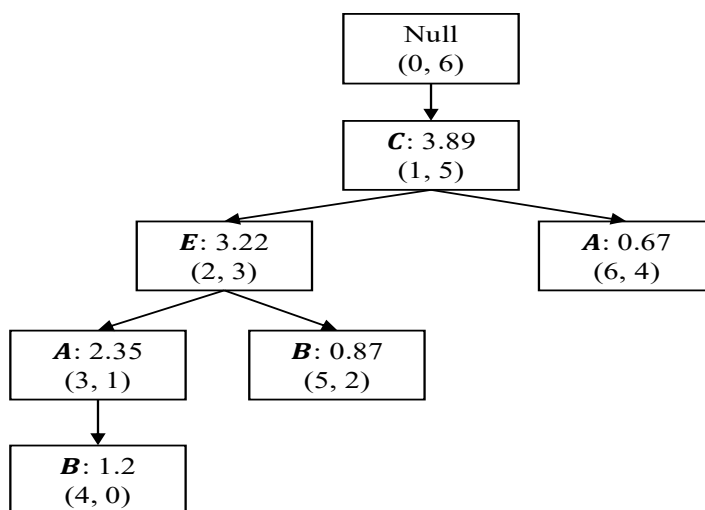
Chi tiết thuật toán có thể tham khảo trong [13]

Bảng 3. $dQDB_E$ với $minwus = 0.5$.

A						
CSDL sau khi sắp xếp lại các mục						
Giao dịch	C	E	A	B	D	twu
t_1	2	1	1	0	3	0.65
t_2	1	2	0	2	0	0.87
t_3	1	1	2	0	0	0.50
t_4	3	0	1	0	2	0.67
t_5	1	1	3	1	2	0.52
t_6	2	3	1	2	0	0.68
Tổng của các twu ($sumtwu$)						3.89

B	
I_1	
Mục	wus
C	1
E	0.83
A	0.78
B	0.53
D	0.47

WUN-tree được xây dựng từ $dQDB_E$ với $minwus = 0.5$ như sau: đầu tiên, quét CSDL để tính giá trị wus của mỗi mục và lưu FWUPs (1-FWUPs) vào tập I_1 (Bảng 3B). Tiếp theo, mỗi mục trong các giao dịch được sắp xếp giảm dần order dựa trên giá trị wus của chúng, các mục không thuộc I_1 sẽ bị bỏ ra khỏi bảng (Bảng 3A). Mục D bị loại khỏi bảng do không thuộc I_1 . Các giao dịch được sắp xếp sau đó được thêm tuần tự vào cây. Sau đó, duyệt WUN-tree để thêm các giá trị pre và $post$ cho mỗi nút. Cuối cùng, cây WUN kết quả được hiển thị trong Hình 1.



Hình 1. WUN-tree với $minwus = 0.5$.

Định nghĩa 8. (WUN-code) [13] Cây WUN-tree, mỗi WUN-code của một node N là một bộ ba $(N.pre, N.post, N.weight)$.

Định lý 1 [15] Cho hai node N_1 và N_2 trên cây WUN-tree. N_1 được gọi là tổ tiên của N_2 nếu và chỉ nếu $N_1.pre < N_2.pre$ và $N_1.post > N_2.post$.

Ví dụ 5. Từ cây WUN-tree Trong Hình 2, hai node C và E có WUN-codes lần lượt là $(1, 5, 3.89)$ và $(2, 3, 3.22)$. Node C là tổ tiên của node E vì $C.pre (= 1) < E.pre (= 2)$ và $C.post (= 5) > E.post (= 3)$.

Định nghĩa 9. (WUNList of 1-pattern) [13]: WUNList của 1-pattern X , ký hiệu là $wunl(X)$, là một dãy các mã WUN được sắp xếp theo thứ tự tăng dần dựa trên giá trị pre của nút đại diện cho mục X trên cây WUN.

Trong quá trình khai thác mẫu phổ biến, ta cần tính wus của các mẫu có kích thước tăng dần: 1-pattern $\rightarrow$ 2-pattern $\rightarrow$ 3-pattern $\rightarrow \dots$ Nhưng việc tính wus cho mỗi mẫu mới bằng cách quét lại CSDL là rất tốn kém. WUNList là một cấu trúc dữ liệu nén lưu thông tin của một mẫu trên cây WUN-tree. Mỗi WUN-code $(pre, post, weight)$ cho biết: tại nút có trình tự duyệt trước = pre và trình tự duyệt sau = $post$, mẫu đó xuất hiện với trọng số $weight$. Từ WUNList có thể suy ra wus của mẫu mà không cần quét lại CSDL.

Thuật toán tính giao các WUNlist được trình bày cụ thể trong [13].

Định nghĩa 10. (WUNList của k -pattern) [13] Cho Px và Py là hai $(k-1)$ -patterns cùng tiền tố P , và y đứng trước x trong tập I_1 (ký hiệu là $y < xy < x$). $wunl(Px)$ và $wunl(Py)$ lần lượt là các WUNList của Px và Py . WUNList của k -pattern Pxy , ký hiệu $wunl(Pxy)$, được tính như sau:

1. Với mỗi WUN-code $Cu \in wunl(Px)$ và $Cv \in wunl(Py)$, nếu Cv là tổ tiên của Cu , thì thêm WUN-code $(Cv.pre, Cv.post, Cu.weight)$ vào $wunl(Pxy)$.
2. Gộp tất cả các WUN-code trong $wunl(Pxy)$ có cùng giá trị pre lại thành một WUN-code duy nhất, với giá trị $weight$ mới bằng tổng các trọng số ($weight$) của các WUN-code đã được gộp.

Định lý 2 [13]. Giả sử có mẫu P với WUNList là $wunl(P) = \{(pre_1, post_1, weight_1), (pre_2, post_2, weight_2), \dots, (pre_m, post_m, weight_m)\}$. Độ hỗ trợ trọng số hữu ích của P - $wus(P)$, được tính:

$$wus(P) = \frac{\sum_{i=1}^m weight_i}{sumtwu}. \quad (3)$$

Ví dụ 6. WUNList của mẫu B là $wunl(B) = \{(4, 0, 1.2), (5, 2, 0.87)\}$ và $sumtwu = 3.89$. Theo định lý 2, ta có $wus(B) = (1.2 + 0.87)/3.89 \approx 0.53$.

Dựa trên Định lý 2, chúng ta có thể tính toán wus của một mẫu bất kỳ từ WUNList của nó. Một mẫu phổ biến trọng số hữu ích (FWUP) sẽ được xác định là mẫu đóng (FWUCP) nếu nó thỏa mãn Định nghĩa 6 và 7 ở trên.

b. Thuật toán nFWUCP

Dựa trên thuật toán xây dựng cây WUN- tree và thuật toán tính giao của 2 WUNList, chúng tôi đề xuất thuật toán khai thác mẫu đóng trên CSDL định lượng động theo định nghĩa 7 trong Hình 2.

Algorithm 6. nFWUCP
Input: A dynamic quantitative data stream $dQDB$ and a user-given threshold $minwus$; Output: A set of frequent weighted utility closed patterns $FWUCPs$
<ol style="list-style-type: none"> 1. $R \leftarrow \text{Construct-WUN-tree}(dQDB, minwus)$ 2. Traverse the WUN-tree R to create WUNList for items in I_1 3. $FWUCPs \leftarrow \emptyset$ 4. $\text{Mine-FWUCPs}(\emptyset, \emptyset, I_1)$ 5. Return $FWUCPs$
Procedure Mine-FWUCPs ($Prefix, S, C$) <ol style="list-style-type: none"> 1. $Prefix_{next} \leftarrow Prefix$

```

2. for  $i \leftarrow |C| - 1$  downto 0 do
3.   for  $j \leftarrow i - 1$  downto 0 do
4.     if  $PTP[C_i][C_j]/sumtwu < minwus$  then continue
5.     if  $C_j \in C_i.Enc$  then
6.        $S_{next} \leftarrow S_{next} \cup C_j$ 
7.       if  $C_i.wus = C_j.wus$  then
8.         Remove  $C_j$ 
9.          $i--$ 
10.      continue
11.     $Cand \leftarrow C_j$ 
12.     $Cand.wunl \leftarrow \mathbf{WUNList-Intersection}(C_i.wunl, C_j.wunl)$ 
13.    if  $Cand.wunl \neq NULL$  then
14.       $Cand.Enc \leftarrow C_i.Enc \cup C_j.Enc$ 
15.       $C_{next} \leftarrow Cand \cup C_{next}$ 
16.     $P \leftarrow Prefix_{next} \cup S_{next}, P.wus \leftarrow C_i.wus$ 
17.    if  $P$  is closed pattern in  $FWUCPs$  then
18.       $FWUCPs \leftarrow FWUCPs \cup P$ 
19.     $\mathbf{Mine-FWUCPs}(Prefix_{next}, S_{next}, C_{next})$ 
20.    Remove last item of  $Prefix_{next}$ 

```

Hình 2. Thuật toán khai thác mẫu phổ biến đóng trên CSDL định lượng động

Ngoài việc cắt tỉa cận trên trong quá trình tính giao các WUNList (dòng 12) được trình bày chi tiết trong [13]. Thuật toán nFWUCP đề xuất 2 chiến lược cắt tỉa không gian tìm kiếm giúp giảm đáng kể thời gian khai thác mẫu.

(i) Từ dòng 5-11: Cắt tỉa bao đóng dựa trên tập Enc . Mỗi mẫu C duy trì một tập $C.Enc$ chứa các mục bị bao đóng bởi C . Khi xét hai mẫu C_i và C_j với $C_j \in C_i.Enc$, nếu $wus(C_i) = wus(C_j)$ thì C_j bị bao trùm bởi $C_i \cup C_j$, do đó C_j không thể là FWUCP và bị loại khỏi danh sách ứng viên ngay (không cần mở rộng tiếp). Khi sinh mẫu mới $Cand$ từ C_i và C_j , gán $Cand.Enc \leftarrow C_i.Enc \cup C_j.Enc$ để tích lũy thông tin bao đóng cho các bước đệ quy sau. Chiến lược này giảm mạnh số phép giao WUNList cần thực hiện.

(ii) Từ dòng 13-18: Cắt tia dựa vào kiểm tra siêu tập trùng *wus*. Trước khi thêm mẫu *P* vào tập kết quả FWUCPs, thuật toán kiểm tra: nếu tồn tại $Q \in \text{FWUCPs}$ sao cho $P \subset Q$ và $wus(P) = wus(Q)$, thì *P* không phải FWUCP và bị loại. Việc kiểm tra được tăng tốc bằng (1) bảng băm theo *wus* để chỉ so sánh các mẫu cùng *wus*, và (2) tập *P.Enc* đã tích lũy từ Chiến lược (i) — chỉ cần kiểm tra các mẫu $Q \subseteq P \cup P.Enc$ thay vì duyệt toàn bộ FWUCPs.

4. Experiments

Bảng 4. Mô tả dữ liệu thực nghiệm.

CSDL	Số mục	Số giao dịch	Số giao dịch một lô
Pumsb	2,113	49,046	3,000
Accidents	468	340,183	5,000
Retail	16,470	88,162	10,000
BMS-POS	1,657	515,597	50,000
Kosarak	41,270	990,002	100,000
Chainstore	46,086	1,112,949	80,000

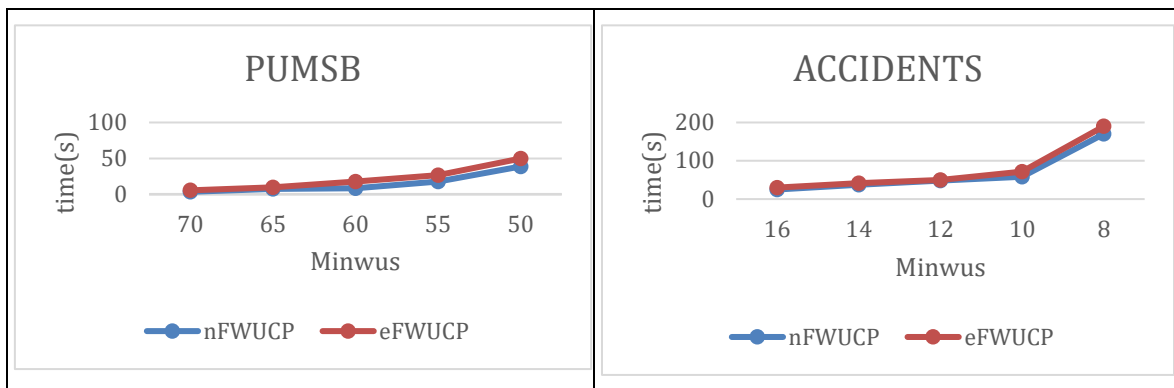
Trong phần này, chúng tôi so sánh thuật toán đề xuất nFWUCP ở trên với một phiên bản điều chỉnh của thuật toán FAST_MINE_FWUI [17], mà chúng tôi tạm gọi là eFWUCP. eFWUCP được sửa đổi để khai thác mẫu đóng và sử dụng cấu trúc EIWS của thuật toán gốc. eFWUCP – là thuật toán khai thác mẫu phổ biến hiệu quả nhất theo chiều dọc với cấu trúc EIWS [17] được đề xuất năm 2022. eFWUCP được điều chỉnh để phù hợp cho việc khai thác mẫu đóng trên CSDL động mà không làm phát sinh thêm độ phức tạp tính toán trong quá trình khai thác.

Các thuật toán được lập trình bằng ngôn ngữ C# 2022 trên nền .NET Framework phiên bản 4.7.2. Việc thử nghiệm được thực hiện trên máy tính sử dụng bộ xử lý Intel Core i5-9400F 2.9GHz, RAM 32GB, và hệ điều hành Windows 11.

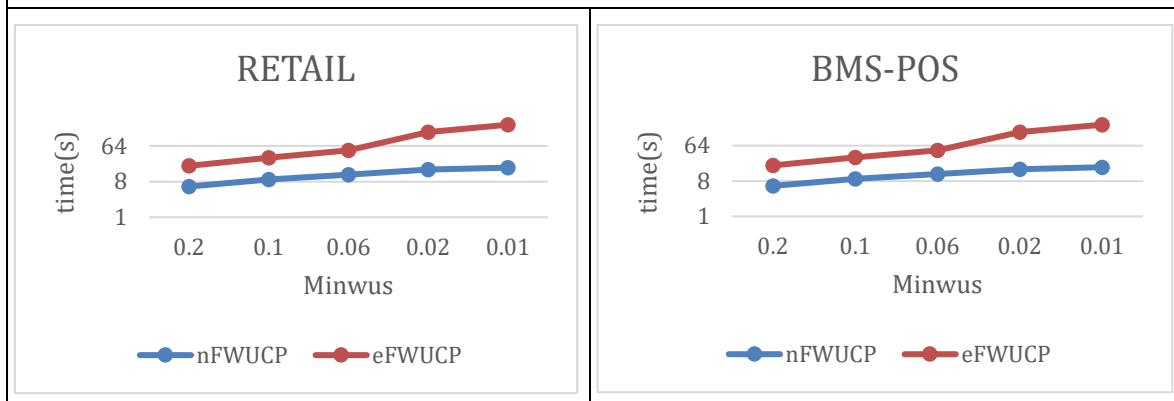
Các tập dữ liệu thực nghiệm được mô tả trong Bảng 4. Tập dữ liệu Chainstore được lấy từ thư viện mã nguồn mở SPMF (<https://www.philippe-fournier-viger.com/spmf/>), còn các tập dữ liệu khác được lấy từ kho dữ liệu FIMI (<http://fimi.uantwerpen.be/data/>).

Để mô phỏng cơ sở dữ liệu định lượng động, số lượng của mỗi mặt hàng xuất hiện trong giao dịch được sinh ngẫu nhiên trong khoảng [1, 10]. Mỗi tập dữ liệu được chia thành nhiều lô (batch), mỗi lô chứa số lượng giao dịch bằng nhau, trừ lô cuối cùng có thể chứa số giao dịch còn lại. Trọng số của từng mặt hàng trong mỗi batch cũng được gán ngẫu nhiên trong khoảng [1, 10].

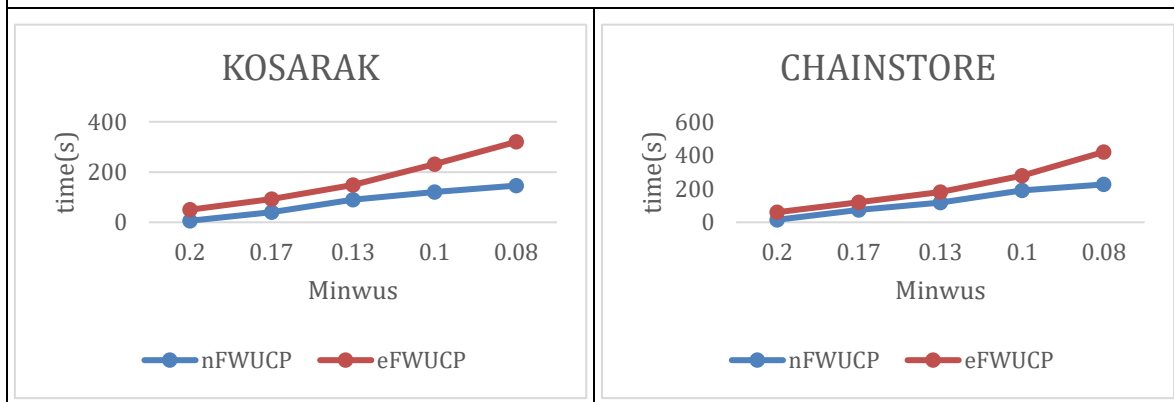
Vì cả hai thuật toán eFWUCP và nFWUCP đều khai thác đầy đủ và chính xác các FWUCP, nên chúng tôi so sánh hiệu năng của chúng dựa trên hai tiêu chí thời gian chạy



Hình 3. So sánh thời gian chạy trên hai CSDL PUMSB và ACCIDENTS



Hình 4. So sánh thời gian chạy trên hai CSDL RETAIL và BMS-POS (thang đo Log2)



Hình 5. So sánh thời gian chạy trên hai CSDL KOSARAK và CHAINSTORE

Kết quả thực nghiệm từ Hình 3 đến Hình 5 cho thấy thời gian khai thác của hai thuật toán tăng khi ngưỡng minwus giảm. Điều này là hợp lý do khi ngưỡng giảm thì không gian tìm kiếm lớn hơn và số lượng FWUCP được sinh ra nhiều hơn do đó đòi hỏi nhiều thời gian tính toán hơn.

Thuật toán nFWUCP thể hiện sự vượt trội ở tất cả các CSDL thực nghiệm, trên cả các dữ liệu dày và thưa. Khi ngưỡng *minwus* giảm xuống khoảng cách chênh lệch của hai thuật

toán càng lớn, chứng tỏ sự hiệu quả vượt trội của nFWUCP với không gian tìm kiếm lớn, khả năng mở rộng trên các CSDL lớn và phức tạp hơn sẽ có hiệu quả tốt hơn. Điều này là do sự hiệu quả của cấu trúc WunList – một mở rộng của cấu trúc N-List nhờ khả năng nén thông tin và tính toán giao tập hợp nhanh chóng. Các chiến lược cắt tia được tích hợp trong thuật toán giao WUNList [13] giúp loại bỏ sớm các nhánh không tiềm năng, từ đó giảm đáng kể không gian tìm kiếm và tiết kiệm thời gian tính toán so với cách tiếp cận của FWUCP,

5. Kết luận

Trong bài báo này, chúng tôi đã giới thiệu bài toán khai thác các mẫu đóng trên CSDL định lượng động. Các FWUCP đóng vai trò như một dạng nén gọn của FWUP, giúp giảm số lượng mẫu được sinh ra nhưng vẫn bảo toàn toàn bộ thông tin quan trọng, từ đó tiết kiệm tài nguyên trong cả quá trình khai thác và lưu trữ. Để giải quyết bài toán này, chúng tôi đã đề xuất một thuật toán mang tên nFWUCP nhằm khai thác FWUCP một cách hiệu quả, sử dụng cấu trúc dữ liệu WUNList.

Sự hiệu quả của nFWUCP thể hiện rõ trên cả các CSDL thưa như *Retail* và *Kosarak*, lẫn các CSDL dày như *Pumsb*. Điều này đặc biệt có ý nghĩa vì các CSDL dày thường là thách thức lớn đối với các thuật toán khai thác mẫu do sự bùng nổ tổ hợp của các mẫu con. Kết quả này cho thấy cấu trúc WUNList và các chiến lược cắt tia của nFWUCP hoạt động hiệu quả trong nhiều kịch bản dữ liệu khác nhau.

Hướng phát triển trong tương lai, chúng tôi dự định sẽ khám phá thêm các kỹ thuật mới để nâng cao hiệu suất khai thác FWUCP. Ngoài ra, chúng tôi sẽ nghiên cứu để triển khai các phương pháp song song và phân tán trong thuật toán nFWUCP nhằm tăng tốc khai thác trên các cơ sở dữ liệu quy mô lớn.

“Lời cảm ơn: Bài báo này là sản phẩm của đề tài nghiên cứu mã số CSA2024-01 được tài trợ bởi Trường Đại học Sài Gòn”.

TÀI LIỆU THAM KHẢO

- R. Agrawal, T. Imielinski and A. Swami, "Mining association rules between sets of items in large databases," in *SIGMOD '93 Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, 1993.
- J. Han, J. Pei and Y. Yin, "Mining frequent patterns without candidate generation," in *The 2000 ACM SIGMOD International Conference on Management of Data*, 2000.
- M. J. Zaki, "Scalable algorithms for association mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 3, pp. 372-390, 2000.
- Z. Deng, Z. Wang and J. Jiang, "A new algorithm for fast mining frequent itemsets using N-lists," *Science China Information Sciences*, vol. 55, no. 9, pp. 2008-2030, Jul. 2012.
- B. Vo, T. Le, F. Coenen and T. Hong, "Mining frequent itemsets using the N-list and subsume concepts," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 2, pp. 253-265, Apr. 2016.

- N. Pasquier, Y. Bastide, R. Taouil and L. Lakhal, "Discovering frequent closed itemsets for association rules," in *Database Theory - ICDT'99*, Jerusalem, Israel, 1999.
- M. J. Zaki and C.-J. Hsiao, "CHARM: An efficient algorithm for closed association rule mining," Technical Report 99, 1999.
- C. Lucchese, S. Orlando and R. Perego, "Fast and memory efficient mining of frequent closed itemsets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 21-36, 2006.
- T. Le and B. Vo, "An N-list-based algorithm for mining frequent closed patterns," *Expert Systems with Applications*, vol. 42, no. 19, pp. 6648-6657, Nov. 2015.
- N. Aryabarzan and B. Minaei-Bidgoli, "Neclatclosed: A vertical algorithm for mining frequent closed itemsets," *Expert Systems with Applications*, p. 174: 114738, 2021.
- M. S. Khan, M. Muyeba and F. Coenen, "A weighted utility framework for mining association rules," in *Computer Modeling and Simulation, 2008.*, 2008.
- H. Nguyen, B. Vo, M. T. H. Nguyen and T. Hong, "MBiS: an efficient method for mining frequent weighted utility itemsets from quantitative databases," *Journal of Computer Science and Cybernetics*, vol. 31, 2015.
- H. Bui, B. Vo and H. Nguyen, "WUN-Miner: A new method for mining frequent weighted utility itemsets," in *The 2016 IEEE Conference on System, Man, and Cybernetics (SMC 2016)*, Budapest, 2016.
- H. Nguyen, N. Le, H. Bui and T. Le, "A new approach for efficiently mining frequent weighted utility patterns," *Applied Intelligence*, pp. 53(1), 121-140, 2023.
- H. Nguyen, N. Le, H. Bui and T. Le, "Mining frequent weighted utility patterns with dynamic weighted items from quantitative databases," *Applied Intelligence*, pp. 53(16), 19629-19646, 2023.
- B. Vo, B. Le and J. J. Jung, "A tree-based approach for mining frequent weighted utility itemsets," in *Computational Collective Intelligence. Technologies and Applications*. Springer, 2012, pp. 114-123.
- H. Nguyen, T. Le, M. Nguyen, P. Fournier-Viger, V. S. Tseng and B. Vo, "Mining frequent weighted utility itemsets in hierarchical quantitative databases," *Knowledge-Based Systems*, vol. 237, p. 107709, 2022.
- J. Pei, J. Han and R. Mao, "CLOSET: An efficient algorithm for mining frequent closet itemsets," in *2000 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, Dallas, Texas, USA, May. 2000.
- J. Wang, J. Han and J. Pei, "CLOSET+ : Searching for best strategies for mining frequent closed itemsets," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, August. 2003.