KHAI PHÁ TẬP SINH TỐI THIỂU CỦA TẬP HIẾM ĐÓNG TỪ DỮ LIỆU GIAO DỊCH CÓ TRỌNG SỐ CỦA ITEMS

ALGORITHM MINING MINIMAL GENERATORS OF CLOSED RARE ITEMSETS FROM TRANSACTIONAL DATABASES WITH WEIGTHS OF ITEMS

Phan Thành Huấn[1], Lê Hoài Bắc[1]

[1]*Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hồ Chí Minh*
huanphan@hcmussh.edu.vn; lhbac@fithcmus.edu.vn

(Nhận bài: 03/9/2020; Chấp nhận đăng: 28/11/2020)

Tóm tắt - Trong khai phá dữ liệu, khai phá luật kết hợp hiếm là một trong những kỹ thuật khai phá quan trọng với nhiều ứng dụng tiềm năng, chẳng hạn như phát hiện các cuộc tấn công mạng, giao tác gian lận trong tài chính, y tế, tin sinh họcvà nhiều ứng dụng khác. Khai phá dữ liệu truyền thống - không có trọng số của từng item. Tuy nhiên, nhiều ứng dụng trong thực tế thì trọng số của mỗi item là khác nhau (cho biết mức độ quan trọng của từng item) – để khai phá luật kết hợp hiếm đầy đủ và không dư thừa trên dữ liệu giao dịch với items có trọng số, cần có giải thuật khai phá tập sinh tối thiểu của tập hiếm đóng.Trong bài viết này, nhóm tác giả đề xuất giải thuật hiệu quả NOV-mGCRSI khai phá tập sinh tối thiểu của tập hiếm đóng trên dữ liệu giao dịch với items có trọng số tiếp cận theo hướng không thỏa tính chất Apriori. Nhóm tác giả tiến hành thực nghiệm đánh giá giải thuật đề xuất dựa trên bộ dữ liệu giả lập và bộ dữ liệu thực, cho thấy giải thuật NOV-mGCRSI hiệu quả.

Từ khóa - Tập hiếm đóng; tập sinh tối thiểu của tập hiếm đóng; giải thuật NOV-mGCRSI; trọng số của items

Abstract - In the data mining, rare association rules mining is one of the important techniques for latent applications such as the finding of network attacks, illegal transactions in financial, medicine, bioinformatics, and other applications. In the out-of-date data mining on transaction databases, which items have no weights (as equal to 1). In spite of this, in the real-life applications are often each item with a different weight (the significance/ importance of each item) - to mining the exact and non-redundant rare association rules on transaction databases with weights of items, we need to mining for minimal generators of closed rare itemsets. In that paper, we suggest an efficient mining algorithm for minimal generators of closed rare itemsets based on dissatisfy the Apriori property. We suggest a novel algorithm named NOV-mGCRSI. The experimental investigational results show that the algorithm NOV-mGCRSI perform quicker than current algorithms on together synthetic datasets and real-life datasets.

Key words - Closed rare itemset; minimal generator itemsets; NOV-mGCRSI algorithm; weights of items

1. Đặt vấn đề

Khai phá luật kết hợp truyền thống được nhiều nhóm tác giả như Agrawal [1], Han [2] đề xuất chỉ dùng một giá trị ngưỡng hỗ trợ tối thiểu *minsupp* với giả định là các *item* trong dữ liệu có cùng tính chất, trong thực tế rất hiếm dạng dữ liệu. Trường hợp ngưỡng *minsupp* được chọn quá cao, kết quả là các itemset được khai phá có số lượng ít và lợi ích sử dụng chưa cao cho người dùng. Ngược lại, nếu chọn *minsupp* quá thấp thì các item được khai phá quá lớn, điều này gây khó khăn cho người dùng khi chọn lựa luật kết hợp sử dụng. Tuy nhiên, trong nhiều ứng dụng thực tế lại cần khai phá các luật kết hợp có ngưỡng hỗ trợ tối đại *maxsupp* nhỏ và độ tin cậy *minconf* cao được gọi là *luật kết hợp hiếm*, chẳng hạn như trong phát hiện tấn công mạng, phát hiện gian lận trong lĩnh vực tài chính, y tế, tin sinh học và nhiều ứng dụng khác. Nhiều nhóm tác giả như Koh, Troiano và Szathmary đã đề xuất giải thuật khai phá *tập hiếm* thỏa *một* hoặc *hai* ngưỡng như giải thuật **Apriori-Inverse** [3], **Rarity** [4] và **Walky-G** [5]. Các giải thuật này còn tồn tại hạn chế như đọc dữ liệu nhiều lần, dùng nhiều bộ nhớ, sử dụng các chiến lược cắt tỉa (*không dùng lại cho lần khai phá kế tiếp*).

Vào năm 2018, nhóm tác giả Borah [8] có tổng luận về thách thức *khai phá mẫu hiếm* trong tương lai. Cùng thời điểm đó, Lu đề xuất giải thuật **RaCloMiner** [9] khai phá *tập hiếm đóng*. Tuy nhiên, để sinh nhanh các *luật kết hợp hiếm đầy đủ* cần có giải thuật hiệu quả khai phá *tập sinh tối thiểu* của tập hiếm đóng.

Song song đó, Cai [6] đã đề xuất mô hình khai phá tập phổ biến có trọng số của *item* (*mức độ quan trọng hay mức ý nghĩa của các item là khác nhau*) chứa nhiều tri thức hơn so với khai phá tập phổ biến truyền thống (*không trọng số*). Nhận thấy được ý nghĩa của vấn đề, nhiều nhóm tác giả đã đề xuất các giải thuật để giải quyết vấn đề này. Phần lớn các giải thuật được đề xuất đều giải quyết theo hướng tiếp cận thỏa ***tính chất Apriori***. Năm 2011, Huai đề xuất giải thuật **WHIUA** [7] giải quyết vấn đề trên dựa theo tiếp cận ***không thỏa tính chất Apriori***, điều này làm gia tăng đáng kể không gian tìm kiếm các *itemset* phổ biến – **đây là một thách thức lớn**.

Trong công trình này, nhóm tác giả trình bày giải thuật đề xuất **NOV-mGCRI** khai phá hiệu quả *tập sinh tối thiểu* của tập hiếm đóng. Điều này, làm giảm đáng kể các kết hợp trong bước sinh *luật kết hợp hiếm*.

2. Vấn đề cơ bản về tập hiếm

Cho $I = \{i_1, i_2, ..., i_m\}$ là tập gồm *m* thuộc tính, mỗi thuộc tính gọi là *item*. Tập $SIG = \{sig_{i1}, sig_{i2}, ..., sig_{im}\}$, $\forall sig_{ik} \in [0, 1]$ là tập các mức ý nghĩa *hay* mức độ quan trọng của từng *item* (*trọng số của từng item*). Tập chứa các item $X = \{i_1, i_2, ..., i_k\}$, $\forall i_j \in I$ $(1 \le j \le k)$ ta gọi là *itemset*, itemset có *k* items gọi là *k-itemset*. *D* là dữ liệu giao dịch,

[1] VNUHCM - University of Science (Phan Thanh Huan, Le Hoai Bac)

gồm *n* mẫu tin gọi là tập các giao dịch T={t_1, t_2,..., t_n}, giao dịch t_k ={i_{k1}, i_{k2},..., i_{km}}, $i_{kj} \in I$ ($1 \le k_j \le m$).

Định nghĩa 1: *Độ hỗ trợ* (support) của *itemset* X ⊆ I, ký hiệu *supp*(X) - tỷ lệ giữa số lượng giao dịch có trong *Ɗ* chứa *itemset* X và *n* giao dịch.

Định nghĩa 2: Mức ý nghĩa của *itemset* $X \subseteq I$ được tính toán $sig(X)=max(sig_{i1}, sig_{i2},..., sig_{ik}), \forall i_j \in X\ (1 \le j \le k)$.

Định nghĩa 3: Cho $X \subseteq I$, *X* gọi là *itemset* hiếm nếu *sigsupp*(*X*) < *maxsigsupp*, *maxsigsupp* - ngưỡng mức ý nghĩa hỗ trợ tối đại (*người dùng cho trước*). Tập hợp chứa các *itemset* hiếm có trọng số gọi là *tập hiếm có trọng số của item*, ký hiệu là **RSI** (Rare Significance Itemsets).

Mức ý nghĩa hỗ trợ của itemset *X*:

$$sigsupp(X) = sig(X) \times supp(X) \quad (1)$$

Định nghĩa 4: Cho $X \in$ **CRSI**, *X* gọi là *itemset* hiếm đóng nếu *X* là *itemset* hiếm và không tồn tại *tập cha cùng độ hỗ trợ*. **CRSI** là ký hiệu tập gồm các *itemset* hiếm đóng có trọng số (Closed Rare Significance Itemsets).

Định nghĩa 5: Cho $X \in$ **CRSI**, tất cả các *itemset* con thực sự của *X* có cùng độ hỗ trợ với *X* được gọi là *itemset* sinh của *itemset* hiếm đóng *X*. Tập hợp chứa các *itemset* sinh của các *itemset* hiếm đóng gọi là tập sinh của tập hiếm đóng có trọng số của item, ký hiệu là **GCRSI** (Generators Rare Significance Itemsets).

Định nghĩa 6: $\forall$ X $\in$ **mGCRSI** $\subseteq$ **CRSI**, không tồn tại tập con có cùng độ hỗ trợ với *X*. Khi đó, **mGCRSI** là tập chứa *itemsest* sinh tối thiểu của *itemsets* hiếm đóng có trọng số (minimal Generators Rare Significance Itemsets).

Cho tập dữ liệu *Ɗ* mô tả ở Bảng 1 và Bảng 2.

Bảng 1. *Tập dữ liệu Ɗ sử dụng cho Ví dụ*

TID	Items							
t1	i_1		i_3	i_4				
t2	i_1		i_3		i_5	i_6		
t3	i_1		i_3	i_4		i_6	i_7	
t4					i_5			i_8
t5					i_5			
t6	i_1		i_3		i_5		i_7	
t7	i_1		i_3				i_7	
t8	i_1	i_2	i_3		i_5		i_7	
t9	i_1		i_3		i_5	i_6	i_7	
t10	i_1	i_2	i_3		i_5			

Dữ liệu ở Bảng 1: 8 *items* I ={i_1; i_2; i_3; i_4; i_5; i_6; i_7; i_8} và 10 giao dịch T= {*t1; t2; t3; t4; t5; t6; t7; t8; t9; t10*}.

Bảng 2. *Mức ý nghĩa tương ứng của mỗi item*

item	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
sig	0,55	0,70	0,50	0,65	0,40	0,60	0,30	0,80

Bảng 3. *CRSI vàm mGCRSI trên Ɗ với maxsigsupp = 0,15*

k-itemset	Tập CRSI (#CRSI=9)	Tập mGCRSI (#mGCRSI=8)
1		i_8, i_2, i_4
2	i_5i_8, i_5i_7	i_2i_7, i_4i_6, i_4i_7, i_6i_7
3	$i_4i_1i_3$	$i_6i_5i_7$
4	$i_2i_1i_3i_5$, $i_6i_1i_3i_7$, $i_6i_1i_3i_5$	
5	$i_2i_1i_3i_5i_7$, $i_4i_1i_3i_6i_7$, $i_6i_1i_3i_5i_7$	

Bảng 3, cho thấy tập **CRSI** và **mGCRSI** được gom nhóm theo *k-itemset* với *maxsigsupp* = 0,15 và số lượng các itemset hiếm đóng |**CRSI**| = 9, itemset sinh tối thiểu của itemset hiếm đóng |**mGCRSI**| = 8.

3. Giải thuật đề xuất

3.1. Tập chiếu và items xuất hiện ít nhất trên cùng một giao dịch với item-hạt-nhân có thứ tự [10]

Chiếu *item* i_k lên trên dữ liệu Ɗ: $\pi(i_k)=\{\forall t_j \in Ɗ i_k \in t_j\}$ đây là tập hợp các giao dịch có chứa i_k, *tập chiếu* của i_k.

$$supp(i_k) = |\pi(i_k)| \quad (2)$$

Phương trình (2): độ hỗ trợ của i_k bằng lực lượng của tập chiếu của i_k trên dữ liệu Ɗ.

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

$$supp(X) = |\pi(X)| \quad (3)$$

Để *không gian sinh* được rút gọn, nhóm tác giả đưa ra Định nghĩa 7 và 8 ($P_k(X)$ – *powerset của X có k item*):

Định nghĩa 7: Cho item $i_k \in I$ ($i_1 \succ i_2 \succ \dots \succ i_m$) có thứ tự giảm dần theo mức ý nghĩa, gọi i_k là *item-hạt-nhân*. Itemset $X_{lexicooc} \subseteq I$ gồm các item *xuất hiện đồng thời* với i_k và $\pi(i_k) \equiv \pi(i_k \cup i_j), \forall i_j \in X_{lexicooc}$, $i_k \succ i_j$. Ký hiệu, $lexicooc(i_k) = X_{lexicooc}$.

Định nghĩa 8: Cho item $i_k \in I$ ($i_1 \succ i_2 \succ \dots \succ i_m$) có thứ tự giảm dần theo mức ý nghĩa, gọi i_k là *item-hạt-nhân*. Itemset $Y_{lexilooc} \subseteq I$ gồm các *item* xuất hiện trong ít nhất một giao dịch cùng với i_k, nhưng không xuất hiện đồng thời: $1 \le |\pi(i_k \cup i_j)| < |\pi(i_k)|, \forall i_j \in Y_{lexilooc}$, $i_k \succ i_j$. Ký hiệu, $lexilooc(i_k) = Y_{lexilooc}$.

Giải thuật sinh mảng **IndexCOOC**. Từng phần tử của mảng **IndexCOOC** có 4 trường thông tin:

- **IndexCOOC[k].***item*: lưu trữ *item-hạt-nhân* i_k;
- **IndexCOOC[k].***supp*: *độ hỗ trợ của* i_k;
- **IndexCOOC[k].***cooc*: *items* xuất hiện đồng thời cùng với i_k;
- **IndexCOOC[k].***looc*: *items* xuất hiện cùng với i_k trong *ít nhất là một giao dịch*;

Giải thuật 1. Tạo dựng mảng IndexCOOC

Đầu vào*:* Tập dữ liệu *Ɗ*

Đầu ra*:* **IndexCOOC**

1. **For each** *IndexCOOC* **do**
2. **IndexCOOC**[k].*item* = i_k; **IndexCOOC**[k].*supp = 0*
3. **IndexCOOC**[k].*cooc*=$2^m - 1$; **IndexCOOC**[k].*looc*=0
4. **For** $t_i \in$ T **do**
5. **For** $i_k \in t_i$ **do**
6. ***IndexCOOC***[k].*cooc* &= *vectorbit*(t_i)
7. ***IndexCOOC***[k].*looc* |= *vectorbit*(t_i)
8. ***IndexCOOC***[k].*supp* + +
9. sort **IndexCOOC** in *descending by sig*
10. **For each IndexCOOC do**
11. **IndexCOOC**[k].*cooc*= *lexicooc*(i_k)
12. **IndexCOOC**[k].*looc*= *lexilooc*(i_k)
13. return **IndexCOOC, BiM**

Minh họa giải thuật 1: thực hiện từ *dòng 1* đến *8*

Khởi tạo đầu tiên cho mảng **IndexCOOC**: (*cooc* và *looc được minh họa theo hexa*) số *item* từ dữ liệu $\mathcal{D}$ đã cho ở Bảng 1 là m = 8

item	$\mathbf{i_1}$	$\mathbf{i_2}$	$\mathbf{i_3}$	$\mathbf{i_4}$	$\mathbf{i_5}$	$\mathbf{i_6}$	$\mathbf{i_7}$	$\mathbf{i_8}$
supp	0	0	0	0	0	0	0	0
cooc	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
looc	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Duyệt giao dịch t_1: {i_1, i_3, i_4} có dạng bit tương ứng là **10110000** (0xB0)

item	$\mathbf{i_1}$	$\mathbf{i_2}$	$\mathbf{i_3}$	$\mathbf{i_4}$	$\mathbf{i_5}$	$\mathbf{i_6}$	$\mathbf{i_7}$	$\mathbf{i_8}$
supp	1	0	1	0	1	1	0	0
cooc	**0xB0**	0xFF	**0xB0**	**0xB0**	0xFF	0xFF	0xFF	0xFF
looc	**0xB0**	0x00	**0xB0**	**0xB0**	0xFF	0xFF	0x00	0x00

Tương tự, duyệt giao dịch t_{10}: {i_1, i_2, i_3, i_5} có dạng bit tương ứng là **11101000** (0xE8)

item	$\mathbf{i_1}$	$\mathbf{i_2}$	$\mathbf{i_3}$	$\mathbf{i_4}$	$\mathbf{i_5}$	$\mathbf{i_6}$	$\mathbf{i_7}$	$\mathbf{i_8}$
sup	8	2	8	2	7	3	5	1
cooc	**0xA0**	**0xE8**	**0xA0**	0xB0	**0x08**	0xA4	0xA2	0x09
looc	**0xFE**	**0xEA**	**0xFE**	0xB6	**0xEF**	0xBE	0xFE	0x01

Dòng *9*, sắp xếp **IndexCOOC** giảm dần theo *sig* của từng item, ta có kết quả:

item	$\mathbf{i_8}$	$\mathbf{i_2}$	$\mathbf{i_4}$	$\mathbf{i_6}$	$\mathbf{i_1}$	$\mathbf{i_3}$	$\mathbf{i_5}$	$\mathbf{i_7}$
supp	1	2	2	3	8	8	7	5
cooc	i_5	i_1,i_3,i_5	i_1, i_3	i_1, i_3	i_3	i_1	∅	i_1, i_3
looc	∅	i_7	i_6, i_7	i_4,i_5,i_7	i_2,i_4,i_5,i_6,i_7	i_2,i_4,i_5,i_6,i_7	i_1,i_2,i_3,i_6,i_7,i_8	i_2,i_4,i_5,i_6

Từ *dòng 10 đến 12* – cho kết quả rút gọn ở Bảng 4:

Chỉ có *itemset* đồng xuất hiện của *item* i_3 cần hiệu chỉnh. Ta có, $cooc(i_3) = \{i_1\}$ và $i_1 \succ i_3$, nên $lexicooc(i_3) = \{\varnothing\}$. Tương tự, ta có $looc(i_1) = \{ i_2, i_4, i_5, i_6, i_7\}$ và $i_2 \succ i_4 \succ i_6 \succ i_1 \succ i_5 \succ i_7$, nên $lexilooc(i_1) = \{ i_5, i_7\}$. Dòng *10, 11 và 12* được thực hiện, ta nhận được kết quả ở Bảng 4. Nhóm tác giả bổ sung vào **IndexCOOC** trường *sig* - minh họa **IndexCOOC** có trường *sig* được xếp giảm dần.

Bảng 4. *IndexCOOC có thứ tự giảm dần theo mức ý nghĩa sig của item, đồng thời cooc và looc cũng có thứ tự*

item	$\mathbf{i_8}$	$\mathbf{i_2}$	$\mathbf{i_4}$	$\mathbf{i_6}$	$\mathbf{i_1}$	$\mathbf{i_3}$	$\mathbf{i_5}$	$\mathbf{i_7}$
sig	0,80	0,70	0,65	0,60	0,55	0,50	0,40	0,30
supp	0,10	0,20	0,20	0,30	0,80	0,80	0,70	0,50
cooc	i_5	i_1,i_3,i_5	i_1, i_3	i_1, i_3	i_3	∅	∅	∅
looc	∅	i_7	i_6, i_7	i_5, i_7	i_5, i_7	i_5, i_7	i_7	∅

3.2. Giải thuật sinh cây nLOOCTree

Từ **IndexCOOC** xây dựng các cây lưu trữ các mẫu *xuất hiện cùng với item-hạt-nhân trong ít nhất* một giao dịch. Nút gốc của cây là *item-hạt-nhân*, các nút con là *items xuất hiện với item-hạt-nhân* trong ít nhất trong một giao dịch. Mỗi nút có 2 trường thông tin:

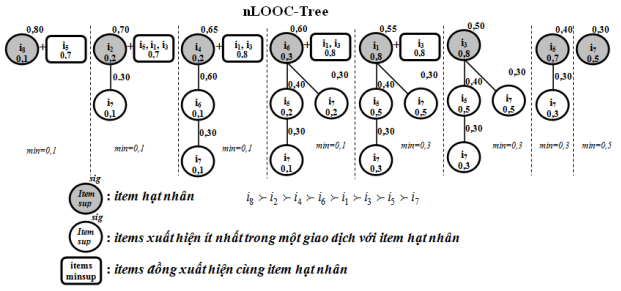
- **nLOOCTree[k].***item*: lưu trữ item xuất hiện cùng với item-hạt-nhân trong ít nhất một giao dịch;
- **nLOOCTree[k].***supp*: lưu trữ độ hỗ trợ của item xuất hiện cùng với item-hạt-nhân;

Giải thuật 2: Tạo sinh nLOOCTree

Đầu vào*:* $\mathcal{D}$, **IndexCOOC**

Đầu ra*:* các **nLOOCTree**

1. **For each IndexCOOC do**
2. nLOOCTree[k].*item* = ***IndexCOOC***[k].*item*
3. nLOOCTree[k].*supp* = ***IndexCOOC***[k].*supp*
4. **For each** $i_k \in t_\ell$ **do**
5. For each $i_j \in$ **IndexCOOC**[k].*looc* do
6. **If** $i_j \notin$ child node of **nLOOCTree**[k]
7. Add child node i_j to **nLOOCTree**[k]
8. **Else**
9. Update *supp* of child node i_j on **nLOOCTree**[k]
10. return **nLOOCTree**



Hình 1. *Các nLOOCTree theo IndexCOOC ở Bảng 4*

Đặc trưng của mỗi **nLOOCTree**:

- Độ cao tương ứng của mỗi cây *không lớn hơn* số *item xuất hiện cùng với item-hạt-nhân trong ít nhất* là một giao dịch (*items có thứ tự theo supp*).
- Một đường đi đơn (*single-path*): *itemset* thứ tự xác định từ nút gốc cho đến nút lá và *supp* của *itemset* chính là *supp* của nút lá ($i_k \rightarrow i_{k+1} \rightarrow \ldots \rightarrow i_\ell$).
- Phân đoạn của đường đi đơn (*sub-single-path*): từ nút gốc đi đến nút con tùy ý của một đường đi đơn là *itemset* thứ tự; *supp* của *itemset* đó là *supp* của nút con nằm ở cuối của phân đoạn.
- Mỗi **nLOOCTree** lưu trữ thêm *độ hỗ trợ nhỏ nhất* (ký hiệu là *min*) trong các nút lá.

3.3. Giải thuật khai phá tập sinh tối thiểu của tập hiếm đóng NOV-mGCRSI

Giải thuật **NOV-mGCRSI** (**NOV**el - **m**inimal **G**enerators **C**losed **R**are **S**ignificance **I**temsets): khai phá tuần tự *tập sinh tối thiểu* dựa trên cây **nLOOCTree** chứa *items cùng xuất hiện với item-hạt-nhân trong ít nhất là một giao dịch*.

Các bổ đề và hệ quả dùng để loại bỏ những *item-hạt-nhân* không thể khai phá *itemset sinh tối thiểu* của *tập hiếm đóng*:

Bổ đề 1: $X_{lexicooc} = lexicooc(i_k)$ thì $supp(i_k \cup x_{sub}) = supp(i_k)$, $\forall\, x_{sub} \in \mathcal{P}_{\geq 1}(X_{lexicooc})$.

Chứng minh: $lexicooc(i_k) = X_{lexicooc}$, $\forall\, x_{sub} \in \mathcal{P}_{\geq 1}(X_{lexicooc})$. Từ Định nghĩa 7, ta có $\pi(i_k \cup x_{sub}) = \pi(i_k) \cap \pi(x_{sub}) = \pi(i_k)$; theo (2) và (3) thì $supp(i_k \cup x_{sub}) = supp(i_k)$, $\forall\, x_{sub} \in \mathcal{P}_{\geq 1}(X_{lexicooc})$■.

Bổ đề 2: $Y_{lexilooc} = lexilooc(i_k)$ thì $supp(i_k \cup y_{lexilooc}) < supp(i_k)$, $\forall\, y_{lexilooc} \in \mathcal{P}_{\geq 1}(Y_{lexilooc})$.

Chứng minh: $supp(i_k \cup y_{lexilooc}) < supp(i_k)$, từ định nghĩa 8 thì $\pi(i_k \cup y_{lexilooc}) = \pi(i_k) \cap \pi(i_1) \cap \ldots \cap \pi(i_j) \subset \pi(i_k)$, $\forall\, i_{1,j} \in y_{lexilooc}$■.

Hệ quả 1: (*bổ đề 1, 2* và *định nghĩa 6*) $\forall\, ssp_j \in$

nLOOCTree(i_k) $\subseteq$ $\mathcal{P}_{\geq 1}$(*lexilooc*(i_k)), nếu *sigsupp*(ssp_j) < *maxsigsupp* và *supp*(ssp_{j-1}) ≠ *supp*(ssp_j) thì ssp_j ∈ **mGCRSI**.

Bổ đề 3: ∀i_k ∈ **mGCRSI**, $X_{lexicooc}$ = *lexicooc*(i_k) và *sigsupp*(i_k) < *maxsigsupp* thì {$i_k \cup x_{sub}$} ∉ **mGCRSI**, ∀ $x_{sub} \in \mathcal{P}_{\geq 1}(X_{lexicooc})$.

Chứng minh: *lexicooc*(i_k) = $X_{lexicooc}$, ∀ $x_{sub} \in \mathcal{P}_{\geq 1}(X_{lexicooc})$. Dựa vào *bổ đề 1*, *supp*($i_k \cup x_{sub}$) = *supp*(i_k) và *sigsupp*($i_k \cup x_{sub}$) < *maxsigsupp* mà i_k ∈ **mGCRSI**, nên {$i_k \cup x_{sub}$} ∉ **mGCRSI** (*Định nghĩa 6*)■.

Hệ quả 2: *sigsupp*(i_k) < *maxsigsupp* và *lexicooc*(i_k) = {∅} thì i_k ∉ **mGCRSI** (theo *bổ đề 3*).

Giải thuật khai phá *tập sinh tối thiểu của tập hiếm đóng* **mGCRSI** từ **nLOOCTree** (i_k ≡ **IndexCOOC**[k]):

Giải thuật 3: Sinh tập mGCRSI

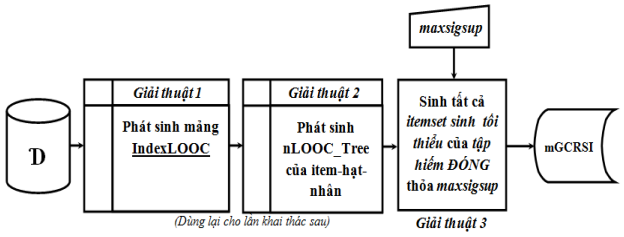
Đầu vào: **IndexCOOC**, *maxsigsupp*

Đầu ra: Tập sinh tối thiểu **mGCRSI**

1. **For each** IndexCOOC[*k*].*item*
2. **If**(*sigsupp*(i_k)<*maxsigsupp*)∨ (*sig*(i_k)×*min*(nLOOC_Tree(i_k)) < *maxsigsupp*)
3. **If** (IndexCOOC[*k*].*cooc* ≠{∅}) *//hệ quả 1*
4. mGCRSI[*k*] = mGCRI[*k*] ∪ IndexCOOC[*k*].*item*
5. **If** (IndexCOOC[*k*].*looc* <> {∅}) //theo *bổ đề 2*
6. nLOOCTree(IndexCOOC[*k*].item)
7. *SSP* = Gen_Path(IndexCOOC[*k*].item)
8. **For each** ssp_j ∈ *SSP* *//hệ quả 2*
9. **If** (*sigsupp*(ssp_j)<*maxsigsupp*)∧ (*supp*(ssp_{j-1})≠*supp*(ssp_j))
10. mGCRSI[*k*] = mGCRSI[*k*]∪{ssp_j}
11. return **mGCRSI**

4. Minh họa giải thuật NOV-mGCRSI

Lưu đồ giải thuật **NOV-mGCRSI** khai phá *tập sinh tối thiểu* của tập hiếm đóng trên dữ liệu giao dịch có trọng số của item, được trình bày ở Hình 2.



Hình 2. *Lưu đồ khai phá tập sinh tối thiểu*

Cho Đ như ở Bảng 1 và 2 với *maxsigsupp* = 0,15. Kết thúc *giải thuật 1*, cho **IndexCOOC** như Bảng 4.

Xét dòng 1-2: các item {i_8, i_2, i_4, i_6, i_5} tiềm năng cho khai phá *itemset sinh tối thiểu* của *itemset hiếm đóng*;

Xây dựng lần lượt các **nLOOCTree** cho *items* tiềm năng: i_8, i_2, i_4, i_6 *và* i_5;

Xét *item* i_8, *lexicooc*(i_8) = {i_5} và *lexilooc*(i_8) = {∅} sinh tập **mGCRSI**$_{[i8]}$ = {(***i*****8**;0,10;0,08)} (*dòng 3*).

Xét *item* i_2, *lexicooc*(i_2) = {i_5, i_1, i_3} sinh tập **mGCRSI**$_{[i2]}$ = {(***i*****2**;0,20;0,14)} (*dòng 3*), cây **nLOOCTree**(i_2): đường đi đơn {$i_2 \rightarrow i_7$} có *sigsupp*(i_2i_7) = 0,70×0,10 < *maxsigsupp*. Ta có, **mGCRSI** $_{[i2]}$ = ∪{(***i*****2*i*****7**;0,10;0,07)} (*dòng 4 đến 10*).

Xét *item* i_4, *lexicooc*(i_4) = {i_1, i_3} sinh tập **mGCRSI**$_{[i4]}$ = {(***i*****4**;0,20;0,13)} (*dòng 3*), cây **nLOOCTree**(i_4): sinh hai phân đoạn đường đi đơn {$i_4 \rightarrow i_6$},{$i_4 \rightarrow i_7$} và *sigsupp*(i_4i_6) = *sigsupp*(i_4i_7) = 0,650×0,10 < *maxsigsupp*. Ta có, **mGCRSI** $_{[i4]}$ = ∪{(***i*****4*i*****6**;0,10;0,065), (***i*****4*i*****7**;0,10;0,065)}.

Xét *item* i_6, *lexicooc*(i_6) = {i_1, i_3} và cây **nLOOCTree**(i_6): có hai đường đi đơn {$i_6 \rightarrow i_7$},{$i_6 \rightarrow i_5 \rightarrow i_7$} và *sigsupp*($i_6i_7$) = 0,60×0,20 < *maxsigsupp* và *sigsupp*($i_6i_5i_7$) = 0,60×0,10 < *maxsigsupp*. Ta có, **mGCRSI**$_{[i6]}$={(***i*****6*i*****7**;0,20 ;0,12), (***i*****6*i*****5*i*****7**;0,10;0,06)}.

Xét *item* i_5, *lexicooc*(i_5) = {∅} và **nLOOCTree**(i_5) có một đường đi đơn {$i_5 \rightarrow i_7$} và *sigsupp*(i_5i_7) = 0,40×0,30 < *maxsigsupp*, sinh tập **mGCRSI**$_{[i5]}$ = {∅}.

Tập sinh tối thiểu **mGCRSI** từ dữ liệu Đ ở Bảng 1 và 2 với *maxsigsupp* = 0,15:

Bảng 5. *Tập* ***mGCRSI*** *trên Đ với maxsigsupp = 0,15*

item	Tập sinh tối thiểu mGCRSI (#mGCRSI = 8)		
i**8**	(i_8; 0,10; 0,08)		
i**2**	(i_2; 0,20; 0,14)	(i_2i_7;0,10;0,07)	
i**4**	(i_4; 0,20; 0,13)	(i_4i_6;0,10;0,065)	(i_4i_7;0,10;0,065)
i**6**	(i_6i_7; 0,20; 0,12)	($i_6i_5i_7$;0,10;0,06)	

5. Thực nghiệm

Giải thuật **NOV-mGCRSI** được thực nghiệm cài đặt trên máy tính cấu hình: Core i7-3540M 3.0 GHz, bộ nhớ 4 GB; ngôn ngữ lập trình C# (Visual Studio 2015).

Thực nghiệm sử dụng 2 loại dữ liệu:

– Dữ liệu thu thập thực tế: 2 tập **Chess** và **Mushroom** từ kho lưu trữ UCI.

– Dữ liệu chạy giả lập: 2 tập dữ liệu giả lập **T10I4D100K** và **T40I10D100K** từ trung tâm Almaden của IBM.

Bảng 6. *Dữ liệu thực nghiệm*

Tập dữ liệu	Số giao dịch	Số items	Số item trung bình
Chess	3.196	75	37
Mushroom	8.142	119	23
T10I4D100K	100.000	870	10
T40I10D100K	100.000	942	40

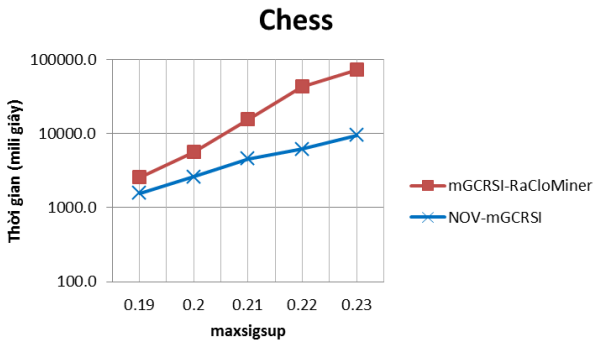
Trong công trình này, nhóm tác giả đề xuất giải thuật hiệu quả khai phá *tập sinh tối thiểu* của *tập hiếm đóng* trên dữ liệu giao dịch *có trọng số của items*. Đây là đề xuất đầu tiên, nên chưa có giải thuật cùng hướng tiếp cận để so sánh hiệu năng giải thuật. Vì vậy, nhóm tác giả đề xuất so sánh hiệu năng giải thuật theo *2 thực nghiệm*:

5.1. Thực nghiệm 1

Khai phá *tập sinh tối thiểu* của *tập hiếm đóng* trên dữ liệu giao dịch *có trọng số items*, *mức ý nghĩa* (trọng số) của các item được phát sinh ngẫu nhiên trong [0, 1].

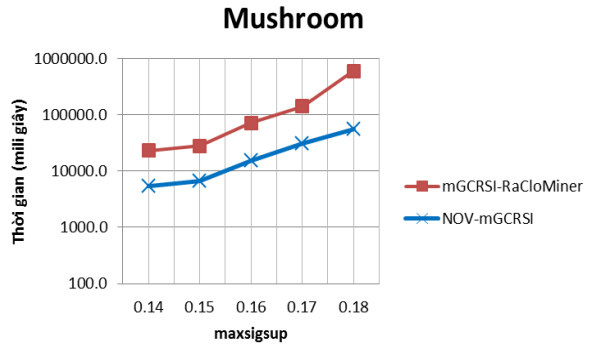
Trong *thực nghiệm 1*, nhóm tác giả dựa vào giải thuật

RaCloMiner [9] khai phá *tập hiếm đóng* trên dữ liệu giao dịch nhị phân do *Lu* và đồng sự đề xuất năm 2018 và cải tiến thành giải thuật khai phá *tập sinh tối tối thiểu*, gọi là **mGCRSI-RaCloMiner**. Trên cơ sở này, nhóm tác giả so sánh hiệu năng giải thuật **mGCRSI-RaCloMiner** với giải thuật đề xuất **NOV-mGCRSI** theo từng ngưỡng *maxsigsupp* và cả 2 giải thuật đều cho cùng kết quả.



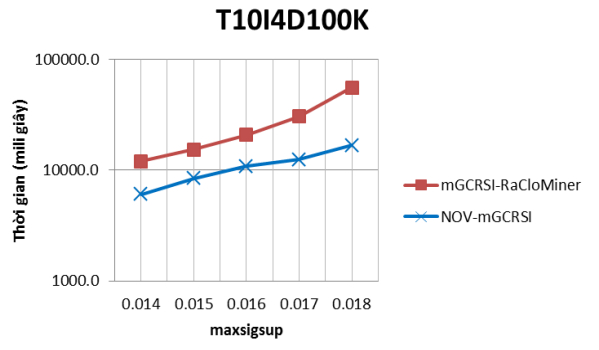
Hình 3.** Biểu đồ khai phá **mGCRSI** trên **Chess

Hình 3 - thực nghiệm so sánh hiệu quả về mặt thời gian từ tập dữ liệu **Chess** mật độ dày đặc (49,3%), cho thấy giải thuật **NOV-mGCRSI** nhanh hơn giải thuật **mGCRSI-RaCloMiner**.



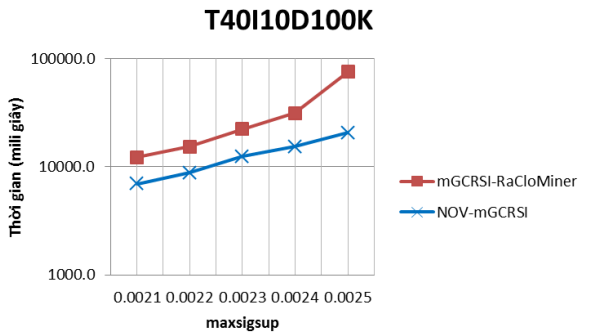
Hình 4.** Biểu đồ khai phá **mGCRSI** trên **Mushroom

Hình 4 - thực nghiệm so sánh hiệu quả về mặt thời gian từ tập dữ liệu **Mushroom** mật độ dày đặc (19,3%), giải thuật **NOV-mGCRSI** nhanh hơn **mGCRSI-RaCloMiner**.



Hình 5.** Biểu đồ khai phá **mGCRSI** trên **T10I4D100K

Hình 5 - thực nghiệm so sánh hiệu quả về mặt thời gian từ tập dữ liệu **T10I4D100K** mật độ rất thưa (1,1%), giải thuật **NOV-mGCRSI** nhanh hơn **mGCRSI-RaCloMiner**.



Hình 6.** Biểu đồ khai phá **mGCRSI** trên **T40I10D100K

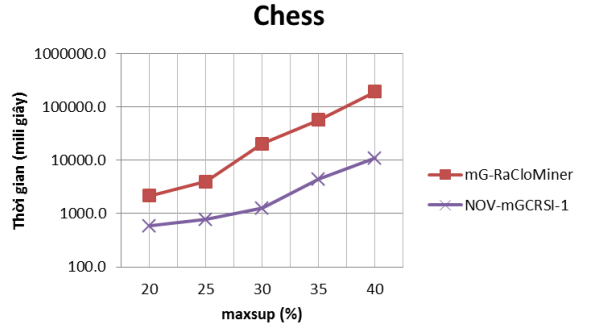
Hình 6 - thực nghiệm so sánh hiệu quả về mặt thời gian từ tập dữ liệu **T40I10D100K** mật độ rất thưa (4,2%), giải thuật **NOV-mGCRSI** nhanh hơn **mGCRSI-RaCloMiner**.

5.2. Thực nghiệm 2

Khai phá *tập sinh tối thiểu* của *tập hiếm đóng*, mức ý nghĩa của items bằng 1 (*maxsigsupp* trở thành *maxsupp*).

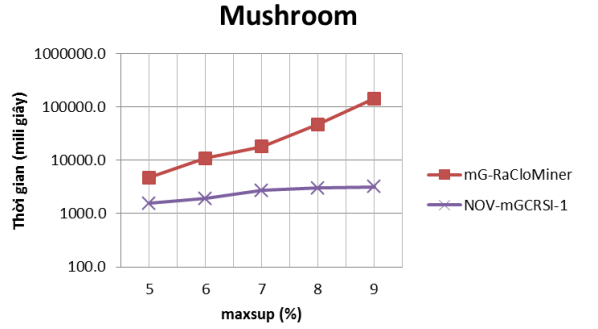
$$sig_{i_1}=sig_{i_2}=...=sig_{i_m}=1 \quad (4)$$

Trong *thực nghiệm 2*, nhóm tác giả so sánh giải thuật đề xuất **NOV-mGCRSI-1** (*trọng số của các item bằng 1*) với giải thuật **mG-RaCloMiner**, đây là giải thuật khai phá tập sinh tối thiểu của tập hiếm đóng được hiệu chỉnh từ giải thuật **RaCloMiner [9]**.Trên cơ sở này, nhóm tác giả so sánh hiệu năng giải thuật **mG-RaCloMiner** với giải thuật đề xuất **NOV-mGCRSI-1**.



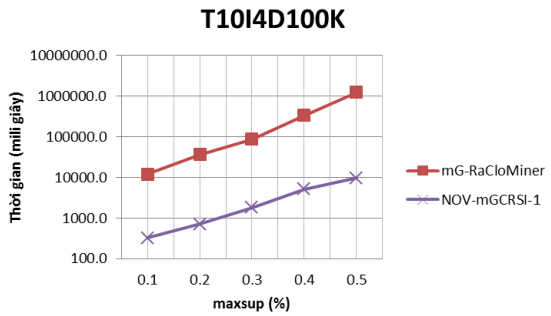
Hình 7.** Biểu đồ khai phá **mGCRSI** trên **Chess

Hình 7 - thực nghiệm so sánh hiệu quả về mặt thời gian từ tập dữ liệu **Chess** mật độ dày đặc (49,3%), cho thấy giải thuật **NOV-mGCRSI-1** cũng nhanh hơn giải thuật **mG-RaCloMiner**.



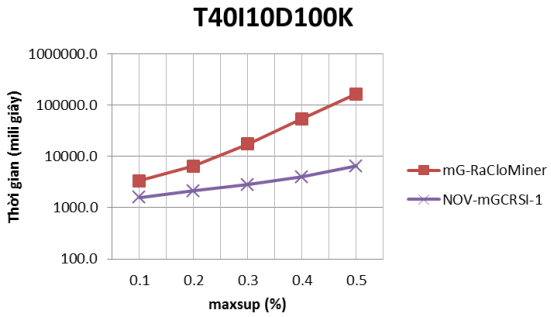
Hình 8.** Biểu đồ khai phá **mGCRSI** trên **Mushroom

Hình 8 - thực nghiệm so sánh hiệu quả về mặt thời gian từ tập dữ liệu **Mushroom** mật độ dày đặc (19,3%), giải thuật **NOV-mGCRSI-1** cũng nhanh hơn giải thuật **mG-RaCloMiner**.



Hình 9. *Biểu đồ khai phá mGCRSI trên T10I4D100K*

Hình 9 - thực nghiệm so sánh hiệu quả về mặt thời gian từ tập dữ liệu **T10I4D100K** mật độ rất thưa (1,1%), cho thấy giải thuật **NOV-mGCRSI-1** cũng nhanh hơn giải thuật **mG-RaCloMiner**.



Hình 10. *Biểu đồ khai phá mGCRSI trên T40I10D100K*

Hình 10 - thực nghiệm so sánh hiệu quả về mặt thời gian từ tập dữ liệu **T40I10D100K** mật độ rất thưa (4,2%), cho thấy giải thuật **NOV-mGCRSI-1** cũng nhanh hơn giải thuật **mG-RaCloMiner**.

Qua hai thực nghiệm trên, cho thấy giải thuật khai phá *tập sinh tối thiểu* **NOV-mGCRSI** hiệu quả hơn rất nhiều so với giải thuật **mGCRSI-RaCloMiner**. Giải thuật **NOV-mGCRSI** cần được thực nghiệm mở rộng trên các dữ liệu giao dịch có kích cỡ lớn.

6. Kết luận

Bài viết đã trình bày giải thuật **NOV-mGCRSI** khai phá hiệu quả *tập sinh tối thiểu của tập hiếm đóng* gồm ba bước: *đầu tiên* là phát sinh nhanh cấu trúc mảng **IndexCOOC** có chứa *items* xuất hiện đồng thời với *item-hạt-nhân* và *items* xuất hiện ít nhất với *item-hạt-nhân* trong một giao dịch; *bước thứ hai*: xây dựng **nLOOCTree** dựa vào mảng **IndexCOOC**; *giai đoạn thứ ba*: khai phá hiệu quả *tập sinh tối thiểu của tập hiếm đóng* dựa trên cây **nLOOCTree**. Kết quả thực nghiệm cho thấy giải thuật đề xuất hiệu quả hơn.

Trong các nghiên cứu tiếp theo, nhóm tác giả hướng đến việc nâng cao hiệu năng giải thuật tuần tự **NOV-mGCRSI** để khai phá hiệu quả *tập sinh tối thiểu của tập hiếm đóng có trọng số* trên bộ xử lý đa lõi, hệ thống phân tán phổ biến hiện nay như Hadoop, Spark.

TÀI LIỆU THAM KHẢO

[1] R. Agrawal, T. Imilienski and A. Swami, *Mining association rules between sets of large databases*, Proc. of the ACM SIGMOD Int Conf on Management of Data., 1993, pp. 207-216.

[2] J. Han, J. Pei, Y. Yin, R. Mao, "Mining frequent patterns without candidate generation: A FP-tree approach". *Data Mining Knowl Discovery*, 8(1), 2004, pp.53–87.

[3] Y. S. Koh, N. Rountree, *Finding sporadic rules using apriori-inverse*. In PAKDD'05, 3518, Springer, 2005, pp.97–106.

[4] L. Szathmary, P. Valtchev, A. Napoli, R. Godin, *Efficient vertical mining of minimal rare itemsets*. 19th Int Conf on Concept Lattices and Their Apps, 2012, pp.269–280.

[5] L. Troiano, C. Birtolo, *A fast algorithm for mining rare itemsets*. 19th Int Conf on Intell Syst Design & App, 2009, pp.1149-1155.

[6] C.H. Cai, A.W. Fu, C.H. Cheng, W.W. Kwong, *Mining association rules with weighted items*. Proc of Int Database Engineering and App Symp (IDEAS 98), 1998, pp.68–77.

[7] Z. Huai, M. Huang, *A weighted frequent itemsets incremental updating algorithm base on hash table*. In 3rd Int Conf on Comm Soft and Networks (ICCSN), IEEE, 2011, pp.201–204.

[8] A. Borah, B. Nath, "Rare pattern mining: challenges and future perspectives". *Complexi Intell Syst*, Springer, 2018, pp.1–23.

[9] Y. Lu, T. Seidl, *Towards Efficient Closed Infrequent Itemset Mining Using Bi-Directional Traversing*. IEEE 5th DSAA, Turin, Italy, 2018, pp. 140-149.

[10] Phan Thành Huấn, "Giải thuật hiệu năng cao khai thác tập sinh của tập phổ biến đóng". *Tạp chí Khoa học và Công nghệ - Đại học Đà Nẵng*, 18(5.2), 2020, pp. 55-60.