KHAI THÁC LUẬT TUẦN TỰ TRÊN CƠ SỞ DỮ LIỆU TUẦN TỰ

Nguyễn Thành Ngô

Trường Đại học Công nghiệp Thực phẩm Tp. HCM

Ngày gửi bài: 15/9/2014 **Ngày chấp nhận đăng: 15/5/2015**

TÓM TẮT

Bài báo đề xuất phương pháp khai thác luật tuần tự từ cơ sở dữ liệu tuần tự. Trước tiên xây dựng luật tuần tự hợp lệ từ các tập phổ biến dựa trên khai thác luật kết hợp của thuật toán Apriori (Apriori-TID) và sau đó loại bỏ các luật dư thừa. Kết quả đạt được các tập luật tuần tự không dư thừa nhưng vẫn đảm bảo tính đầy đủ.

Từ khóa: Khai thác luật tuần tự, Khai thác dữ liệu, Khai thác luật kết hợp, Tập phổ biến

ABSTRACT

This paper proposes a method for mining sequential rules in sequential datasets. Firstly, we create valid sequential rules from the frequent itemsets based on association rule mining by the Apriori (Apriori-TID) and then pruning the redundant rules. The result of the sets of sequential rules are not redundant but still ensure complete.

Key words: Data mining, sequential rule mining, association rule mining, frequent itemsets.

1. GIỚI THIỆU

Ngày nay, một lượng lớn thông tin tuần tự được lưu trữ trong cơ sở dữ liệu (dữ liệu thị trường chứng khoán, hồ sơ y tế, dữ liệu y học, dữ liệu của khách hàng, v.v.). Khám phá mối quan hệ tuần tự trong cơ sở dữ liệu rất quan trọng trong nhiều lĩnh vực, vì nó cung cấp sự hiểu biết tốt hơn về dữ liệu, và thiết lập một cơ sở cho việc dự đoán.

Nhiều phương pháp đã được đề nghị để khai thác mối quan hệ tuần tự trong dữ liệu.

Trong lĩnh vực data mining, kỹ thuật phổ biến nhất để phát hiện các mối quan hệ tuần tự trong chuỗi thời gian rời rạc là khai thác mẫu tuần tự. Thuật toán SPM [4, 9, 10], tìm các mẫu tuần tự, chuỗi nhỏ xuất hiện thường xuyên trong cơ sở dữ liệu tuần tự (một tập hợp của các trình tự). Tuy nhiên, một chuỗi xuất hiện thường xuyên trong một cơ sở dữ liệu trình tự không đủ để dự đoán.

Trong hầu hết các thuật toán khai thác luật, các tác giả đặc biệt chú ý đến vấn đề làm thế nào để khai thác nhanh tập phổ biến. Chính vì vậy, có khá nhiều tác giả chỉ tập trung vào việc nghiên cứu nhằm tìm ra thuật toán hiệu quả nhất cho bài toán tìm tập phổ biến. Tuy nhiên, với các CSDL đặc (mật độ trùng lặp các item giữa các dòng dữ liệu cao) hoặc khi minsup nhỏ dẫn đến số lượng tập phổ biến khá lớn thì thời gian khai thác và khối lượng bộ nhớ yêu cầu để lưu trữ tập phổ biến và luật kết hợp khá lớn. Vì vậy, các tác giả M. Zaki [11] và Bastide [5] đã đưa ra một cách tiếp cận mới nhằm làm giảm khối lượng lưu trữ và thời gian khai thác đó chính là khai thác luật kết hợp không dư thừa dựa vào tập đóng.

Trong bài báo này chúng tôi đề xuất phương pháp khai thác luật tuần tự từ dữ liệu chuỗi thông qua thuật toán CMRules[8] để khai thác luật tuần tự và kết hợp thuật toán MNARS[6] khai thác luật kết hợp không dư thừa tối thiểu có ứng dụng tập dàn để làm gọn chuỗi dữ liệu tuần tự.

2. CÁC KHÁI NIỆM CƠ BẢN VÀ MỘT SỐ ĐỊNH NGHĨA

- Một CSDL giao dịch D gồm:

  + Tập các giao dịch $T = \{t_1, t_2, \ldots, T_n\}$ và các giao dịch định danh (Tids) trong một cơ sở dữ liệu D. Cơ sở dữ liệu đầu vào là nhị phân mối quan hệ $\delta \subseteq I \times T$. Nếu một mục i xảy ra trong một t giao dịch, chúng tôi viết nó như là $(i, t) \in \delta$ hoặc $i\delta t$.

  + Tập các item I = {i1, i2, ..., im}, trong đó: t1, t2, ... , Tn ⊆ I.

- Một itemset Fk được gọi là một tập phổ biến nếu và chỉ nếu độ hỗ trợ của nó lớn hơn hoặc bằng một giá trị ngưỡng minsup: sup(Fk) ≥ minsup và độ tin cậy của luật Confidence ≥ minconf.

  + Độ hỗ trợ của một luật X→Y được định nghĩa là sup(X∪Y)/|T|

  + Độ tin cậy của một luật X→Y được định nghĩa là conf(X→Y)=sup(X∪Y)/sup(X).

  + Tập phổ biến đóng: X là đóng nếu không có bất kỳ tập phổ biến Y mà X⊂Y và δ(X) =δ(Y).

- Tập sinh tối thiểu [12]: Cho X là một phổ biến thường xuyên đóng. X'≠∅ được gọi là một tập sinh của X khi và chỉ khi:

  + X'⊆ X

  + sup(X)=sup(X').

  + Cho G(X) biểu thị tập hợp các tập sinh của X. Chúng ta nói rằng X' ∈ G(X) là một tập sinh nhỏ nếu nó không có tập con trong G(X). Cho mGs (X) là tập tất cả các tập sinh tối thiểu của X. Theo định nghĩa, mGs(X) ≠∅ vì nếu không có tập sinh thích hợp sau đó X là một mG của X.

- Các luật tuân theo tiêu chí tối thiểu là luật kết hợp hợp lệ.

- Nguyên tắc chung: Cho hai các luật R1: X1→ Y1 và R2: X2→ Y2 , R1 nói tổng quát hơn R2, Ký hiệu là $R_1 \propto R_2$, , khi và chỉ khi $X_1 \subseteq X_2$ và $Y_2 \subseteq Y_1$.

Các tính chất của luật kết hợp

- **Tính chất 1**: Đối với luật kết hợp bất kỳ r: X→Y tìm được trong CSDL giao dịch T, ta có quan hệ conf (r) ≥ sup (r) .

Theo định nghĩa, $sup(X \cup Y)/|T|$ và $conf(X \rightarrow Y)=sup(X \cup Y)/sup(X)$ bởi vì $|T| \geq sup(X)$.

- Luật kết hợp khai thác từ cơ sở dữ liệu giao dịch, bao quát một cơ sở dữ liệu giao dịch có chứa các thông tin CSDL tuần tự [2]. Một CSDL trình tự SD được định nghĩa là một tập hợp các trình tự S = {S1, S2, …,Sn} và tập hợp các items I={i1, i2, …, in}, trong đó mỗi trình tự SX là một danh sách các giao dịch (tập phổ biến). sx= {X1, X2, . . Xm} sao cho X1, X2, …, Xn ⊆ I.

Các định nghĩa bên dưới của luật tuần tự được nêu ra trong CSDL trình tự.

- Luật tuần tự X →Y là mối quan hệ giữa hai tập phổ biến X, Y sao cho (X,Y⊆ I, X ∩ Y = Ø).

Để đánh giá một luật tuần tự có hai biện pháp :

+ Độ hỗ trợ tuần tự: seqSup(X→Y) = Sup (X→Y)/|S|

+ Độ tin cậy tuần tự: seqConf(X→Y) = Sup(X→Y)/Sup(X)

- Giải thích:

+ Luật tuần tự X⟶Y: nếu các item trong X xảy ra trong một số giao dịch của một trình tự, các item trong Y sẽ xảy ra trong một số giao dịch sau đó cùng một trình tự [1, 5, 7]. Lưu ý rằng các item trong X và Y có thứ tự (không bắt buộc phải xảy ra trong cùng một giao dịch trình tự).

+ sup(X→Y) biểu thị số lượng tuần tự từ CSDL trình tự, tất cả item của X xuất hiện trước tất cả item của Y (lưu ý item X hoặc Y không cần phải trong cùng một giao dịch).

+ sup (X) biểu thị số lượng các tuần tự có chứa X.

- Tính chất 2: Cho luật kết hợp bất kỳ r: X→Y trong CSDL S, quan hệ seqConf (r) ≥ seqSup(r).

+ Một luật tuần tự hợp lệ khi: seqSup(X→Y) = sup(X→Y)/|S|

và seqConf(X→Y) = sup(X→Y)/sup(X). Bởi vì |S| ≥ Sup(X).

Khai thác luật tuần tự chung cho nhiều trình tự, bao gồm việc tìm kiếm các luật hợp lệ trong CSDL trình tự. một luật hợp lệ là luật có độ hỗ trợ tuần tự và độ tin cậy tuần tự tương ứng không nhỏ hơn ngưỡng *minSeqSup* và *minSeqConf* do người dùng định nghĩa.

+ Ví dụ: Xét trên một CSDL trình tự với minSeqSup = 0.5 và minSeqConf = 0.5

Bảng 3.1. Cơ sở dữ liệu tuần tự

STT	Trình tự
1	(a b); (c);(f);(g);(e)
2	(a d);(c);(b);(e f)
3	(a);(b);(f);(e)
4	(b);(f g)

Bảng 3. 2 Tập luật thu được là các luật tuần tự hợp lệ

Tập luật	Các tập luật	Độ hỗ trợ tuần tự	Độ tin cậy tuần tự
R1	(a) => (b,e,f)	0.5	1
R2	(a) => (c,e,f)	0.5	0.66
R3	(a,b) => (e,f)	0.5	1
R4	(b) => (e,f)	0.75	0.75
R5	(a b c) => (e)	0.5	1
R6	(a) => (e,f)	0.75	1
R7	(c) => (f)	0.5	1
R8	(a) => (b)	0.5	0.66
R9	(c) => (e,f)	0.5	1
R10	(b,c) => (e,f)	0.5	1

- Giải thích:

+ Xét luật {a, b, c} →{e}. Độ hộ trợ tuần tự là sup({a, b, c}→{e})/|S| =2/4 và độ tin cậy tuần tự sup({a, b, c}→{e})/sup (X) = 2/2 = 1. Vì các giá trị không nhỏ hơn minSeqSup và minSeqConf, luật là hợp lệ.

+ Lưu ý vấn đề khai thác luật tuần tự như định nghĩa khác với khai thác mẫu tuần tự. Vì vậy, không thể sửa lại thuật toán khai thác mẫu tuần tự để sinh ra các luật dạng này.

- Tác giả đề xuất thuật toán SApriori nhận xét: Nếu ta bỏ qua thông tin về thời gian của CSDL tuần tự SD, ta có CSDL giao dịch SD'. Đối với mỗi CSDL tuần tự SD và CSDL giao dịch SD' tương ứng, mỗi luật tuần tự r: X=> Y của S có một luật kết hợp r': X→Y tương ứng trong S'.

- Tính chất 3: Cho luật kết hợp bất kỳ r: X→Y trong CSDL trình tự SD, ta có quan hệ sup(r') ≥ seqSup(r).

- Tính chất 4: Cho luật kết hợp bất kỳ r: X→Y trong CSDL trình tự SD, ta có conf(r') ≥ seqConf(r).

• Các luật dư thừa và các định lý liên quan

Độ hỗ trợ tuần tự của r và độ hỗ trợ r' lần lượt là $sup(X \rightarrow Y)/ |SD|$ và $sup(X \cup Y)/|SD'|$. Vì $|SD|=|SD'|$ và $sup(X \cup Y) \geq sup(X \rightarrow Y)$ ta có quan hệ $sup(r') \geq seqSup(r)$.

Độ tin cậy tuần tự của tập r và độ tin cậy r' lần lượt là $sup(X \rightarrow Y)/sup(X)$ và $sup(X \cup Y)/sup(X)$. Bởi vì $sup(X \cup Y) \geq sup(X \rightarrow Y)$ ta có quan hệ $conf(r') \geq seqConf(r)$.

Tập $R = \{R_1, R_2, \ldots, R_n\}$ là tập hợp các các luật mà giống nhau độ hỗ trợ và độ tin cậy. Luật R_j là dư thừa nếu trong R tồn tại các luật R_i như rằng $R_i \propto R_j$ $(i \neq j)$.

Định lý 1: MNARs [6] với độ tin cậy = 100% và chỉ được tạo ra từ $X' \rightarrow X$ ($\forall X' \in mGs(X)$, X là FCI).

Chứng minh: Vì $sup(mG(X)) = sup(X)$, vì vậy $conf(mG(X) \rightarrow X) = 100\%$. Để chứng minh định lý này, chúng ta cần phải chứng minh ba bài toán:

i. Luật mG $(X) \rightarrow X'$ là dư thừa ($\forall X'$ là tập sinh của X, $X' \subset X$): *Bởi vì* $X' \subset X$, nên $mG(X) \rightarrow X \propto mg(X) \rightarrow X' \Rightarrow mG(X) \rightarrow X'$ là luật dư thừa.

ii. Luật $X' \rightarrow X$ *là dư thừa* (($\forall X'$ là tập sinh của X, $X' \notin$ mGs(X)): bởi vì $mG(X) \subset X' \Rightarrow mG(X) \rightarrow X \propto X' \rightarrow X \Rightarrow X' \rightarrow X$ là luật dư thừa.

iii. Không có các luật $X' \rightarrow Y'$ với độ tin cậy = 100% ($\forall X'$ là tập sinh của X, Y' là tập sinh của Y, X, $Y \in$ FCIs, X ⊂ Y): bởi vì $sup(X') = sup(X)$, $sup(Y')=sup(Y)$ và $X,Y \in FCIs(X \subset Y)$, nên $sup(X') \neq sup(Y') \Rightarrow conf(X' \rightarrow Y') = sup(Y')/sup(X')=sup(Y)/sup(X) < 100\%$.

Định lý 2: MNARs [6] với độ tin cậy <100% chỉ được tạo ra từ $X' \rightarrow$ Y ($\forall X' \in mGs(X)$, X, $Y \in FCI$ và $X \subset Y$).

Chứng minh: Để chứng minh định lý này, chúng ta cần phải chứng minh hai bài toán:

i. $Conf(X' \rightarrow Y) < 100\%$: bởi vì $X,Y \in FCIs$ và $X \subset Y \Rightarrow sup(X') \neq sup(Y) \Rightarrow conf(X' \rightarrow Y) = sup(Y)/sup(X') = sup(Y)/sup(X) < 100\%$.

ii. Tất cả các các luật ở dạng $X'' \rightarrow Y'$ là dư thừa ($\forall X''$ là tập sinh của X, $\forall Y$ là tập sinh Y, $Y' \neq Y$): vì X'' là tập sinh của $X \Rightarrow X'' \subseteq X$ và $sup(X'') = sup(X)$. Tương tự như vậy, Y' là một tập sinh của $Y \Rightarrow Y' \subset Y$ và $sup(Y') = sup(Y)$. Do đó, $conf(X'' \rightarrow Y') = conf(X' \rightarrow Y) \Rightarrow X'' \rightarrow Y' \propto X' \rightarrow Y \Rightarrow X'' \rightarrow Y'$ là luật dư thừa.

Dựa trên Định lý 1 và 2, tác giả thay đổi cấu trúc của FIL bằng cách thêm một trường đánh dấu để cho biết một tập phổ biến trong nút dàn là mG hoặc không, và các trường khác để đánh dấu cho biết một tập phổ biến trong nút dàn là tập phổ biến đóng hay không. Nó cần thêm thời gian để xây dựng dàn, nhưng nó giúp tiết kiệm rất nhiều thời gian để xem xét nếu một nút là *mG* hoặc FCI.

3. THUẬT TOÁN SAPRIORI – KHAI THÁC LUẬT TUẦN TỰ KHÔNG DƯ THỪA

INPUT: *CSDL tuần tự, minSeqSup, minSeqConf*

OUTPUT: tất cả các luật tuần tự hợp lệ và không dư thừa

PROCEDURE:

1. Xem CSDL tuần tự như một CSDL giao dịch. (SApriori bỏ qua các thông tin tuần tự từ CSDL tuần tự để có được một CSDL giao dịch).

2. Tìm tất cả các luật kết hợp đúng từ CSDL giao dịch bằng cách áp dụng một thuật toán khai thác luật kết hợp (như Apriori). Chọn *minsup = minSeqSup* và *minconf = minSeqConf.*

3. Quét CSDL tuần tự ban đầu để tính SeqSup và seqConf của từng luật kết hợp tìm thấy trong bước trên. Loại bỏ các luật *r* mà *seqSup(r) <minSeqSup* hoặc *seqConf (r) <minSeqConf* .

4. Lọc bỏ các tập luật dư thừa để hạn chế các tập luật phát sinh không cần thiết làm cô đọng thông tin.

3.1. Chứng minh về tính đầy đủ của thuật toán

Để chứng minh rằng SApriori có thể tìm thấy tất cả các luật tuần tự hợp lệ → chứng minh tập luật tuần tự hợp lệ là con của tập luật kết hợp hợp lệ.

Định lý 1. Thuật toán sẽ tìm ra tất cả các luật tuần tự với *minSeqConf* và *minSeqSup* cho trước, nếu ta chọn *minconf* ≤ *minSeqConf* và *minsup*≤ *minSeqSup.*

Hệ quả 1. Thuật toán hiệu quả hơn nếu chúng ta chọn *minconf = minSeqConf* và *minsup = minSeqSup.*

Ví dụ: Với tập thuộc tính *ABC* thì ta suy ra được các luật kết hợp hợp lệ như sau: *A→B, A →C, B →A, B→C, C →A, C→B, AB→C, AC→B, BA→C, BC→A, CA→B, CB→A* nhưng chỉ có *A→B, A→C* là luật tuần tự hợp lệ.

3.2. Chứng minh về tính chính xác của thuật toán

Chứng minh SApriori là một thuật toán chính xác →chứng minh thuật toán không tạo ra các luật tuần tự không hợp lệ.

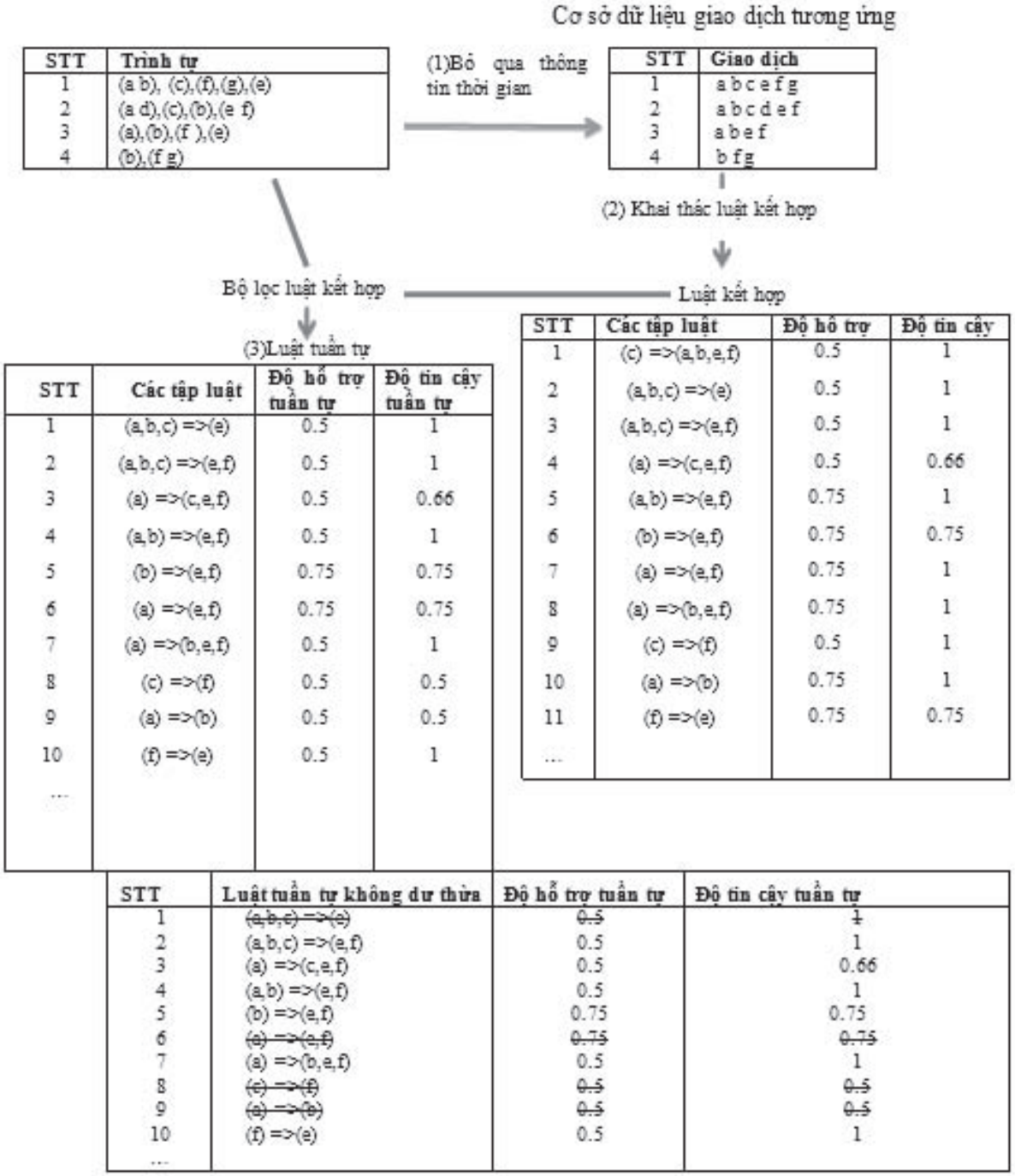
Định lý 2. Thuật toán SApriori không tạo ra các luật không hợp lệ.

Theo Định lý 1 nếu ta chọn *minconf ≤ minSeqConf và minsup ≤ minSeqSup* khi khai thác luật kết hợp, thì tập các luật kết hợp có chứa tất cả các luật tuần tự hợp lệ thỏa *minSeqSup và*

minSeqConf. Nhưng, do *sup (r ') ≥ seqSup (r)* và *conf (r') ≥ seqconf (r)* (với luật tuần tự bất kỳ *r* (Tính chất 3 và 4)), tập hợp các luật kết hợp cũng có thể chứa luật tuần tự không hợp lệ.

Tuy nhiên, SApriori thường không tạo ra các luật không hợp lệ vì Bước 3 của thuật toán sẽ lọc chúng ra. Do đó, thuật toán là chính xác.

Bảng 3.3. Bảng so sánh kết quả thuật toán Sapriori sau khi khai thác luật không dư thừa tập dữ liệu Kosarak-1



STT	Trình tự
1	(a b), (c),(f),(g),(e)
2	(a d),(c),(b),(e f)
3	(a),(b),(f),(e)
4	(b),(f g)

STT	Giao dịch
1	a b c e f g
2	a b c d e f
3	a b e f
4	b f g

STT	Các tập luật	Độ hỗ trợ	Độ tin cậy
1	(c) =>(a,b,e,f)	0.5	1
2	(a,b,c) =>(e)	0.5	1
3	(a,b,c) =>(e,f)	0.5	1
4	(a) =>(c,e,f)	0.5	0.66
5	(a,b) =>(e,f)	0.75	1
6	(b) =>(e,f)	0.75	0.75
7	(a) =>(e,f)	0.75	1
8	(a) =>(b,e,f)	0.75	1
9	(c) =>(f)	0.5	1
10	(a) =>(b)	0.75	1
11	(f) =>(e)	0.75	0.75
...			

STT	Các tập luật	Độ hỗ trợ tuần tự	Độ tin cậy tuần tự
1	(a,b,c) =>(e)	0.5	1
2	(a,b,c) =>(e,f)	0.5	1
3	(a) =>(c,e,f)	0.5	0.66
4	(a,b) =>(e,f)	0.5	1
5	(b) =>(e,f)	0.75	0.75
6	(a) =>(e,f)	0.75	0.75
7	(a) =>(b,e,f)	0.5	1
8	(c) =>(f)	0.5	0.5
9	(a) =>(b)	0.5	0.5
10	(f) =>(e)	0.5	1
...			

STT	Luật tuần tự không dư thừa	Độ hỗ trợ tuần tự	Độ tin cậy tuần tự
1	~~(a,b,e) =>(e)~~	~~0.5~~	~~1~~
2	(a,b,c) =>(e,f)	0.5	1
3	(a) =>(c,e,f)	0.5	0.66
4	(a,b) =>(e,f)	0.5	1
5	(b) =>(e,f)	0.75	0.75
6	~~(a) =>(e,f)~~	~~0.75~~	~~0.75~~
7	(a) =>(b,e,f)	0.5	1
8	~~(c) =>(f)~~	~~0.5~~	~~0.5~~
9	~~(a) =>(b)~~	~~0.5~~	~~0.5~~
10	(f) =>(e)	0.5	1
...			

3.3. Chứng minh về tính hiệu quả của thuật toán

Bài toán đã khai thác được các luật tuần tự không dư thừa tối thiểu (MNARs)[6] làm giảm đi nhiều các luật tuần tự không cần thiết.

Bằng các thực nghiệm, chúng tôi nhận thấy kết quả thu gọn số lượng đáng kể các luật làm tăng hiệu quả quản lý các dữ liệu dư thừa.

Việc ứng dụng dàn tập đóng trong khai thác luật thiết yếu nhất [5] phương pháp thích hợp để khai thác với số luật ít hơn nhưng vẫn bảo đảm tích hợp đầy đủ tất cả các luật của phương pháp khai thác truyền thống. Chỉ lưu lại các luật có vế trái tối tiểu và vế phải tối đại (theo quan hệ cha – con). sử dụng quan hệ cha – con trên dàn để giảm chi phí xét quan hệ cha con và vì vậy làm giảm được thời gian khai thác luật.

4. KẾT QUẢ THỰC NGHIỆM

4.1. Cơ sở dữ liệu và môi trường thực nghiệm

Dữ liệu chuỗi là loại dữ liệu phổ biến trong nhiều lĩnh vực ứng dụng. Do vậy, đề tài tiến hành thực nghiệm trên hai loại CSDL. Với loại CSDL thứ nhất, mỗi itemset của một chuỗi là một tập các item. Loại CSDL này thường có trong lĩnh vực ứng dụng giao dịch thương mại và gọi là CSDL giao dịch. Với loại CSDL thứ hai, mỗi item đóng vai trò là một itemset, loại CSDL này khá phổ biến như: dữ liệu sinh học, dữ liệu web, dữ liệu vết thực thi chương trình, v.v.

Việc thực nghiệm được tiến hành trên trên máy tính *Intel (R), Core (TM) i3 CPU 2.27GHz,* sử dụng *Eclipse Standard/SDK - Version: Kepler Service Release 1.*

Trong đó, với loại CSDL thứ nhất cơ sơ dữ liệu *Kosarak:* đây là một bộ dữ liệu rất lớn chứa 990,000 trình tự của các dữ liệu được lấy từ một cổng thông tin của các báo điện tử.

Kosarak-1 là phiên bản nhỏ hơn của *Kosarak* chỉ chứa 70.000 chuỗi đầu tiên của Kosarak. Nhóm tác giả chỉ giữ 70.000 chuỗi đầu tiên để thực hiện nhanh hơn.

Bộ dữ liệu thứ hai là: *BMSWebView1 (Gazelle) (KDD CUP 2001)*. Bộ dữ liệu này có 59.601 chuỗi các dữ liệu trong thương mại điện tử. Nó chứa 497 mặt hàng khác nhau. Độ dài trung bình của chuỗi có một độ lệch chuẩn. Trong bộ dữ liệu này, có một số dãy dài. Ví dụ, 318 chuỗi chứa hơn 20 mặt hàng.

Bộ dữ liệu thứ ba là: Leviathan. Bộ dữ liệu này là một sự chuyển đổi của Leviathan cuốn tiểu thuyết của *Thomas Hobbes (1651)* như là một cơ sở dữ liệu chuỗi (mỗi từ là một mục). Nó chứa 5834 chuỗi và 9025 mặt hàng khác nhau. Số lượng trung bình các mặt hàng ở mỗi chuỗi là: 33,8. Số lượng trung bình của mặt hàng riêng biệt cho mỗi chuỗi là 26.34.

Các bộ dữ liệu tổng hợp này có được từ trang web :
http://www.cs.rpi.edu/~zaki/Workshops/FIMI/data/

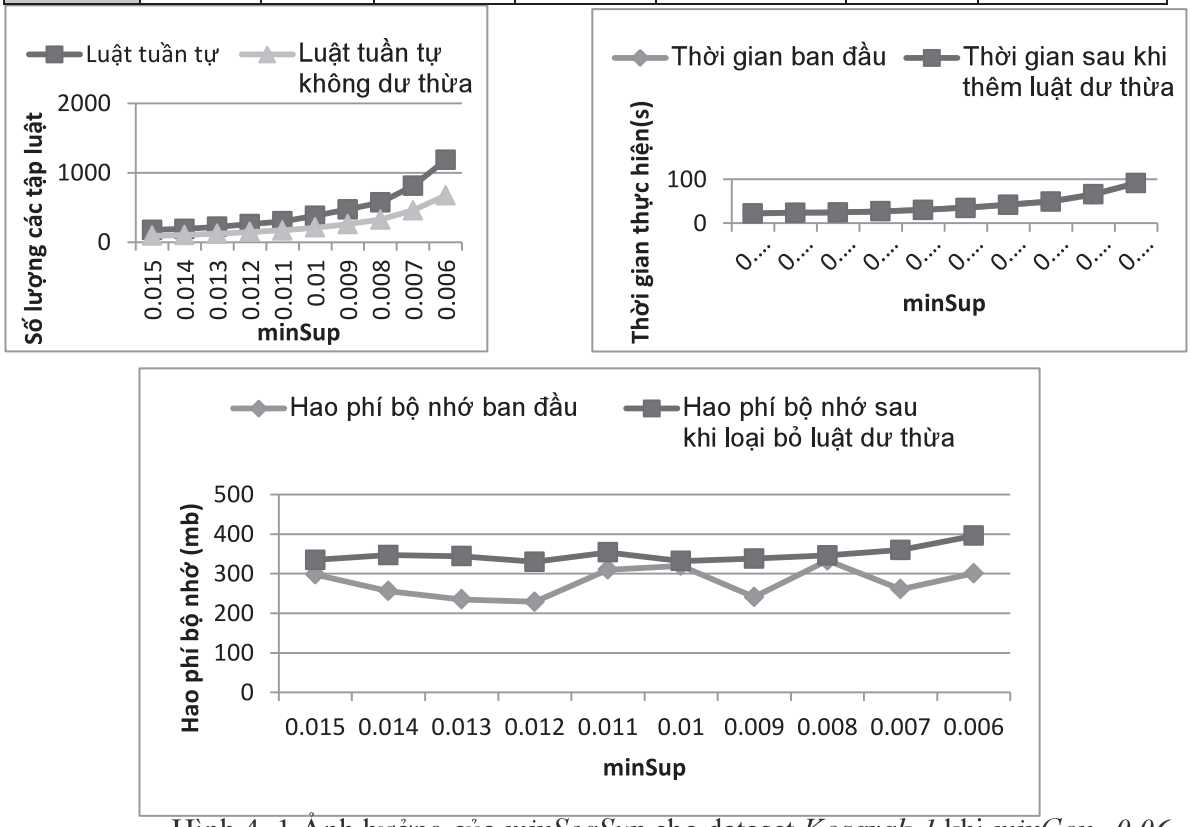
4.2. Kết quả thực nghiệm đánh giá mức độ ảnh hưởng của minsup

Các thực nghiệm đầu tiên chạy thuật toán Sapriori cho các tập dữ liệu với các giá trị minsup khác nhau trên các tập phổ biến khác nhau để so sánh số lượng các luật kết hợp, luật tuần tự và các luật tuần tự không dư thừa.

Thực nghiệm trên tập Kosarak-1 áp dụng với *minconf=0.3* và *0.6* trong khi *minsup* sẽ thay đổi từ *(0.009→0.015)* ta được kết quả như sau:

Bảng 4. 1 Bảng so sánh kết quả thuật toán *Sapriori* sau khi khai thác luật không dư thừa tập dữ liệu *Kosarak-1*

minSup	Luật kết hợp	Luật tuần tự	Luật tuần tự không dư thừa	Thời gian ban đầu	Thời gian sau khi loại bỏ luật dư thừa	Hao phí bộ nhớ ban đầu	Hao phí bộ nhớ sau khi loại bỏ luật dư thừa
0.006	5989	1184	674	90816	91441	301	396
0.007	3928	811	459	65351	65833	261	360
0.008	2614	571	326	49321	49611	333	346
0.009	2119	472	260	41987	42346	241	338
0.01	1600	380	212	34836	35093	320	332
0.011	1179	300	169	30228	30509	310	354
0.012	1004	261	148	27145	27407	229	330
0.013	815	220	122	24203	24449	235	344
0.014	680	192	103	23615	23834	256	347
0.015	621	177	92	22022	22220	298	335

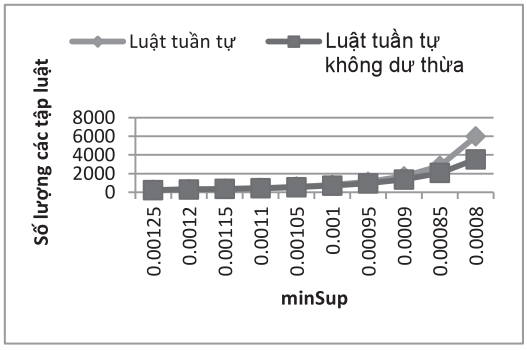


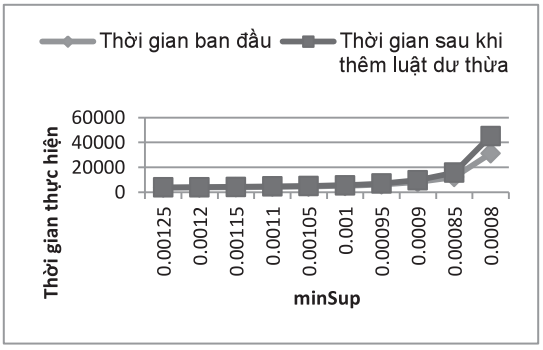
Hình 4. 1 Ảnh hưởng của *minSeqSup* cho dataset *Kosarak-1* khi *minCon=0.06*

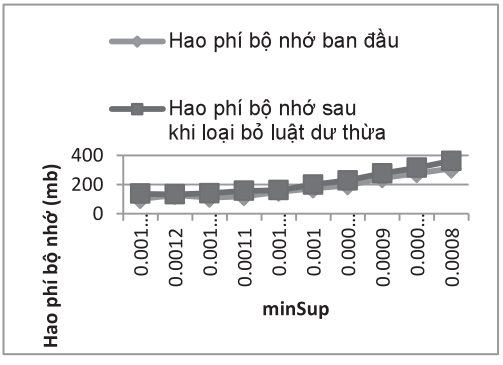
Thực nghiệm trên tập *BMS-Webview1*, áp dụng với *minconf=0,3* và *0,6* trong khi *minsup* sẽ thay đổi từ *(0,0008, 0,00085, ..., 0,00125)* ta được kết quả như sau:

Bảng 4. 2 Bảng so sánh kết quả thuật toán *Sapriori* sau khi khai thác luật không dư thừa tập dữ liệu *BMS-Webview1*

minsup	Luật kết hợp	Luật tuần tự	Luật tuần tự không dư thừa	Thời gian ban đầu	Thời gian sau khi loại bỏ luật dư thừa	Hao phí bộ nhớ ban đầu	Hao phí bộ nhớ sau khi loại bỏ luật dư thừa
0.00125	926	247	235	3697	3931	101	137
0.00120	1160	308	293	3823	4088	129	133
0.00115	1389	357	342	3963	4259	110	141
0.00110	1785	448	421	4415	4711	120	157
0.00105	2396	581	542	4586	5039	150	161
0.00100	3448	789	709	5070	5631	174	200
0.00095	5495	1122	955	6240	7020	194	228
0.00090	9422	1701	1368	8174	9719	244	276
0.00085	18408	2814	2063	12168	15803	278	315
0.00080	59469	5981	3518	31168	45162	309	362





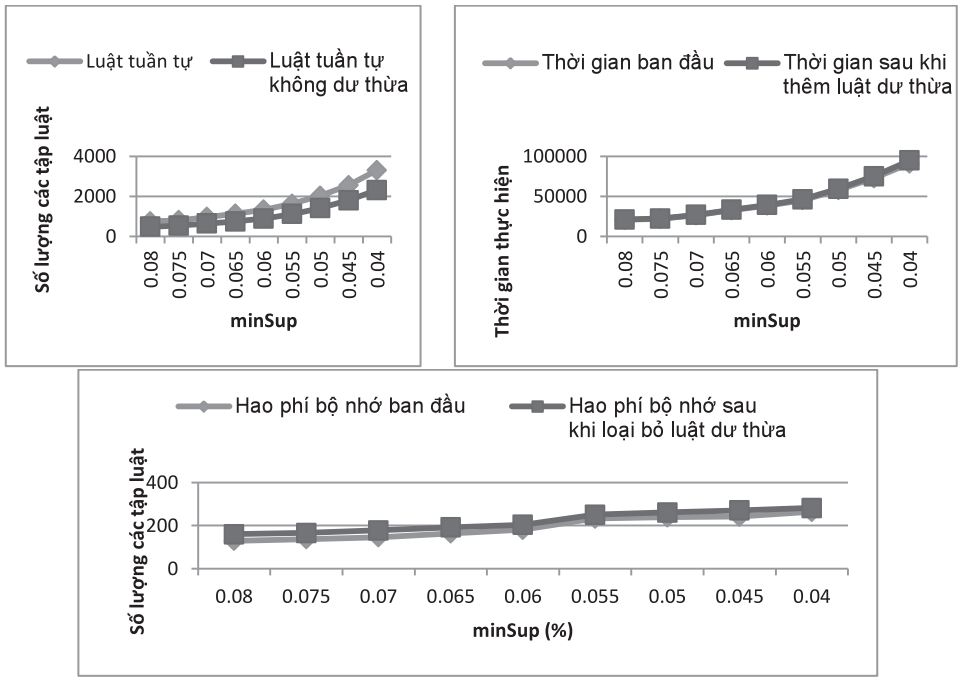


Hình 4. 2 Ảnh hưởng của *minSeqSup* cho dataset *BMS-Webview1* khi *minConf=0.06*

Thực nghiệm trên tập dữ liệu *Leviathan,* áp dụng với *minconf=0,3* và *0,5* trong khi *minsup* sẽ thay đổi từ *(0,04, 0,045 ... 0,008)* ta được kết quả như sau:

Bảng 4. 3 Bảng so sánh kết quả thuật toán *Sapriori* sau khi khai thác luật không dư thừa tập dữ liệu *Leviathan*

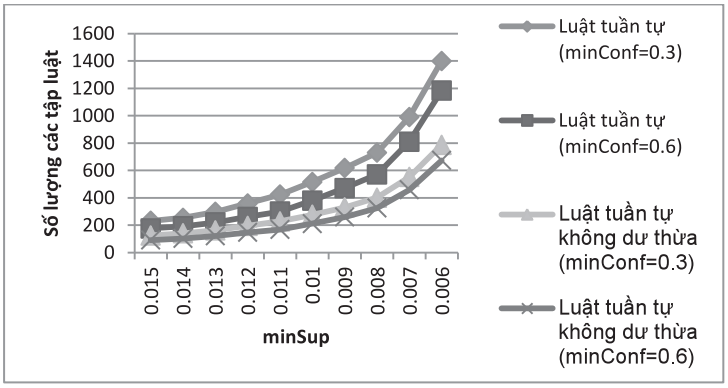
minsup	Luật kết hợp	Luật tuần tự	Luật tuần tự không dư thừa	Thời gian ban đầu	Thời gian sau khi loại bỏ luật dư thừa	Hao phí bộ nhớ ban đầu	Hao phí bộ nhớ sau khi loại bỏ luật dư thừa
0.08	6132	751	487	20843	21280	130	160
0.075	6914	812	547	21996	22386	137	166
0.07	8543	969	647	26317	26925	146	178
0.065	10678	1137	754	32713	33447	164	192
0.06	13126	1337	900	38579	39421	181	204
0.055	16994	1627	1126	45396	46426	233	251
0.05	22626	2027	1422	58032	59560	239	261
0.045	29896	2559	1804	72742	75145	242	231
0.04	41421	3308	2321	91136	94848	264	282



Hình 4. 3. Ảnh hưởng của minSeqSup cho dataset Leviathan khi *minconf* =0.06

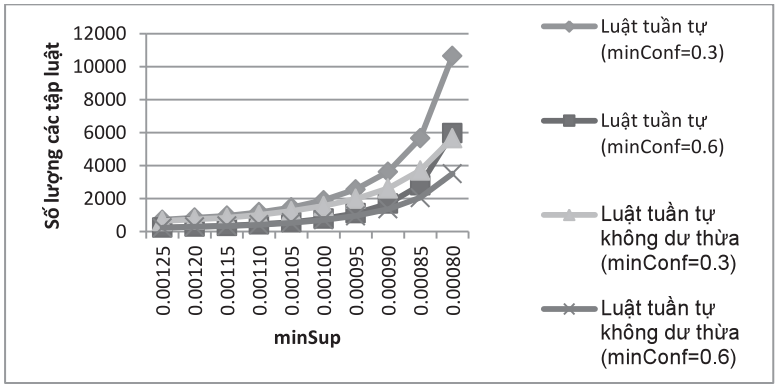
4.3. Kết quả thực nghiệm đánh giá mức độ ảnh hưởng của minconf

Vì trọng tâm thực nghiệm tìm ra tập không dư thừa nên chúng tôi chỉ quan tâm đến số lượng các tập luật. Biểu đồ hình 4.4 thực nghiệm trên tập Kosarak-1 áp dụng với *minconf=0.3* và *0.6* trong khi *minsup* sẽ thay đổi từ *(0.009→0.015)* ta được mô hình như sau:



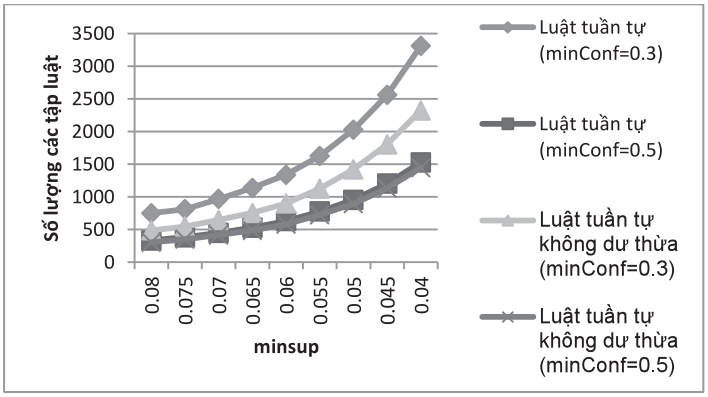
Hình 4. 4 Ảnh hưởng của *minconf* lên số lượng tập luật sử dụng tập luật *Kosarak-1*

Thực nghiệm trên tập dữ liệu BMS-Weview1 áp dụng với *minconf=0.3* và *0.6* trong khi *minsup* sẽ thay đổi từ *(0.0008→0.00125)* ta được mô hình bên dưới.



Hình 4. 5 Ảnh hưởng của *minconf* lên số lượng tập luật sử dụng tập luật *BMS-Webview1*

Thực nghiệm trên tập dữ liệu ***Leviathan*** áp dụng với *minconf=0.3* và *0.5* trong khi *minsup* sẽ thay đổi từ *(0.0008→0.00125)* ta được mô hình bên dưới.



Hình 4. 6 Ảnh hưởng của *minconf* lên số lượng tập luật sử dụng tập luật *BMS-Leviathan*

4.2. Tổng kết

Phần này trình bày các kết quả thực nghiệm thuật toán Sapriori khai thác trên các tập CSDL (*Kosarak-1, BMS-Webview1, Leviathan)* Ko lấy từ trang web

http://www.cs.rpi.edu/~zaki/Workshops/FIMI/data/

Qua các kết quả thực nghiệm trên các cơ sở dữ liệu khác có nhận xét chung như sau:

- Số lượng các tập luật được rút gọn đáng kể tùy theo bộ CSDL

• Bộ dữ liệu *BMS-Webview1* số lượng tập luật không dư thừa so với tập luật ban đầu là *0.79*.

• Bộ dữ liệu *Kosarak-1* số lượng tập luật không dư thừa so với tập luật ban đầu là *0.55*.

• Bộ dữ liệu *Kosarak-1* số lượng tập luật không dư thừa so với tập luật ban đầu. *0.93*

+ Thời gian xấp xỉ với thời gian xử lý ban đầu

+ Tốn thêm bộ nhớ cho việc xử lý dư thừa nhưng không đáng kể

Nhìn chung thuật toán đã hoàn thành được mục tiêu đặt ra là loại bỏ các tập luật dư thừa mà vẫn đầy đủ các tập luật, tiết kiệm được thời gian xử lý và hạn chế bộ nhớ.

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Bài báo đã tìm hiểu cơ sở lý thuyết về khai thác luật tuần tự trên CSDL tuần tự. Kết quả thực nghiệm trên các CSDL cho thấy tính hiệu quả của các phương pháp đề xuất so với thuật toán CMRules[8]. Mục đích của bài báo là đưa ra phương pháp hiệu quả để khai thác luật tuần tự trên CSDL tuần tự.

Luật tuần tự rất hữu ích trong việc khám phá những tri thức tiềm ẩn trong các nguồn dữ liệu ở dạng tuần tự. Tuy nhiên, với tình trạng bùng nổ thông tin hiện nay, khối lượng dữ liệu ngày càng trở nên đồ sộ, việc khai thác tập luật tuần tự sao cho hiệu quả và tốn ít thời gian nhất là cần thiết. Do vậy, bài báo hướng tới việc cải thiện thuật toán hơn nữa để đạt tốc độ tối ưu hơn.

Trong một số lĩnh vực ứng dụng, cần thiết phải khai thác những luật tuần tự thiết yếu nhất đó là luật tuần tự không dư thừa. Vì vậy, nghiên cứu phương pháp khai thác luật tuần tự không dư thừa dựa trên cây tiền tố. Hơn nữa, phát triển phương pháp sinh luật dựa trên cây tiền tố vào bài toán khai thác luật thú vị. Đi sâu vào tính ứng dụng của bài toán khai thác luật tuần tự trên CSDL chuỗi.

TÀI LIỆU THAM KHẢO

[1]. Agrawal, R., Imielinski, T., Swami, A. (1993), *Mining association rules between sets ofitems in large databases*, Proceedings of the 1993 ACM SIGMOD Conference Washington DC, USA, May 1993.

[2]. Agrawal, R., Srikant, R. (1994), Fast algorithms for mining association rules, in: Proceedings of International Conference on Very Large Data Bases.

[3]. Agrawal, R., Srikant, R. (1996), Mining Sequential Patterns: Generalizations and Performance Improvements, in: Proc. 5th Int'l Conf. Extending Database Technology.

[4]. Ayres, J., Gehrke, J.E., Yiu, T., Flannick, J. (2002), Sequential Pattern Mining using a Bitmap Representaion, in SIGKDD Conf.

[5]. Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, L. Lakhal(2000), Mining Minimal NonRedundant Association Rules using Closed Frequent Itemsets, In 1st International Conference on Computational Logic.

[6]. Bay Vo, Bac Le(2011), Mining minimal non-redundant association rules using frequent itemsets lattice. IJISTA 10(1).

[7]. Dong, G., Pei, J. (2007), Sequence Data Mining, Springer Science + Business Media, LLC.

[8]. P. Fournier-Viger, U. Faghihi, R. Nkambou, and E.M. Nguifo(2012), CMRULES: Mining Sequential Rules Common to Several Sequences, Knowledge-based Systems 25(1).

[9]. Lo, D., Khoo, S.-C. (2006), SMArTIC: Toward Building an Accurate, Robust and Scalable Specification Miner, in: Proceedings of SIGSOFT Sympo- sium on the Foundations of Software Engineering.

[10]. Zaki, M.J. (2000), SPADE: An Efficient Algorithm for Mining Frequent Sequences, Machine Learning Journal 42(1/2).

[11]. M. J. Zaki (2004), Mining Non-Redundant Association Rules, Data Mining and Knowledge Discovery, 9, Kluwer Academic Publishers. Manufactured in The Netherlands.

[12]. Zhang, M., Hsu, W., Lee, M. -L. (2006), Mining progressive confident rules, in: Proceedings of SIGKDD Conference on Knowledge Discovery and Data Mining.