FHNM: THUẬT TOÁN KHAI PHÁ TẬP MỤC HỮU ÍCH CAO TỪ CƠ SỞ DỮ LIỆU GIAO TÁC CÓ GIÁ TRỊ HỮU ÍCH ÂM

FHNM: HIGH UTILITY ITEMSETS MINING ALGORITHM FROM TRANSACTION DATABASE WITH NEGATIVE UTILITY VALUE

Huỳnh Triệu Vỹ[1], Lê Quốc Hải[2], Phạm Khánh Bảo[1]

[1]Trường Đại học Phạm Văn Đồng; htrvy@yahoo.com, pkbao@pdu.edu.vn
[2]Trường Cao đẳng Sư phạm Quảng Trị; hailq79@gmail.com

Tóm tắt - Các thuật toán khai phá tập mục hữu ích cao thường có xu thế khai thác được các tập mục có nhiều mục [1, 2, 3]. Tuy nhiên, các tập mục có nhiều mục thường là các tập mục hiếm nên không có nhiều ý nghĩa đối với người sử dụng [5]. Thuật toán FHM+ [5] khai phá tập mục hữu ích cao, nhưng thu gọn được độ dài của các tập mục với điều kiện giá trị hữu ích của các mục là dương, nhưng trong thực tế có nhiều cơ sở dữ liệu giao tác có chứa các mục có giá trị hữu ích ngoại âm. Vấn đề đặt ra, là làm thế nào để khai phá tập mục hữu ích cao từ cơ sở dữ liệu có chứa các mục có giá trị hữu ích ngoại là âm, dựa trên ràng buộc về độ dài của tập mục. Để giải quyết vấn đề đã đặt ra, trong bài báo này, chúng tôi đề xuất một thuật toán mới được xây dựng từ sự cải tiến của thuật toán FHM+ và FHN [4] có tên là FHNM.

Từ khóa - cơ sở dữ liệu giao tác; tập mục hữu ích cao; khai phá tập mục hữu ích cao; hữu ích ngoại âm; ràng buộc độ dài

Abstract - Algorithms for mining high utility itemset normally aims at discovering itemsets that contain more items [1, 2, 3]. However, the itemsets that contain more items are rare in the database and have little meaning to users [5]. Therefore, the algorithm FHM+ [5] discovers high utility itemsets and reduces their length while maintains the condition that the foreign utility of those items is positive. The problem addressed here is how to discover high utility itemsets constrained by their length from database containing items that have negative foreign utility value. In order to solve the addressed problem, this paper proposes an algorithm named FHNM by improving FHM+ and FHN [4].

Key words - transaction database; high utility itemsets; high utility itemsets mining; external negative utility; length constraints

1. Giới thiệu

Các kỹ thuật tỉa không gian tìm kiếm, được phát triển trong khai phá tập mục phổ biến không áp dụng trực tiếp được trong khai phá tập mục hữu ích cao [3], do tính chất của tập phổ biến không giống như tập hữu ích cao. Vì vậy, năm 2004, Hong Yao, Howard J. Hamilton [6] đã đề xuất một mô hình nền tảng để giải quyết bài toán khai phá tập mục hữu ích cao. Trong mô hình này, họ đã định nghĩa hai đơn vị đo lường hữu ích cho mỗi mục, đó là hữu ích giao tác (transaction utility) và hữu ích ngoại (external utility). Mô hình toán học trong [6] được định nghĩa dựa trên cơ sở của hai tính chất, đó là ràng buộc hữu ích và ràng buộc hỗ trợ. Tính chất ràng buộc hữu ích có thể được áp dụng vào trong chiến lược tỉa không gian tìm kiếm. Dựa trên mô hình này, Hong Yao, Howard J. Hamilton [7] đã đề xuất các thuật toán Uming và UmingH. Các kỹ thuật tỉa không gian tìm kiếm mà các thuật toán này áp dụng có khả năng thu gọn một phần tập ứng viên. Năm 2005, Liu. Y, Liao. W, A. Choudhary [8] đã đề xuất một thuật toán hai pha TwoPhase để khai phá tập mục hữu ích cao. Các tác giả đã đưa ra khái niệm về hữu ích của giao tác và hữu ích của tập mục, tính theo hữu ích của giao tác chứa nó gọi là TWU (Transaction-Weighted-Utilization). Các tác giả đã chứng minh được TWU có tính chất phản đơn điệu, là yếu tố cốt lõi để thuật toán hai pha rút gọn nhanh không gian tìm kiếm. Trên cơ sở này, một số thuật toán sau đó đã được đề xuất hiệu quả hơn [3, 4, 6] về độ phức tạp tính toán. Tuy nhiên, tính chất của đơn vị TWU chỉ còn đúng khi tất cả giá trị hữu ích của các mục là dương, tức không thể xuất hiện bất cứ mục nào trong cơ sở dữ liệu có giá trị hữu ích ngoại là âm. Trong thực tế, nhiều cơ sở dữ liệu có các giao tác chứa các mục có giá trị hữu ích ngoại là âm. Nếu các mục này được khai thác thì mang lại một giá trị có hữu ích cao. Chẳng hạn như trong lĩnh vực kinh doanh có những mặt hàng được bán ra chấp nhận lỗ để có thể bán kèm theo mặt hàng khác, và kết quả của việc bán kèm theo như thế sẽ đem lại lợi nhuận cao. Để khai thác những giá trị hữu ích này, Chu, C.-J., Tseng, V. S., Liang [1] và Philippe Fournier-Viger [4] đã đề xuất các thuật toán để khai phá tập mục hữu ích cao trong cơ sở dữ liệu có giá trị hữu ích ngoại là âm.

Các thuật toán khai phá tập mục hữu ích cao trước đây có xu thế khai phá được các tập mục có chiều dài lớn, tuy nhiên, các mục này thường là các mục hiếm, nên ít có ý nghĩa đối với người sử dụng [6]. Để khắc phục hạn chế này, các tác giả trong [6] đề xuất thuật toán FHM+ để khai phá các tập mục hữu ích cao dựa theo ràng buộc về độ dài của tập mục. FHM+ cho thấy hiệu quả hơn các thuật toán trước đây. Tuy nhiên, FHM+ cũng chỉ áp dụng để khai phá tập mục hữu ích cao từ cơ sở dữ liệu không chứa bất cứ mục nào có giá trị hữu ích âm. Để giải quyết hạn chế này, trong bài báo chúng tôi đề xuất một thuật toán có tên là FHNM (cải tiến từ thuật toán FHN và FHM+) để khai phá tập mục hữu ích cao từ cơ sở dữ liệu có chứa các mục có giá trị hữu ích ngoại âm hiệu quả hơn thuật toán FHN. FHNM áp dụng chiến lược tỉa không gian tìm kiếm dựa vào ràng buộc về độ dài của tập mục.

Nội dung tiếp theo của bài báo được tổ chức như sau: Phần 2 trình bày về khai phá tập mục hữu ích cao dựa trên ràng buộc về độ dài của tập mục, Phần 3 trình bày thuật toán FHNM, Phần 4 trình bày kết quả đạt được và so sánh với thuật toán khác, Phần 5 kết luận.

2. Khai phá tập mục hữu ích cao dựa trên ràng buộc về độ dài của tập mục

Định nghĩa 1 (Cơ sở dữ liệu giao tác và giá trị hữu ích của tập mục): *Cho* $I=\{i_1, i_2, \ldots, i_m\}$ là một tập các mục. $D=\{T_1, T_2, \ldots, T_m\}$ là cơ sở dữ liệu giao tác, ở đây, mỗi

$T_c \in D$ là tập con của *I*. Mỗi mục $i \in T_c$ có một giá trị dương, ký hiệu là *q(i,Tc)* được gọi là giá trị hữu ích nội của *i (tương ứng với số lượng của i trong mỗi T_c)*. Mỗi mục $i \in I$ có một giá trị hữu ích ngoại, ký hiệu là *p(i) (tương ứng với giá trị hữu ích của mục i)*.

Hữu ích của mục $i \in T_c$, *được định nghĩa là* $u(i,T_c) = q(i,T_c) \times p(i)$. Hữu ích của tập mục *X* trong giao tác T_c, được định nghĩa $u(X,T_c) = \sum_{i \in X \wedge X \subseteq T_c} u(i,T_c)$. Hữu ích của tập mục *X* trong cơ sở dữ liệu *D*, được định nghĩa $u(X) = \sum_{T_c \in D \wedge X \subseteq T_c} u(X,T_c)$.

Định nghĩa 2 (Bài toán Khai phá tập mục hữu ích cao theo ràng buộc về độ dài của tập mục): Cho *minutil*, *minlength*, và *maxlength* là các tham số do người dùng thiết lập. Vấn đề khai phá tập mục hữu ích cao với ràng buộc về độ dài của tập mục cho trước là tìm ra tất cả các tập mục có độ hữu ích không nhỏ hơn *minutil* và số lượng các mục trong mỗi tập mục không nhỏ hơn *minlength* và không lớn hơn *maxlength*. Trong bài báo này, giả sử rằng bốn tham số trên mặc nhiên đã được thiết lập bởi người sử dụng. Các định nghĩa đưa ra trong bài báo đều sử dụng các tham số *minlength* và *maxlength* để ràng buộc về độ dài tập mục.

Định nghĩa 3 (Tập hữu ích lớn nhất trong giao tác): Cho giao tác $T_c=\{i_1,i_2,...,i_k\}$. Tập hữu ích lớn nhất của giao tác T_c là một tập có đúng *maxlength* mục được chọn từ tập $\{u(i_1,T_c), u(i_2,T_c), \ldots, u(i_k,T_c)\}$ sao cho tổng giá trị hữu ích của chúng là lớn nhất, ký hiệu là $L(T_c)$.

Định nghĩa 4 (Giá trị hữu ích lớn nhất của giao tác): Cho giao tác $T_c=\{i_1,i_2,...,i_k\}$. Giá trị hữu ích lớn nhất của giao tác T_c là tổng giá trị hữu ích của các mục trong $L(T_c)$, được định nghĩa như sau: $RTU(T_c) = \sum_{u(i,T_c) \in L(T_c)} u(i,T_c)$

Định nghĩa 5 (Trọng số hữu ích lớn nhất của giao tác trong cơ sở dữ liệu): Trọng số hữu ích lớn nhất của một tập mục *X* trong cơ sở dữ liệu *D* là tổng giá trị hữu ích lớn nhất của các giao tác chứa tập *X* theo ràng buộc về độ dài tập mục, được định nghĩa như sau: $RTWU(X) = \sum_{T_c \in D \wedge X \subseteq T_c} RTU(T_c)$.

Tính chất 1: Trọng số hữu ích của một tập mục *X* luôn luôn lớn hơn hoặc bằng giá trị hữu ích của chính nó theo ràng buộc về độ dài của tập mục, tức là: $RTWU(X) \geq u(X)$ [6].

Tính chất 2 (Tỉa không gian tìm kiếm dựa vào RTWU): Cho *X* là một tập mục, nếu *RTWU*(*X*)<*minutil* thì tập mục *X* và tất cả tập cha của *X* không phải là tập mục hữu ích cao [6].

Định nghĩa 6 (Hữu ích lớn nhất trong giao tác của tập mục dùng để mở rộng tập mục): Cho $\succ$ là một quan hệ sắp thứ tự toàn phần trên tập các mục từ *I*; giao tác T_c và tập mục *X*. Gọi $V(T_c,X)=\{v_1,v_2,...,v_k\}$ là tập các mục xuất hiện trong T_c mà có thể bổ sung vào *X*, ký hiệu: $V(T_c,X) = \{v \in T_c \mid v \succ x, \forall x \in X\}$. Số mục tối đa có thể bổ sung vào trong *X* sao cho tập mục kết quả đảm bảo về ràng buộc độ dài tối đa của tập mục được định nghĩa: *maxExtend*(*X*)=*maxlength* - |*X*|, ở đây |*X*| là lực lượng của *X*. Hữu ích lớn nhất trong giao tác T_c của tập mục có thể bổ sung vào trong *X* là một tập gồm *maxExtend(X)* phần tử lớn nhất từ tập $\{u(v_1,T_c), u(v_2,T_c),\ldots, u(v_k,T_c)\}$, ký hiệu: $L(T_c,X)$.

Định nghĩa 7 (Giá trị hữu ích còn lại): Cho giao tác T_c và tập mục X. Giá trị hữu ích còn lại của tập mục X trong giao tác T_c được định nghĩa như sau: $rru(T_c,X) = \sum_{u(v_j,T_c) \in L(T_c,X)} u(v_j,T_c)$. Giá trị hữu ích còn lại của tập mục X trong cơ sở dữ liệu D được định nghĩa như sau: $rreu(X) = \sum_{T_c \in D \wedge X \subseteq T_c} rru(T_c,X)$.

Tính chất 2 (Tỉa không gian tìm kiếm dựa vào giá trị hữu ích còn lại): Cho tập mục *X*. Nếu tổng *u*(*X*)+*rreu*(*X*) < *minutil* thì tập mục *X* cũng như tập mở rộng của *X* không phải là tập hữu ích cao dựa theo ràng buộc về độ dài của tập mục [6].

Định nghĩa 8 (Cấu trúc danh sách hữu ích): *Danh sách hữu ích* của một tập mục *X* (ký hiệu *rul*(*X*)) trên cơ sở dữ liệu *D* là một tập của các bộ *tuple(tid, iutil, llist)* cho mỗi giao tác T_{tid} chứa *X*. Trong đó: *iutil* = $u(X,T_{tid})$; *llist*=$L(T_{tid},X)$.

Tính chất 3 (Tỉa không gian tìm kiếm sử dụng cấu trúc rul): Cho tập mục *X*. Nếu $\sum iutil + \sum llist$ của rul(X) nhỏ hơn *minutil* thì tập *X* cũng như tập mở rộng của X không phải là tập mục hữu ích cao [6].

3. Thuật toán FHNM

Thuật toán FHM+ hay các thuật toán khai phá tập mục hữu ích cao trước đây sử dụng tính chất phản đơn điệu của Trọng số hữu ích giao tác (*TWU*) để tỉa không gian tìm kiếm [2, 3, 4, 7, 8]. Tuy nhiên, tính chất này chỉ đúng đối với cơ sở dữ liệu chứa các mục có giá trị hữu ích ngoại là dương. Trong trường hợp áp dụng các thuật toán này để khai phá tập mục hữu ích cao từ cơ sở dữ liệu có chứa các mục có đơn vị lợi tức ngoại âm sẽ cho ra kết quả sai, điều này được minh chứng qua ví dụ 1.

Bảng 1. *Cơ sở dữ liệu giao tác có đơn vị hữu ích nội*

TID	a	b	c	d	e	f	g
T1	1	0	1	1	0	0	0
T2	2	0	6	0	2	0	5
T3	1	2	1	6	1	5	0
T4	0	4	3	3	1	0	0
T5	0	2	2	0	1	0	2

Bảng 2. *Bảng hữu ích ngoại*

Item	a	b	c	d	e	f	g
Profit	-5	2	1	2	3	1	1

Ví dụ 1: Xét cơ sở dữ liệu cho ở Bảng 1 và 2 với *minutil*=15 thì: *u*({*c*,*e*,*g*})=24; *RTWU*({*c*,*e*,*g*})=14.

Suy ra, *RTWU*({*c*,*e*,*g*})<*minutil*, nhưng *u*({*c*,*e*,*g*})>*minutil*, như vậy không thỏa mãn tính chất 2.

Để giải quyết hạn chế này trong Mục 3.1 tác giả đưa ra các định nghĩa và tính chất để có thể áp dụng tính chất phản đơn điệu của *RTWU* vào trong chiến lược tỉa không gian tìm kiếm để có thể khai phá được tập mục hữu ích cao trong cơ sở dữ liệu giao tác có chứa các mục có giá trị hữu ích âm.

3.1. Các định nghĩa và tính chất sử dụng trong thuật toán FHNM

Định nghĩa 9 (Định nghĩa hữu ích lớn nhất trong giao tác của CSDL có chứa mục có hữu ích ngoại âm)

Cho giao tác T_c={i_1,i_2,..., i_k}. Hữu ích lớn nhất của giao tác T_c là một tập có tối đa *maxlength* phần tử có giá trị lớn nhất không âm trong tập {$u(i_1,T_c)$,$u(i_2,T_c)$,..., $u(i_k,T_c)$}, ký hiệu là $RL(T_c)$

Ví dụ 2: Xét cơ sở dữ liệu ở ví dụ 1, nếu *maxlength* = 3 thì hữu ích lớn nhất của T_1, T_2, T_3, T_4, T_5 sẽ là: $RL(T_1)$={1,2}; $RL(T_2)$={6,6,5}; $RL(T_3)$={12,5,4}; $RL(T_4)$={8,6,3}; $RL(T_5)$={4,3,2}.

Dựa vào định nghĩa này sẽ đảm bảo các tính chất 1 và 2 là đúng để áp dụng vào trong chiến lược tỉa không gian tìm kiếm. Điều này được chứng minh như sau:

Chứng minh tính chất 1 là đúng với cơ sở dữ liệu có chứa mục có giá trị hữu ích âm:

Ta có: $u(X)=\sum_{T_c\in D\wedge X\subseteq T_c} u(X,T_c)=\sum_{T_c\in D\wedge X\subseteq T_c}\sum_{i\in X} u(i,T_c)$

$RTWU(X)=\sum_{T_c\in D\wedge X\subseteq T_c} RTU(T_c)=\sum_{T_c\in D\wedge X\subseteq T_c}\sum RL(T_c)$

Theo định nghĩa 8 thì $\sum_{i\in X} u(i,T_c)\le\sum RL(T_c)$

Suy ra $RTWU(X)\ge u(X)$

Chứng minh tính chất 2 là đúng với cơ sở dữ liệu có chứa mục giá trị hữu ích âm:

Cho I^k là tập mục có k mục và I^{k-1} là tập mục có k-1 mục, như vậy $I^{k-1}\subset I^k$ (1)

Cho T_{I^k} là tập các giao tác chứa I^k và $T_{I^{k-1}}$ là tập các giá tác chứa I^{k-1}. Từ (1) suy ra, $T_{I^k}\subseteq T_{I^{k-1}}$ (2)

$RTWU(I^{k-1})=\sum_{T_c\in D\wedge I^{k-1}\subseteq T_c} RTU(T_c)=\sum_{T_c\in D\wedge I^{k-1}\subseteq T_c}\sum RL(T_c)$

$RTWU(I^k)=\sum_{T_c\in D\wedge I^k\subseteq T_c} RTU(T_c)=\sum_{T_c\in D\wedge I^k\subseteq T_c}\sum RL(T_c)$

Vì $T_{I^k}\subseteq T_{I^{k-1}}\Rightarrow RTWU(I^{k-1})\ge RTWU(I^k)$

Theo tính chất 1, nếu $RTWU(I^{k-1})<\min util$ thì $u(I^{k-1})<\min util$.

$\Rightarrow RTWU(I^k)<\min util\wedge u(I^k)<\min util$

Định nghĩa 10 (Tập mục có giá trị hữu ích ngoại âm, dương): Cho tập mục *X* có chứa các mục có giá trị hữu ích ngoại âm hoặc dương hoặc cả hai, $up(X)\subseteq X$ là tập tất cả các mục dương trong tập *X* và $un(X)\subseteq X$ là tập tất cả các mục âm trong *X*.

Tính chất 3: Với tập mục *X*, thì $u(X)\le u(up(X))$ và $u(X)\ge u(un(X))$.

Chứng minh: Theo định nghĩa 9, ta có

$u(X)=u(up(X))+u(un(x))$

Vì $u(un(X))\le 0\wedge u(up(X))\ge 0$

$\Rightarrow u(X)\le u(up(X))$ và $u(X)\ge u(un(X))$

Tính chất 4: Cho tập mục *X* và mục z có giá trị hữu ích ngoại âm, $z\notin X$, ta có $u(up(X)\cup\{z\})<u(up(X))$.

Chứng minh: Vì z là mục có giá trị hữu ích âm nên $u(\{z\})<0\Rightarrow u(up(X)\cup\{z\})<u(up(X))$.

Tính chất 5: Cho X là một tập mục, Y là tập mục mở rộng của tập mục X từ các mục có giá trị hữu ích ngoại âm, ta có: $u(Y)<u(X)$.

Chứng minh: Dựa vào tính chất 4 suy ra được tính chất 5.

Tính chất 6 (Điều kiện cắt tỉa không gian tìm kiếm): Cho *X* là một tập mục, nếu $u(up(X))<minutil$ và chỉ có các mục âm có thể được sử dụng để mở rộng *X* thì tất cả các tập mở rộng này điều có giá trị hữu ích thấp.

Chứng minh: Dựa vào tính chất 5 suy ra được tính chất 6.

Việc sử dụng điều kiện cắt tỉa này trong thuật toán với yêu cầu là để có thể tính $u(up(X))$ một cách hiệu quả, ta cần định nghĩa lại cấu trúc danh sách hữu ích bằng cách tách giá trị *iutil* thành hai giá trị *iputil* và *inutil* tương ứng với $u(up(X),T_c)$ và $u(un(X),T_c)$.

Định nghĩa 11 (Định nghĩa lại cấu trúc Danh sách hữu ích): Cho $\succ$ là một quan hệ thứ tự toàn phần trên các mục từ *I*, mà các mục $i\in I$ có thể có giá trị hữu ích ngoại âm. Danh sách hữu ích của một tập mục *X* (ký hiệu *rrul*(*X*)) trên cơ sở dữ liệu *D* là một tập của các bộ *tuple(tid, {iputil,inutil}, llist)* cho mỗi giao tác T_{tid} chứa *X*. Trong đó: $iputil=u(up(X),T_{tid})$; $inutil=u(un(X),T_{tid})$; $llist=L(T_{tid},X)$.

Ví dụ 5: Xét cơ sở dữ liệu ở ví dụ 1, maxlength=3:

rrul({*a*})={(T_1,{0,-5},{2,1}),(T_2,{0,-10},{6,6}, (T_3,{0,-5},{12,5})}

rrul({*b*})={(T_3,{4,0},{12,5}),(T_4,{8,0},{6,3}), (T_5,{4,0},{3,2})}

rrul({a,b})={(T_3,{4,-5},{12})}

Tính chất 7 (Tỉa không gian tìm kiếm sử dụng cấu trúc rrul): Cho tập mục *X*. Tập mở rộng của *X* là tập sau khi bổ sung vào tập *X* một mục y, sao cho $y\succ i$, $\forall i\in X$. Nếu $\sum iputil+\sum llist$ của *rrul*(*X*) nhỏ hơn *minutil* thì tập *X* cũng như tập mở rộng của *X* không phải là tập mục hữu ích cao dựa trên ràng buộc về độ dài của tập mục.

Chứng minh: Cho *Y* là tập mở rộng của tập mục *X* (*X*⊂*Y*) thỏa mãn ràng buộc về độ dài của tập mục, ta có:

$X\subset Y\subseteq T_c\Rightarrow (Y/X)\subseteq T_c/X$

$u(Y,T_c)=u(X,T_c)+u((Y/X),T_c)$

$=u(X,T_c)+\sum_{i\in(Y/X)} u(i,T_c)\le u(X,T_c)+\sum_{i\in(T_c/X)} u(i,T_c)$

$\le u(up(X),T_c)+\sum_{i\in(T_c/X)} u(i,T_c)=u(up(X),T_c)+L(T_c,X)$

Cho T_X là tập các giao tác chứa X và T_Y là tập các giao tác chứa Y, ta có:

$$X \subset Y \Rightarrow T_Y \subseteq T_X$$

$$u(Y) = \sum_{T_c \in T_Y} u(Y, T_c) \le \sum_{T_c \in T_Y} u(X, T_c) + \sum_{T_c \in T_Y} L(X, T_c)$$
$$\le \sum_{T_c \in T_Y} u(up(X), T_c) + \sum_{T_c \in T_Y} L(X, T_c)$$
$$\le \sum_{T_c \in T_X} u(up(X), T_c) + \sum_{T_c \in T_X} L(X, T_c) < \min util$$

3.2. Thuật toán FHNM

Thuật toán FHNM bao gồm một thủ tục chính có tên là FHNM và 2 thủ tục phụ có tên là Search và Construct. Sau đây là mô tả chi tiết về thuật toán.

3.2.1. Thuật toán FHNM

Mô tả thuật toán

Đầu tiên duyệt qua cơ sở dữ liệu *D* để tính giá trị *RTWU* của từng mục *i* (áp dụng định nghĩa 9 vào trong công thức tính *RTWU)*. Tiếp đến, xây dựng tập I^* của tất cả các mục *i* có giá trị *RTWU(i)* ≥ *minutil*, tức là loại bỏ đi các mục có giá trị hữu ích thấp (sử dụng tính chất 2). Sau khi có I^*, thiết lập $\succ$ là bộ sắp thứ tự toàn phần các giá trị *RTWU* tăng dần trên I^*. Duyệt qua cơ sở dữ liệu lần thứ 2 để xây dựng danh sách hữu ích (áp dụng định nghĩa 10) của từng mục $i \in I^*$ và xây dựng cấu trúc *EUCS* (*EUCS* được đề xuất trong FHNM [3] nhằm mục đích chứa các *RTWU* của các cặp mục để hỗ trợ cho việc tính *RTWU* được nhanh chóng). Kiểm tra nếu minlength ≤ 1, xuất các mục $i \in I^*$ sao cho tổng giá trị hữu ích của mục {*i*} không nhỏ hơn *minutil*, ngược lại gọi thủ tục đệ qui *Search* để thăm dò, tìm kiếm theo chiều sâu của một tập mục, bắt đầu với tập rỗng để tìm tập hữu ích cao.

Thuật toán

Vào: Cơ sở dữ liệu *D*, *minutil*, *minlength*, *maxlength*;

Ra: Tập các mục hữu ích cao;

1. Duyệt cơ sở dữ liệu *D* để tính *RTWU* của các mục đơn;
2. Xây dựng I^* từ các mục *i* sao cho *RTWU*(*i*) ≥ *minutil*;
3. Sử dụng quan hệ $\succ$ để sắp thứ tự toàn phần các giá trị *RTWU* tăng dần trên I^*;
4. Duyệt *D* để xây dựng rrul của mỗi mục $i \in I^*$ và xây dựng cấu trúc *EUCS*;
5. if (*minlength* ≤ 1);
6. Xuất các mục $i \in I^*$ sao cho

   SUM({*i*}.*rrul.iputil*) ≥ *minutil*;
7. if (*maxlength*>1);
8. Search(ϕ , I^*, minutil, minlegth, maxlength, EUCS);

3.2.2. Thủ tục Search

Mô tả thủ tục

Với mỗi phần mở rộng *Px* của *P*, nếu *iputil* + *llist* trong danh sách hữu ích của tập *Px* không nhỏ hơn *minutil* thì *Px* và phần mở rộng của *Px* cần được khai thác (sử dụng tính chất 9). Điều này được thực hiện bằng cách kết hợp *Px* với tất cả các phần mở rộng *Py* của *P*, sao cho $y \succ x$ để hình thành *Pxy* có |*Px*| + *1* mục. Danh sách hữu ích của của *Pxy* được xây dựng bằng cách gọi thủ tục *Construct(P,Px,Py)* để liên kết danh sách hữu ích của *P*, *Px*, *Py*, sau đó kiểm tra nếu tổng giá trị hữu ích của *Pxy* không nhỏ hơn *minutil* và *Pxy* vẫn thỏa mãn ràng buộc về độ dài của tập mục thì xuất *Pxy* (sử dụng tính chất 7). Tiếp đến, nếu độ dài của tập mục *Pxy* vẫn còn nhỏ hơn maxlength thì tiếp tục mở rộng tập mục *Pxy* bằng cách gọi thủ tục *Search*.

Thủ tục

- **Vào**: Tập mục P, tập mở rộng từ P ExtendsionsOfP, minutil, minlength, maxlength, EUCS;
- **Ra:** Tập mục hữu ích cao;

1. foreach ($Px \in$ ExtendsionsOfP) do
2. if (*SUM*(*Px.rrul.iputil*) + *SUM*(*Px.rrul.llist*) ≥ minutil) then
3. ExtensionsOfPx ← Φ;
4. foreach (tập mục $Py \in$ ExtensionsOfPx sao cho $y \succ x$) do
5. if $(\exists (x, y, c) \in EUCS)$ sao cho $c \ge$ minutil) then
6. $Pxy \leftarrow Px \cup Py$;
7. Pxy.utilitylist ← Construct(*P*,*Px*,*Py*);
8. ExtensionsOfPx ← ExtensionsOfPx ∪ *Pxy*;
9. if(*SUM*(*Pxy.rrul.iputils*+ *Pxy.rrul.inutils*) ≥ *minutil* và *minlength* ≤ |*Pxy*| ≤ *maxlength*) then xuất *Pxy*;
10. end
11. end
12. if(|*Pxy*| <*maxlength*) then
13. Search(*Px*,*ExtensionsOfPx*, *minutil*)
14. end
15. end

3.2.3. Thủ tục Construct

Thủ tục này nhằm mục đích tính danh sách hữu ích của tập mục Pxy.

Vào: Tập mục *P*, *Px*: Tập mở rộng của *P* với mục *x*, *Py*: Tập mở rộng của *P* với mục *y*;

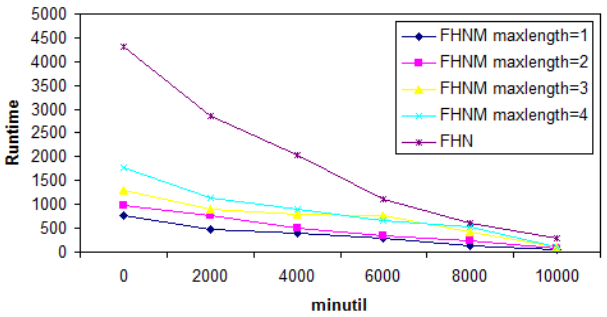
Ra: Danh sách hữu ích của *Pxy*;

1. UtilityListOfPxy ← ϕ;
2. foreach (*tuple ex*∈*Px.rrul*) do
3. if ($\exists$ *ex* ∈ *Py.rrul* and *ex.tid*=*exy.tid*) then
4. if (*P.rrul* ≠ ϕ) then
5. Tìm phần tử *e* ∈ *P.rrul* sao cho *e.tid*=*ex.tid*;
6. *exy* ← (*ex.tid*, *ex.iputil* + *ey.iputil*-*e.iputil*-*e.iputil*, *ey.llist*);
7. end
8. else
9. *exy* ← (*ex.tid*,*ex.iputil*+*ey.iputil*,*ey.llist*);
10. end
11. UtilityListOfPxy ← UtilityListOfPxy ∪ {*exy*};
12. end

13. end

14. return UtilityListPxy;

4. Đánh giá thuật toán



Hình 1. *Thời gian thực hiện của thuật toán FHNM và FHN với các ngưỡng hữu ích và maxlength khác nhau*

Trong phần này chúng tôi so sánh kết quả thực nghiệm của thuật toán FHNM so với thuật toán FHN trên cùng cơ sở dữ liệu Retail mẫu được lấy từ nguồn "*http://www.philippe-fournier-viger.com/spmf/datasets/ndatasets/retail_negative.txt, 12/2016*" gồm hơn 121 nghìn giao tác có chứa các mục có giá trị hữu ích ngoại âm, đây là bộ dữ liệu được sử dụng trong thuật toán FHN. Kết quả thực nghiệm khi chạy trên cùng một hệ thống máy tính cho thấy thuật toán FHNM có thời gian thực thi nhanh hơn thuật toán FHN do FHNM không vét sạch tất cả tập mục hữu ích cao thỏa mãn ngưỡng hữu ích tối thiểu, mà chỉ khai phá các tập mục hữu ích cao thỏa mãn độ dài của tập mục cho trước. Với *maxlength* càng nhỏ thì FHNM thực thi càng nhanh hơn FHN. Trường hợp *maxlength* lớn hơn hoặc bằng với độ dài lớn nhất của tập mục khai phá được thì thời gian xử lý của FHNM tương đương với FHN.

5. Kết luận

Khai phá tập mục hữu ích cao là một hướng nghiên cứu rất được quan tâm hiện nay và được ứng dụng rộng rãi trong bài toán hỗ trợ ra quyết định. Tuy nhiên, các kết quả nghiên cứu đã được công bố chủ yếu tập trung vào vấn đề khai phá tập mục hữu ích cao dựa vào tính chất của đơn vị TWU để tỉa không gian tìm kiếm, nên chỉ có thể áp dụng được trên cơ sở dữ liệu giao tác không chứa giá trị hữu ích ngoại âm. Hiện nay có rất ít công trình nghiên cứu về khai phá tập mục hữu ích cao cho trên cơ sở dữ liệu giao tác có chứa giá trị hữu ích ngoại âm, trong khi đó, trong thực tế có rất nhiều cơ sở dữ liệu giao tác mà trong đó có chứa đơn vị hữu ích ngoại âm cần được khai thác. Trong bài báo này, nhóm tác giả đã đề xuất thuật toán FHNM được cải tiến từ thuật toán FHM+ [6] và FHN [4] để khai phá các tập mục hữu ích cao trong CSDL có chứa hoặc không chứa hữu ích ngoại âm. Thuật toán này ưu điểm hơn thuật toán FHN là tốc độ xử lý nhanh và ít tốn bộ nhớ hơn, bởi vì thuật toán FHNM được xây dựng dựa trên ràng buộc độ dài của tập mục cần khai thác.

TÀI LIỆU THAM KHẢO

[1] Chu, C.-J., Tseng, V. S., Liang, An efficient algorithm for mining high utility itemsets with negative item values in large databases, *In: Applied Math. Comput*, pp. 767-778 (2009).

[2] Erwin A., Gopalan R., Achutan (2008), "Efficient Mining of High Utility Itemsets from Large Datasets", *T. Washio: PAKDD 2008, LNAI 5012*, pp. 554-561.

[3] Fournier-Viger, P., Wu, C.-W., Zida, S., Tseng, V. S., FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning, *In: Proc. 21st Intern. Symp. on Methodologies for Intell. Syst.*, pp. 83{92 (2014).

[4] Philippe Fournier-Viger, FHN: Efficient Mining of High-Utility Itemsets with Negative Unit Profits, Advanced Data Mining and Applications, *Volume 8933 of the series Lecture Notes in Computer Science*, 2014, pp 16-29.

[5] Philippe Fournier Viger, Chun-Wei Jerry Lin, Quang-Huy Duong, Thu-Lan Dam, FHM+: Faster High-Utility Itemset Mining Using Length Upper-Bound Reduction, *29th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems*, IEA/AIE 2016, pp 115-127.

[6] Hong Yao, Howard J. Hamilton, *A foundational Approach to Mining Itemset Utilities from Databases*, In: 4th SIAM International Conference on Data Mining, Florida USA (2004).

[7] Hong Yao, Howard J. Hamilton, "Mining Itemset Utilities from Transaction Databases", *Journal Data & Knowledge Engineering*, Volume 59 Issue 3, December 2006, pp 603 - 626.

[8] Liu. Y, Liao. W, A. Choudhary, *A fast high utility itemsets mining algorithm*, in: Proceedings of the Utility-Based Data Mining Workshop, August 2005.

(BBT nhận bài: 04/05/2017, hoàn tất thủ tục phản biện: 26/05/2017)