THUẬT TOÁN TOP-K MẪU TUẦN TỰ TỐI ĐẠI

ALGORITHM OF TOP-K MAXIMAL SEQUENTIAL PATTERNS

ĐỖ THANH TÙNG(*), TRẦN THỊ YẾN NHI(**) và LÝ HẢI SƠN(***)

TÓM TẮT: *Khai thác mẫu tuần tự là một phần quan trọng của khai thác dữ liệu với các ứng dụng rộng rãi. Tuy nhiên, việc tùy chỉnh thông số minsup để phù hợp trong các thuật toán khai thác mẫu tuần tự nhằm tạo ra đúng số mẫu mà người dùng mong muốn là điều rất khó khăn và tốn thời gian. Để giải quyết vấn đề này, thuật toán khai thác k mẫu tuần tự tối đại TSP (Top-K Closed Sequential Patterns) [7, tr.438-457] đã đưa ra phương án giới hạn lại số lượng k mẫu cần khai thác, nhưng thời gian thực hiện và bộ nhớ sử dụng của thuật toán cao. Bài viết đề xuất thuật toán MTKS (Max Top-K Sequential pattern mining) tìm k mẫu tuần tự tối đại dựa trên thuật toán TKS [2, tr.109-120]. Với k mẫu nhập vào thuật toán sẽ trả về k mẫu có độ hỗ trợ cao nhất trong cơ sở dữ liệu.*
Từ khóa: *khai thác Top-K mẫu tuần tự; Top-K mẫu tuần tự tối đại; thuật toán TKS; TSP.*

ABSTRACT: *Mining the sequential pattern is an important part of data mining with wide-range applications. However, it is very difficult and time-consuming to customize the minsup parameters to fit in a sequential pattern exploring algorithms to create the right number of samples desired by the user. To solve this problem, the Top-K closed Sequential Pattern (TSP) algorithm* gave a *method to limit the number of k patterns to be exploited, but the running time and usage memory of this algorithm is high. Therefore, the article proposes the Max Top-K Sequential pattern (MTKS) algorithm to find k maximum sequential patterns based on the algorithm Top-K Sequential pattern mining (TKS). With input k patterns, the algorithm returns k patterns highest degree of support in the database.*
Key words: *Top-K sequential patterns mining; Top-K maximal sequential patterns; TKS algorithm; TSP algorithm.*

1. ĐẶT VẤN ĐỀ

Trong lĩnh vực khai thác dữ liệu, nhất là trên cơ sở dữ liệu chuỗi. Việc khai thác mẫu tuần tự là một nhiệm vụ khai thác dữ liệu quan trọng đã được nghiên cứu rộng rãi [1, tr.3-14], [3, tr.512-521], [4, tr.259-289], [5, tr.554-560], [6, tr.3-17]. Cho một tập các chuỗi, trong đó mỗi chuỗi bao gồm một danh sách các tập phổ biến và một ngưỡng hỗ trợ tối thiểu do người dùng chỉ định (Minsup), khai thác mẫu tuần tự là để tìm tất cả các mẫu phổ biến có độ hỗ trợ không thấp hơn minsup. Khai thác mẫu tuần tự được ứng dụng trong nhiều lĩnh vực như: phân tích thị trường, phân tích mẫu truy cập web, dự đoán nhu cầu mua sắm của khách hàng…

Khi khai thác mẫu tuần tự tồn tại một số vấn đề như sau: khai thác mẫu tuần tự thường tạo ra một số lượng lớn các mẫu, vấn đề đó không thể tránh khỏi khi cơ sở dữ liệu bao gồm các chuỗi phổ biến dài. Nó sẽ tạo ra các mẫu phổ biến mà các mẫu đó có thể có cùng độ hỗ trợ hoặc là cha của mẫu phổ biến khác; Nếu chọn minsup quá cao, tạo ra ít các mẫu bỏ qua các thông tin có giá trị còn ngược lại, quá nhiều

(*) ThS. Trường Đại học Văn Lang, *tung.dt@vlu.edu.vn*
(**) ThS. Trường Đại học Văn Lang, *nhi.tty@vlu.edu.vn*
(***) ThS. Trường Đại học Văn Lang, *son.ly@vlu.edu.vn, Mã số: TCKH25-05-2021*

mẫu dẫn đến thuật toán chậm. Để chọn một giá trị minsup hợp lý đòi hỏi phải biết về dữ liệu; Mỗi một sản phẩm mà khách hàng mua lại có thể có giá khác nhau. Tương tự mỗi một hạng mục trong giao dịch cũng có các trọng số khác nhau tùy theo từng loại cơ sở dữ liệu cụ thể. Nhiều nghiên cứu đã được thực hiện và nhiều thuật toán đã được đề xuất trong lĩnh vực này. Thuật toán TKS [2, tr.109-120] được đánh giá cao bởi vì chi phí thực hiện thấp hơn so với các thuật toán khác trong việc khai thác k mẫu tuần tự phổ biến. Dựa vào đó để làm nền tảng tiến hành nghiên cứu bài toán khai thác Top K mẫu tuần tự tối đại.

2. NỘI DUNG

2.1. Các khái niệm về chuỗi dữ liệu

Cho I = *{i1, i2,…, ik}* là một tập các item. Tập con của I gọi là Itemset. Chuỗi *s= <t1, t2,…, tm > (ti ⊆I)* là một danh sách có thứ tự. Chúng ta giả sử rằng, các item trong mỗi itemset được nhóm theo thứ tự.

Ví dụ: Xét cơ sở dữ liệu như bảng 1.

Bảng 1. Cơ sở dữ liệu D

SID	Sequences
1	<(a)(bc)(ac)(d)(c)>
2	<(ab)(bc)(c)(a)>
3	<(cd)(ab)(df)(c)(b)>
4	<(ae)(bf)(c)(b)(c)>
5	<(bf)(ad)(d)(e)(ac)(f)>

Chuỗi *s*1 có 5 itemset xảy ra theo thứ tự <(a)(bc)(ac)(d)(c)>. Chiều dài của *s*, *l(s)* là tổng số các item trong s còn được gọi là *l-sequence*.

Ví dụ: *Chuỗi <(ac)(d)> là một* 3-sequence *có kích thước là 2.*

Chuỗi α = *<a1, a2,…, am*> là một chuỗi con của chuỗi khác β = *<b1, b2,…,bn*> ký hiệu là $\alpha \subseteq \beta$ nếu và chỉ nếu ∃*i1, i2,…im,* sao cho *1* ≤ *i1* < *i2* <… < *im* ≤ *n* và $a_1 \subseteq b_{i1}$, $a_2 \subseteq b_{i2}$, ... $a_m \subseteq b_{im}$. Chúng gọi β là chuỗi cha của α.

Ví dụ: Chuỗi <(a)(c)> là chuỗi con của <(ad)(c)(bc)(ae)> nhưng <(a)(d)> *không phải là chuỗi con của chuỗi* <(ad)> *và ngược lại*.

2.2. Độ hỗ trợ

Xét cơ sở dữ liệu chuỗi *D*, mỗi chuỗi có một chỉ số định danh duy nhất. Độ hỗ trợ tuyệt đối của một mẫu tuần tự α là tổng số chuỗi trong *D* có chứa p, ký hiệu sup_D *(p)* = l{s| s∈D và α⊆s}l. Độ hỗ trợ tương đối của *p* là tỷ lệ phần trăm chuỗi trong *D* chứa *p*. Ở đây, mức hỗ trợ tuyệt đối hoặc tương đối sẽ được sử dụng chuyển đổi qua lại, ký hiệu là *sup(p).*

Ví dụ: Xét cơ sở dữ liệu như bảng 1, chuỗi p = <(a)(d)> xuất hiện trong chuỗi s1, s3, s5. Vậy, độ hỗ trợ của chuỗi p là 3.

2.3. Mẫu

Mẫu là một chuỗi con của một chuỗi dữ liệu. Mỗi itemset trong một mẫu còn được gọi là một thành phần (element).

Ví dụ: *Mẫu α* = <(ab)> *là chuỗi con của chuỗi* s1

2.4. Mẫu tuần tự

Cho trước ngưỡng hỗ trợ tối thiểu (*Minsup*) xác định bởi người dùng, *minsup* ∈ (0,1]. Một mẫu α được coi là phổ biến nếu độ hỗ trợ của nó lớn hơn hoặc bằng *minsup*: *sup(α)*≥ *minsup*, khi đó α được gọi là mẫu tuần tự.

Ví dụ: Xét cơ sở dữ liệu như bảng 1.

Có tập các item phân biệt là *{a, b, c, d, e, f }*. Xét chuỗi *s1* = <(a)(bc)(ac)(d)(c)> chuỗi *s1* có 5 itemset là: <(a),(bc),(ac),(d),(c)> và có 7 item. Vậy *s1* có kích thước là 5 và có độ dài là 7. Trong chuỗi *s1*, item *a* xuất hiện 2 lần nhưng nếu tính độ hỗ trợ thì độ hỗ trợ của item *a* chỉ được tính là 1 đối với chuỗi *s1* đó. Chuỗi *p* = *<(a)*> xuất hiện trong chuỗi s1, s2, s3, s4 s5. Vậy, độ hỗ trợ của mẫu *p* là 5. Vì *sup(p)* > *minsup* nên *p* là mẫu tuần tự.

2.5. Kỹ thuật khai thác tập phổ biến trên TKS

Ví dụ: Cho cơ sở dữ liệu như bảng 1 với k =5

Bảng 2. Cơ sở dữ liệu mẫu

SID	Sequences
1	<(a)(bc)(ac)(d)(c)>
2	<(ab)(bc)(c)(a)>
3	<(cd)(ab)(df)(c)(b)>
4	<(ae)(bf)(c)(b)(c)>
5	<(bf)(ad)(d)(e)(ac)(f)>

Đầu tiên: thuật toán khởi tạo 2 tập rỗng R, L và đặt mipsup =0. Duyệt cơ sở dữ liệu D (bảng 2) để tạo ra các bit vector và đếm độ hỗ trợ của các item trên bit vector đó. Sau đó, thuật toán sẽ đi xét độ hỗ trợ của từng item xem chúng có thỏa điều kiện minsup hay không. Nếu thỏa, lưu vào tập L và sắp xếp chúng tăng dần theo độ hỗ trợ như danh sách trong từ điển. Kế tiếp là mở rộng các item có độ hỗ trợ triển vọng nhất (cao nhất). Các item thỏa điều kiện minsup gồm: e:2, d:3, f:3, a:5, b:5, c:5. Lúc này, thuật toán đã tìm ra các mẫu phổ biến và gán minsup bằng độ hỗ trợ lớn nhất của các item trong tập L với độ hỗ trợ là 5. Lưu các item có độ hỗ trợ bằng 5 vào tập L và lần lượt đi mở rộng chúng theo 2 hướng đó là s-extension và i-extension.

Mở rộng item đầu tiên là (a) mở rộng theo s-extension, ta thu được các mẫu sau:

{(a)(a)}:3,{(a),(b)}:4,{(a),(c)}:5,{(a),(d)}3,{(a),(e)}:1,{(a),(f)}:3. Ta được 1 item thỏa điều kiện minsup đề ra đó là {(a),(c)} với độ hỗ trợ là 5. Lưu tập {(a),(c)} vào L.

Tiếp đến mở rộng (a) theo hướng i-extension với minsup =5 ta có tập kết quả như sau:

{(a,b)}:2,{(a,c)}:2,{(a,d)}:1,{(a,e)}:1,{(a,f)}:0. Không có item nào thỏa điều kiện thỏa minsup dừng thuật toán mở rộng (a) theo i-extension.

Dựa vào kết quả đạt được khi mở rộng (a) s-extension ta thu đươc item {(a),(c)}. Và đi mở rộng chúng theo s-extension và i-extension lần lượt ta có kết quả như sau:

Mở rộng theo s-extension{(a),(c),(a)}:2,{(a),(c),(b)}:2,{(a),(c),(c)}:3,{(a),(c),(d)}:1,{(a),(c),(e)}:0,{(a),(c),(f)}:1. Không có item nào thỏa điều kiện thỏa minsup dừng thuật toán mở rộng {(a),(c)}, theo s-extension.

Tiếp đến mở rộng {(a),(c)} theo i-extension ta được tập kết quả như sau:

{(a),(ca)}:0,{(a),(cb)}:0,{(a),(cd)}:0,{(a),(ce)}:0,{(a),(cf)}:0. Không có item nào thỏa điều kiện thỏa minsup dừng thuật toán mở rộng {(a),(c)} theo i-extension.

Tiếp tục đi mở rộng (b) theo s-extension và i-extension với minsup =5 ta được các tập kết quả như sau:

Đối với mở rộng s-extension ta được:

{(b)(a)}:3,{(b)(b)}:3,{(b)(c)}:5,{(b)(d)}:3,{(b)(e)}:1, {(b)(f)}:2. Ta được 1 item thỏa điều kiện minsup đề ra đó là {(b),(c)} với độ hỗ trợ là 5.

Vì chúng không thỏa điều kiện minsup. Lưu tập item {(b),(c)} vào tập L.

Kế tiếp, đi mở rộng (b) theo hướng i-extension và thu được kết quả như sau:

{(ba)}:0,{(bc)}:2,{(bd)}:0,{(be)}:0,{(bf)}:1. Không có item nào thỏa điều kiện thỏa minsup dừng thuật toán mở rộng (b) theo i-extension.

Dựa vào kết quả đạt được khi mở rộng (b) s-extension ta thu đươc item {(b),(c)}và đi mở rộng chúng theo s-extension và i-extension lần lượt ta có kết quả như sau:

Đối với s-extension: {(b),(c),(a)}:1,{(b),(c),(b)}:2,{(b),(c),(c)}:3,{(b),(c),(d)}:1,{(b),(c),(e)}:0,{(b),(c),(f)}:1. Không có item nào thỏa điều kiện thỏa minsup dừng thuật toán mở rộng {(b),(c)}, theo s-extension.

Tiếp đến mở rộng {(b),(c)} theo i-extension ta được tập kết quả như sau: {(b),(ca)}:0,{(b),(cb)}:0,{(b),(cd)}:0,{(b),(ce)}:0,{(b),(cf)}:0. Không có item nào thỏa điều kiện thỏa minsup dừng thuật toán mở rộng{(b),(c)}, theo i-extension.

Cuối cùng đi mở rộng item (c) theo s-extension và i-extension với minsup =5 ta được các tập kết quả như sau:

Đối với mở rộng s-extension ta được: {(c)(a)}:3,{(c)(b)}:2,{(c)(c)}4:,{(c)(d)}:2,{(c)(e)}:0,{(c)(f)}:2. Không có item nào thỏa điều kiện thỏa minsup dừng thuật toán mở rộng (c) theo s-extension.

Kế tiếp, đi mở rộng (c) theo hướng i-extension và thu được kết quả như sau: {(c,a)}:0,{(c,b)} :0,{(c,d)}:1,{(c,e)}:0,{(c,f)}:0. Không có item nào thỏa điều kiện thỏa minsup dừng thuật toán mở rộng (c) theo i-extension.

Kết quả trả về tập L chứa 5 item có độ hỗ trợ lớp nhất là 5 đó là {(a)},{(b)},{(c)},{(a),(c)}, {(b),(c)} và bằng với mẫu k.

Kết quả trả về mẫu tuần tự phổ biến với k =5.

Bảng 3. Kết quả mẫu tuần tự phổ biến TKS

Mẫu tuần tự phổ biến	Độ hỗ trợ
<(a)>	5
<(b)>	5
<(c)>	5
<(a),(c)>	5
<(b),(c)>	5

3. Thuật toán MTKS

Thuật toán TKS [2, tr.109-120] đã tìm được k mẫu tuần như ví dụ ở bảng 3. Dựa vào ý tưởng trên, bài viết đề xuất toán thuật toán MTKS tìm mẫu tuần tự đóng dựa trên thuật toán TKS.

3.1. Thuật toán MTKS

Thực hiện: Đầu tiên, thuật toán MTKS khởi tạo các biến *R, L* là các tập rỗng và đặt *minsup* = 0 (dòng 1). Sau đó, quét cơ sở dữ liệu chuỗi *D* để tạo ra *V(D*) (dòng 2), đồng thời một danh sách của tất cả các items trong *D* được tạo ra (*Sinit*) (dòng 3). Với mỗi item *s*, độ hỗ trợ của nó được tính toán dựa trên bit vector *bv(s)* của nó trong *V(D)*. Nếu item là phổ biến thì thủ tục SAVE được gọi để lưu ⟨*s*⟩ vào *L*, với ⟨*s*⟩ và *L* là các đối số đầu vào (dòng 4 và 5). Ngoài ra, bộ ba <*s, Sinit,* các items từ *Sinit* lớn hơn *s* theo thứ tự từ điển> được lưu vào *R* để chỉ định rằng ⟨*s*⟩ có thể được mở rộng để tạo các ứng viên (dòng 6). Sau đó, một vòng lặp WHILE được thực thi. Thực hiện đệ quy bộ ba các mẫu đại diện cho mẫu *r* có độ hỗ trợ cao nhất trong *R* sao cho *sup(r)* ≥ *minsup* (dòng 7 và 8). Tiếp đó, thuật toán sử dụng bộ ba để tạo ra các mẫu bằng cách gọi thủ tục SEARCH (dòng 9) và loại bỏ bộ ba khỏi *R* ngay khi tất cả các bộ ba cho các mẫu đã trở nên không phổ biến (dòng 11). Ý tưởng của vòng lặp WHILE là luôn mở rộng các mẫu có độ hỗ trợ cao nhất đầu tiên bởi vì nó hầu như luôn tạo ra các mẫu có độ hỗ trợ cao và vì vậy, cho phép tăng *minsup* nhanh hơn để cắt tỉa nhiều không gian tìm kiếm hơn. Vòng lặp kết thúc khi không có mẫu nào trong *R c*ó độ hỗ trợ cao hơn *minsup*. Sau đó, xét nếu có một mẫu s1 khác mẫu s1 và là con của mẫu s trong tập L thì tiến hành loại bỏ mẫu s1 ra khỏi tập L. Lúc này, *L* chứa *k* mẫu tuần tự phổ biến tối đại.

Ví dụ: Cho cơ sở dữ liệu D với k=10

Bảng 4. Cơ sở dữ liệu thuật toán MTKS

SID	Sequences
1	<(a)(bc)(ac)(d)(c)>
2	<(ab)(bc)(c)(a)>
3	<(cd)(ab)(df)(c)(b)>
4	<(ae)(bf)(c)(b)(c)>
5	<(bf)(ad)(d)(e)(ac)(f)>

Cho bảng cơ sở dữ liệu như bảng 4 với 5 mẫu giao dịch (SID) chưa 6 items riêng biệt I = {a,b,c,d,e,f}. Kết quả trả về ở bảng 5, ta thấy các mẫu trả luôn có độ hỗ trợ là cao nhất trong bảng cơ sở dữ liệu.

Bảng 5. Kết quả thuật toán MTKS

Mẫu tuần tự tối đại	Độ hỗ trợ
<(a)(c)>	5
<(b)(c)>	5
<(c)(c)>	4
<(a)(b)>	4
<(b)(b)>	3
<(b)(a)>	3
<(a)(d)(c)>	3
<(b)(d)(c)>	3
<(f)(c)>	3
<(a)(c)(c)>	3

3.2. Nhận xét

So với thuật toán TSP, thuật toán MTKS đạt hiệu quả cao hơn TSP về mặt thời gian và cũng như bộ nhớ cấp phát khi sử dụng. Nhờ sử

dụng phương pháp sắp xếp các item có độ hỗ trợ cao nhất theo thứ tự tăng dần và sau đó lấy các item có độ hỗ trợ cao nhất lần lượt đi mở rộng. Với những ưu thế từ thuật toán gốc TKS mang lại, thêm vào đó là các thuật giải mới nên thuật toán MTKS đã có những bước tiến vượt trội hơn so với TSP. Trong phần 4 sẽ thực hiện hiệu suất khai thác của MTKS so với thuật toán TSP.

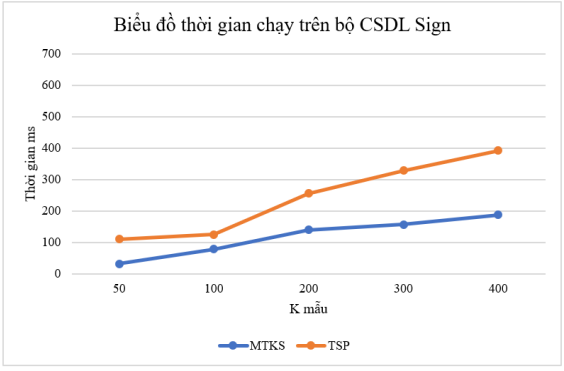
MTKS (*CSDL chuỗi D,k*)
1. $R := \emptyset$. $L := \emptyset$. *minsup* := 0.
2. Quét CSDL *để tạo V*(*D*).
3. Cho S_{init} là danh sách các items trong *V*(*D*).
4. **FOR** each item $s \in S_{init}$, **IF** *s* là phổ biến tương ứng với *bv(s)* **THEN**
5. **SAVE**(*s*, *L*, *k*, *minsup*).
6. $R := R \cup \{<s, S_{init}$, các items từ S_{init} lớn hơn *s* theo thứ tự từ điển$>\}$.
7. **WHILE** $\exists$ $<r, S1, S2> \in R$ AND sup(*r*) $\geq$ *minsup* DO
8. Chọn bộ ba $<r, S1, S2>$ với mẫu *r* có độ hỗ trợ cao nhất trong *R*.
9. **SEARCH**(*r*, *S1*, *S2*, *L*, *R*, *k*, *minsup*).
10. Loại bỏ $<r, S1, S2>$ từ R.
11. Loại bỏ từ *R* tất cả các bộ $<r, S1, S2> \in R \mid sup(r) < minsup$.
12. **END WHILE**
13. **IF** (s chứa s1 AND s != s1)
14. Loại s1 ra khỏi tập L
15. **RETURN** *L*.

Hình 1. Thuật toán MTKS

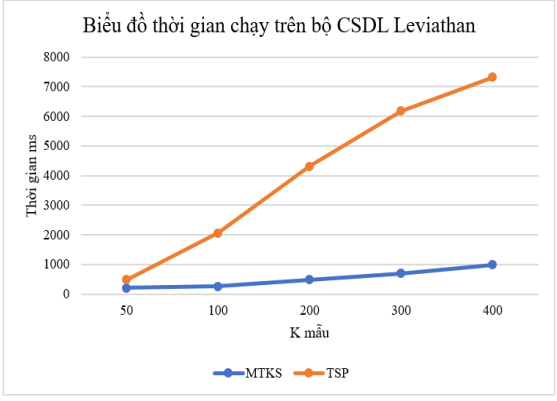
4. KẾT QUẢ THỰC NGHIỆM

Dữ liệu chuỗi là loại dữ liệu phổ biến trong nhiều lĩnh vực ứng dụng hiện nay. Do đó, bài viết tiến hành thực nghiệm trên cơ sở dữ liệu được lấy từ địa chỉ web [9] loại cơ sở dữ liệu này khá phổ biến và đa dạng như: Dữ liệu sinh học, dữ liệu web, dữ liệu viết thực thi chương trình...

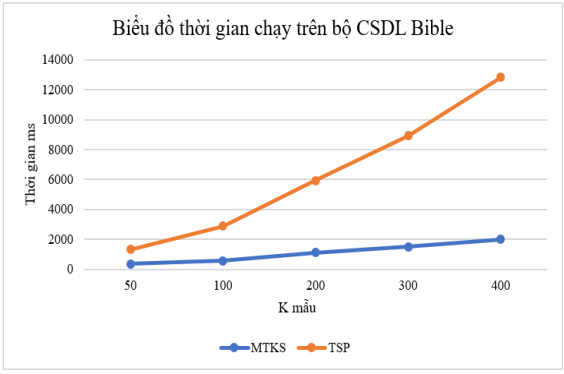
Thực nghiệm được tiến hành trên một máy tính xách tay HP, cấu hình CPU Intel core i7-8550U, 8G RAM, sử dụng hệ điều hành Microsoft Windows 10, cài đặt trên ngôn ngữ lập trình Java. Bộ dữ liệu chạy thực nghiệm từ dataset sequence bao gồm: Leviathan, Bible, Sign, FIFA. Kết quả thực nghiệm giữa MTKS và TSP; Chạy thực nghiệm 2 thuật toán MTKS và TSP với bộ dữ liệu Sign và Leviathan, Bible, FIFA lần lượt cho các mẫu k = 50, 100 , 200, 300, 400 ta có bảng kết quả như sau:



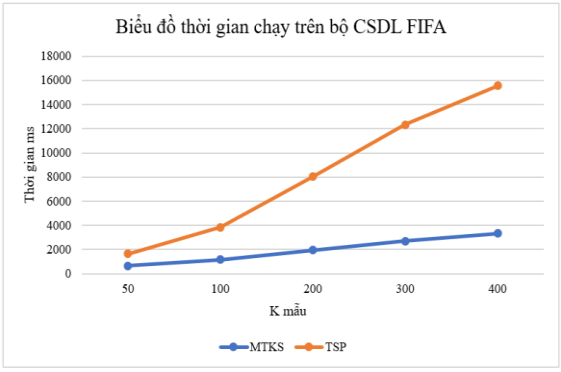
Hình 2. So sánh thời gian thực hiện giữa MTKS và TSP trên cơ sở dữ liệu Sign



Hình 3. So sánh thời gian thực hiện giữa MTKS và TSP trên cơ sở dữ liệu Leviathan



Hình 4. So sánh thời gian thực hiện giữa MTKS và TSP trên cơ sở dữ liệu Bible



Hình 5. So sánh thời gian thực hiện giữa MTKS và TSP trên cơ sở dữ liệu FIFA

Với kết quả chạy thực nghiệm như ta nhận thấy rằng thuật toán MTKS có thời gian thực thi nhanh hơn hẳn so với TSP. Nhất là khi người dùng ngưỡng k mẫu cần tìm càng lớn thời gian chạy giữa MTKS và TSP càng lớn. Theo kết quả so sánh ở hình 2 với cơ sở dữ liệu Sign thì khi nhập k = 50 đến 400 ta thấy trung bình thời gian chạy của MTKS nhanh gấp 2.2 lần so với TSP. Với các bộ cơ sở dữ liệu khác nhau độ chênh lệch về thời gian cũng khác nhau. Đối với các cơ sở dữ liệu lớn và có nhiều item thời gian thực thi của TSP cũng sẽ tăng dần theo. Như hình 4 cùng với mẫu k = 50 nhưng TSP có thời gian chạy lâu hơn và gấp 3.67 MTKS với mẫu k = 400 gấp 6.38 lần. Như vậy thuật toán MTKS cho ta thấy được khi thực thi trên cơ sở dữ liệu lớn và có số lượng item nhiều thì thời gian thực hiện tốt hơn so với TSP.

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Bài viết đã tìm hiểu cơ sở lý thuyết khai thác mẫu tuần tự và mẫu tuần tự tối đại. Qua đó, cũng thấy được tầm quan trọng trong khai thác mẫu tuần tối đại hiện nay. Khai thác mẫu tuần tự tối đại là tìm tất cả các chuỗi cha loại các chuỗi con ứng với k mẫu nhập vào. Tìm ra những mẫu có độ hỗ trợ cao nhất và loại bỏ các trường hợp tìm ra các mẫu bị trùng lắp. Cho đến nay, rất nhiều thuật toán được đưa ra trong đó có thuật toán TSP đã giải quyết được vấn đề trên nhưng khi thực thi thuật toán còn nhiều hạn chế về tốc độ cũng như dung lượng lưu trữ. Để giải quyết những khuyết điểm trên, bài viết đã đề xuất triển khai thuật toán MTKS. Với những ưu thế từ thuật toán gốc TKS mang lại, thêm vào đó là các thuật giải mới nên thuật toán MTKS đã có những bước tiến vượt trội hơn so với TSP. Kết quả thực nghiệm đã chứng minh thuật toán MTKS không chỉ tiết kiệm được bộ nhớ sử dụng mà còn có thời gian thực thi nhanh hơn hẳn so với TSP.

5.2. Hướng phát triển

Khai thác mẫu tuần tự tối đại rất hữu ích trong việc khai thác những tri thức tiềm ẩn trong nguồn dữ liệu ở dạng tuần tự. Trong thời đại ngày nay, khối lượng dữ liệu khai thác rất lớn, đòi hỏi chúng ta khai thác làm sao cho hiệu quả với thời gian thực thi là ngắn nhất và dung lượng sử dụng nhỏ nhất.

Để tìm k mẫu tuần tự tối đại thuật toán đã sử dụng tính năng giao bit vector làm rất tốn nhiều thời gian cũng như bộ nhớ để xử lý. Do đó, hướng phát triển tiếp theo là khi khai thác thuật toán MTKS ta nên kết hợp thêm thuật toán mã hóa khối nguyên tố (Dynamic Bit-Vector) [8, tr.7196-7206] để thời gian chạy nhanh nhất và dung lượng được nhỏ nhất.

TÀI LIỆU THAM KHẢO

[1] Agrawal and R. Srikant (1995), *Mining sequential patterns, Proc. 11th Int. Conf, Data Eng,*.

[2] Fournier-Viger, A Gomariz, T Gueniche, E Mwamikazi, R Thomas (2013), *International Conference on Advanced Data Mining and Applications*.

[3] Guha, R. Rastogi, and R. K. Shim (2009), *A robust clustering algorithm for categorical attributes, In ICDE'99*.

[4] Mannila, H. Toivonen, and A. Verkamo (1997), *Discovery of frequent episodes in event sequences, Data Min. Knowl.…,*.

[5] Myra Spiliopoulou (1999), *Managing Interesting Rules in Sequence Mining*, Proceedings of European Conference on Principles of Data Mining and Knowledge Discovery.

[6] Srikant and R. Agrawal (1996), *Mining Sequential Patterns: Generalizations and Performance Improvements, Proc. 5th Int. Conf. Extending Database Technol. Adv. Database Technol.*

[7] Tzvetkov, X. Yan, and J. Han (2005), *TSP: Mining top-k closed sequential patterns,* Knowl. Inf. Syst., vol. 7, no. 4.

[8] Bay Vo, Tzung-Pei Hong, Bac Le (2012) DBV-Miner: A Dynamic Bit-Vector approach for fast mining frequent closed itemsets Expert Systems with Applications, Volume 39, Issue 8, 15 June 2012.

[9] http://www.philippe-fournier-viger.com/spmf/index.php?link=download.php.

Ngày nhận bài: 23-10-2020. Ngày biên tập xong: 11-01-2021. Duyệt đăng: 22-01-2021