

Sử dụng Bitmap trong khai thác tập có thể xóa trên dữ liệu tăng cường

NGUYỄN THỊ HOÀI LINH *

Trường Đại học Kinh tế - Tài chính TP.HCM

Nhận bài: 04/08/2022 - Duyệt đăng: 15/10/2022

(*) Liên hệ: linhnth@uef.edu.vn; ĐT: 0945110590

Tóm tắt:

Khai thác tập có thể xóa là một trong những lĩnh vực thú vị trong khai thác tập phổ biến. Phân tích dữ liệu tăng cường trở nên quan trọng hơn vì dữ liệu thú vị liên tục được tích lũy trong các lĩnh vực ứng dụng khác nhau bao gồm cả các lĩnh vực công nghiệp. Bài viết này cung cấp một phương pháp tiếp cận sử dụng Bitmap để khai thác các tập có thể xóa trên dữ liệu tăng cường. Không giống như ý nghĩa Bitmap truyền thống cho các tập phổ biến, một Bitmap cho tập các tập có thể xóa được khai thác ở đây biểu thị mối quan hệ mà một sản phẩm bao gồm ít nhất một nguyên liệu (item) trong một vật phẩm. Sử dụng biểu diễn Bitmap có thể dễ dàng tìm thấy các sản phẩm mong muốn để kiểm tra, do đó giảm số lần quét cơ sở dữ liệu.

Từ khóa: Khai thác dữ liệu, Tập có thể xóa, Khai thác tăng dần, Biểu diễn Bitmap

Abstract:

Erasable itemset mining is one of the interesting areas in frequent set mining. Incremental data analytics becomes more important as interesting data is continuously accumulated in various application areas including industrial sectors. This article provides an approach using Bitmap to mine erasable itemset on incremental data. Unlike the traditional Bitmap meaning for frequent itemsets, a Bitmap for erasable itemset mining here denotes the relationship that a product includes at least one material (item) in a specified itemset. Using the Bitmap representation can easily find the desired products to check, thus reducing the number of database scans.

Keywords: Data mining, Erasable itemsets, Incremental mining, Bitmap representation.

1. Giới thiệu

Bài toán khai thác tập có thể xóa (EI – erasable itemset) được đề xuất bởi Deng và các đồng tác giả vào năm 2009 (Z. Deng, G. Fang, và Z. Wang, 2009) với thuật toán META (mining erasable itemsets with the anti-monotone property), một thuật toán dựa vào thuật toán Apriori. Sau đó, nhiều thuật toán được phát triển để cải thiện hiệu suất

khai thác (thời gian, bộ nhớ) như VME (vertical-format-based algorithm for mining erasable itemsets) được đề nghị năm 2010 (Z. Deng và X. Xu, 2010), MERIT (fast mining erasable itemsets) được đề nghị năm 2012 (Z. Deng và X. Xu, 2012), và MEI (mining erasable itemsets) được đề nghị năm 2014 (T. Le và B. Vo, 2014) và BREM (TP. Hong, GC. Lan, MC. Chiang và

J. CW. Lin, 2021).

Năm 2016, Lee et al. đã đề xuất thuật toán IWEI khai thác EI cho cơ sở dữ liệu (CSDL) tăng cường (G. Lee, U. Yun, H. Ryang và D. Kim, 2016). Sau đó, một số thuật toán khác cũng đã được đề xuất như FUP-ERASABLE (TP Hong, KY Lin, CW Lin và B. Vo, 2017), LINE (G. Lee và U. Yun, 2018) và WEIL (H. Nam, U. Yun, E. Yoon, JCW Lin, 2020).

Hầu hết các thuật toán này đều sử dụng chiến thuật chia để trị, nghĩa là các tập mục với một phần tử được xử lý trước, sau đó các tập mục chứa hai phần tử được sử dụng, quy trình này lặp lại, với mỗi lần lặp số phần tử tăng lên một cho đến khi đáp ứng tiêu chí nào đó. Quá trình này tốn nhiều thời gian để khai thác. Một số cấu trúc cây sau đó đã được đề xuất để giảm thời gian thực hiện. Tuy nhiên cấu trúc cây này phức tạp hơn và cần nhiều bộ nhớ để lưu trữ toàn bộ cây để khai thác. Trong bài báo này, chúng tôi phát triển một cách tiếp cận khác dựa trên Bitmap[6] để giảm thời gian xử lý nhưng không tiêu tốn nhiều không gian bộ nhớ để khai thác tập phổ biến có thể xóa trên cơ sở dữ liệu tăng trưởng.

Phần còn lại của bài báo như sau: phần 2. Các nghiên cứu liên quan và một số định nghĩa được nêu trong Phần 3. Thuật toán đề xuất được mô tả trong Phần 4. Cuối cùng, các kết luận và hướng phát triển được đưa ra trong Phần 5.

2. Các nghiên cứu liên quan

Bài toán khai thác tập có thể xóa được đề xuất bởi Deng và các đồng tác giả vào năm 2009 (Z. Deng, G. Fang, và Z. Wang, 2009), đồng thời, họ cũng đề xuất thuật toán META (mining erasable itemsets with the anti-monotone property) (Deng ZH, Fang G, Wang Z, Xu X. 2009), một thuật toán dựa vào thuật toán Apriori (R. Agrawal and R. Srikant, 1994). Sau đó, nhiều thuật toán được phát triển để cải thiện hiệu suất khai thác (thời gian, bộ nhớ) như VME (vertical-format-based algorithm for mining erasable itemsets)

được đề nghị năm 2010 (Z. Deng và X. Xu, 2010), MERIT (fast mining erasable itemsets) được đề nghị năm 2012 (Z. Deng và X. Xu, 2012), MEI (mining erasable itemsets) được đề nghị năm 2014 (T. Le và B. Vo, 2014), và BREM (TP. Hong, GC. Lan, MC. Chiang và J. CW. Lin, 2021) được đề nghị năm 2021. BREM sử dụng phương pháp Bitmap để khai thác tập có thể xóa trên cơ sở dữ liệu tĩnh thay vì dùng tDifset (T. Le, B. Vo, and F. Coenen, 2013). **Tập có thể xóa trên CSDL tăng trưởng**

Xử lý luồng dữ liệu động là một vấn đề thiết yếu trong khai thác tập có thể xóa. Vấn đề này đã thu hút sự nghiên cứu trong khai thác tập đại diện, khai thác tập tiện ích cao, khai thác tập Top-K. Ngoài ra, đã có những nỗ lực để xử lý với dữ liệu động trong lĩnh vực khai thác tập có thể xóa và hai phương pháp liên quan đã được đề xuất. WEPS (U. Yun, G. Lee, 2016) là một cách tiếp cận dựa trên cây tập trung vào dữ liệu mới nhất bằng cách áp dụng mô hình cửa sổ trượt vào khai thác mẫu có thể xóa. Nếu kích thước cửa sổ được cung cấp bởi người dùng, thuật toán thường xuyên chèn dữ liệu mới vào cửa sổ và xóa dữ liệu cũ. Cấu trúc dữ liệu để xử lý dữ liệu luồng được xây dựng với một lần quét cơ sở dữ liệu và được duy trì thông qua tái cấu trúc cây. IWEI (G. Lee, U. Yun, H. Ryang và D. Kim, 2016) là một thuật toán dựa trên cây hiện có để xử lý dữ liệu gia tăng, tuân theo dMERIT + (T. Le, B. Vo, và F. Coenen, 2013). Ngược lại với cách tiếp cận tĩnh trước đó, thuật toán xây dựng cấu trúc cây riêng, cây IWEI, quét cơ sở

dữ liệu một lần để xử lý dữ liệu gia tăng động. Mặc dù IWEI có thể thực hiện các công việc khai thác mẫu có thể xóa được từ cơ sở dữ liệu sản phẩm gia tăng, nhưng vẫn có những hạn chế về hiệu suất vì những lý do sau:

1) Chi phí xây dựng và tái cấu trúc cây phức tạp.

2) Hiệu suất sử dụng cấu trúc cây thấp so với chi phí bảo trì cao.

Trong khi đó, thuật toán được đề xuất có thể trích xuất các mẫu có thể xóa được từ CSDL tăng trưởng theo cách hiệu quả hơn thông qua kỹ thuật dữ liệu mới được đề xuất.

3. Khai thác tập có thể xóa trên CSDL tăng trưởng

3.1. Các định nghĩa liên quan

Cho một CSDL sản phẩm gốc $PDB_0 = \{P_1, P_2, P_3, \dots, P_n\}$ trong đó mỗi sản phẩm P_i chứa một tập các thành phần (itemset) và giá thành (val) để sản xuất sản phẩm tương ứng. Và 2 CSDL sản phẩm được thêm vào $PDB_1 = \{P_1', P_2', P_3', \dots, P_n'\}$ và $PDB_2 = \{P_1'', P_2'', P_3'', \dots, P_n''\}$ Ví dụ về CSDL sản phẩm được minh họa trong Bảng 1.

Định nghĩa 1. Một Bitmap của 1 itemset X là một chuỗi bit nhị phân một chiều, ký hiệu là $Bitmap_X = \{b_1, b_2, \dots, b_z\}$ thể hiện mối quan hệ của một item trong P_i . Giá trị bit thứ i là 1 nếu item đó tồn tại trong sản phẩm thứ i và bằng 0 nếu không.

Ví dụ item a được chứa trong P_1, P_2 và P_6 , và item b được chứa trong P_4 , độ dài Bitmap của mỗi item là 6 vì có 6 sản phẩm trong PDB_0 . Biểu diễn Bitmap của a là (1, 1, 0, 0, 0, 1) và b là (0, 0, 0, 1, 0, 0).

Bảng 1. Ví dụ về một CSDL sản phẩm PDB₀ và 2 CSDL tăng cường

PID	Items	Val (\$)
P1	a d e	1100
P2	a c d e	300
P3	c d g	500
P4	b d	400
P5	e g h	600
P6	a g	200
P PDB₀		
P7	a b g	700
P8	b d e	500
P9	c f	300
P PDB₁		
P10	b d g h	600
P11	g h i	500
P12	e f	400
P PDB₂		

Định nghĩa 2. Cho X là một itemset với $|X| > 2$. Bitmap của X được định nghĩa là:

$$\text{Bitmap}_X = \{b_1, b_2, \dots, b_z\}$$

$$= (OR_{j=1}^{|X|} b_1^{x_j}, OR_{j=1}^{|X|} b_2^{x_j}, \dots, OR_{j=1}^{|X|} b_i^{x_j}, \dots, OR_{j=1}^{|X|} b_z^{x_j}),$$

Với $|X|$ là số item trong X . $b_i^{x_j}$ là giá trị bit thứ i của item thứ j trong X , và $OR_{j=1}^{|X|} b_i^{x_j}$ thực hiện OR trên $b_i^{x_j}$ giá trị của tất cả các item trong X .

Ví dụ 1. Giả sử $\{a\}$ và $\{b\}$ là hai tập 1-itemset có thể xóa được. Theo quy trình xử lý theo cấp độ, tập hợp 2-itemset có thể xóa ứng viên được tìm thấy từ chúng. Trong ví dụ này, một tập hợp 2-itemset ứng viên là $\{a, b\}$. Cùng với biểu diễn Bitmap, có thể dễ dàng thu được các sản phẩm bao gồm 2 item

ứng viên bằng cách sử dụng phép toán OR logic trên Bitmap của các thành phần. Ví dụ: Bitmap của tập 2-itemset ứng viên $\{a, b\}$ có thể được tìm thấy bằng cách thực hiện phép toán OR logic trên a (1, 1, 0, 0, 0, 1) và b (0, 0, 0, 1, 0, 0). Kết quả là (1, 1, 0, 1, 0, 1).

Định nghĩa 3. Gain của một Bitmap biểu diễn BitmapX là tổng lợi nhuận của các sản phẩm có giá trị bit tương ứng là 1 trong BitmapX.

$$\text{gain}(X) = \sum_{i=1}^z (b_i^X * Val_i)$$

Nghĩa là, gain của X được tính bằng tổng các val của các sản phẩm có chứa ít nhất một thành phần (item) có trong X (vị trí bit bằng 1).

Ví dụ 2. Xét CSDL PDB₀ như Bảng 1, Bitmap của $\{ab\}$ là (1, 1, 0, 1, 0, 1) nên $\text{gain}(\{a,b\}) = 1100 + 300 + 400 + 200 = 2000$ (\$).

Định nghĩa 4. Cho một ngưỡng ξ và một CSDL sản phẩm P . Khi đó, một tập X được gọi là tập có thể xóa nếu

$$\text{gain}(X) \leq \xi \times \sum_{P_k \in P} P_k.\text{val} = \text{MVT}$$

Ví dụ 3. Từ CSDL P trong Bảng 1, giả sử $\xi = 45\%$, từ ví dụ 1, ta có $\text{gain}(\{a,b\}) = 2000 >> 3100 \times 45\% = 1395$ nên ab không là tập có thể xóa.

Định nghĩa 5. Bitmap-table. BMT-table là một cấu trúc bảng bao gồm các chỉ mục (PID) và lợi nhuận (Val) của các giao dịch P_k thuộc cơ sở dữ liệu.

Bảng 2. Kiến trúc chung của Bitmap-table

PID	P_1	P_2	...	P_k
Val	$P_1.\text{Val}$	$P_2.\text{Val}$	...	$P_k.\text{Val}$

Định nghĩa 6. BitmapEI-list. BitmapEI-list là một cấu trúc dữ liệu để lưu trữ thông tin được tinh chỉnh từ một cơ sở dữ liệu sản phẩm nhất định.

Bảng 3. Kiến trúc chung của BitmapEI-list

Item	i_k
Bitmap	$b_1, b_2, \dots, b_z$
Gain	$\text{gain}(i_k)$

Các bước xử lý của thuật toán sẽ như sau:

1. Tích lũy dữ liệu (Dữ liệu gốc và dữ liệu tăng cường).

2. Xây dựng và cập nhật Bitmap-table và BitmapEI-list.

Khi có dữ liệu được thêm vào, thuật toán sẽ lặp lại các bước trên.

Sau đó thuật toán chờ yêu cầu khai thác EI từ người dùng và tiến hành khai thác EI dựa trên Bitmap-table và BitmapEI-list đã xây dựng và cập nhật được ở trên và cung cấp các tập có thể xóa cho người dùng.

3.2. Xây dựng cấu trúc dữ liệu ban đầu

Trước tiên, thuật toán khởi tạo 1 Bitmap-table rỗng và tiến hành quét CSDL để xây dựng Bitmap-table. Khi 1 sản phẩm được thêm vào, PID và val của nó được thêm vào Bitmap-table. Sau đó, thuật toán xây dựng BitmapEI-list của tất cả các item trong sản phẩm đầu tiên. BitmapEI-list bao gồm tên item, Bitmap và gain của sản phẩm hiện hành. Giá trị thu được của một EI là kết quả của việc

tích lũy giá trị val của các giao dịch có chứa item tương ứng. Sau khi lặp lại các bước trên, ta thu được Bitmap-table và BitmapEI-list lưu trữ toàn bộ thông tin của CSDL hiện tại.

Ví dụ 4. Hình 1 thể hiện trạng thái của Bitmap-table và BitmapEI-list được xây dựng từ CSDL sản phẩm gốc PDB₀ trong Bảng 1. Hình 1(a) thể hiện trạng thái của Bitmap-table và BitmapEI-list khi sản phẩm đầu tiên được thêm vào. Khi tất cả các sản phẩm được thêm vào thành công, ta thu được Bitmap-table và BitmapEI-list như hình 1(b)

Lúc này thuật toán đã sẵn sàng để khai thác các tập có thể xóa.

Xử lý dữ liệu gia tăng với cấu trúc dữ liệu được đề xuất

Khi dữ liệu bổ sung được thêm vào, thuật toán sẽ xử lý chúng một cách hiệu quả theo quy trình sau. Như trong trường hợp xử lý dữ liệu gốc, dữ liệu gia tăng bổ sung cũng phải được xử lý trong một lần quét cơ sở dữ liệu. Chúng tôi kiểm soát từng giao dịch của cơ sở dữ liệu gia tăng đã nhập

trong hai trường hợp sau. Đối với mỗi item của một giao dịch, trường hợp đầu tiên là khi item đó được chứa trong một trong các BitmapEI-list đã xây dựng trước đó. Trường hợp thứ hai ngược lại với trường hợp đầu tiên, tức là khi item không thuộc bất kỳ list nào. Trong trường hợp đầu tiên, thuật toán sửa đổi BitmapEI-list tương ứng với item đó bằng cách cập nhật bitmao và gain của nó thay vì tạo mới. Trường hợp thứ 2, thuật toán sẽ tiến hành tạo mới 1 BitmapEI-list tương ứng cho item và thêm các thông tin cần thiết vào danh sách.

Ví dụ 5. Giả sử, CSDL PDB₁ ở Bảng 1 được thêm vào, thuật toán tiến hành xây dựng Bitmap-table và BitmapEI-list. Hình 2 thể hiện Bitmap-table và BitmapEI-list sau khi PDB₁ được thêm vào.

3.4. Khai thác tập có thể xóa trên CSDL gia tăng

Nếu Bitmap-table và BitmapEI-list được xây dựng đối với dữ liệu sản phẩm gia tăng đã thêm, thuật toán sẽ đợi yêu cầu khai thác của người dùng hoặc dữ

Hình 1. Bitmap-table và BitmapEI-list khi quét CSDL PDB₀.

PID		P ₁	
Val		1100	

a	d	e
(1)	(1)	(1)
1100	1100	1100

(a) Sau khi thêm P₁

PID	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆
Val	1100	300	500	400	600	200

a	d	e	c	g	b	h
(1,1,0,0,0,1)	(1,1,1,1,0,0)	(1,1,0,0,1,0)	(0,1,1,0,0,0)	(0,0,1,0,1,1)	(0,0,0,1,0,0)	(0,0,0,0,1,0)
1600	2300	2000	800	1300	400	600

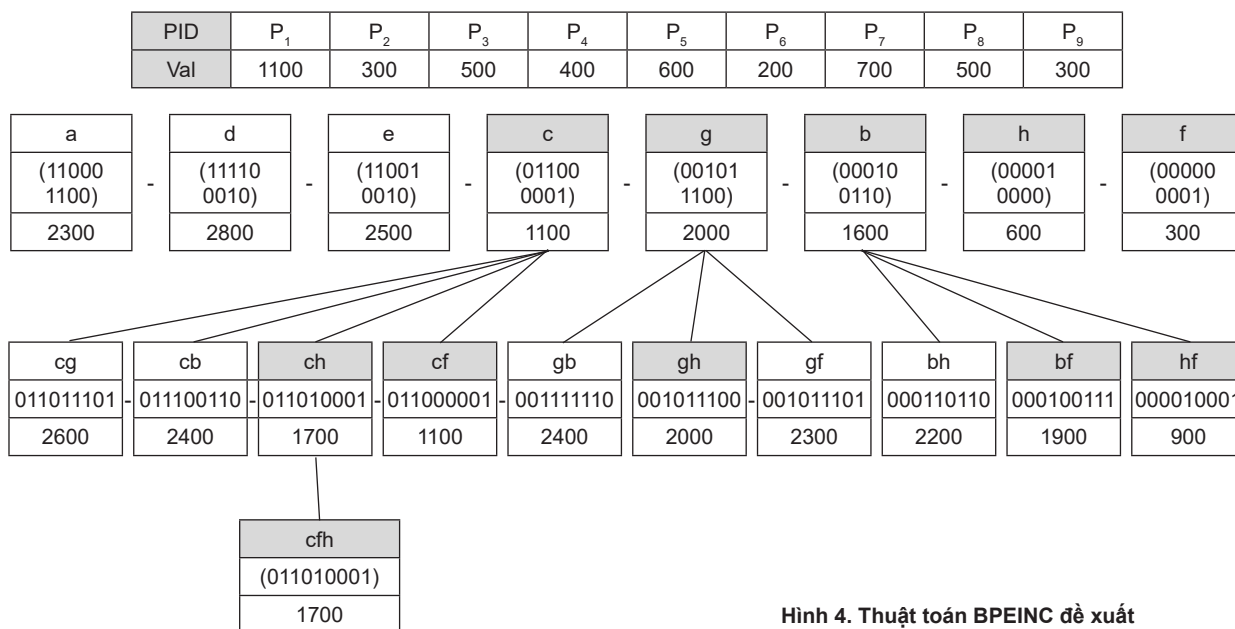
(b) Sau khi thêm P₁ đến P₆

Hình 2. Bitmap-table và BitmapEI-list sau khi thêm CSDL PDB_0 và PDB_1

PID	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
Val	1100	300	500	400	600	200	700	500	300

a	d	e	c	g	b	h	f
(11000 1100)	(11110 0010)	(11001 0010)	(01100 0001)	(00101 1100)	(00010 0110)	(00001 0000)	(00000 0001)
2300	2800	2500	1100	2000	1600	600	300

Hình 3. Danh sách các EI sau khi thêm PDB_1 (các tập được tô màu xám)



Hình 4. Thuật toán BPEINC đề xuất

liệu mới được thêm vào. Nếu người dùng yêu cầu các hoạt động khai thác mẫu có thể xóa, thuật toán tiên hành trích xuất các mẫu có thể xóa hợp lệ bằng cách kết hợp đệ quy các cặp BitmapEI-list. Trong quá trình này, danh sách BitmapEI-list được tạo đệ quy, mỗi danh sách có thông tin về một tập có thể xóa được với độ dài từ hai item trở lên.

Ví dụ 6. Giả sử dữ liệu hiện tại được thêm vào bao gồm PDB_0 và PDB_1 và Bitmap-table và BitmapEI-list như hình 2. Nếu có 1 yêu cầu khai thác từ người dùng với ngưỡng $\xi = 45\% = 4600 * 45\% = 2070$. Kết quả sẽ thu các EI được như Hình 3.

4. Thuật toán đề xuất

Dựa trên các định nghĩa trên, chúng tôi đề xuất thuật toán Khai thác có thể xóa bằng biểu diễn Bitmap (BPEINC) để tìm các tập mục có thể xóa trên cơ sở dữ liệu tăng trưởng một cách hiệu quả. Hình 4 mô tả chi tiết từng bước của thuật toán được đề xuất.

BPEINC

Đầu vào: CSDL gốc, $PDB_0 = \{P_1, P_2, \dots, P_n\}$, CSDL tăng trưởng $IncDB = \{PDB_1, PDB_2, \dots\}$ và một ngưỡng ξ .

Đầu ra: Tập các tập có thể xóa EIs

- [1] *Bitmap-table* $BT = \emptyset$, *BitmapEI-list* $BEI = \emptyset$, $MVT = 0$
- [2] Call *Build-PDB*(PDB_0 , BT , BEI)
- [3] **If** a mining request occurs, **Then**
- [4] $EIs = \emptyset$
- [5] Calculate T
- [6] **For each** item $s \in BEI$, **do**
- [7] $EIs' = \emptyset$
- [8] **If** $gain(s) \leq T$, **Then** $EIs' \leftarrow s$
- [9] **End For**
- [10] $EIs = EIs \cup EIs'$
- [11] Call *Mine_BEINC*(BT , EIs' , T , EIs , ξ)
- [12] Provide the user with EIs , $EIs = \emptyset$
- [13] **End if**
- [14] **If** a insert $PDB_i \in IncDB$ occurs, **Then**
- [15] Call *Build-PDB*(PDB_i , BT , BEI)
- [16] **End if**

Hình 5. Hàm Build-PDB

Build-PDB
Đầu vào: CSDL sản phẩm P, Bitmap-table BT và BitmapEI-list BEI.
Đầu ra: BT và BEI kết quả

```

[1] For each product  $P_k \in P$ , do
[2]    $e.PID = P_k.PID$ 
[3]    $gain(e) = P_k.Val$ 
[4]    $BT \leftarrow e$ 
[5]   For each item  $i_k \in P_k$ , do
[6]     If  $e_i \cap BEI \neq \emptyset$ , Then
[7]        $e_i.Bitmap = e_i.Bitmap \oplus P_k.Bitmap$ 
[8]        $gain(e_i) = gain(e_i) + P_k.Val$ 
[9]     Else
[10]       $e_i.Bitmap = \{b_1, b_2, \dots, b_k, \dots, b_z\} \quad \forall b_k = 1, b_z = 0$ 
[11]       $gain(e_i) = P_k.Val$ 
[12]       $e_i \rightarrow BEI$ 
[13]    End if
[14]  End for
[15] End for

```

Hình 6. Hàm Mine_BEINC

Mine_BEINC
Đầu vào: Bitmap-table BT, tập có thể xóa Els', tổng gain T và ngưỡng ξ
Đầu ra: Tập các tập có thể xóa Els

```

[1]  $Els'' = \emptyset$ 
[2] Calculate MVT:

$$MVT = \xi \times \sum_{e \in BT} gain(e)$$

[3] For each item  $I_i$  in Els', do
[4]   For each item  $I_j$  in Els', do
[5]      $ep.Itemset = I_i.Item \cup I_j.Item$ 
[6]      $ep.Bitmap = I_i.Bitmap \oplus I_j.Bitmap$ 
[7]     For each bitmap  $z \in ep.Bitmap$ 
[8]       If  $z = 1$ , Then
[9]          $gain(ep) += gain(e_z), e_z \in BT$ 
[10]    End For
[11]    If  $gain(ep) \leq MVT$ , Then
[12]       $Els'' = Els'' \cup ep$ 
[13]       $Els = Els \cup ep$ 
[14]    End if
[15]  End for
[16]  If  $count(Els'') > 1$ , Then
[17]    Call Mine_BEINC(BT, Els'', T,  $\xi$ )
[18] End for

```

5. Kết luận và hướng phát triển

Trong bài báo này, chúng tôi đã đề xuất phương pháp khai thác tập có thể xóa trên dữ liệu tăng cường dựa trên biểu diễn Bitmap. Quá trình xác định k-itemsets có thể xóa dựa trên (k-1)-itemsets có thể được tiến hành nhanh chóng với sự hỗ trợ của việc giữ lại Bitmap của các tập mục.

Tiếp theo, chúng tôi sẽ tiến hành chạy thực nghiệm để so sánh đánh giá kết quả của thuật toán đề xuất so với các thuật toán hiện có. Ngoài ra, chúng tôi cũng sẽ tiến hành nghiên cứu phương pháp hiệu quả để cập nhật lại EIs trên CSDL luồng.

TÀI LIỆU THAM KHẢO

- G. Lee, U. Yun, H. Ryang, and D. Kim. Erasable itemset mining over incremental databases with weight conditions. *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 213-234, 2016.
- G. Lee, and U. Yun. Single-pass based efficient erasable pattern mining using list data structure on dynamic incremental databases. *Future Generation Computer Systems*, 80, 12-28, 2018.
- H. Nam, U. Yun, E. Yoon, J. C. W. Lin. Efficient approach for incremental weighted erasable pattern mining with list structure. *Expert Syst. Appl.* 143, 2020.
- R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. *Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 487-499, Sep. 1994.
- T. Le, B. Vo, and F. Coenen. An Efficient Algorithm for Mining Erasable Itemsets Using the Difference of NC-Sets. *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2270- 2274, Oct. 2013.
- T. Le and B. Vo. MEI: An efficient algorithm for mining erasable itemsets. *Engineering Applications of Artificial Intelligence*, vol. 27, pp. 155-166, Jan. 2014.
- T.P. Hong, K. Y. Lin, C. W. Lin, and B. Vo. An incremental mining algorithm for erasable itemsets. *IEEE International Conference on innovations in intelligent systems and applications*, pp. 286-289, 2017.
- T.P. Hong, GC. Lan, MC. Chiang và J. CW. Lin. A Bitmap Approach for Mining Erasable Itemsets. *IEEE Access*, vol. 9, 2021.
- U. Yun, G. Lee. Sliding window based weighted erasable stream pattern mining for stream data applications. *Future Gener. Comput. Syst.* 59 (2016) 1-20.
- Z. Deng, G. Fang, and Z. Wang. Mining erasable itemsets. *Proceedings of the 8th International Conference on Machine Learning and Cybernetics*, vol. 1, pp. 67-73, Jul. 2009.
- Z. Deng and X. Xu. An Efficient Algorithm for Mining Erasable Itemsets. *Advanced Data Mining and Applications*, pp. 214-225, Nov. 2010.
- Z. Deng and X. Xu. Fast mining erasable itemsets using NC_sets. *Expert Systems with Applications*, vol. 39, no. 4, pp. 4453-4463, Mar. 2012.