Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Đà Nẵng, ngày 17-18/08/2017
DOI: 10.15625/vap.2017.000108

THUẬT TOÁN KHAI THÁC TẬP PHỔ BIẾN TRÊN DỮ LIỆU GIAO DỊCH VỚI NHIỀU NGƯỠNG PHỔ BIẾN TỐI THIỂU

Phan Thành Huấn[1], Lê Hoài Bắc[2]

[1]Bộ môn Tin học, Trường Đại học Khoa học Xã hội và Nhân văn, Đại học Quốc gia Tp. Hồ Chí Minh
[2]Khoa Công nghệ thông tin, Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Tp. Hồ Chí Minh

huanphan@hcmussh.edu.vn, lhbac@fithcmus.edu.vn

TÓM TẮT: *Trong khai thác dữ liệu, kỹ thuật quan trọng và được nghiên cứu nhiều là khai thác luật kết hợp. Khai thác tập phổ biến là một trong những bước cơ bản và chiếm nhiều thời gian trong khai thác luật kết hợp. Hầu hết các thuật toán tìm tập phổ biến thỏa một ngưỡng phổ biến tối thiểu duy nhất. Trong thực tế, độ phổ biến của từng mục hàng phản ánh bản chất, vai trò của mục hàng trong các giao dịch. Trong bài viết này, chúng tôi đề xuất thuật toán hiệu quả khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu (mỗi mục hàng có một ngưỡng phổ biến tối thiểu riêng). Sau cùng, chúng tôi trình bày kết quả thực nghiệm trên bộ dữ liệu thực và giả lập, cho thấy thuật toán đề xuất hiệu quả hơn so với thuật toán hiện hành.*

Từ khóa: *Luật kết hợp, nhiều ngưỡng phổ biến tối thiểu, tập phổ biến.*

I. GIỚI THIỆU

Khai thác luật kết hợp là một kỹ thuật quan trọng trong lĩnh vực khai thác dữ liệu. Mục tiêu khai thác là phát hiện những mối liên hệ giữa các giá trị dữ liệu trong dữ liệu giao dịch. Mô hình đầu tiên của bài toán khai thác luật kết hợp là mô hình nhị phân hay còn gọi là mô hình cơ bản được Agrawal và đồng sự đề xuất vào năm 1993 [1], phân tích dữ liệu giao dịch, phát hiện các mối liên hệ giữa các tập mục hàng hoá đã bán được tại các siêu thị. Từ đó có kế hoạch bố trí, sắp xếp, kinh doanh hợp lý, đồng thời tổ chức sắp xếp các quầy gần nhau như thế nào để có doanh thu trong các phiên giao dịch là lớn nhất.

Bài toán khai thác luật kết hợp là khai phá các luật kết hợp có độ phổ biến (*support*) cũng như độ tin cậy (*confidence*) lớn hơn hoặc bằng một ngưỡng phổ biến tối thiểu (*minsup*) và ngưỡng tin cậy tối thiểu (*minconf*).

Các thuật toán được đề xuất để khai thác luật kết hợp chia thành 2 giai đoạn [1, 2, 3]:

Giai đoạn 1: Tìm tất cả các tập mục phổ biến từ dữ liệu giao dịch thoả *minsup*;

Giai đoạn 2: Sinh các luật tin cậy kết hợp từ các tập mục phổ biến tìm thấy ở giai đoạn thứ nhất.

Giai đoạn *thứ nhất* chiếm hầu hết thời gian cho toàn quá trình khai thác luật kết hợp [1, 2, 3]. Giá trị ngưỡng phổ biến tối thiểu *minsup* là yếu tố quan trọng trong quá trình rút gọn không gian tìm kiếm cũng như giới hạn các luật sinh trong giai đoạn *thứ hai*. Các thuật toán khai thác luật kết hợp truyền thống chỉ dùng **một** giá trị ngưỡng phổ biến tối thiểu *minsup* với ngầm định là các mục hàng có cùng tính chất và tần số trong dữ liệu, điều này không thực tế. Trong kinh doanh bán lẻ, thông thường các mặt hàng thiết yếu, hàng tiêu dùng và các sản phẩm giá rẻ được mua nhiều hơn, trong khi các mặt hàng xa xỉ và các sản phẩm giá trị cao lại ít được mua. Nếu chọn *minsup* quá cao thì các mặt hàng được khai thác thông thường có giá thành thấp và mang lại lợi nhuận không cao cho doanh nghiệp. Ngược lại, nếu chọn *minsup* quá thấp thì các mặt hàng được khai thác quá lớn, điều này làm cho doanh nghiệp khó khăn khi ra quyết định kinh doanh. Vì vậy, Liu và các đồng sự [6] vào năm 1999 đã mở rộng bài toán khai thác luật kết hợp với nhiều ngưỡng phổ biến tối thiểu (*mỗi mục hàng có một ngưỡng phổ biến tối thiểu riêng*) tương ứng mỗi mục hàng khác nhau có tính chất khác nhau và tần số giao dịch khác nhau. Nhóm tác giả này đã đề xuất thuật toán **MSApriori** – khai thác luật kết hợp khác nhau thỏa ngưỡng phổ biến tối thiểu khác nhau phụ thuộc vào các mục hàng có trong luật.

Một số thuật toán điển hình khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu [6, 7, 8]:

Thuật toán MSApriori: Liu và các đồng sự [6] đề xuất khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu khác nhau. Thuật toán này sử dụng phương pháp tiếp cận tựa **Apriori** [1] và tính chất bao đóng được sắp xếp theo item để giảm không gian tìm kiếm, chi phí tính toán.

Thuật toán CFP-growth: Hu và các đồng sự [7] đề xuất hướng tiếp cận tựa thuật toán **FP-growth** trong khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu được gọi là **CFP-growth**. Thuật toán sử dụng cấu trúc **MIS-tree** tựa **FP-tree** [3] để lưu trữ thông tin quan trọng các mẫu phổ biến. Thuật toán khai thác đầy đủ tập phổ biến với một lần quét dữ liệu. Thuật toán này tương đối tốt hơn so với **MSApriori**.

Thuật toán CFP-growth++: Kiran và các đồng sự [8] đề xuất cải tiến thuật toán **CFP-growth** bằng cách rút gọn không gian tìm kiếm và xây dựng *MIS-tree nhỏ gọn* dựa trên *MIS-tree* [7]. Thuật toán đề xuất bốn kỹ thuật rút gọn không gian tìm kiếm: *ngưỡng phổ biến tối thiểu thấp nhất*, *ngưỡng phổ biến tối thiểu có điều kiện*, *tính chất bao đóng có điều kiện* và *tỉa các nút lá không phổ biến*. Thuật toán cải thiện hiệu suất đáng kể so với thuật toán **CFP-growth**.

Các thuật toán trên [6, 7, 8] chưa đáp ứng thực tế, khi cần khai thác luật kết hợp thì người dùng có thể yêu cầu thực hiện khai thác luật kết hợp thỏa nhiều ngưỡng phổ biến tối thiểu trong nhiều chuỗi thao tác liên tiếp khác nhau. Để đáp ứng thực tế, nhóm tác giả đề xuất thuật toán **MMS-FI** khai thác nhanh tập phổ biến từ mảng chứa các *itemset* đồng xuất hiện và không đọc lại dữ liệu cho lần khai thác tiếp theo, bao gồm các thuật toán con sau:

- Xây dựng mảng **Index_COOC** chứa *itemset* đồng xuất hiện và *itemset* xuất hiện ít nhất trong một giao dịch của từng *item* hạt nhân;
- Thuật toán **MMS-FI** khai thác hiệu quả tập phổ biến với nhiều ngưỡng phổ biến tối thiểu dựa trên mảng **Index_COOC**.

Trong phần II, bài báo trình bày các khái niệm cơ bản về khai thác tập phổ biến truyền thống, tập phổ biến với nhiều ngưỡng phổ biến tối thiểu. Phần III, xây dựng thuật toán xác định mảng chứa *itemset* đồng xuất hiện và *itemset* xuất hiện *ít nhất trong một giao dịch* của từng *item* hạt nhân và thuật toán **MMS-FI** khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu. Kết quả thực nghiệm được trình bày trong phần IV và kết luận ở phần V.

II. CÁC VẤN ĐỀ LIÊN QUAN

A. Khai thác tập phổ biến truyền thống

Khai thác tập phổ biến truyền thống là các thuật toán [1, 2, 3] chỉ dùng duy nhất ***một*** giá trị ngưỡng phổ biến tối thiểu *minsup* với ngầm định là các mục hàng có cùng tính chất và độ phổ biến trong dữ liệu. Các hạn chế khi khai thác tập phổ biến truyền thống: giá trị *minsup* cao thì các tập mục hiếm bị bỏ qua hoặc khi giá trị *minsup* thấp thì sinh tập mục phổ biến quá lớn. Sau đây là các khái niệm liên quan:

Cho $I = \{i_1, i_2, ..., i_m\}$ là tập gồm m mục hàng riêng biệt, mỗi mục hàng gọi là *item*. Tập các mục $X = \{i_1, i_2, ... i_k\}, \forall i_j \in I (1 \le j \le k)$ gọi là *itemset*, tập mục có k mục gọi là *k-itemset*. $\mathcal{D}$ là dữ liệu giao dịch, gồm n bản ghi phân biệt gọi là tập các giao dịch T = $\{t_1, t_2, ..., t_n\}$, mỗi giao dịch $t_i = \{i_{k_1}, i_{k_2}, ... i_{k_j}\}, i_{k_j} \in I (1 \le k_j \le m)$.

Định nghĩa 1: *Độ phổ biến* (support) của *itemset* $X \subseteq I$, ký hiệu *sup(X)*, là số các giao dịch trong $\mathcal{D}$ có chứa *X*.

Định nghĩa 2: Cho $X \subseteq I$, *X* gọi là *itemset* phổ biến nếu *sup(X)* ≥ *minsup*, trong đó *minsup* là ngưỡng phổ biến tối thiểu. Ký hiệu **FI** là tập hợp các tập mục phổ biến.

Tính chất 1: $\forall X \subseteq Y$: *sup(Y)* ≥ *minsup* $\Rightarrow$ *sup(X)* ≥ *minsup*;

Tính chất 2: $\forall X \subset Y$: *sup(X)* < *minsup* $\Rightarrow$ *sup(Y)* < *minsup*;

Cho dữ liệu giao dịch $\mathcal{D}$ trong Bảng 1.

Bảng 1. Dữ liệu giao dịch D

Mã giao dịch	Tập mục
t1	A C E F
t2	A C G
t3	E H
t4	A C D F G
t5	A C E G
t6	E
t7	A B C E
t8	A C D
t9	A B C E G
t10	A C E F G

Ví dụ 1: Dữ liệu giao dịch $\mathcal{D}$ trong *Bảng 1*, có 8 *item* riêng biệt *I* = {*A, B, C, D, E, F, G, H*} và 10 giao dịch T = {*t1, t2, t3, t4, t5, t6, t7, t8, t9, t10*} với giá trị ngưỡng *minsup* = 2, ta có:

Tập mục *X* ={A, C, E}, *sup*(ACE) = 5 ≥ *minsup*, ta nói: "*X* ={ACE} phổ biến theo ngưỡng *minsup* = 2";

Theo ***tính chất 1*** thì các tập con của *X* ={ACE} cũng phổ biến, nghĩa là tất cả tập con của *X* đều phổ biến – *sup*(A) = 8, *sup*(C) = 8, *sup*(E) = 7, *sup*(AC) , *sup*(AE) = 5, *sup*(CE) = 5 ≥ *minsup*.

Tương tự, với *Y* = {H} thì *sup*(H) = 1 < *minsup*, ta nói: "*Y* = {H} *không phổ biến* theo ngưỡng *minsup* = 2";

Theo ***tính chất 2*** thì các tập cha của *Y* ={H} cũng *không phổ biến*, nghĩa là *Y* = {EH} cũng *không phổ biến*, với *sup*(EH) = 1 < *minsup* = 2.

B. Khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu

Trong thực tế, hầu hết dữ liệu giao dịch đều không đồng nhất về tính chất của từng mục hàng, cũng như tần số giao dịch của các mục hàng. Các tác giả đã đề xuất thuật toán [6, 7, 8] khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu của từng mục hàng. Các thuật toán này khai thác luật kết hợp khác nhau thỏa ngưỡng phổ biến tối thiểu khác nhau phụ thuộc vào ngưỡng phổ biến của các mục hàng có trong luật. Sau đây là các khái niệm liên quan:

Cho $I = \{i_1, i_2, ..., i_m\}$ là tập gồm m mục hàng riêng biệt, mỗi mục hàng gọi là *item*. Tập $MIS = \{mis_{i_1}, mis_{i_2}, ..., mis_{i_m}\}$ là tập các ngưỡng phổ biến tối thiểu cho từng *item*. $\mathcal{D}$ là dữ liệu giao dịch, gồm n bản ghi phân biệt gọi là tập các giao dịch T = $\{t_1, t_2, ..., t_n\}$, mỗi giao dịch $t_i = \{i_{k_1}, i_{k_2}, ..., i_{k_j}\}, i_{k_j} \in I (1 \le k_j \le m)$.

Định nghĩa 3: Cho $X = \{i_1, i_2, ..., i_k\}, \forall i_j \in I (1 \le j \le k)$, ngưỡng phổ biến tối thiểu của *itemset X* được tính là $mis_X = min(mis_{i_1}, mis_{i_2}, ..., mis_{i_k}), \forall i_j \in X (1 \le j \le k)$.

Định nghĩa 4: Cho $X = \{i_1, i_2, ..., i_k\}, \forall i_j \in I (1 \le j \le k)$, X gọi là *itemset* phổ biến nếu $sup(X) \ge mis_X$.

Bảng 2. Ngưỡng phổ biến tối thiểu của từng mục hàng trong dữ liệu giao dịch $\mathcal{D}$

Mục hàng	A	B	C	D	E	F	G	H
Ngưỡng phổ biến tối thiểu của từng mục hàng (mis_{i_j})	5	2	3	3	2	4	3	2

Ví dụ 2: Dữ liệu giao dịch $\mathcal{D}$ trong *Bảng 1*, và mỗi mục hàng có mỗi giá trị ngưỡng phổ biến tối thiểu được cho trong Bảng 2, ta có:

Tập mục X ={A, C, E}, sup(ACE) = 5 ≥ mis_X = min(mis_A, mis_C, mis_E) = min(5, 3, 2) = 2, ta nói: "Tập mục X ={A, C, E} là tập mục phổ biến";

Tập mục Y ={A, C, F}, sup(ACF) = 3 ≥ mis_Y = min(mis_A, mis_C, mis_F) = min(5, 3, 4) = 3, ta nói: "Tập mục Y ={ACF} là tập mục phổ biến";

Theo ***tính chất 1*** thì các tập con của Y ={ACF} cũng phổ biến, nghĩa là tất cả tập con của Y đều phổ biến – Các con của Y là Y_{sub} = *{(A, 8, 5), (C, 8, 3), (F, 3, 4), (AC, 8, 3), (AF, 3, 4), (CF, 3, 3)}*, tuy nhiên chỉ có các tập mục {A, C, AC, CF} là phổ biến; còn các tập mục {F, AF}, ta có: sup(F) = 3 < mis_F = 4, sup(AF) = 3 < min(mis_A, mis_F) = (5, 4) = 4 là *không phổ biến*. Điều này cho chúng ta thấy: "*Khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu thì* ***tính chất 1*** *là không thỏa*".

Tương tự, với Z = {A, F} thì sup(AF) = 3 < min(mis_A, mis_F) = (5, 4) = 4, ta nói: "Tập mục Z = {A, F} *không là tập mục phổ biến*";

Theo *tính chất 2* thì các tập cha của Z ={A, F} cũng *không phổ biến*. Tuy nhiên, ta có Z ={A, F} ⊂ Y = {A, C, F}, mà sup(ACF) = 3 ≥ mis_Y = min(mis_A, mis_C, mis_F) = min(5, 3, 4) = 3, Y = {A, C, F} là tập mục *phổ biến*. Điều này cho chúng ta thấy: "*Khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu thì* ***tính chất 2*** *là không thỏa*".

Bảng 3. Tập phổ biến FI theo *một* và *nhiều* ngưỡng phổ biến tối thiểu trên dữ liệu giao dịch $\mathcal{D}$

***k*-itemset**	**Tập phổ biến FI với *minsup*=2**	**Tập phổ biến FI với nhiều ngưỡng phổ biến tối thiểu (*theo Bảng 2*)**
1	B, D, F, G, E, A, C	B, G, E, A, C
2	BE, BA, BC, DA, DC, FA, FC, FG, FE, GA, GC, GE, EA, EC, AC	BE, BC, BA, FC, GE, GA, GC, EF, EA, EC, AC
3	BAC, BEA, BEC, DAC, FAC, FAG, FAE, FCG, FCE, GAC, GAE, GCE, EAC	BEC, BEA, BAC, FEA, FEC, FAC, GEA GEC GAC, EAC
4	BEAC, FACG, FACE, GACE	BEAC, FACE, GACE

C. Tổ chức lưu trữ dữ liệu giao dịch

Lưu trữ dữ liệu giao dịch dạng *bit* là cấu trúc dữ liệu hiệu quả trong khai thác tập phổ biến [4, 5, 9]. Chuyển đổi CSDL giao dịch thành một ma trận nhị phân **BiM**, trong đó mỗi dòng tương ứng với một giao dịch và mỗi cột tương ứng với một mục hàng. Nếu mục hàng thứ i_k xuất hiện trong giao dịch t_j thì *bit* thứ i của dòng t_j trong **BiM** sẽ mang giá trị *1*, ngược lại sẽ mang giá trị *0*.

Mã giao dịch	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
t1	1	0	1	0	1	1	0	0
t2	1	0	1	0	0	0	1	0
t3	0	0	0	0	1	0	0	1
t4	1	0	1	1	0	1	1	0
t5	1	0	1	0	1	0	1	0
t6	0	0	0	0	1	0	0	0
t7	1	1	1	0	1	0	0	0
t8	1	0	1	1	0	0	0	0
t9	1	1	1	0	1	0	1	0
t10	1	0	1	0	1	1	1	0

Hình 1. Biểu diễn dạng bit của dữ liệu giao dịch $\mathcal{D}$

III. CÁC THUẬT TOÁN

A. Tập chiếu và itemset đồng xuất hiện

Tập chiếu của mục hàng i_k trên dữ liệu giao dịch $\mathcal{D}$: $\pi(i_k)=\{t\in \mathcal{D} \mid i_k \in t\}$ là tập các giao dịch có chứa mục hàng i_k (π-đơn điệu giảm).

$sup(i_k) = |\pi(i_k)|$ (1)

Tập chiếu của tập mục $X=\{i_1,i_2,...i_k\}, \forall i_j \in I (1\le j\le k)$, $\pi(X) = \pi(i_1)\cap\pi(i_2)\dots\cap\pi(i_k)$.

$sup(X) = |\pi(X)|$ (2)

Ví dụ 3: Theo Bảng 1, có π(A) = {1, 2, 4, 5, 7, 8, 9, 10} và π(B) = {7, 9}. Khi đó, π(AB) = π(A)$\cap\pi$(B)= {1, 2, 4, 5, **7**, 8, **9**, 10}$\cap${**7**, **9**} = {**7**, **9**}, π(B) $\subseteq$ π(A) và π(AB) $\subseteq$ π(A).

Định nghĩa 5: Cho $i_k \in I$, ta gọi i_k là *item* hạt nhân. Tập $X_{cooc} \subseteq I$ gọi đồng xuất hiện với i_k: X_{cooc} là tập các *item* xuất hiện cùng i_k thì $\pi(i_k)\equiv\pi(i_k\cup X_{cooc})$. Ký hiệu, $cooc(i_k) = X_{cooc}$.

Ví dụ 4: Xem *item* B là *item* hạt nhân, ta xác định được *itemset* đồng xuất hiện cùng độ phổ biến với *item* B là *cooc*(B) = {A, C, E} và *sup*(**B**) = *sup*(**B**ACE) = 2 (*theo định nghĩa 5*).

Định nghĩa 6: Cho $i_k \in I$, ta gọi i_k là *item* hạt nhân. Tập $Y_{looc} \subseteq I$ chứa các *item* xuất hiện cùng với i_k ít nhất trong một giao dịch, nhưng không đồng xuất hiện: $1\le|\pi(i_k\cup i_{looc})|<|\pi(i_k)|$, $\forall i_{looc}\in Y_{looc}$. Ký hiệu, $looc(i_k) = Y_{looc}$.

Ví dụ 5: Xem *item* G là *item* hạt nhân, ta xác định được các *item* xuất hiện cùng với *item* B ít nhất trong một giao dịch là *looc*(G) = {B, D, E, F} có π(G) = {2, 4, 5, 9, 10} và π(**G**B) = {9}, π(**G**E) = {5, 9, 10} (*theo định nghĩa 6*).

B. Thuật toán sinh itemset đồng xuất hiện có thứ tự

Nhóm tác giả đã trình bày thuật toán sinh *itemset* đồng xuất hiện trong bài viết [9]. Dưới đây là thuật toán cải tiến (*bổ sung theo định nghĩa 6*) sinh các *item* đồng xuất hiện với từng *item* trong dữ liệu giao dịch và lưu trữ vào mảng **Index_COOC**. Mỗi phần tử trong **Index_COOC** gồm 4 thành phần sau:

- **Index_COOC[j].***item*: *item hạt nhân thứ j*;
- **Index_COOC[j].***sup*: *độ phổ biến* của *item hạt nhân thứ j*;
- **Index_COOC[j].***cooc*: các *item* đồng xuất hiện cùng *item hạt nhân thứ j* dạng bit;
- **Index_COOC[j].***looc*: các *item* xuất hiện cùng *item hạt nhân thứ j ít nhất trong một giao dịch* dạng bit;

Mã giả thuật toán 1. Xây dựng bảng Index_COOC

Đầu vào*:* Dữ liệu giao dịch $\mathcal{D}$

Đầu ra*:* Mảng **Index_COOC**, ma trận **BiM**

1. Với mỗi phần tử *j* của mảng **Index_COOC** thực hiện:
2. ***Index_COOC***[j].*item* = *ij*
3. ***Index_COOC***[j].*sup* = 0
4. ***Index_COOC***[j].*cooc*= 2^m - 1
5. ***Index_COOC***[j].*looc*= 0
6. Với mỗi giao dịch t_i thực hiện:
7. Lưu giao dịch t_i vào ma trận **BiM**
8. Với mỗi item *j* có trong giao dịch t_i thực hiện:
9. ***Index_COOC***[j].*cooc* = ***Index_COOC***[j].*cooc* **AND** *vectorbit*(t_i)
10. ***Index_COOC***[j].*looc* = ***Index_COOC***[j].*looc* **OR** *vectorbit*(t_i)
11. ***Index_COOC***[j].*sup* = ***Index_COOC***[j].*sup* + 1
12. Sắp xếp mảng **Index_COOC** tăng dần theo *sup*
13. Trả về mảng **Index_COOC,** ma trận **BiM**

Từ *dòng 1* đến *dòng 5* là các bước khởi tạo cho mảng **Index_COOC**. *Dòng 6* duyệt dữ liệu giao dịch, ứng với từng giao dịch ta xem xét có chứa *item* thứ *j* thì thực hiện phép toán **AND** trên *bit* để xác định các *item* đồng xuất hiện với *item j* (*dòng 9*) và thực hiện pháp toán **OR** trên bit để xác định các *item* xuất hiện với *item j* ít nhất trong một giao dịch, nhưng không là đồng xuất hiện (*dòng 10*).

Khởi tạo mảng **Index_COOC**: (*thành phần cooc và looc biểu diễn dạng bit*) số *item* là m = 8

item	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
sup	0	0	0	0	0	0	0	0
cooc	*11111111*	*11111111*	*11111111*	*11111111*	*11111111*	*11111111*	*11111111*	*11111111*
looc	*00000000*	*00000000*	*00000000*	*00000000*	*00000000*	*00000000*	*00000000*	*00000000*

Đọc giao dịch t_1: {A, C, E, F} có biểu diễn dạng bit là **10101100**

item	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
sup	**1**	0	**1**	0	**1**	**1**	0	0
cooc	**10101100**	11111111	**10101100**	11111111	**10101100**	**10101100**	11111111	11111111
looc	**10101100**	*00000000*	**10101100**	*00000000*	**10101100**	**10101100**	*00000000*	*00000000*

Đọc giao dịch t_2: {A, C, G} có biểu diễn dạng bit là **10100010**

item	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
sup	**2**	0	**2**	0	1	1	**1**	0
cooc	**10100000**	11111111	**10100000**	11111111	10101100	10101100	**10100010**	11111111
looc	**10101110**	*00000000*	**10101110**	*00000000*	10101100	10101100	**10100010**	*00000000*

Đọc giao dịch t_3: {E, H} có biểu diễn dạng bit là **00001001**

item	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
sup	2	0	2	0	**2**	1	1	**1**
cooc	10100000	11111111	10100000	11111111	**00001000**	10101100	10100010	**00001001**
looc	10101110	*00000000*	10101110	*00000000*	**10101101**	10101100	10100010	**00001001**

Đọc giao dịch t_4: {A, C, D, F, G} có biểu diễn dạng bit là **10110110**

item	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
sup	**3**	0	**3**	**1**	2	**2**	**2**	1
cooc	**10100000**	11111111	**10100000**	**10110110**	00001000	**10100100**	**10100010**	00001001
looc	**10111110**	*00000000*	**10111110**	**10110110**	10101101	**10111110**	**10110110**	00001001

Đọc giao dịch t_5: {A, C, E, G} có biểu diễn dạng bit là **10101010**

item	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
sup	**4**	0	**4**	1	**3**	2	**3**	1
cooc	**10100000**	11111111	**10100000**	10110110	**00001000**	10100100	**10100010**	00001001
looc	10111110	*00000000*	**10111110**	10110110	**10101111**	10111110	**10111110**	00001001

Đọc giao dịch t_6: {E} có biểu diễn dạng bit là **00001000**

item	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
sup	4	0	4	1	**4**	2	3	1
cooc	10100000	11111111	10100000	10110110	**00001000**	10100100	10100010	00001001
looc	10111110	*00000000*	10111110	10110110	10101111	10111110	10111110	00001001

Đọc giao dịch t_7: {A, B, C, E} có biểu diễn dạng bit là **11101000**

item	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
sup	**5**	**1**	**5**	1	**5**	2	3	1
cooc	**10100000**	**11101000**	**10100000**	10110110	**00001000**	10100100	10100010	00001001
looc	**11111110**	**11101000**	**11111110**	10110110	**11101111**	10111110	10111110	00001001

Đọc giao dịch t_8: {A, C, D} có biểu diễn dạng bit là **10110000**

item	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
sup	**6**	1	**6**	**2**	5	2	3	1
cooc	**10100000**	11101000	**10100000**	**10110000**	00001000	10100100	10100010	00001001
looc	11111110	11101000	11111110	10110110	11101111	10111110	10111110	00001001

Đọc giao dịch t_9: {A, B, C, E, G} có biểu diễn dạng bit là **11101010**

item	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
sup	**7**	**2**	**7**	2	**6**	2	**4**	1
cooc	**10100000**	**11101000**	**10100000**	10110000	**00001000**	10100100	**10100010**	00001001
looc	11111110	**11101010**	11111110	10110110	11101111	10111110	**11111110**	00001001

Đọc giao dịch t_{10}: {A, C, E, F, G} có biểu diễn dạng bit là **10101110**

item	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
sup	**8**	2	**8**	2	**7**	**3**	**5**	1
cooc	**10100000**	11101000	**10100000**	10110000	**00001000**	**10100100**	**10100010**	00001001
looc	11111110	11101010	11111110	10110110	11101111	10111110	11111110	00001001

Thuật toán 1, trả về mảng **Index_COOC** sắp tăng theo độ phổ biến của item theo Bảng 4.

Bảng 4. Trả về mảng Index_COOC sắp tăng theo độ phổ biến của *item*.

item	**H**	**B**	**D**	**F**	**G**	**E**	**A**	**C**
sup	**1**	**2**	**2**	**3**	**5**	**7**	**8**	**8**
cooc	E	A, C, E	A, C	A, C	A, C	Ø	C	A
looc	Ø	G	F, G	D, E, G	B, D, E, F	A, B, C, F, G, H	B, D, E, F, G	B, D, E, F, G

Định nghĩa 7: Cho $i_k \in I\ (i_1 \prec i_2 \prec \ldots \prec i_m)$ thứ tự theo độ phổ biến, ta gọi i_k là *item hạt nhân.* Tập $X_{lexcooc} \subseteq I$ gọi *đồng xuất hiện có thứ tự* với *item* i_k: $X_{lexcooc}$ là tập các *item* xuất hiện cùng i_k và $\pi(i_k) \equiv \pi(i_k \cup X_{lexcooc})$, $i_k \prec i_j, \forall i_j \in X_{lexcooc}$. Ký hiệu, $lexcooc(i_k) = X_{lexcooc}$.

Định nghĩa 8: Cho $i_k \in I\ (i_1 \prec i_2 \prec \ldots \prec i_m)$ thứ tự theo độ phổ biến, ta gọi i_k là item hạt nhân. Tập $Y_{lexlooc} \subseteq I$ chứa các item xuất hiện có thứ tự cùng với i_k ít nhất trong một giao dịch, nhưng không đồng xuất hiện: $1 \le |\pi(i_k \cup i_{lexlooc})| < |\pi(i_k)|, \forall i_{lexlooc} \in Y_{lexlooc}$. Ký hiệu, $lexlooc(i_k) = Y_{lexlooc}$.

Bổ đề 1: $\forall\ i_k \prec i_j$, nếu $i_j \in lexlooc(i_k)$ thì $sup(i_k \cup i_j) < sup(i_k)$.

Chứng minh: $sup(i_k \cup i_j) < sup(i_k)$, hiển nhiên $\pi(i_k \cup i_j) = \pi(i_k) \cap \pi(i_j) \subset \pi(i_k)$ ■.

Ví dụ 6: Xét *item* B $\prec$ G, G$\in$ *lexlooc*(B)={G}. Ta có, *sup*(**B**G) = 1 < *sup*(**B**)=2.

Bổ đề 2: $lexcooc(i_k) = X_{lexcooc}$ thì $sup(i_k \cup Z_{sub}) = sup(i_k), \forall\ Z_{sub} \subseteq X_{lexcooc}$.

Chứng minh: $lexcooc(i_k) = X_{lexcooc}$, giả sử $X_{lexcooc}$ gồm ℓ *item* thì có $2^{\ell} - 1$ tập con. Với $Z_{sub} \subseteq X_{lexcooc}$ thì ta có $\pi(i_k \cup Y_{sub}) = \pi(i_k) \cap \pi(Y_{sub}) = \pi(i_k)$ ■.

Ví dụ 7: Xét *item* G, với *sup*(G)=5. Ta có, *lexcooc*(**G**) = {A, C} thì 3 *itemset* kết hợp {A, C, AC} và *sup*(**G**) = *sup*(**G**A) = *sup*(**G**C) = *sup*(**G**AC) = 5.

Bổ đề 3: $\forall\ i_k \prec i_j$ và $i_j \in Y_{lexlooc}$: $sup(i_k \cup i_j) = sup(i_k \cup Z_{sub} \cup i_j)$ và $sup(i_k \cup Z_{sub} \cup i_j) < sup(i_k \cup Z_{sub}), \forall\ Z_{sub} \subseteq X_{lexcooc}$.

Chứng minh: (theo Bổ đề 2) $\pi(i_k \cup Z_{sub}) = \pi(i_k)$ thì $\pi(i_k \cup Z_{sub} \cup i_j) = \pi(i_k \cup i_j) \subset \pi(i_k)$ ■.

Ví dụ 8: Xét *item* G, với *sup*(G)=5 và i_j = E . Ta có, *lexcooc*(**G**) = {A, C} thì 3 *itemset* kết hợp {A, C, AC} và *sup*(**GE**) = *sup*(**GE**A) = *sup*(**GE**C) = *sup*(**GE**AC) = 3 < *sup*(**G**AC) = 5.

Bổ sung dòng lệnh 14, 15 và 16 vào thuật toán 1:

14. Với mỗi phần tử *j* của mảng **Index_COOC**:

15. ***Index_COOC***[j].*cooc*= *lexcooc*(i_j)

16. ***Index_COOC***[j].*looc*= *lexlooc*(i_j)

Chỉ có *itemset* đồng xuất hiện của *item* C cần hiệu chỉnh. Ta có, *cooc*(C) = {A} và A $\prec$ C, nên *lexcooc*(C) = {∅}. Tương tự, ta có *looc*(G) = {B, D, E, F} và B, D $\prec$ F $\prec$ G $\prec$ E, nên *lexlooc*(G) = {E}.

Sau khi thực hiện *dòng 14, 15 và 16,* ta có kết quả trong Bảng 5.

Bảng 5. Trả về mảng **Index_COOC** sắp tăng theo độ phổ biến của *item*, thành phần *cooc* và *looc* có thứ tự

item	**H**	**B**	**D**	**F**	**G**	**E**	**A**	**C**
sup	**1**	**2**	**2**	**3**	**5**	**7**	**8**	**8**
cooc	E	E, A, C	A, C	A, C	A, C	Ø	C	Ø
looc	Ø	G	F, G	G, E	E	A, C	Ø	Ø

C. Thuật toán khai thác tập phổ biến MMS-FI

Thuật toán 2 - Khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu dựa trên mảng **Index_COOC** chứa các *item* đồng xuất hiện và *xuất hiện ít nhất* trong một giao dịch với *item* hạt nhân (*có thứ tự theo độ phổ biến*).

Định nghĩa 9: Cho $i_k \in I\ (i_1 \prec i_2 \prec \ldots \prec i_m)$ thứ tự theo độ phổ biến, $lexcooc(i_k) = X_{lexcooc}$ gồm ℓ *item* thì có $2^{\ell} - 1$ tập con. Tập chứa các kết hợp $F_{po_{i_k}} = \{i_k \cup Z_j\}, sup(i_k) = sup(i_k \cup Z_j), \forall Z_j \subset X_{lexcooc} (1 \le j \le 2^{\ell} - 1)$, gọi là tập chứa các *itemset tiềm năng* theo *item* hạt nhân i_k. Ký hiệu, $potent(i_k) = F_{po_{i_k}}$.

$min\_mis(I) = min(mis_{i_1}, mis_{i_2}, \ldots, mis_{i_m}), \forall i_j \in I\ (1 \le j \le m)$: ngưỡng phổ biến nhỏ nhất của *m item* trong Ɗ;

Tính chất 3: $\forall i_j \in I\ (1 \le j \le m),\ sup(i_j) \ge mis_{i_j}$ thì $i_j \in FI$;

Tính chất 4: $\forall i_j \in I\ (1 \le j \le m),\ sup(i_j) < min\_mis(I)$ thì $i_j \notin FI$;

Bổ đề 4: $sup(i_k) <$ ***min_mis***$(X_{lexcooc}(i_k) \cup i_k)$ và $sup(i_k) <$ ***min_mis***$(Y_{lexlooc}(i_k))$ thì *không sinh tập mục phổ biến* từ i_k

Chứng minh: (theo Bổ đề 2) $\forall\ Z_{sub} \subseteq X_{lexcooc}$, $sup(i_k \cup Z_{sub}) = sup(i_k) <$ ***min_mis***$(lexcooc(i_k) \cup i_k)$, $2^{\ell} - 1$ tập con sinh từ *item* hạt nhân i_k là *không phổ biến*; (theo Bổ đề 1) $\forall\ i_k \prec i_j$, nếu $i_j \in lexlooc(i_k)$ thì $sup(i_k \cup i_j) < sup(i_k) <$ ***min_mis***$(Y_{lexlooc}(i_k))$ – các tập mục kết hợp sau i_k cũng *không phổ biến*■.

Ví dụ 9: Xét *item* D, có *lexcooc*(D) ={A, C} = *sup*(D) = 2 < *min_mis*(*lexcooc*(D)∪D) = *min*(mis_A=5, mis_C=3, mis_D=3) = 3. Khi đó, các tập mục tiềm năng sinh từ *item hạt nhân* D không phổ biến $\mathbf{F_{po}}$ ={(**D**, 2, 3), (**D**A, 2, 3), (**D**C, 2, 3), (**D**AC, 2, 3)}. Và *lexlooc*(D) ={F, G} với *sup*(D) = 2 < *min_mis*(mis_F=4, mis_G=3) = 3, nên các kết hợp từ *item hạt nhân D* và các tập con của *lexlooc*(D) là $\mathbf{L_{sub}}$ = {F, G, FG} lần lượt là (**D**F, 1, 3), (**D**G, 1, 3), (**D**FG, 1, 3) không là tập mục phổ biến.

Bổ đề 5: $sup(i_k)$= ***min_mis***$(X_{lexcooc}, i_k)$ và $sup(i_k) \le$ ***min_mis***(Z_{sub}), $\forall Z_{sub} \subset Y_{lexlooc}$ thì $\forall f_j \in \mathbf{F}_{po}$, $Z_{sub} \cup f_j \notin FI$.

Chứng minh: (theo Bổ đề 1, 2) $\forall Z_{sub} \subset Y_{lexlooc}$ thì $sup(i_k \cup Z_{sub}) = sup(f_j \cup Z_{sub}) < sup(i_k) =$ ***min_mis***(f_j, Z_{sub})■.

Ví dụ 10: Xét *item* F, có *lexcooc*(F) ={A, C} = *sup*(F) = 3 = *min_mis*(*lexcooc*(F), F) = *min*(mis_A=5, mis_C=3, mis_F=4) = 3. Khi đó, các tập mục tiềm năng sinh từ *item hạt nhân* F là $\mathbf{F_{po}}$ ={(**F**A, 3, 4), (**F**C, 3, 3), (**F**AC, 3, 3)}. Và *lexlooc*(F) ={G, E} với $\mathbf{L_{sub}}$ = {G, E, GE} loại bỏ G vì mis_G= 3≥ *sup*(F): *sup*(FG)=2<mis_{FG}=min(mis_F=4, mis_G=3)=3.

Mã giả thuật toán 2. Khai thác tập phổ biến MMS-FI

Đầu vào: Mảng **Dataset**, **Index_COOC** và tập $MIS = \{ mis_{i_1}, mis_{i_2}, \ldots, mis_{i_m} \}$

Đầu ra: Tập phổ biến **FI**

1. *Loại các item theo **tính chất 4** và **Bổ đề 4***
2. Với mỗi **Index_COOC**[*k*].*sup* ≥ $min\_mis = min(mis_{i_1}, mis_{i_2}, \ldots, mis_{i_m})$
3. Nếu **Index_COOC**[*k*].sup ≥ mis_{i_k}
4. **FI**[*k*] = **FI**[*k*] ∪{i_k}//(*tính chất 3*)
5. ***Co*** = **Index_COOC**[*k*].*cooc*
6. ***Lo*** = **Index_COOC**[*k*].*looc*
7. $\boldsymbol{C_{sub}} \leftarrow$ ***các tập con của Co***
8. Với mỗi *itemset* $IS_i \in \boldsymbol{C_{sub}}$
9. $\mathbf{F_{po}}[k] = \mathbf{F_{po}}[k] \cup \{i_k \cup IS_i\}$
10. $\boldsymbol{L_{sub}} \leftarrow$ ***các tập con của Lo***
11. Nếu (**Index_COOC**[*k*].*sup* = *min_mis*(***Co***)) thì
12. $\boldsymbol{L_{sub}} \leftarrow \boldsymbol{L_{sub}} \setminus \{ \forall \boldsymbol{z_{sub}} \in \boldsymbol{L_{sub}} \mid$ **Index_COOC**[*k*].*sup*≤*min_mis*($\boldsymbol{z_{sub}}$)**}**
13. $\boldsymbol{F_{sub}} \leftarrow$ ***các kết hợp giữa*** $\boldsymbol{L_{sub}}$ ***và*** $\boldsymbol{i_k}$
14. Với mỗi $f_i \in \boldsymbol{F_{po}}$
15. Với mỗi $f_j \in \boldsymbol{F_{sub}}$
16. **FI**[*k*] = **FI**[*k*] ∪ {$f_i \cup f_j$}
17. **FI**[*k*] = **FI**[*k*] ∪ $\mathbf{F}_{po}[k]$
18. Sắp xếp **FI** giảm dần theo *sup*
19. Trả về tập phổ biến **FI**

Ví dụ 10: Cho dữ liệu giao dịch Ɗ trong Bảng 1 và tập các ngưỡng phổ biến tối thiểu của từng *item* theo Bảng 2. Sau khi thực hiện thuật toán 1, ta có mảng chứa các *itemset* đồng xuất hiện như Bảng 5.

Ta có: *MIS* = {mis_A=5, mis_B=2, mis_C=3, mis_D=3, mis_E=2, mis_F=4, mis_G=3, mis_H=2} và ***min_mis(I)*** = 2;

Dòng 1, loại các *item* theo tính chất 4 – có *item* H; theo Bổ đề 4 – có *item* D ;

Với *item* H, có *sup*(H) = 1 < ***min_mis*** : loại bỏ *item* H khỏi danh sách các *item* khai phá; (*tính chất 4*)

Xét *item* D là *item* cơ sở – *sup*(D) = 2 < ***min_mis***(lexcooc(D), lexlooc(D))= ***min_mis***(mis_A=5, mis_C=3, mis_F=4, mis_G=3)=mis_D = 3. $\mathbf{FI}_{[D]}$ = {∅}: loại bỏ item D khỏi danh sách các *item* khai phá; (*bổ đề 4*)

Xét *item* B có *lexcooc*(B) ={E, A, C}: sinh tập tiềm năng $\mathbf{F_{po}}$ = {(**B**, 2, 2), (**B**E, 2, 2), (**B**A, 2, 2), (**B**C, 2, 2), (**B**EA, 2, 2), (**B**EC, 2, 2), (**B**AC, 2, 2), (**B**EAC, 2, 2)}. Với **Lo** = {G} và $\mathbf{L_{sub}}$ = {∅} (*dòng 11, 12*), $\mathbf{F_{sub}}$ ={∅}. Ta có, $\mathbf{FI_{[B]}}$ = {(**B**, 2, 2), (**B**E, 2, 2), (**B**A, 2, 2), (**B**C, 2, 2), (**B**EA, 2, 2), (**B**EC, 2, 2), (**B**AC, 2, 2), (**B**EAC, 2, 2)}.

Xét *item* F có *lexcooc*(F) = {A, C}: sinh tập tiềm năng $\mathbf{F_{po}}$ ={(**F**A, 3, 4), (**F**C, 3, 3), (**F**AC, 3, 3)}. Với **Lo**={G, E}, $\mathbf{L_{sub}}$ = {~~G~~, E, GE} (*dòng 11, 12*), và sinh $\mathbf{F_{sub}}$ ={(**F**E, 2, 2), (**F**GE, 1, 2)}. Sinh tập phổ biến $\mathbf{FI}_{[F]}$ ={(**F**C, 3, 3), (**F**E, 2, 2), (**F**AC, 3, 3), (**FE**A, 2, 2), (**FE**C, 2, 2), (**FE**AC, 2, 2)}.

Xét *item* G có *lexcooc*(G) = {A, C}: sinh tập tiềm năng $\mathbf{F_{po}}$ ={(**G**, 5, 3), (**G**A, 5, 3), (**G**C, 5, 3), (**G**AC, 5, 3)}. Với **Lo** = {E}, $\mathbf{L_{sub}}$ ={E} và sinh $\mathbf{F_{sub}}$ = {(**G**E, 3, 2)}. Sinh tập phổ biến $\mathbf{FI}_{[G]}$ ={(**G**, 5, 3), (**G**A, 5, 3), (**G**C, 5, 3), (**G**AC, 5, 3), (**G**E, 3, 2), (**GE**A, 3, 2), (**GE**C, 3, 2), (**GE**AC, 3, 2)}.

Xét *item* E có *lexcooc*(E) ={∅}: sinh tập tiềm năng $\mathbf{F_{po}}$ ={(**E**, 7, 2)}. Với **Lo** ={A, C}, $\mathbf{L_{sub}}$ ={A, C, AC} và sinh $\mathbf{F_{sub}}$ ={(**E**A, 5,2), (**E**C, 5, 2), (**E**AC, 5, 2)}. Sinh tập phổ biến $\mathbf{FI}_{[E]}$ ={(**E**, 7, 2), (**E**A, 5,2), (**E**C, 5, 2), (**E**AC, 5, 2)}.

Xét *item* A, có *lexcooc*(A) = {C}: sinh tập tiềm năng $\mathbf{F_{po}}$ = {(**A**, 8, 5), (**A**C, 8, 3)}, **Lo** = {∅}, $\mathbf{L_{sub}}$ = {∅}, $\mathbf{F_{sub}}$ = {∅}. Ta có các *itemset* khi xét *item* A là $\mathbf{FI}_{[A]}$ = {(**A**, 8, 5), (**A**C, 8, 3)}.

Xét *item* C, có *lexcooc*(C) = {∅}, $\mathbf{F_{po}}$ = {(**C**, 8, 3)}, **Lo** = {∅}, $\mathbf{L_{sub}}$ = {∅}, $\mathbf{F_{sub}}$ = {∅}. Ta có các *itemset* khi xét *item* C là $\mathbf{FI_{[C]}}$ = {(**C**, 8, 3)}.

Tập phổ biến **FI** trên dữ liệu giao dịch Ɗ ở Bảng 1 và ngưỡng phổ biến tối thiểu của từng item trong Bảng 2.

$\mathbf{FI_{[B]}}$	(**B**, 2, 2)	(**B**E, 2, 2)	(**B**A, 2, 2)	(**B**C, 2, 2)	(**B**EA, 2, 2)	(**B**EC, 2, 2)	(**B**AC, 2, 2)	(**B**EAC, 2, 2)
$\mathbf{FI_{[F]}}$	(**F**C, 3, 3)	(**F**E, 2, 2)	(**F**AC,3,3)	(**F**EA, 2,2)	(**F**EC, 2, 2)	(**F**EAC, 2, 2)		
$\mathbf{FI_{[G]}}$	(**G**, 5, 3)	(**G**A, 5, 3)	(**G**C, 5, 3)	(**G**E, 3, 2)	(**G**AC, 5, 3)	(**G**EA, 3, 2)	(**G**EC, 3, 2)	(**G**EAC, 3,2)
$\mathbf{FI_{[E]}}$	(**E**, 7, 2)	(**E**A, 5, 2)	(**E**C, 5, 2)	(**E**AC, 5, 2)				
$\mathbf{FI_{[A]}}$	(**A**, 8, 5)	(**A**C, 8, 3)						
$\mathbf{FI_{[C]}}$	(**C**, 8, 3)							

IV. KẾT QUẢ THỰC NGHIỆM

Thực nghiệm trên máy tính Panasonic CF-74, Core Duo 2.0 GHz, 4GB RAM, thuật toán cài đặt trên C#, Microsoft Visual Studio 2010.

Nghiên cứu thực nghiệm trên hai nhóm dữ liệu:

Nhóm dữ liệu thực có mật độ dày: sử dụng dữ liệu thực từ kho dữ liệu về học máy của trường Đại học California (Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science) gồm 2 tập **Chess** và **Mushroom**.

Nhóm dữ liệu giả lập có mật độ thưa: sử dụng phần mềm phát sinh dữ liệu giả lập của trung tâm nghiên cứu IBM Almaden (IBM Almaden Research Center, San Joe, California 95120, U.S.A [http://www.almaden.ibm.com]) gồm 2 tập **T10I4D100K** và **T40I10D100K**.

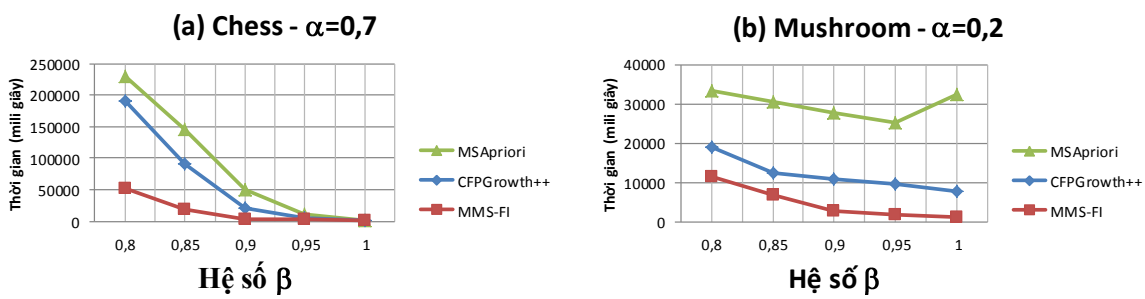
Bảng 6. Dữ liệu thực nghiệm

Tên dữ liệu	Số mục	Số giao dịch	Số mục nhỏ nhất/ giao dịch	Số mục lớn nhất/ giao dịch	Số mục trung bình/ giao dịch	Mật độ (%)
Chess	75	3.196	37	37	37	49,3%
Mushroom	119	8.142	23	23	23	19,3%
T10I4D100K	870	100.000	1	29	10	1,1%
T40I10D100K	942	200.000	4	77	40	4,2%

Phần đánh giá kết quả thực nghiệm: để nhất quán khi so sánh với các thuật toán trước, nhóm tác giả sử dụng phương thức gán các giá trị ngưỡng phổ biến tối thiểu cho từng mục hàng theo các thuật toán [6, 7, 8]. Phương pháp gán các giá trị ngưỡng phổ biến tối thiểu cho từng mục hàng theo công thức tính như sau:

$$mis_{i_j} = MAX(\alpha, \beta \times sup(i_j)), \forall i_j \in I \quad (3)$$

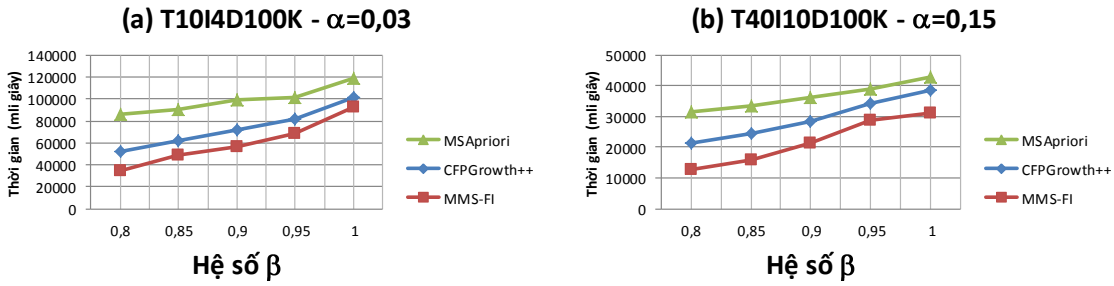
Trong đó, giá trị α là ngưỡng phổ biến nhỏ nhất có thể. Hệ số $\beta (0 \le \beta \le 1)$ là hệ số điều khiển, nhằm xác định giá trị các ngưỡng phổ biến tối thiểu của từng mục hàng theo độ phổ biến của từng mục hàng. Trường hợp β=0, lúc này trở thành bài toán khai thác tập phổ biến với *một* ngưỡng phổ biến tối thiểu là α. Dưới đây, chúng tôi so sánh thuật toán đề xuất **MMS-FI** với các thuật toán **MSApriori**, **CFPGrowth++** cùng cho kết quả giống nhau.



Hình 2. Thời gian thực hiện **MMS-FI, CFPGrowth++** và **MSApriori** trên dữ liệu **Chess**, **Mushroom**

Hình 2a và 2b là kết quả thực nghiệm trên nhóm dữ liệu có mật độ *cao*, giá trị α cố định và thay đổi hệ số β (0,8; 0,85; 0,9; 0,95; 1), ta thấy thuật toán **MMS-FI** nhanh hơn thuật toán **CFPGrowth++** và **MSApriori.**

Hình 3a và 3b là kết quả thực nghiệm trên nhóm dữ liệu giả lập có mật độ *thấp*, giá trị α cố định và thay đổi hệ số β (0,8; 0,85; 0,9; 0,95; 1), ta thấy thuật toán **MMS-FI** nhanh hơn thuật toán **CFPGrowth++** và **MSApriori.** Ngoài ra, ta thấy khi tăng hệ số β thì thời gian xử lý tăng theo (*dữ liệu mật độ thấp*).



Hình 3. Thời gian thực hiện MMS-FI, CFPGrowth++ và MSApriori trên dữ liệu T10I4D100K, T40I10D100K

V. KẾT LUẬN

Nhóm tác giả đã đề xuất kiến trúc khai thác tập phổ biến với nhiều ngưỡng phổ biến tối thiểu gồm hai giai đoạn: *giai đoạn thứ nhất* là tính nhanh mảng **Index_COOC** chứa các *itemset* đồng xuất hiện và xuất hiện với *item hạt nhân* ít nhất trong một giao dịch, đây là thuật toán cải tiến từ thuật toán của chính nhóm tác giả [9]; *giai đoạn thứ hai*: đề xuất thuật toán **MMS-FI** khai thác hiệu quả tập phổ biến với nhiều ngưỡng phổ biến tối thiểu dựa trên mảng **Index_COOC**. Với kiến trúc như trên, khi người dùng khai thác tập phổ biến với *bộ ngưỡng khác* thì thuật toán đề xuất chỉ thực hiện khai thác tập phổ biến trên mảng **Index_COOC** đã tính ở lần khai thác trước làm giảm thời gian xử lý đáng kể.

Với kiến trúc đề xuất như trên: Tương lai, nhóm tác giả sẽ mở rộng thuật toán để có thể khai thác tập phổ biến đóng, phổ biến tối đại trên dữ liệu giao dịch có *trọng số* với nhiều ràng buộc theo từng mức, đây là hướng nghiên cứu đang được quan tâm vì khả năng ứng dụng vào nhiều lĩnh vực, đặc biệt là trong kinh doanh.

LỜI CẢM ƠN

Nhóm tác giả cảm ơn sự hỗ trợ từ Trường Đại học Khoa học Xã hội và Nhân văn, Đại học Quốc gia Tp. HCM.

TÀI LIỆU THAM KHẢO

[1] R. Agrawal, T. Imilienski, A. Swami, "Mining association rules between sets of large databases". Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington, DC, pp. 207-216, 1993.

[2] R. Agrawal, R. Srikant, "Fast algorithms for mining association rules". Proceedings of International Conference on Very Large Data Base, Santiago, Chile, pp.478-499, 1994.

[3] J. Han, J. Pei, Y. Yin, R. Mao, "Mining frequent patterns without candidate generation: A frequent pattern tree approach". Data Mining and Knowledge Discovery, 8(1) pp.53–87, 2004.

[4] J. Dong, M. Han, "BitTableFI: An efficient mining frequent itemsets algorithm". Knowledge-Based Systems 20(4), pp. 329–335, 2007.

[5] W. Song, B. Yang, "Index-BitTableFI: An improved algorithm for mining frequent itemsets". Knowledge-Based Systems 21, pp. 507-513, 2008.

[6] B. Liu, W. Hsu, Y. Ma, "Mining association rules with multiple minimum supports". Proceedings of the fifth ACM SIGKDD International Conference on Knowledge discovery and Data mining, pp.337–341, 1999.

[7] Y.-H. Hu, Y.-L. Chen, "Mining association rules with multiple minimum supports: a new mining algorithm and a support tuning mechanism". Decision Support Systems, 42(1) pp.1–24, 2006.

[8] R. U. Kiran, P. K. Reddy. "Novel techniques to reduce search space in multiple minimum supports based frequent pattern mining algorithms". In EDBT, pp.11–20, 2011.

[9] Lê Hoài Bắc, Phan Thành Huấn, "DYN-FI: Thuật toán hiệu quả khai thác tập phổ biến trên dữ liệu giao dịch với ngưỡng phổ biến tối thiểu động". Một số vấn đề chọn lọc về CNTT và TT lần thứ 19, pp.98-103, 2016.

MINING FREQUENT ITEMSETS IN TRANSACTIONAL DATABASES WITH MULTIPLE MINIMUM SUPPORT THRESHOLD

Phan Thanh Huan, Le Hoai Bac

ABSTRACT: *Association rule mining, one of the most important and well-researched techniques of Data Mining. Mining frequent itemsets is one of the most fundamental problems and most time-consuming in association rule mining. Most of the algorithms in literature used to find frequent itemsets satisfy single minimum support threshold. In practice, frequentcy of each item reflects the nature and role of items in transactional databases. In this paper, we propose an efficient mining algorithm for frequent itemsets with multiple minimum support threshold (a different minimum item support for each item). Finally, we present result experiments on both synthetic and real-life datasets, which shows the proposed algorithm has better than the existing algorithms.*

Keywords: *Association rule mining, multiple minimum support threshold, frequent itemsets.*