DUP-APRIORI: THUẬT TOÁN HIỆU QUẢ KHAI THÁC TẬP PHỔ BIẾN DỰA TRÊN GIAO DỊCH TRÙNG LẶP

DUP-APRIORI: AN EFFICIENT ALGORITHM FOR MINING FREQUENT ITEMSETS BASED ON DUPLICATE TRANSACTIONS

Phan Thành Huấn*

Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia tp. Hồ Chí Minh[1]

*Tác giả liên hệ: huanphan@hcmussh.edu.vn

(Nhận bài: 15/9/2022; Chấp nhận đăng: 03/11/2022)

Tóm tắt - Thuật toán Apriori là thuật toán kinh điển được dùng cho khai thác tập phổ biến từ dữ liệu giao dịch nhị phân – giai đoạn quan trọng trong khai thác luật kết hợp. Đây là thuật toán được nhiều nhóm nghiên cứu quan tâm cải tiến, cũng như sử dụng khai thác trên nhiều loại dữ liệu khác nhau. Trong bài viết này, tác giả trình bày tiếp cận mới trong cải tiến hiệu quả thuật toán Apriori dựa trên giao dịch trùng lặp - giúp đẩy nhanh tốc độ tính toán và giảm thiểu quá trình truy xuất dữ liệu. Thuật toán cải tiến được gọi là **DUP-Apriori**. Tác giả tiến hành thực nghiệm thuật toán trên bộ dữ liệu thực của UCI và dữ liệu giả lập của trung tâm nghiên cứu IBM Almaden, cho thấy thuật toán cải tiến hiệu quả so với thuật toán gần đây.

Từ khóa - Luật kết hợp; tập phổ biến; thuật toán DUP-Apriori

Abstract - The Apriori algorithm is the classic algorithm used for frequent itemset mining from binary dataset - important phase in association rule mining. This is an algorithm that many research groups are interested in improving, as well as using mining on many different types of dataset. In this paper, the author presents a new approach in improving the efficiency of the Apriori algorithm based on duplicate transactions - to speed up computation and reduce database access. The improved algorithm is called DUP-Apriori. Experimenting the algorithm on real dataset of UCI and simulated dataset of IBM Almaden research center, shows that the algorithm improves efficiency compared to the recent algorithm.

Key words - Association rules; frequent itemsets; DUP-Apriori algorithm

1. Đặt vấn đề

Năm 1993, Agrawal cùng đồng sự đề xuất mô hình đầu tiên của bài toán khai thác luật kết hợp – khai thác luật kết hợp trên dữ liệu giao dịch (DLGD) nhị phân [1]. 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*). Bài toán được chia thành hai pha [1-15]:

Pha 1: Tìm tất cả các kết hợp thỏa ngưỡng phổ biến tối thiểu *minsup* (sinh tập phổ biến **FI** - **F**requent **I**temset);

Pha 2: Sinh luật kết hợp lần lượt từ các kết hợp thỏa *minsup* ở pha 1 và các luật kết hợp này phải thỏa ngưỡng tin cậy tối thiểu *minconf*.

Năm sau đó, Agrawal cùng đồng sự tập trung hướng giải quyết cho *pha 1* và nhóm đã đề xuất thuật toán Apriori [2] cho khai thác tập phổ biến. Đây là thuật toán then chốt, quan trọng trong khai thác luật kết hợp. Thuật toán tiếp cận sinh các kết hợp phổ biến với chiến lược tìm kiếm theo chiều rộng (**B**readth **F**irst **S**earch – BFS) dễ dàng cài đặt và song song hóa nhằm nâng cao hiệu năng; Thuật toán tốn nhiều lần quét dữ liệu và có độ phức tạp dạng hàm mũ. Chính vì vậy, Apriori là thuật toán được nhiều nhà nghiên cứu cải tiến và áp dụng khai phá trên nhiều loại dữ liệu khác nhau: *Chuỗi* [4], *định lượng* [5], *đồ thị* [6], *thuộc tính có trọng số* [7],…

Hai hướng tiếp cận chính của các nghiên cứu liên quan đến cải tiến thuật toán Apriori:

- *Định dạng dữ liệu theo chiều ngang*: Đây là định dạng theo thuật toán Apriori gốc. Các thuật toán cải tiến Apriori thường sử dụng chiến lược rút gọn giao dịch và rút gọn không gian sinh các ứng viên tiềm năng k-*itemset*. Tuy nhiên, vấn đề tính độ phổ biến của k-*itemset* vẫn chưa thật sự hiệu quả. Một số thuật toán cải tiến Apriori áp dụng định dạng dữ liệu theo chiều ngang: SOT-Apriori [10], MBAT [11], CBTRA [12], LOT-Apriori [13], NOV-Apriori [15]…

- *Định dạng dữ liệu theo chiều dọc*: Định dạng này, giúp tính độ phổ biến dễ dàng và hạn chế đối với DLGD có mật độ cao. Một số thuật toán cải tiến Apriori áp dụng định dạng dữ liệu theo chiều dọc: Parition [8], IApriori [9], MD-Apriori [14]…

Quá trình khảo sát, tác giả thấy rằng: DLGD thực tế có tần số trùng lặp của giao dịch trước và sau khi loại bỏ các item không thỏa ngưỡng *minsup* là tương đối cao. Vì vậy, tác giả đề xuất tiếp cận mới trong cải tiến hiệu quả thuật toán Apriori dựa trên giao dịch trùng lặp.

2. Các vấn đề liên quan

2.1. Khai thác tập phổ biến

Cho I = $\{i_1, i_2, ..., i_m\}$ là tập gồm *m* thuộc tính, mỗi thuộc tính gọi là *item*. Với $X \subseteq I$, $X = \{i_1, i_2, ..., i_k\}$, $\forall i_j \in I$ $(1 \le j \le k)$ gọi là *itemset*, itemset có *k* item gọi là *k-itemset*. 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 $t_k = \{i_{k1}, i_{k2}, ..., i_{km}\}$, $i_{kj} \in I$ $(1 \le k_j \le m)$.

Định nghĩa 1: *Độ phổ biến* (support) của *itemset* $X \subseteq I$, ký hiệu *sup*(X) - tỷ lệ giữa số giao dịch có chứa itemset X và *n* giao dịch.

$$sup(X) = \left|\{t \in T \mid X \subseteq t\}\right| / n$$

[1] Vietnam National University Ho Chi Minh City - University of Science (Huan Phan)

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

Các *tính chất bao đóng giảm* trong khai thác tập phổ biến trên DLGD:

Tính chất 1: $\forall X \subseteq Y$: $sup(X) \geq sup(Y)$;

Tính chất 2: $\forall X \subset Y, sup(Y) \geq minsup$: $sup(X) \geq minsup$;

Tính chất 3: $\forall X \subset Y, sup(X) < minsup$: $sup(Y) < minsup$.

Cho dữ liệu giao dịch sử dụng ở các Ví dụ.

Bảng 1. *Dữ liệu giao dịch T*

TID	Items
t1	A B C E F
t2	A C G
t3	E H
t4	A C D G
t5	A C E G
t6	E H
t7	A B C E F
t8	A C D
t9	A C E G
t10	A C E G

Ví dụ 1: Dữ liệu giao dịch trong Bảng 1, có 8 *item* riêng biệt $I = \{A, B, C, D, E, F, G, H\}$ và tập giao dịch $T = \{t1, t2, t3, t4, t5, t6, t7, t8, t9, t10\}$ với giá trị ngưỡng phổ biến tối thiểu *minsup* = 0,50, ta có:

Theo ***tính chất 1***: $X = \{G, A, C\}$, $sup(GAC) = 0,50$ – độ phổ biến lần lượt các tập con của X: $sup(A) = sup(C) = sup(AC) = 0,80$; $sup(G) = sup(GA) = sup(GC) = 0,50$.

Theo ***tính chất 2***: Các tập con của $X = \{G, A, C\}$ cũng phổ biến; ta thấy độ phổ biến của các tập con của X đều lớn hơn hoặc bằng ngưỡng *minsup*.

Theo ***tính chất 3***: $Y = \{F\}$ thì $sup(F) = 0,20 < minsup$ - "$Y = \{F\}$ *itemset không phổ biến* ngưỡng *minsup*". Khi đó, các tập cha của Y cũng không phổ biến, nghĩa là $Z = \{F, E\}$ cũng *không phổ biến*, $sup(FE) = 0,20 < minsup$.

Bảng 2. *FIs trên dữ liệu giao dịch T, minsup = 0,50*

k-itemset	Tập phổ biến FIs (#FIs = 11)
1	(G; 0,50), (E; 0,70), (A; 0,80), (C; 0,80)
2	(GA; 0,50), (GC; 0,50), (EA; 0,50), (EC; 0,50), (AC; 0,80)
3	(GAC; 0,50), (EAC; 0,50)

Ở Bảng 2, trình bày *k-itemset* phổ biến trên DLGD với ngưỡng *minsup* = 0,50; các *k-itemset* phổ biến được sắp xếp tăng dần theo độ phổ biến của các items ($B \prec D \prec H \prec F \prec G \prec E \prec A \prec C$) và có 11 itemset phổ biến.

2.2. Thuật toán NOV-Apriori

2.2.1. Đóng góp của thuật toán

Trong phần này, tác giả trình bày thuật toán **NOV-Apriori** [15] cải tiến dựa trên tiếp cận thuật toán **AprioriTID** [2] và cho thấy hiệu quả khả quan so với thuật toán cải tiến gần đây, các đóng góp của thuật toán:

Thứ nhất, sắp xếp các item theo thứ tự tăng dần của độ phổ biến – sử dụng *tính chất 3* cho việc rút gọn các kết hợp ở bước tiếp theo (item đầu tiên trong các kết hợp là item có độ phổ biến nhỏ nhất).

Thứ hai, cải tiến thủ tục AprioriGen sinh các ứng viên bằng cách sắp xếp các (k-1)-*itemset* phổ biến theo thứ tự và sinh các kết hợp mới giúp giảm dư thừa và trùng lặp.

Thứ ba, thực hiện tính độ phổ biến cho các ứng viên tiềm năng C_k theo nhóm item đầu dựa trên ma trận bit T_k tương ứng được rút gọn theo dòng (giao dịch) ở mỗi bước sinh k-*itemset* phổ biến.

Một số ký hiệu trong thuật toán NOV-Apriori:

- L_k: Tập thành viên chứa k-*itemset* thỏa *minsup*, mỗi thành viên có 4 trường thông tin là itemset và độ phổ biến *sup*, bổ sung thêm thứ tự nhỏ nhất (*min*) và lớn nhất (*max*) của item trong mỗi itemset thuộc L_k;

- C_k: Tập ứng viên chứa k-*itemset* tiềm năng, mỗi ứng viên có 4 trường thông tin là itemset biểu diễn dạng bit, độ phổ biến *sup*, thứ tự nhỏ nhất (*min*) và lớn nhất (*max*) của item trong mỗi itemset thuộc C_k;

- T_k: Tập giao dịch được biểu diễn dạng bit, mỗi giao dịch dạng bit có thêm 3 trường thông tin là |t| số lượng item trong giao dịch, thứ tự nhỏ nhất (*min*) và thứ tự lớn nhất (*max*) là thứ tự item *đầu*, *cuối* mỗi giao dịch.

Mã giả thuật toán NOV-Apriori

Đầu vào: Tập giao dịch **T**, ngưỡng *minsup*

Đầu ra: Tập phổ biến **FI**

1. L_1 = {1-itemset}//thứ tự tăng theo *sup*
2. T_1 = tập T chỉ chứa các item có trong L_1 và có |t| > 1 //T_k biểu diễn dạng bit và có thứ tự theo min, max
3. $C_2 = \{L_1 \times L_1\}$//sinh ứng viên tiềm năng 2-itemset
4. k = 2
5. **Do**
6. **For each** c ∈ C_k **do** //tính sup theo nhóm giao dịch
7. i = 1
8. **Do** //t ∈ T_{k-1}
9. **If** (t[i].*min* ≤ c.*min* ∧ c.*max* ≤ t[i].*max*) **then**
10. **If**(c.*itemset*==c.*itemset* AND t[i].*itemset*)**then**
11. c.*sup* += 1/*n*
12. i++
13. **While** (t[i].*min* ≤ c.*min*)
14. $L_k = \{c \in C_k | c.sup \geq minsup\}$//lọc ứng viên thỏa
15. $T_k = T_{k-1} - \{t \in T_{k-1} | |t| \leq k\}$//rút gọn dữ liệu
16. k++
17. C_k = **AprioriGen***(L_{k-1})
18. **While** ($|L_{k-1}| \geq k$)
19. Trả về **FI**= $\cup_k L_k$

Mô tả thuật toán NOV-Apriori:

Dòng 1 và 2, sinh tập L_1 chứa các item thỏa ngưỡng *minsup* và rút gọn tập giao dịch biểu diễn dạng bit thành T_1 (loại bỏ các giao dịch chỉ có 1 item). Dòng 3, sinh tập C_2 chứa ứng viên tiêm năng 2-*itemset* từ L_1. Từ dòng 6 đến 13, mỗi ứng viên tiềm năng được tính tần suất trên các giao

dịch có chứa item đầu tiên (*min*) – không quét trên tất cả giao dịch như một số thuật toán cải tiến khác. Dòng 14 và 15, lọc các ứng viên thỏa *minsup* và rút gọn dữ liệu cho bước lặp tiếp theo. Dòng 17, sinh tập C_k ứng viên tiềm năng k-*itemset* từ L_{k-1} cho bước lặp thứ k.

Thủ tục AprioriGen* - sinh các ứng viên *k-itemset* tiềm năng C_k từ tập (k-1)-*itemset* L_{k-1}:

Mã giả thủ tục AprioriGen*

Đầu vào: Tập chứa các (*k-1*)-*itemset* phổ biến L_{k-1}

Đầu ra: Tập ứng viên *k-itemset* C_k

1. $C_k = \varnothing$
2. i = 1
3. **While** (i < $|L_{k-1}|$) **do**
4. j = i + 1
5. **Do**
6. **If** (X_i.*min* == X_j.*min*) **then**//$X_i, X_j \in L_{k-1}$
7. $C_k = \{C_k \cup \{X_i \cup X_j\} | \{X_i \cup X_j\} \notin C_k\}$
8. j++
9. **Else**
10. i = j
11. **While** (i ≠ j)
12. Trả về C_k

2.2.2. Minh họa thuật toán NOV-Apriori

Ví dụ 2: Cho tập giao dịch T trong Bảng 1, ngưỡng *minsup* = 0,50.

Bảng 3. *Dữ liệu giao dịch T được rút gọn theo item*

TID	G	E	A	C	min	max	|t|
t2	**1**	0	**1**	**1**	1	4	3
t4	**1**	0	**1**	**1**	1	4	3
t5	**1**	**1**	**1**	**1**	1	4	4
t9	**1**	**1**	**1**	**1**	1	4	4
t10	**1**	**1**	**1**	**1**	1	4	4
t3	0	**1**	0	0	2	2	1
t6	0	**1**	0	0	2	2	1
t1	0	**1**	**1**	**1**	2	4	3
t7	0	**1**	**1**	**1**	2	4	3
t8	0	0	**1**	**1**	3	4	2

Dữ liệu T_1 được sắp xếp theo *min, max*

T_1 ={t2, t4, t5, t9, t10, t1, t7, t8} – *loại* {t3, t6};

C_2: Tập ứng viên tiềm năng 2-*itemset*

GE	GA	GC	EA	EC	AC
1 2	1 3	1 4	2 3	2 4	3 4

Bước lặp k = 2: Tính *sup* cho từng ứng viên C_2;

GE	GA	GC	EA	EC	AC
1 2	1 3	1 4	2 3	2 4	3 4
0,30	0,50	0,50	0,50	0,50	0,80

L_2: Tập 2-*itemset* phổ biến

GA	GC	EA	EC	AC
1 3	1 4	2 3	2 4	3 4
0,50	0,50	0,50	0,50	0,80

T_2 ={t2, t4, t5, t9, t10, t1, t7} – *loại* {t8};

C_3: Tập ứng viên tiềm năng 3-*itemset*

GAC	EAC
1 4	2 4

Bước lặp k = 3: Tính *sup* cho từng ứng viên C_3;

GAC	EAC
1 4	2 4
0,50	0,50

L_3: Tập 3-*itemset* phổ biến

GAC	EAC
1 4	2 4
0,50	0,50

T_3 ={t5, t9, t10} – *loại* {t2, t4, t1, t7};

C_4: Tập ứng viên tiềm năng 4-*itemset*

GEAC
1 4

Bước lặp k = 4: Tính *sup* cho từng ứng viên C_4;

GEAC
1 4
0,30

L_4 = {∅}, thuật toán kết thúc.

Tổng số giao dịch duyệt ở 4 bước lặp: 8 + 7 + 3 = 18.

3. Thuật toán cải tiến

Trong phần này, tác giả trình bày thuật toán cải tiến **DUP-Apriori** và minh họa thuật toán, cho thấy cải tiến đề xuất là hiệu quả.

3.1. Thuật toán DUP-Apriori

Thuật toán **NOV-Apriori** [15] đã rút gọn giao dịch sao mỗi bước sinh k-*itemset* tiềm năng dựa vào 3 trường thông tin là *min*, *max* và |*t*|. Tuy nhiên, trong thực tế các dữ liệu giao dịch luôn tồn tại nhiều giao dịch trùng lặp. Vì vậy, tác giả đề xuất phương pháp tính nhanh độ phổ biến của k-*itemset* dựa vào *tần số trùng lặp* của các giao dịch trong dữ liệu.

- T_k: Tập giao dịch được biểu diễn dạng bit, mỗi giao dịch dạng bit có 3 trường thông tin là |*t*| số lượng items trong giao dịch, thứ tự nhỏ nhất (*min*), thứ tự lớn nhất (*max*) là thứ tự item *đầu, cuối* trong mỗi giao dịch như thuật toán **NOV-Apriori** và được bổ sung thêm trường thông tin *dup* (≥ 1) lưu trữ *tần số trùng lặp* của giao dịch trong dữ liệu.

Mã giả thuật toán DUP-Apriori

Đầu vào: Tập giao dịch **T**, ngưỡng *minsup*

Đầu ra: Tập phổ biến **FI**

1. L_1 = {1-itemset}//thứ tự tăng theo *sup*
2. T_1 = tập T chứa các item có trong L_1 và có |t| > 1 và *gom các giao dịch trùng lặp*;

…

11. c.*sup* += t[i].*dup*/*n*

…

Mô tả thuật toán DUP-Apriori

Thuật toán **DUP-Apriori** được cải tiến từ **NOV-Apriori**, chi tiết cải tiến: Dòng 2 – gồm các giao dịch trùng lặp từ dữ liệu T, mỗi giao dịch sẽ có thêm 4 trường thông tin là *min*, *max*, |*t*| và *dup*; Ở dòng 11 – tính nhanh độ phổ biến của itemset tiềm năng thông qua trường thông tin *dup* (tần số trùng lặp của giao dịch) của từng dòng giao dịch.

3.2. Minh họa thuật toán DUP-Apriori

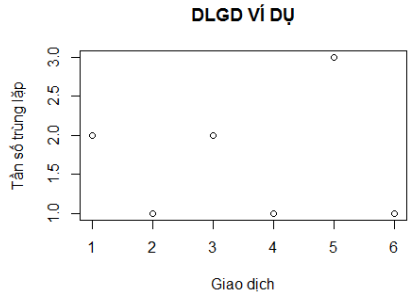
Trong phần này, tác giả minh họa thuật toán **DUP-**

Apriori khai thác tập phổ biến trên DLGD, cho thấy thuật toán cải tiến hiệu quả được so sánh thông qua không gian duyệt các giao dịch ở mỗi bước sinh các itemset phổ biến.

Ví dụ 3: Cho tập giao dịch T trong Bảng 1, ngưỡng *minsup* = 0,50.

Bảng 4. *Dữ liệu T được gom theo giao dịch trùng lặp*

TID	Items	Trùng lặp
t1,t7	A B C E F	2
t2	A C G	1
t3,t6	E H	2
t4	A C D E G	1
t5,t9,t10	A C E G	3
t8	A C D	1



Hình 1. *Minh họa giao dịch trùng lặp trên dữ liệu Ví dụ*

Hình 1, cho thấy DLGD *T* sau khi thực hiện thuật toán 1: có 3 giao dịch tần số xuất hiện 1 lần là giao dịch {t2}, {t4}, {t8}, giao dịch có tần số xuất hiện là 2 lần là giao dịch {t1, t7}, {t3, t6} và giao dịch có tần số xuất hiện 3 lần là giao dịch {t5, t9, t10}.

Bảng 5. *Dữ liệu T được rút gọn và gom theo giao dịch*

TID	G	E	A	C	min	max	|t|	dup
t2	**1**	0	**1**	**1**	1	4	3	1
t4	**1**	0	**1**	**1**	1	4	3	1
t5 (t9, t10)	**1**	**1**	**1**	**1**	1	4	4	3
t3 (t6)	0	**1**	0	0	2	2	1	2
t1 (t7)	0	**1**	**1**	**1**	2	4	3	2
t8	0	0	**1**	**1**	3	4	2	1

Dữ liệu T_1 được sắp xếp theo *min, max*

T_1 ={t2, t4, t5, t1, t8} – *loại* {t3};

C_2: Tập ứng viên tiềm năng 2-*itemset*

GE	GA	GC	EA	EC	AC
1 2	1 3	1 4	2 3	2 4	3 4

Bước lặp k = 2: Tính *sup* cho từng ứng viên C_2;

GE	GA	GC	EA	EC	AC
1 2	1 3	1 4	2 3	2 4	3 4
0,30	0,50	0,50	0,50	0,50	0,80

L_2: Tập 2-*itemset* phổ biến

GA	GC	EA	EC	AC
1 3	1 4	2 3	2 4	3 4
0,50	0,50	0,50	0,50	0,80

T_2 ={t2, t4, t5, t1} – *loại* {t8};

C_3: Tập ứng viên tiềm năng 3-*itemset*

GAC	EAC
1 4	2 4

Bước lặp k = 3: Tính *sup* cho từng ứng viên C_3;

GAC	EAC
1 4	2 4
0,50	0,50

L_3: Tập 3-*itemset* phổ biến

GAC	EAC
1 4	2 4
0,50	0,50

T_3 ={t5} – *loại* {t2, t4, t1};

C_4: Tập ứng viên tiềm năng 4-*itemset*

GEAC
1 4

Bước lặp k = 4: Tính *sup* cho từng ứng viên C_4;

GEAC
1 4
0,30

L_4: Tập 4-*itemset* phổ biến

Tổng số giao dịch duyệt ở 4 bước lặp: 5 + 4 + 1 = 10, so với tổng số giao dịch duyệt theo thuật toán **NOV-Apriori** là ít hơn 44,44% (tương ứng 8/18).

4. Kết quả thực nghiệm

Thực nghiệm trên máy tính Core i7-3540M 3.0 GHz, 4GB RAM, thuật toán cài đặt trên MSVC# 2015.

4.1. Mô tả dữ liệu thực nghiệm

Nghiên cứu thực nghiệm trên 2 *nhóm dữ liệu*

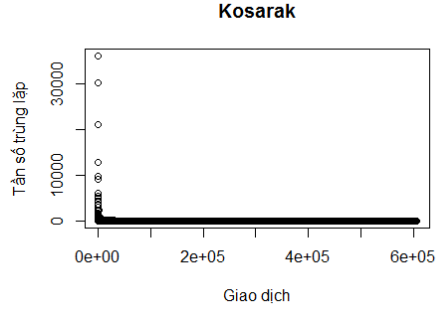
- Nhóm dữ liệu thực: Từ kho dữ liệu về học máy UCI của trường Đại học California gồm **Kosarak** và **Retail**.

- Nhóm dữ liệu giả lập: Dùng phần mềm phát sinh dữ liệu giả lập của trung tâm nghiên cứu IBM Almaden gồm **T10I4D100K** và **T40I10D100K**.

Bảng 6. *Dữ liệu thực nghiệm*

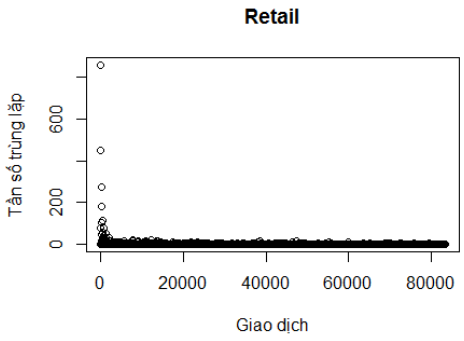
Dữ liệu	Số item	Số lượng giao dịch	Mật độ (%)	Trùng lặp (%)
Kosarak	41.270	990.002	0,02	38,71
Retail	16.470	88.162	0,06	5,30
T10I4D100K	870	100.000	1,16	10,87
T40I10D100K	942	100.000	4,20	0,07

Bảng 6, mô tả 4 tập dữ liệu sử dụng trong thực nghiệm, gồm các thông số như số lượng các item, số lượng giao dịch, mật độ của tập dữ liệu và mức độ trùng lặp của các giao dịch trong từng tập dữ liệu.



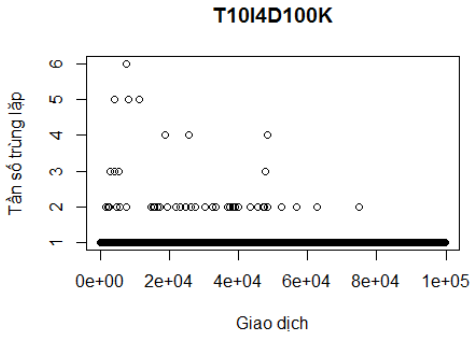
Hình 2. *Minh họa giao dịch trùng lặp trên Kosarak*

Hình 2 cho thấy, dữ liệu **Kosarak** được gom theo tần số trùng lặp; **Kosarak** chứa 990.002 giao dịch và có 383.232 giao dịch trùng lặp, xấp xỉ 38,71% dữ liệu.



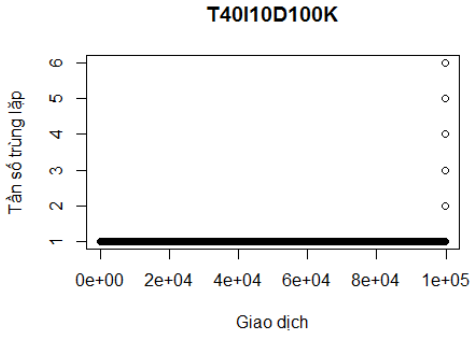
Hình 3. *Minh họa giao dịch trùng lặp trên Retail*

Hình 3, cho thấy dữ liệu **Retail** được gom theo tần số trùng lặp; **Retail** chứa 88.162 giao dịch và có 4.672 giao dịch trùng lặp, xấp xỉ 5,30% giao dịch trên dữ liệu.



Hình 4. *Minh họa giao dịch trùng lặp trên T10I4D100K*

Hình 4, cho thấy dữ liệu **T10I4D100K** được gom theo tần số trùng lặp; **T10I4D100K** chứa 100.000 giao dịch và có 10.865 giao dịch trùng lặp, xấp xỉ 10,87% dữ liệu.

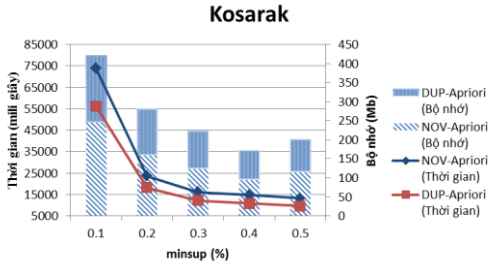


Hình 5. *Minh họa giao dịch trùng lặp trên T40I10D100K*

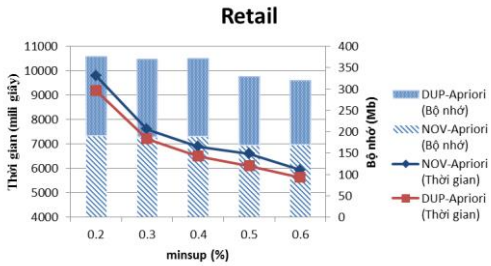
Hình 5, cho thấy dữ liệu **T40I10D100K** được gom theo tần số trùng lặp; **T40I10D100K** chứa 100.000 giao dịch và có 69 giao dịch trùng lặp, xấp xỉ 0,07% dữ liệu.

4.2. Thực nghiệm

Để đánh giá mức độ hiệu quả của thuật toán **DUP-Apriori**, tác giả so sánh thuật toán **DUP-Apriori** khai thác tập phổ biến trên DLGD với thuật toán **NOV-Apriori** [15] cùng hướng tiếp cận chiến lược *tìm kiếm theo chiều rộng*. Cả hai thuật toán đều cho cùng kết quả trên các ngưỡng *minsup* khác nhau.

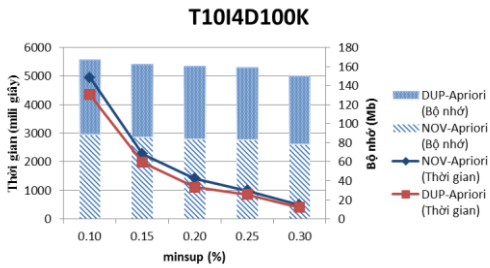


Hình 6. *Thời gian thực hiện và bộ nhớ sử dụng trên Kosarak*



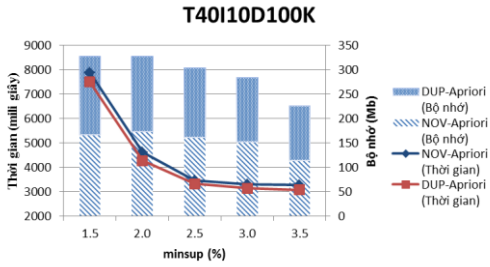
Hình 7. *Thời gian thực hiện và bộ nhớ sử dụng trên Retail*

Hình 6 và 7 là kết quả thực nghiệm trên nhóm dữ liệu thực, cho thấy thuật toán **DUP-Apriori** có thời gian thực hiện nhanh hơn và bộ nhớ sử dụng ít hơn trên các ngưỡng *minsup* với thuật toán **NOV-Apriori.**



Hình 8. *Thời gian thực hiện và bộ nhớ sử dụng trên T10I4D100K*

Hình 8 là kết quả thực nghiệm trên nhóm dữ liệu giả lập **T10I4D100K**, cho thấy thuật toán **DUP-Apriori** có thời gian thực hiện nhanh hơn và bộ nhớ sử dụng ít hơn với thuật toán **NOV-Apriori.**



Hình 9. *Thời gian thực hiện và bộ nhớ sử dụng trên dữ liệu T40I10D100K*

Hình 9 là kết quả thực nghiệm trên nhóm dữ liệu giả lập **T40I10D100K** cho thấy, thuật toán **DUP-Apriori** chưa thực sự hiệu quả so với thuật toán **NOV-Apriori** khi thực hiện trên dữ liệu có tỷ lệ giao dịch trùng lặp thấp**.**

Kết quả thực nghiệm cho thấy, thuật toán cải tiến **DUP-**

Apriori hiệu quả hơn thuật toán **NOV-Apriori** và mức độ hiệu quả phụ thuộc vào tỷ lệ trùng lặp giao dịch của tập dữ liệu. Ngoài ra, thuật toán cũng cần thực nghiệm so sánh thêm với các thuật toán theo hướng tiếp cận theo chiều sâu (**D**epth **F**irst **S**earch - DFS), cùng với nhiều tập dữ liệu có mật độ cao khác.

5. Kết luận và hướng phát triển

Trong bài viết này, tác giả đề xuất phương pháp gom các giao dịch trùng lặp, giúp thuật toán tính nhanh độ phổ biến của các itemset ở mỗi bước sinh k-*itemset* tiềm năng - giảm số lần duyệt giao dịch. Phần thực nghiệm, cho thấy tính hiệu quả của thuật toán **DUP-Apriori** cả về mặt thời gian thực hiện và bộ nhớ sử dụng so với thuật toán cải tiến gần đây. Tuy nhiên, hiệu suất của thuật toán phụ thuộc vào tỷ lệ trùng lặp của giao dịch (tỷ lệ thuận), trước và sau khi loại bỏ các item không thỏa *minsup*.

Nghiên cứu trong thời gian tới của tác giả là nghiên cứu và đề xuất kỹ thuật hiệu quả tính nhanh độ phổ biến của các item, cũng như mở rộng thuật toán khai thác tập phổ biến hiệu quả cho dữ liệu lớn dựa trên nền tảng điện toán phân tán như Hadoop, Spark,...

TÀI LIỆU THAM KHẢO

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

[2] R. Agrawal, R. Srikant, *Fast Algorithms for Mining Association Rules in Large Databases*, VLDB 1994, pp. 487-499.

[3] P. Huan, L. Bac, *A Novel Algorithm for Frequent Itemsets Mining in Transactional Databases*, PAKDD 2018. LNCS, 11154, Springer Cham, 2018, pp. 243–255.

[4] R. Agrawal, R. Srikant, *Mining sequential patterns*, Proc of the 11th Inter Conf on Data Engineering, 1995, pp. 3-14.

[5] C.L. Carter, H.J. Hamilton, N. Cercone, *Share Based Measures for Itemsets*, PKDD1997, 1997, pp. 14-24

[6] A. Inokuchi, T. Washio, H. Motoda, *An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data*, PKDD'00, 1910, 2000, pp. 13-23

[7] G. C. Lan, T. P. Hong, H. Y. Lee, and C. W. Lin, *Mining Weighted Frequent Itemsets*, Proc of the 30th workshop on Combinatorial Mathematics and Computation Theory (Alg'30), 2013, pp. 85-89.

[8] A. Savasere, E. Omiecinski, S.B. Navathe, *An Efficient Algorithm for Mining Association Rules in Large Databases*, VLDB1995, 1995, pp. 432-444.

[9] Y. Guo, Z. Wang, *A vertical format algorithm for mining frequent itemsets*, 2nd International Conference on Advanced Computer Control, 4, 2010, pp. 11-13.

[10] J. Singh, H. Ram, "Improving Efficiency of Apriori Algorithm Using Transaction Reduction", *Int Journal of Scientific and Research Publications*, 3(1), 2013, pp.1-4.

[11] H. Singh, R. Dhir, "A New Efficient Matrix Based Frequent Itemset Mining Algorithm with Tags", *Int Journal of Future Computer and Communication*, 2013, pp. 355-358.

[12] V. Vijayalakshmi, A. Pethalakshmi, "An Efficient Count Based Transaction Reduction Approach for Mining Frequent Patterns", *Procedia Computer Science*, 47, 2015, pp. 52-61.

[13] S. Aditya, M. Hemanth, C.K. Lakshmikanth, K. Suneetha, *Effective algorithm for frequent pattern mining*, 2017 Inter Conf on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2017, pp. 704-708.

[14] L. Xu, L. Qiao, F. Zhao, B. Yang, Q. Wang, P. Ding, L. Li, *Improvement and Application of Apriori Algorithm Based on Equalization*, IEEE Fourth International Conference on Data Science in Cyberspace (DSC), 2019, pp. 635-641.

[15] Phan Thành Huấn, Lê Hoài Bắc, *Tiếp cận mới trong cải tiến hiệu quả thuật toán Apriori cho khai thác luật kết hợp*, Hội thảo Quốc gia lần thứ XXIV - Một số vấn đề chọn lọc của CNTT và Truyền thông, 2021, pp. 478-483.