Tạp chí Tin học và Điều khiển học, T.23, S.4 (2007), 364–373

KHÁM PHÁ TẬP MỤC LỢI ÍCH CAO TRONG CƠ SỞ DỮ LIỆU

NGUYỄN THANH TÙNG

Viện Công nghệ thông tin, Viện Khoa học và Công nghệ Việt Nam

Abstract. Recently, to address the limitations of traditional association rules mining methods, a utility mining model was proposed. Utility is a measure of how "usefull" (i.e. "profitable") an itemset is. The goal of utility mining is to identify all high utility itemsets that drive a large portion of the overall utility.

In this paper, we propose an algorithm to efficiently prune down the number of candidates and precisely obtain the complete set of high utility itemsets. The experimental results indicate that this algorithm performs very efficiently in term of speed and memory cost, even on large databases.

Tóm tắt. Nhằm khắc phục những hạn chế của phương pháp khai phá luật kết hợp truyền thống, gần đây người ta đã đề xuất mô hình khai phá lợi ích. Lợi ích là số đo lợi nhuận mà một tập mục có thể đem lại. Mục đích của khai phá lợi ích là phát hiện tất cả các tập mục đem lại phần lợi ích cao cho nhà kinh doanh.

Trong bài này, chúng tôi đề xuất một thuật toán có khả năng tỉa một cách hiệu quả các ứng viên, phát hiện chính xác tập tất cả các tập mục lợi ích cao. Các kết quả thực nghiệm cho thấy thuật toán hoạt động rất hiệu quả cả về thời gian lẫn bộ nhớ, ngay cả đối với cơ sở dữ liệu lớn.

1. MỞ ĐẦU

Bài toán cơ bản (hay còn gọi bài toán nhị phân) khai phá luật kết hợp do R. Agrawal, T. Imielinski và A. N. Swami đề xuất và nghiên cứu lần đầu tiên vào năm 1993 [1, 2]. Mục tiêu của nó là phát hiện các tập mục thường xuyên, từ đó tạo các luật kết hợp. Cho đến nay, bài toán cơ bản khai phá luật kết hợp có nhiều ứng dụng, tuy vậy do tập mục thường xuyên chỉ mang ngữ nghĩa thống kê nên nó chỉ đáp ứng được phần nào nhu cầu ứng dụng thực tiễn, (xem chẳng hạn [3, 6, 15, 18, 19, 20]).

Gần đây, nhằm khắc phục hạn chế của bài toán cơ bản khai phá luật kết hợp, người ta đã mở rộng nó theo nhiều hướng khác nhau, xem [3, 4, 5, 6, 15, 18, 19, 20]. Trong [19, 20], H. Yao, H. J. Hamilton và cộng sự đã đề xuất bài toán khai phá tập mục lợi ích cao. Lợi ích của một tập mục là số đo lợi nhuận mà nó có thể mang lại trong kinh doanh, được tính toán dựa trên giá trị khách quan và giá trị chủ quan của các mục thành viên. Khai phá tập mục lợi ích cao là khám phá tất cả tập mục đem lại lợi ích không nhỏ hơn ngưỡng quy định bởi người sử dụng.

Trong khai phá luật kết hợp cơ bản, các thuật toán khám phá tập mục thường xuyên được xây dựng theo phương pháp tìm kiếm từng bước. Cơ sở của các thuật toán này là tính chất Apriori (hay còn gọi là tính chất phản đơn điệu - Anti monotone) của tập mục thường xuyên. Đáng tiếc là ràng buộc lợi ích cao không thỏa mãn tính chất Apriori. Do đó, việc rút

gọn không gian tìm kiếm, phát hiện tập mục lợi ích cao không thể thực hiện được như đối với khai phá tập mục thường xuyên. Trong [18] Hong Yao và Howard J. Hamilton. đã đề nghị hai thuật toán khám phá tập mục lợi ích cao. Đó là các thuật toán UMining và UMining-H. Các thuật tỉa mà hai thuật toán này áp dụng có khả năng thu gọn phần nào tập ứng viên, tuy vậy có những nhược điểm nên hiệu quả không cao.

Bài này trình bày một thuật toán hiệu quả khám phá tập mục lợi ích cao trong cơ sở dữ liệu lớn: thuật toán HUMining. Việc tỉa các tập mục ứng viên, thu gọn không gian tìm kiếm được thực hiện thông qua giá trị lợi ích kéo theo của chúng. Nội dung tiếp theo của bài báo gồm: Mục 2 nêu định nghĩa của một số thuật ngữ và phát biểu bài toán khai phá tập mục lợi ích cao. Mục 3 tóm tắt nội dung và nêu những nhược điểm của hai thuật toán UMining và UMining-H . Mục 4 trình bày thuật toán mới HUMining. Mục 5 là phần đánh giá thuật toán và kết luận dựa trên việc phân tích cấu trúc thuật toán và các tính toán thử nghiệm.

2. BÀI TOÁN KHAI PHÁ TẬP MỤC LỢI ÍCH CAO

Trước hết ta nêu định nghĩa của một số thuật ngữ (theo [18]).

Cho cơ sở dữ liệu giao tác T. Ký hiệu $I = \{i_1, ..., i_p, i_q, ..., i_m\}$ là tập tất cả các mục (thuộc tính) của T. Mỗi giao tác (bản ghi) t_q trong T là một tập con của I, được gán một định danh $\langle TID \rangle$. Một tập con của I, gồm k mục phân biệt được gọi là một k-tập mục, ký hiệu là S^k. Để đơn giản, đôi khi thay vì $\{i_1, ..., i_k\}$ ta viết $i_1, ..., i_k$; chẳng hạn tập mục $\{A, B, C, D\}$ được viết ngắn gọn là $ABCD$.

Định nghĩa 2.1. Ta gọi số đơn vị mục i_p có trong giao tác t_q (giá trị có sẵn trong cột i_p hàng t_q của cơ sở dữ liệu) là giá trị khách quan (objective value) của mục i_p tại giao tác t_q, ký hiệu bằng x_{pq}.

Định nghĩa 2.2. Ta gọi giá trị y_p do nhà kinh doanh gán cho mục i_p trong cơ sở dữ liệu, dựa trên đánh giá lợi nhuận mà mỗi đơn vị mục có thể đem lại, là giá trị chủ quan (subjective value) của mục i_p. Dĩ nhiên, nếu i_p được đánh giá cao hơn i_q thì $i_p > i_q$.

Thông thường, giá trị chủ quan của các mục được cho trong một bảng kèm theo cơ sở dữ liệu. Dưới đây là một ví dụ về cơ sở dữ liệu giao tác T (Bảng 1 và 2).

TID \ Mục	A	B	C	D	E
t_1	0	0	16	0	1
t_2	0	12	0	2	1
t_3	2	0	1	0	1
t_4	1	0	0	2	1
t_5	0	0	4	0	2
t_6	1	2	0	0	0
t_7	0	20	0	2	1
t_8	3	0	25	6	1
t_9	1	2	0	0	0
t_{10}	0	12	2	0	2

Bảng 1. Cơ sở dữ liệu giao tác

Mục	lợi nhuận ($/đơn vị)
A	3
B	5
C	1
D	3
E	5

Bảng 2. Giá trị lợi ích chủ quan của các mục trong Bảng 1.

Trong cơ sở dữ liệu này, ta có giá trị khách quan của mục D tại giao tác t_4 là $x_{44} = 2$, giá trị chủ quan của D bằng 3.

Khi đã có các giá trị khách quan và chủ quan, lợi ích của một mục trong một giao tác được đánh giá thông qua một hàm hai biến, gọi là hàm lợi ích $f(x, y)$.

Định nghĩa 2.3. Ký hiệu x là giá trị khách quan, y là giá trị chủ quan của một mục. Một hàm hai biến $f(x, y) : R \otimes R \to R$, đơn điệu tăng theo x và theo y, được gọi là hàm lợi ích.

Không mất tính tổng quát, có thể giả thiết hàm lợi ích là hàm không âm. Thông thường $f(x, y)$ được định nghĩa như sau: $f(x, y) = x \times y$.

Định nghĩa 2.4. Cho hàm lợi ích $f(x, y)$. Lợi ích của mục i_p tại giao tác t_q là giá trị của $f(x, y)$ tại x_{pq} và y_p, tức $f(x_{pq}, y_p)$.

Định nghĩa 2.5. Cho giao tác t_q chứa tập mục S. Giá trị lợi ích của S tại t_q, ký hiệu bằng $l(S, t_q)$, là tổng giá trị lợi ích tại giao tác t_q của tất cả các mục i_p thuộc S, tức là

$$l(S, t_q) = \sum_{i_p \in S \subseteq t_q} f(x_{pq}, y_p). \tag{2.1}$$

Ký hiệu T_S là tập của tất cả các giao tác chứa tập mục S, tức $T_S = \{t_q | S \subseteq t_q, t_q \in T\}$.

Định nghĩa 2.6. Cho tập mục $S \subseteq I$. Giá trị lợi ích của S trong T, ký hiệu bằng $u(S)$, là tổng các giá trị lợi ích của S tại tất cả các giao tác thuộc T_S, nghĩa là

$$u(S) = \sum_{t_q \in T_S} l(S, t_q). \tag{2.2}$$

Thay (2.1) vào (2.2), thu được

$$u(S) = \sum_{t_q \in T_S} \sum_{i_p \in S} f(x_{pq}, y_p) = \sum_{i_p \in S} \sum_{t_q \in T_S} f(x_{pq}, y_p). \tag{2.3}$$

Định nghĩa 2.7. Cho ngưỡng lợi ích *minutil* (> 0) và xét tập mục S. S được gọi là tập mục lợi ích cao nếu $u(S) > minutil$. Trường hợp ngược lại, S được gọi là tập mục lợi ích thấp.

Định nghĩa 2.8. Với ràng buộc lợi ích *minutil*, bài toán khai phá tập mục lợi ích cao là việc tìm tập H của tất cả các tập mục lợi ích cao, tức là tìm tập

$$H = \{S | S \subseteq I,\ u(S) \geq minutil\}.$$

Có thể coi bài toán cơ bản khai phá tập mục thường xuyên là trường hợp đặc biệt của bài toán khai phá tập mục lợi ích cao, trong đó tất cả các mục đều có giá trị khách quan bằng 0 hoặc 1 và giá trị chủ quan bằng 1.

Định nghĩa 2.8. cho thấy người sử dụng đóng vai trò quan trọng trong quá trình khai phá tập mục có tính đến lợi ích. Người sử dụng chính là người quyết định các tập mục cần khai phá thông qua việc ấn định ngưỡng *minutil* và hàm $f(x, y)$.

Tính chất cơ bản được khai thác để xây các thuật toán khai phá tập mục thường xuyên và luật mạnh là tính chất Apriori của độ hỗ trợ, xem [2, 7, 11]. Tính chất này được phát biểu như sau: Nếu một tập mục là thường xuyên thì mọi tập con khác rỗng của nó cũng là thường xuyên. Điều này có nghĩa các $(k+1)$-tập mục thường xuyên chỉ có thể sinh ra từ các k-tập mục thường xuyên. Dễ thấy, tính chất Apriori này không còn thoả mãn đối với ràng buộc lợi ích. (Ví dụ, trong cơ sở dữ liệu Bảng 1, ta có $u(BC) = 62 < 72 = u(BCE)$) trong khi đó $u(BC) = 62 > 0 = u(BCE)$. Do đó không thể áp dụng các thủ pháp hiện có phát hiện tập mục thường xuyên cho việc khám phá tập mục lợi ích cao.

Trong [18], H. Yao và H. J. Hamilton đã đề xuất hai thuật toán khai phá tập mục lợi ích cao, đó là các thuật toán UMining và UMining-H . Mục tiếp theo dưới đây trình bày nội dung cơ bản của hai thuật toán này cùng với một số nhận xét.

3. THUẬT TOÁN UMINING VÀ UMINING-H

3.1. Thuật toán UMining

Cơ sở lý thuyết của thuật toán UMining là các Định lý 3.1. và 3.2 sau đây (xem [18]).

Định lý 3.1. (Cận trên của lợi ích - Utility upper bound) *Với $k > 1$, đặt*

$$L^{k-1} = \{S^{k-1} / S^{k-1} \subset S^k\}.$$

Giá trị lợi ích $u(S^k)$ của k-tập mục S^k thoả mãn bất đẳng thức sau

$$u(S^k) \leq \frac{1}{k-1} \sum_{S^{k-1} \in L^{k-1}} u(S^{k-1}).$$

Định lý 3.2. *Cho k-tập mục S^k. Gọi C^{k-1} là tập các $(k-1)$-tập mục mà S^k có thể được tạo ra bằng cách kết nối hai tập mục thuộc C^{k-1}, (như vậy $C^{k-1} \subseteq L^{k-1}$). Đặt*

$$b(S^k) = \frac{1}{card(C^{k-1} - 1)} \sum_{S^{k-1} \in C^{k-1}} u(S^{k-1}). \tag{3.1}$$

Nếu mọi $(k-1)$-tập mục $S^{k-1} \in (L^{k-1} - C^{k-1})$ là tập mục lợi ích thấp và $b(S^k) < minutil$ thì S^k cũng là tập mục lợi ích thấp, tức là $u(S^k) < minutil$, $b(S^k)$ được gọi là giá trị lợi ích ước lượng của S^k.

Dựa vào Định lý 3.1. và 3.2., thuật toán UMining thực hiện việc phát hiện các tập mục lợi ích cao theo cách tiếp cận từng bước.

Bước 1. Tính lợi ích của tất cả các mục (1-tập mục) theo công thức (2.3), lưu các mục này cùng với lợi ích của nó vào tập C_1. Tuyển chọn các mục lợi ích cao nạp vào tập H.

Bước 2. Kết nối các cặp mục trong C_1 tạo thành các 2-tập mục, lưu vào C_2. Rút gọn C_2 thành C_2^* bằng cách loại bỏ các 2-tập mục có giá trị tính theo (3.1) nhỏ hơn ngưỡng $minutil$. Tập C_2^* sẽ bao gồm các ứng viên cho 2-tập mục lợi ích cao và được sử dụng để tạo các 3-tập mục ở bước thứ 3. Tính giá trị lợi ích thực của các mục trong C_2^* theo (2.3), tuyển chọn các tập mục lợi ích cao lưu vào H.

Các bước $k = 3, 4, ...$ tiếp theo tương tự như Bước 2.

Thuật toán dừng khi không còn tập ứng viên nào nữa, (tức khi $C_k = \phi$), hoặc khi đã đạt đến bước K định trước bởi người sử dụng. Việc định trước số bước K cho phép điều chỉnh thời gian thực hiện chương trình.

Ví dụ, xét cơ sở dữ liệu Bảng 1. Giả sử $minutil = 130$. Thuật toán UMining cho:

$C_1 = \{A, B, C, D, E\}$, $C_2 = \{AB, AC, AD, AE, BC, BD, BE, CD, CE, DE\}$,

$C_2^* = \{AB, BC, BD, BE\}$, $C_3 = \{ABC, ABD, ABE, BCD, BCE, BDE\}$,

$C_3^* = \{ABD, ABE, BCD, BCE, BDE\}$,

$C_4 = \{ABCD, ABCE, ABDE, BCDE\}$, $C_4^* = \phi$, $C_5 = \phi$.

Không gian tìm kiếm gồm 25 tập mục $C_1 \cup C_2 \cup C_3 \cup C_4$ và $H = \{B, BD, BE, BDE\}$.

Có thể thấy, nếu T chứa nhiều mục lợi ích thấp, khi đó sẽ có nhiều tập mục bị loại khỏi không gian tìm kiếm, thuật toán UMining hoạt động hiệu quả. Ngược lại, khi tất cả các tập

mục của T đều là tập mục lợi ích cao, thuật toán phải kiểm tra mọi tổ hợp mục có thể, UMining hoạt động không hiệu quả.

3.2. Thuật toán UMining-H

Thuật toán UMining-H cũng là thuật toán từng bước [18]. Các bước của thuật toán UMining-H cũng như trong thuật toán UMining. Điểm khác ở đây là, từ bước $k \geq 2$ trở đi, thay vì sử dụng ước lượng $b(S^k)$ tính theo (3.1) để tỉa các mục ứng viên trong C_k, UMining-H sử dụng giá trị dự đoán $b^*(S^k)$ dưới đây cho lợi ích của S^k:

$$b^*(S^k) = \frac{supp_{\min}}{card(C^{k-1}) - 1} \sum_{S^{k-1} \in C^{k-1}} \frac{u(S^{k-1})}{supp(S^{k-1})} \tag{3.2}$$

trong đó $supp(S^k)$ là độ hỗ trợ của S^k và $supp_{\min} = \min_{S^{k-1} \in C^{k-1}} \{supp(S^{k-1})\}$. $b^*(S^k)$ là giá trị lợi ích dự đoán Heuristic dựa vào Định lý 3.1, 3.2. và tính chất Apriori của độ hỗ trợ

$$supp(S^k) \geq \min_{S^{k-1} \in C^{k-1}} \{supp(S^{k-1})\}.$$

Do đó UMining-H là thuật toán Heuristic.

Ví dụ, xét cơ sở dữ liệu Bảng 1. Giả sử $minutil = 130$. Thuật toán Umining-H cho:
$C_1 = \{A, B, C, D, E\}$, $C_2 = \{AB, AC, AD, AE, BC, BD, BE, CD, CE, DE\}$,
$C_2^* = \{AB, BC, BD, BE\}$, $C_3 = \{ABC, ABD, ABE, BCD, BCE, BDE\}$,
$C_3^* = \{BCD, BCE, BDE\}$, $C_4 = \{BCDE\}$, $C_4^* = \phi$, $C_5 = \phi$.
Không gian tìm kiếm gồm 22 mục $C_1 \cup C_2 \cup C_3 \cup C_4$ và $H = \{B, BD, BE, BDE\}$.

Nhược điểm của UMining-H là nó có thể bỏ sót một số mục lợi ích cao. Việc này xảy ra khi một mục lợi ích cao S lại chứa một mục con lợi ích thấp và độ hỗ trợ của mục con này nhỏ hơn rất nhiều so với độ hỗ trợ của những mục con khác của S (xem [18]).

Nhằm khắc phục những hạn chế của UMining và UMining-H, chúng tôi đề nghị một thuật toán mới khai phá mục lợi ích cao, gọi là thuật toán HUMining.

4. THUẬT TOÁN HUMINING

4.1. Cơ sở lý thuyết

Định nghĩa 4.1. Ta gọi tổng lợi ích của tất cả các mục có mặt trong t_q là lợi ích của giao tác t_q. Ký hiệu lợi ích của giao tác t_q bằng $tu(t_q)$, ta có:

$$tu(t_q) = \sum_{i_p \in t_q} f(x_{pq}, y_p) \tag{4.1}$$

Ví dụ, cho cơ sở dữ liệu Bảng 1 và Bảng 2, ta có $tu(t_3) = 2 \times 3 + 1 \times 1 + 1 \times 5 = 12$.

Định nghĩa 4.2. Cho mục S và T_S là tập tất cả các giao tác chứa S. Ta gọi tổng lợi ích của tất cả các giao tác trong T_S là lợi ích kéo theo của S. Ký hiệu lợi ích kéo theo của S là $wu(S)$, ta có:

$$wu(S) = tu(T_S) = \sum_{t_q \in T_S} tu(t_q) = \sum_{t_q \in T_S} \sum_{i_p \in t_q} f(x_{pq}, y_p). \tag{4.2}$$

Ví dụ, xét dữ liệu cho trong Bảng 1 và 2. Với $S = A$, ta có $T_A = \{t_3, t_4, t_6, t_8, t_9\}$, $wu(\{A\}) = tu(t_3) + tu(t_4) + tu(t_6) + tu(t_8) + tu(t_9) = 12 + 14 + 13 + 57 + 13 = 109$.

Với $S = AD$ thì $T_{AD} = \{t_4, t_8\}$, $wu(\{A, D\}) = tu(t_4) + tu(t_8) = 14 + 57 = 71$.

Định nghĩa 4.4. Cho ngưỡng $\varepsilon > 0$. Tập mục S được gọi là tập mục có lợi ích kéo theo cao theo ngưỡng ε nếu $wu(S) \geq \varepsilon$.

Định lý 4.1. (Tính chất Apriori của lợi ích kéo theo) *Cho S^k là một k-tập mục, S^{k-1} là một $(k-1)$-tập mục con của S^k ($S^{k-1} \subset S^k$). Nếu S^k có lợi ích kéo theo cao thì S^{k-1} cũng có lợi ích kéo theo cao.*

Chứng minh. Vì S^k ($S^{k-1} \subset S^k$), nên $T_{S^k} \subseteq T_{S^{k-1}}$. Theo (4.2)

$$wu(S^{k-1}) = \sum_{t_q \in T_{S^{k-1}}} tu(t_q) \geq \sum_{t_q \in T_{S^k}} tu(t_q) = wu(S^k).$$

Suy ra, nếu $wu(S^k) \geq \varepsilon$ thì $wu(S^{k-1}) \geq \varepsilon$.

Nhận xét 4.1. Tính chất Apriori của lợi ích kéo theo có nghĩa là nếu một $k-$tập mục S^k có chứa tập mục con S^{k-1} mà S^{k-1} là tập mục có lợi ích kéo theo thấp thì S^k cũng sẽ là tập mục có lợi ích kéo theo thấp. Từ đó suy ra, để phát hiện tập mục có lợi ích kéo theo cao ta có thể tiến hành từng bước theo độ dài từ nhỏ đến lớn của các tập mục. Tại bước $k = 2, 3, ...$ các ứng viên chỉ có thể là các kết nối của các $(k-1)$-tập mục có lợi ích kéo theo cao.

Định lý 4.2. *Ký hiệu WH là lớp tất cả các tập mục X có lợi ích kéo theo cao, H là lớp tất cả các tập mục lợi ích cao. Nếu chọn $\varepsilon = minutil$ thì $H \subseteq WH$.*

Chứng minh. Với mọi tập mục S và $t_q \in T_S$, đều có $S \subseteq t_q$. Do đó

$$u(S) = \sum_{t_q \in T_S} \sum_{i_p \in S} f(x_{pq}, y_p) \leq \sum_{t_q \subseteq T_S} \sum_{i_p \in t_q} f(x_{pq}, y_p) = wu(S).$$

Vậy nếu $u(S) \geq minutil = \varepsilon$ thì $u(S) \geq \varepsilon = minutil$, tức là nếu $S \in H$ thì $S \in WH$.

Nhận xét 4.2. Từ Định lý 4.2 suy ra, nếu $\varepsilon = minutil$ thì để tìm các tập mục lợi ích cao ta chỉ cần tìm trong số các tập mục có lợi ích kéo theo cao.

4.2. Thuật toán HUMining

Kết hợp Nhận xét 4.1 và 4.2 trên đây, ta có thuật toán sau đây phát hiện tập mục lợi ích cao trong cơ sở dữ liệu cỡ lớn. Thuật toán gồm hai công đoạn:

Công đoạn 1. Tìm tất cả các tập mục có lợi ích kéo theo cao với ngưỡng $\varepsilon = minutil$.

Công đoạn 2. Phát hiện tất cả các tập mục lợi ích cao trong số các tập mục có lợi ích kéo theo cao.

Có thể thấy, công đoạn 1 là công đoạn chính, tiêu tốn phần lớn thời gian trong tiến trình khai phá tập mục lợi ích cao. Vì ràng buộc lợi ích kéo theo thỏa mãn tính chất Apriori, để phát hiện tập mục có lợi ích kéo theo cao, ta có thể sử dụng bất kỳ thuật toán nào trong số các thuật toán khai phá tập mục thường xuyên xây dựng dựa trên tính chất Apriori, chỉ cần thay việc đếm độ hỗ trợ của các tập mục bằng việc tính lợi ích kéo theo của chúng. (Hiện nay đã có rất nhiều thuật toán hiệu quả khai phá tập mục thường xuyên được xây dựng dựa trên tính chất Apriori, xem [2, 7, 11, 21]).

Vì tất cả các bước phát hiện các k-tập mục có lợi ích kéo theo cao đều cần sử dụng lợi ích của các giao tác, việc làm đầu tiên mà HUMining thực hiện là duyệt toàn bộ cơ sở dữ liệu, tính lợi ích của mỗi giao tác. Ta giả thiết, mỗi giao tác trong cơ sở dữ liệu T được gán thêm một trường, ký hiệu là $tu(t)$, để lưu giá trị lợi ích của nó.

Khi đã khám phá được tập tất cả các tập mục có lợi ích kéo theo cao thì việc phát hiện các tập mục lợi ích cao trong số các tập mục này là rất đơn giản, chỉ cần quét cơ sở dữ liệu một lần duy nhất.

Dưới đây là thuật toán khai phá tập mục lợi ích cao được xây dựng với phần tìm tập các tập mục có lợi ích kéo theo cao dựa theo thuật toán Apriori trong khai phá tập mục thường xuyên (xem [2, 7]).

Thuật toán HUmining Tìm tất cả các tập mục lợi ích cao với ngưỡng *minutil*.
Input: Cơ sở dữ liệu giao tác T, hàm lợi ích f, ngưỡng lợi ích *minutil*.
Output: Tập H bao gồm các tập mục lợi ích cao.
Method:

1. for mỗi giao tác $t \in T$
2. Tính lợi ích $tu(t)$ theo công thức (4.1);
3. end for;

// Công đoạn 1: Phát hiện các tập mục có lợi ích kéo theo cao.

4. $k = 1$;
5. $WH_k = \{i/i \in I\}$ và $wu(i) \geq minutil\}$;// Phát hiện 1-tập mục có lợi ích kéo theo cao.
6. repeat
7. $k = k + 1$;
8. $C_k = apriori\_gen(WH_{k-1})$;//Tạo các tập mục ứng viên.
9. for mỗi giao tác $t \in T$ do
10. $C_t = subset(C_k, t)$; //Nhận diện các ứng viên thuộc t.
11. for mỗi ứng viên $c \in C_t$ do
12. $wu(c) = wu(c) + tu(t)$; //Cộng thêm lợi ích của giao tác t vào lợi ích kéo theo của c.
13. end for;
14. end for;
15. $WH_k = \{c/c \in C_k \wedge wu(c) \geq minutil\}$; //Tuyển chọn các k-tập mục có lợi ích kéo theo cao.
16. until $C_k \neq \phi$;
17. $WH = \cup WH_k$;

// Công đoạn 2: Phát hiện tập mục lợi ích cao.

18. for mỗi giao tác $t \in T$ do
19. $C_t = subset(WH, t)$;//Nhận diện các tập mục có lợi ích kéo theo cao thuộc t.
20. For mỗi ứng viên $s \in C_t$ do
21. $u(s) = u(s) + u(s, t)$; //Cộng thêm lợi ích của c tại giao tác t vào lợi ích của c.
22. end for;
23. end for;
24. $H = \{s/s \in WH \wedge u(s) \geq minutil\}$;//Lọc các tập mục lợi ích cao.

Thuật toán HUMining sử dụng hàm $apriori\_gen(WH_{k-1})$ tại bước 8. Hàm này hoàn toàn giống như hàm $Apriori\_gen(F_{k-1})$ trong thuật toán *Apriori* khai phá tập mục thường xuyên. Nó tạo ra các k-tập mục ứng viên tại bước k bằng cách kết nối các $(k-1)$-tập mục có lợi ích kéo theo cao đã tìm được ở bước $k-1$, với giả thiết các tập mục đã được sắp thứ tự từ điển, (xem [2]).

Ví dụ, cho cơ sở dữ liệu giao tác Bảng 1. Giả sử giá trị chủ quan của các mục cho trong Bảng 2, hàm lợi ích $f(x, y) = x \times y$ và $minutil = 130$. Thuật toán HUMining thực hiện việc phát hiện các tập mục lợi ích cao trong cơ sở dữ liệu này như sau:

Các dòng lệnh 1-3 cho $tu(t_1) = 21$, $tu(t_2) = 71$, $tu(t_3) = 12$, $tu(t_4) = 14$, $tu(t_5) = 14$, $tu(t_6) = 13$, $tu(t_7) = 111$, $tu(t_8) = 57$, $tu(t_9) = 13$, $tu(t_{10}) = 72$.

Công đoạn 1. Các dòng lệnh 4-16 cho:

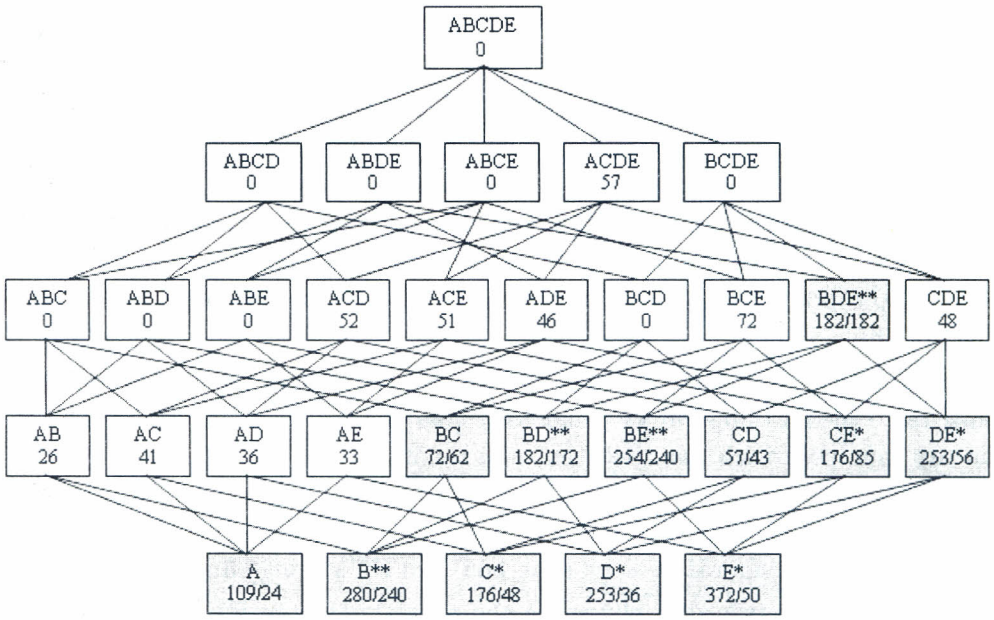
Bước $k = 1, C_1 = \{A, B, C, D, E\}, wu(A) = 109, wu(B) = 280, wu(C) = 176, wu(D) = 253, wu(E) = 372, WH_1 = \{B, C, D, E\}$.

Bước $k = 2, C_2 = \{BC, BD, BE, CD, CE, DE\}, wu(BC) = 72, wu(BD) = 182, wu(BE) = 254, wu(CD) = 57, wu(CE) = 176, wu(DE) = 253, WH_2 = \{BD, BE, CE, DE\}$.

Bước $k = 3, C_3 = \{BDE\}, wu(BDE) = 182, WH_3 = \{BDE\}$.

Công đoạn 1 kết thúc chỉ sau 3 bước, ($C_4 = \phi$). Ta có

$$WH = \{B, C, D, E, BD, BE, CE, DE, BDE\}.$$



Hình 1

Hình 1 minh họa không gian tìm kiếm tập mục lợi ích cao theo thuật toán HUMining (gồm 12 tập mục tô đen) trong dàn tập mục của cơ sở dữ liệu Bảng 1, với $minutil = 130$. Số ghi trong ô là lợi ích của tập mục hoặc lợi ích kéo theo / lợi ích của tập mục. Tập mục đánh dấu một hoặc hai sao là tập mục có lợi ích kéo theo cao, trong đó tập mục hai sao là tập mục lợi ích cao tìm được.

Công đoạn 2. Các dòng lệnh 18-23 cho $u(B) = 240$, $u(C) = 48$,$u(D) = 36$, $u(E) = 50$, $u(BD) = 172$,$u(BE) = 240$, $u(CE) = 83$, $u(DE) = 56$, $u(BDE) = 182$.

Dòng lệnh 24 cho $H = \{B, BD, BE, BDE\}$.

5. ĐÁNH GIÁ THUẬT TOÁN VÀ KẾT LUẬN

Chúng tôi đã tiến hành thử nghiệm thuật toán HUMining trên một số cơ sở dữ liệu giao tác tạo được bằng phương pháp tạo ma trận số ngẫu nhiên. Căn cứ vào các kết quả thử nghiệm và việc phân tích cấu trúc thuật toán, chúng tôi nhận thấy HUMining có những ưu điểm sau đây so với UMining và UMining-H.

1. Thu gọn một cách đáng kể không gian tìm kiếm. Hình vẽ trên đây minh hoạ không gian tìm kiếm tập mục lợi ích cao trong cơ sở dữ liệu bảng 1 theo thuật toán HUMining. Có thể thấy, ở đây không gian tìm kiếm chỉ gồm 12/31 tập mục con của I.

2. Khi *minutil* lớn, không gian tìm kiếm được thu gọn ngay từ bước 2 . Ví dụ, đối với cơ sở dữ liệu bảng 1, tại bước 2 đã có 4 tập mục chứa A bị loại khỏi không gian tìm kiếm do A không phải là mục có lợi ích kéo theo cao, trong khi các tập mục này hoàn toàn không bị loại bởi thuật tìa trong UMining và UMining-H. Với những cơ sở dữ liệu cỡ lớn hơn, có thể hy vọng HUMining sẽ thu gọn không gian tìm kiếm nhiều hơn.

3. Căn cứ vào Định lý 4.2, nếu chọn $\varepsilon = minutil$ thì tập tất cả các tập mục lợi ích cao là tập con của tập các tập mục có lợi ích kéo theo cao. Do đó, sẽ không có tập mục lợi ích cao nào bị bỏ sót khi thuật toán kết thúc.

4. Trong quá trình thực hiện thuật toán, tuy phải tính toán nhiều lần lợi ích kéo theo của các tập mục theo công thức (4.2), nhưng để xác định các giá trị này, phép toán chủ yếu phải thực hiện là phép cộng. Do đó, tổng thời gian tính nhỏ hơn nhiều so với tổng thời gian cần thiết để tính các giá trị lợi ích ước lượng khi áp dụng thuật toán UMining hoặc UMining-H .

5. Để thực hiện công đoạn 2 (phát hiện các tập mục lợi ích cao trong số các tập mục có lợi ích kéo theo cao), ta chỉ cần quét tập dữ liệu đúng một lần. Hơn nữa, nếu chọn ngưỡng *minutil* lớn , số tập mục có lợi ích kéo theo cao sẽ ít, thời gian dành cho việc thực hiện công đoạn 2 sẽ rất nhỏ.

6. Cuối cùng, nhưng lại là một ưu điểm nổi bật của HUMining là, với chỉ một vài thay đổi nhỏ, chúng ta có thể sử dụng bất kỳ thuật toán khai phá tập mục thường xuyên nào vào việc khai phá tập mục có lợi ích kéo theo cao, nếu thuật toán đó được xây dựng dựa trên tính chất Apriori của tập mục thường xuyên. Như đã biết, hiện nay có rất nhiều thuật toán hiệu quả khai phá tập mục thường xuyên, (xem [2, 7, 11, 21, 22]). Hơn nữa, vì các thuật toán khai phá tập mục thường xuyên rất dễ song song hóa, nên HUMining cũng là thuật toán như thế.

Với những ưu điểm trên đây, chúng tôi cho rằng thuật toán HUMining hiệu quả hơn so với hai thuật toán UMining và UMining-H trong [18].

TÀI LIỆU THAM KHẢO

[1] R. Agrawal, T. Imielinski, and A. N. Swami, Mining association rules between sets of items in large databases, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, Washington, D.C., 1993.

[2] R. Agrawal and R. Srikant, Fast algorithms for mining association rules, *Proceedings of 20th International Conference on Very Large Databases*, Santiago, Chile, 1994.

[3] Y. Aumann, and Y. Lindell, A statistical theory for quantitative association rules, *Proc. of the 5th KDD*, San Diego, Canada, 1999.

[4] R. J. Bayardo, R. Agrawal, and D. Gunopulos, Constraint based rule mining in large dense databases, *Proceedings of the 15rd International Conference on Data Engineering*, Sydney, Australia, 1999.

[5] C. H. Cai, A. W. C. Fu, C. H. Cheng, and W. W. Kwong, Mining rules with weighted items, *Proceedings of IEEE International Database Engeneering and Applications Symposium*, Cardiff, United Kingdom, 1998.

[6] R. Chan, Q. Yang, and Y. D. Shen, Mining high utility itemsets, *Proceedings of 3rd IEEE International Conference on Data Mining*, Melbourne, Florida, 2003.

[7] B. Goethals, "Survey on Frequent Parttern Mining", Technical Report, Helsinki, Institute for Information Technology, 2003.

[8] IBM Synthetic data, http://www.almaden.ibm.com/software/quest/Resources/index.shtml, 2004.

[9] T. Y. Lin, Y. Y. Yao, and E. Louie, Value added association rules, *Advances in Knowledge Discovery and Data Mining*, 6th Pacific-Asia Conference, Taipei, 2002.

[10] S. Lu, H. Hu, and F. Li, Mining weighted association rules, *Intelligent Data Analysis* **5** (3) (2001).

[11] H. Mannila, and H. Toivonen, and A. I. Verkamo, Efficient algorithms for discovering association rules, *AAAI Workshop on Knowledge Discovery in Databases (KDD'94)*, Seattle, 1994.

[12] G. D. Ramkumar, S. Ranka, and S. Tsur, Weighted Association Rules: Model and Algorithm, http://www.cs.ucla.edu/'czdemo/tsur/, 1997.

[13] Y. D. Shen, Q. Yang, and Z. Zhang, Objective-oriented utility-based association mining, *Proceedings of the 13th ACM SIGMOD International Conference on Knowledge Discovery and Data mining*, San Jose, California, USA, 2007.

[14] R. Srikant, and R. Agrawal, Mining generalized association rules, *Proceedings of 21th International Conference on Very Large Databases*, Databases, Zurich, Switzerland, 1995.

[15] R. Srikant, and R. Agrawal, Mining quantitative association rules in large rational tables, *Proc. of 1996 ACM-SIGMOD, International Conference on Management of Data*, Montreal, Canada, 1996.

[16] R. Srikant, Q. Vu, and R. Agrawal, Mining Association with Item Constraints, *Proc. of the 3rd International Conference on Knowledge Discovery and Data Mining*, Newport Beach, Canada, 1997.

[17] F. Tao, F. Murtagh, and M. Farid, Weighted association rule mining using weighted support and significance framework, *Proc. of International Conference on Knowledge Discovery and Data mining*, Washington DC, USA, 2003.

[18] H. Yao, H. J. Hamilton, Mining itemsets utilities from transaction databases, *Data and Knowledge Engeneering* **59**(3) (2006).

[19] H. Yao, H. J. Hamilton, and C. J. Butz, A foundational approach to mining itemset utilities from databases, *Proceedings of the 4th SIAM International Conference on Data Mining*, Florida, USA, 2004.

[20] H. Yao, H. J. Hamilton, and L. Geng, A unified framework for utility based measures for mining itemsets, *UBDM'06* Philadelphia, Pennsylvania, USA, August 2006.

[21] Q. L. Zhao, and S. S. Bhowmik, "Association Rules Mining: A Survey", Technical Report, CAIS, Nanyang Technological University, Singapore, No. 2003116, 2003.

[22] M. J. Zaki, Parallel and distributed association mining: A survey, *IEEE concurrency, Special Issue on Parallel Mechanisms for Data Mining* **7** (4) (December 1999).

Nhận bài ngày 29 - 11 - 2007
Nhận lại sau sửa ngày 18 - 3 -2008