ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

TRẦN ANH DUY

KỸ THUẬT CẮT NHÁNH TRONG QUÁ TRÌNH PHÁT SINH ỨNG VIÊN KHI KHAI THÁC MẪU TUẦN TỰ

Ngành : KHOA HỌC MÁY TÍNH
Mã số : 60480101

ĐỒ ÁN THẠC SĨ KHOA HỌC MÁY TÍNH

TP. Hồ Chí Minh-2016

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

TRẦN ANH DUY

KỸ THUẬT CẮT NHÁNH TRONG QUÁ TRÌNH PHÁT SINH ỨNG VIÊN KHI KHAI THÁC MẪU TUẦN TỰ

Ngành : KHOA HỌC MÁY TÍNH
Mã số : 60480101

ĐỒ ÁN THẠC SĨ KHOA HỌC MÁY TÍNH

NGƯỜI HƯỚNG DẪN KHOA HỌC
PGS.TS. LÊ HOÀI BẮC
TP. Hồ Chí Minh-2016

LỜI CẢM ƠN

Tôi xin gởi lời cảm ơn chân thành đến tất cả các thầy cô khoa Công Nghệ Thông Tin trường đại học Khoa Học Tự Nhiên TP.HCM đã giảng dạy bằng tất cả sự nhiệt tình trong quá trình tôi học tập tại đây.

Xin cảm ơn các cán bộ, nhân viên của phòng Đào Tạo Sau Đại Học đã hỗ trợ và hướng dẫn tỉ mĩ các thủ tục cần thiết trong suốt quá trình học.

Đặc biệt tôi xin gởi lời cảm ơn chân thành nhất và lòng biết ơn sâu sắc nhất đến giảng viên hướng dẫn, PGS.TS. Lê Hoài Bắc vì đã tận tình chỉ dạy tôi trong quá trình học, cũng như hướng dẫn và giúp đỡ tôi trong quá trình làm đồ án.

Con xin cảm ơn Cha Mẹ đã làm tất cả cho con để có thể hoàn thành khóa học này.

Cuối cùng, tôi xin gửi lời cảm ơn đến nhóm bạn của tôi ở lớp cao học khóa 24 trường đại học Khoa Học Tự Nhiên TP.HCM vì đã hỗ trợ, giúp đỡ, động viên tôi rất nhiều để tôi có thể hoàn thành được đồ án tốt nghiệp này.

Trần Anh Duy

MỤC LỤC

DANH MỤC CÁC BẢNG

DANH MỤC HÌNH VẼ VÀ ĐỒ THỊ

DANH MỤC CÁC TỪ VIẾT TẮT

STT	Từ viết tắt	Ý nghĩa
01	CSDL	Cơ sở dữ liệu
02	minsup	Ngưỡng hỗ trợ nhỏ nhất
03	SDB	Cơ sở dữ liệu tuần tự
04	SPADE	Khai thác mẫu tuần tự dựa trên lớp tương đương
05	CMAP	Ánh xạ đồng xuất hiện
06	SID	Định danh chuỗi

MỞ ĐẦU

Từ những năm đầu của thế kỷ 21, máy tính và những ứng dụng của nó đã trở nên phổ biến với con người ở mọi tầng lớp xã hội. Các ứng dụng chạy trên máy tính đã xâm nhập vào rất nhiều lĩnh vực đời sống như lĩnh vực kinh tế thông qua các ứng dụng kế toán hay thương mại điện tử, hay lĩnh vực du lịch thông qua các ứng dụng hướng dẫn đường đi và đặt vé trực tuyến, lĩnh vực giải trí thông qua các ứng dụng xem phim hay đọc truyện …

Ngày nay, mọi thông tin liên quan đến con người đều có khả năng được thể hiện và lưu trữ, thông qua các ứng dụng máy tính, dưới dạng dữ liệu. Dữ liệu đó bao gồm : doanh thu của doanh nghiệp qua nhiều tháng, điểm thi của một cá nhân trong những bài thi thử trực tuyến, lộ trình du lịch hay lộ trình di chuyển của một đoàn khách, thể loại phim được xem nhiều của một nhóm người nào đó … Từ việc khai thác được thông tinh chứa trong dữ liệu, chúng ta có thể sử dụng chúng phục vụ cho cuộc sống chúng ta như : tăng sức cạnh tranh của doanh nghiệp, hiểu khách hàng hơn, nắm rõ những quy luật tìm ẩn … Do đó, lĩnh vực khai thác dữ liệu trở thành một trong những lĩnh vực nghiên cứu quan trọng nhất hiện nay.

Khai thác dữ liệu là tìm ra những thông tin hữu ích tìm ẩn trong khối lượng lớn dữ liệu được lưu trữ. Có rất nhiều loại dữ liệu khác nhau như : dữ liệu giao tác (transaction), dữ liệu chuỗi (sequence), dòng dữ liệu (data stream), dữ liệu chuỗi thời gian (time-series), dữ liệu ảnh số, … Ứng với mỗi loại dữ liệu khác nhau sẽ có những phương pháp khai thác khác nhau. Nội dung đồ án tập trung vào một trong các phương pháp khai thác dữ liệu chuỗi hiệu quả nhất hiện nay là phương pháp SPADE.

Phương pháp SPADE là phương pháp được sử dụng để khai thác những chuỗi tuần tự phổ biến (frequent sequences) trong CSDL tuần tự, do Zaki đề xuất năm 2001. Bài toán khai thác chuỗi tuần tự phổ biến, hay còn gọi là mẫu tuần tự (Sequential Patterns), như sau : “Cho một CSDL tuần tự và một ngưỡng hỗ trợ nhỏ

nhất (minsup), yêu cầu tìm ra tất cả các chuỗi con có số lần xuất hiện trong CSDL lớn hơn hay bằng ngưỡng hỗ trợ nhỏ nhất.” Ứng dụng của bài toán này có thể tìm thấy trong : phân tích chuỗi mua hàng của khách hàng của một siêu thị để xác định những mặc hàng thường được mua nhất, phân tích chuỗi ký tự đặc biệt trong một văn bản để xác định nội dung hay mục đích của văn bản, phân tích chuỗi thông tin di truyền (ADN hay ARN) để xác định nguyên nhân bệnh ung thư, phân tích chuỗi truy cập ứng dụng Web của người dùng để xác định nhu cầu sử dụng Web của họ ... .Với số lượng chuỗi cần xữ lý là rất lớn và chiều dài chuỗi là không xác định nên công việc tìm ra tất cả các chuỗi con, thỏa ngưỡng hỗ trợ nhỏ nhất, trong CSDL là một công việc đầy thách thức.

Có rất nhiều thuật toán được đề xuất cho việc khai thác dữ liệu chuỗi như AprioriAll[1], GSP[4], SPADE[3], PrefixSpan[2], SPAM[5], ClaSP[6], … Thuật toán SPADE đang xét (và một số thuật toán khác như GSP, SPAM …) sử dụng chiến lượt khai thác là phát sinh nên các chuỗi ứng viên, sau đó kiểm tra và loại bỏ các chuỗi ứng viên không thỏa ngưỡng hỗ trợ do người dùng đặt ra. Với CSDL lớn và độ hỗ trợ nhỏ, số lượng ứng viên được phát sinh trong quá trình chạy thuật toán là rất lớn. Trong số các ứng viên đó, có những ứng viên không tồn tại trong CSDL hoặc có độ hỗ trợ nhỏ hơn ngưỡng hỗ trợ nhỏ nhất. Vì thế tốn rất nhiều thời gian để phát sinh và phát hiện các mẫu ứng viên dư thừa như trên. Việc loại bỏ các ứng viên dư thừa như vậy từ gốc sẽ giúp tiết kiệm rất nhiều thời gian và bộ nhớ trong quá trình thực thi. Mặc dù đã có những phương pháp hiệu quả để làm giảm số lượng ứng viên dư thừa được phát sinh song số lượng ứng viên dư thừa vẫn rất lớn. Nhu cầu phát hiện và chặn trước việc phát sinh các ứng viên dư thừa đang là một thách thức đối với các nhà nghiên cứu khai thác dữ liệu.

Từ yêu cầu giảm bớt việc phát sinh nên các ứng viên dư thừa đã nêu, đồ án này đề xuất một bước cải tiến cho thuật toán SPADE cho phép làm giảm số lượng

ứng viên dư thừa phát sinh. Kết quả thực nghiệm cho thấy thời gian thực thi lẫn bộ nhớ tiêu hao của phương pháp đề xuất ít hơn nhiều so với phương pháp gốc.

Bố cục của đồ án như sau :

- Chương 1 : Trình bày tổng quan về khai thác dữ liệu chuỗi, những ứng dụng của nó và một số bài toán khai thác phổ biến hiện nay.
- Chương 2 : Trình bày cơ sở lý thuyết của các phương pháp khai thác dữ liệu chuỗi, chi tiết về thuật toán SPADE và cấu trúc cải tiến CMAP.
- Chương 3 : Trình bày về phương pháp khai thác mẫu tuần tự dựa trên cấu trúc bảng trạng thái.
- Chương 4 : Trình bày về kết quả thực nghiệm và so sánh với các thuật toán gốc SPADE và thuật toán cải tiến CM-SPADE.
- Chương 5 : Kết luận và hướng phát triển trong tương lai.

CHƯƠNG 1: TỔNG QUAN

1.1 Giới thiệu

Khái niệm khai thác dữ liệu chuỗi được Rakesh Agrawal và Ramakrishnan Srikant giới thiệu lần đầu tiên vào năm 1995 [1]. Vấn đề đầu tiên được nhắc đến thuộc lĩnh vực phân tích kinh doanh, mục đích chính là tìm ra các mẫu phổ biến trong chuỗi mua hàng của khách hàng trong một siêu thị, với các hóa đơn được sắp xếp theo thời gian. Sau đó, khái niệm khai thác chuỗi được ứng dụng rộng rãi trong nhiều lĩnh vực y tế, giáo dục, quân sự, kinh tế v.v…

Dữ liệu chuỗi có các đặc điểm phân biệt với các loại dữ liệu khác (dữ liệu giao tác, dữ liệu ảnh, video …) như sau :

- Kích thước chuỗi hay chiều dài của chuỗi đang xét là không cố định. Trong cùng một CSDL, kích thước mỗi chuỗi có thể khác nhau thậm chí có sự chênh lệch lớn. Ví dụ chuỗi các từ trong mỗi câu trong một văn bản: Có những câu rất dài, gồm đầy đủ các thành phần chủ ngữ-vị ngữ, nhưng có những câu rất ngắn chỉ gồm một vài từ cảm thán.
- Vị trí của các thành phần trong chuỗi là thông tin rất quan trọng. Ví dụ chuỗi XY sẽ hoàn toàn khác chuỗi YX. Mối quan hệ về thứ tự hay vị trí của các thành phần trong chuỗi là đặc điểm chỉ có ở dữ liệu chuỗi. Đây là điểm khác biệt cơ bản so với dữ liệu khác.

1.2 Ví dụ về dữ liệu chuỗi

- **Chuỗi văn bản** : Dữ liệu văn bản là dữ liệu chuỗi bao gồm các từ đã được sắp xếp theo những vị trí cố định để tạo thành câu có nghĩa. Chuỗi văn bản là loại dữ liệu chúng ta thường thấy, nhất là trên báo, trên các trang web, trên các bản hướng dẫn, tờ rơi … Việc khai thác chuỗi văn bản cho phép chúng ta xác nhận được nội dung, chủ đề, đối tượng mà văn bản đề cập đến, và cũng

cho phép chúng ta xác định được cảm xúc của người viết ra văn bản đó trong một số trường hợp đặc biệt. Việc khai thác chuỗi văn bản mang lại nhiều thông tin hữu ích cho doanh nghiệp, ví dụ như trong việc lấy thông tin phản hồi từ người dùng về sản phẩm hay dịch vụ, từ những bài viết họ đăng trên các trang mạng xã hội, các bình luận báo chí, hay trong các bài viết cá nhân.

- **Chuỗi sự kiện** : Là một dạng dữ liệu bao gồm các sự kiện hay sự vật cụ thể gắng liền với một chuỗi thời gian. Ví dụ như các hàng hóa được mua từ một khách hàng tại 1 siêu thị của nhiều tháng trong năm, hay sự thay đổi của nhiệt độ của các ngày trong tháng, hay sự biến động của giá vàng trong nhiều tháng, hay thông tin truy cập web của một cá nhân nào đó. Việc khai thác chuỗi sự kiện cho phép chúng ta phát hiện ra các mẫu thông tin hữu ích tiềm ẩn trong dữ liệu như: quy luật của thời tiết, sở thích khách hàng, lưu thông thông tin trên web. ...
- **Chuỗi dữ liệu sinh học** : là dữ liệu mô tả chuỗi các thành phần cấu thành nên các chất trong cơ thể sống ví dụ như deoxyribonucleic acid (DNA), amino acid (Protein), hay ribonucleic acid (RNA) … Việc khai thác chuỗi sinh học cho chúng ta thông tin hữu ích về cấu thành của một số chuỗi đặc biệt chỉ xuất hiện ở một số cá nhân hay tập thể khác thường nào đấy. Điều này giúp ích nhiều trong việc phát hiện và điều trị các bệnh nan y.

1.3 Một số bài toán khai thác dữ liệu chuỗi

Có rất nhiều bài toán khác nhau phát sinh trong việc khai thác dữ liệu chuỗi. Các bài toán này đến từ các yêu cầu khác nhau và cũng có những phương pháp khác nhau. Ở đây liệt kê 2 dạng bài toán thông dụng nhất là :

- **Khai thác mẫu tuần tự (sequential pattern mining)** : Là quá trình tìm kiếm các chuỗi dữ liệu phổ biến trong CSDL. Bài toán cho trước một CSDL tuần tự và một ngưỡng hỗ trợ nhỏ nhất. Yêu cầu đặt ra là tìm ra tất cả các

chuỗi con có số lần xuất hiện lớn hơn hoặc bằng ngưỡng hỗ trợ nhỏ nhất, hay còn gọi là mẫu tuần tự.

- **Khai thác luật tuần tự (sequential rule mining)** : Là quá trình tìm ra mối quan hệ giữa các sự kiện trong CSDL. Mối quan hệ giữa các sự kiện này được biểu thị bởi một luật có dạng X→ Y, nghĩa là với một sự kiện X xảy ra trước thì sẽ có một sự kiện Y xảy ra ngay sau đó.

1.4 Đóng góp của đồ án

Nội dung đồ án tập trung vào bài toán khai thác mẫu tuần tự. Cụ thể là với phương pháp khai thác mẫu tuần tự phổ biến gọi là SPADE[3] do Zaki đề xuất năm 2001, và phương pháp cải tiến do Fournier-Viger đề xuất năm 2014 với tên gọi CM-SPADE[7], tác giả đồ án đề xuất một bước bổ sung cho phép cải thiện tốc độ xữ lý và tiết kiệm vùng nhớ trên một số bộ dữ liệu có tính chất riêng biệt.

Đề xuất được nói đến trong đồ án bao gồm 2 phần :

- **Phần 1** : Điều chỉnh lại các bước khai thác trong thuật toán SPADE sao cho phù hợp với đề xuất, đồng thời tiết kiệm vùng nhớ đang thao tác. Cụ thể, tác giả đồ án vẫn sử dụng cấu trúc biểu diễn dữ liệu của Zaki để lưu trữ thông tin về dữ liệu chuỗi, nhưng tác giả thay đổi phương pháp khai thác trên tập dữ liệu lưu trữ đó bằng cách bổ sung thêm cấu trúc dữ liệu biểu diễn trạng thái hiện thời của mẫu đang xét và tiến hành khai thác dựa trên cấu trúc trạng thái hiện thời đó.
- **Phần 2** : Bổ sung thêm một bước kiểm tra ngay sau phương pháp CM-SPADE cho phép tăng tốc độ xữ lý của thuật toán. Cụ thể, tác giả đồ án vẫn sử dụng cấu trúc CMAP[7] của Fournier-Viger nhưng bổ sung thêm một bước kiểm tra dựa trên thông tin lưu trữ trong cấu trúc trạng thái hiện thời. Từ đó cải thiện được tốc độ xữ lý trên một số bộ dữ liệu có tính chất riêng biệt.

CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

2.1 Những khái niệm cơ bản

Cơ sở dữ liệu tuần tự : Cho tập hợp $I=\{i_1,i_2,\ldots,i_l\}$ là tập các sự kiện (item). Một tập hợp sự kiện (itemset) $I_x=\{i_1,i_2,\ldots,i_m\}$ là một tập hợp, không xét thứ tự, các sự kiện không trùng nhau và $I_x \subseteq I$. Chuỗi tuần tự là một danh sách đã sắp xếp các tập sự kiện $s=<I_1,I_2,\ldots,I_n>$ với $I_k \subseteq I$ $(1 \leq k \leq n)$. CSDL tuần tự SBD là danh sách các chuỗi tuần tự $SDB=<s_1,s_2,\ldots,s_p>$ với chỉ số xác định của mỗi chuỗi là 1,2,…,p. Ví dụ một CSDL tuần tự trong hình 2.1: CSDL trong hình bao gồm 4 chuỗi với định danh là 1,2,3,4 trong cột SID. Chuỗi <{D},{A},{B}> là một chuỗi tuần tự chứa 3 tập sự kiện (itemset),mỗi tập sự kiện có 1 phần tử, gồm có sự kiện D, trong tập thứ nhất, xảy ra đầu tiên, sau đó là sự kiện A trong tập thứ 2, sau cùng là sự kiện B, trong tập thứ 3, diễn ra.

SID	Sequences
1	<{A},{B},{C},{A},{D},{B}>
2	<{A},{B},{D},{A},{B}>
3	<{D},{B}>
4	<{D},{A},{B}>

Hình 2.1 CSDL tuần tự

Chuỗi con tuần tự: Một chuỗi $s_a=<A_1,A_2,\ldots,A_n>$ được gọi là chuỗi con của $s_b=<B_1,B_2,\ldots,B_m>$ nếu và chỉ nếu tồn tại dãy số $1 \leq i_1 < i_2 < \ldots < i_n \leq m$ sao cho $A_1 \subseteq B_{i1}$, $A_2 \subseteq B_{i2},\ldots,A_n \subseteq B_{in}$. Khi đó ta nói chuỗi s_b chứa chuỗi s_a.

Kích thước chuỗi (size of a sequence) : Ta ký hiệu kích thước chuỗi s là |s|, bằng số tập sự kiện (itemset) có trong chuỗi s.

Chiều dài chuỗi (length of a sequence): Ta ký hiệu chiều dài chuỗi s là l bằng số sự kiện (item) có trong chuỗi s. Chuỗi có chiều dài k gọi là k-sequence.

Tiền tố (prefix): Chuỗi $s_a=<A_1,A_2,\ldots,A_n>$ gọi là tiền tố của chuỗi $s_b=<B_1,B_2,\ldots,B_m>$ nếu và chỉ nếu với mọi $n < m$ và $A_1=B_1$, $A_2=B_2$, … , $A_n=B_n$.

Độ hỗ trợ (support): Độ hỗ trợ của chuỗi s_a trong CSDL tuần tự được định nghĩa là số lượng chuỗi bên trong CSDL có chứa chuỗi s_a. Ký hiệu là $sup_{SDB}(s_a)$.

Mở rộng chiều dài chuỗi : Với chuỗi $s=<A_1,A_2,\ldots,A_n>$ và sự kiện i_k. Chuỗi s' được gọi là mở rộng theo chiều dài của chuỗi s với sự kiện i_k nếu và chỉ nếu $s'=<A_1,A_2,\ldots,A_n,A_{n+1}>$ và $A_{n+1}=\{i_k\}$.

Khai thác mẫu tuần tự (sequential pattern mining): Với CDSL tuần tự SDB cho trước và một ngưỡng hỗ trợ nhỏ nhất (minsup) do người dùng tự định nghĩa, một chuỗi s được gọi là phổ biến, hay còn gọi là mẫu tuần tự, nếu và chỉ nếu $sup_{SDB}(s) \geq$ minsup. Vấn đề của khai thác chuỗi tuần tự phổ biến là tìm ra tất cả các mẫu tuần tự có trong CSDL.

Định dạng dữ liệu dạng ngang : Là định dạng dữ liệu mà mỗi mẫu tin (entry) là một chuỗi tuần tự, gắng liền với mỗi chuỗi là định danh của chuỗi (SID). Hình 2.1 mô tả định dạng dữ liệu dạng ngang.

Định dạng dữ liệu dạng dọc : Là định dạng dữ liệu mà mỗi mẫu tin chứa thông tin của một sự kiện (item). Thông tin của sự kiện bao gồm định danh chuỗi mà sự kiện ấy xảy ra và vị trí hay tập sự kiện, trong chuỗi đó, có chứa sự kiện đang xét.

2.2 Thuật toán SPADE

Thuật toán SPADE (**S**equential **PA**ttern **D**iscovery using **E**quivalence classes) do Zaki đề xuất năm 2001 để giải bài toán khai thác mẫu tuần tự phổ biến. Trước đó đã có các thuật toán như AprioriAll[1], GSP[4] được đề xuất để giải bài toán này, nhưng các thuật toán trên đều phải dựa trên thao tác truy xuất CSDL nhiều lần để tìm ra các mẫu tuần tự phổ biến và sử dụng nhiều cấu trúc phức tạp với hiệu

quả thấp. Vì thế, Zaki đề xuất thuật toán SPADE để khắc phục vấn đề truy xuất CSDL nhiều lần, đồng thời ông cũng chỉ ra cách thức để tìm ra các mẫu tuần tự phổ biến dựa trên thao tác kết đơn giản. Lợi ích của thuật toán SPADE là kết hợp các đặc điểm của dữ liệu để chia nhỏ bài toán gốc thành các bài toán con, từ đó có thể giải quyết bài toán hiệu quả với thao tác lọc và thao tác kết. Những đặc điểm chính của thuật toán SPADE như sau:

- Định dạng dữ liệu được sử dụng trong thuật toán SPADE là định dạng dọc. Với việc sử dụng dữ liệu định dạng dọc, tất cả các mẫu tuần tự có thể được liệt kê thông qua thao tác kết thông tin của các sự kiện với nhau.
- Lý thuyết dàn được sử dụng để có thể chia nhỏ không gian tìm kiếm.Bên cạnh đó, thuật toán chỉ cần duyệt CSDL 3 lần, do đó làm tối thiểu hóa chi phí truy xuất dữ liệu.
- Thuật toán sử dụng 2 chiến lượt duyệt theo chiều rộng và duyệt theo chiều sâu để khai thác dữ liệu tuần tự.

2.2.1 Cấu trúc dữ liệu lưu trữ thông tin sự kiện (id-list)

Trong định dạng dữ liệu dạng dọc, thông tin của một sự kiện i_k, thuộc *I,* được lưu trữ theo dạng <sid,pos(s)>. Trong đó: s là chuỗi đang xét, sid là định danh của chuỗi s và pos(s) là vị trí xuất hiện của sự kiện i_k trong chuỗi s. Thuật toán SPADE lưu trữ một danh sách các <sid,pos(s)> cho từng sự kiện khác nhau.

Ví dụ : Ta biểu diễn lại CSDL dạng ngang trong hình 2.1 thành CSDL dạng dọc như hình 2.2. Khi đó, thông tin của các sự kiện được lưu trữ bằng danh sách A={<1,1>,<1,4>,<2,1>,<2,4>,<4,2>},B={<1,2>,<1,6>,<2,2>,<2,5>,<3,2>,<4,3>}, C={<1,3>}, D={<1,5>,<2,3>,<3,1>,<4,1>}

Độ hỗ trợ của từng sự kiện được tính bằng số định danh có xuất hiện sự kiện ấy. Như trong CSDL ở hình 2.2 ta có $sup_{SDB}(A)=3$, $sup_{SDB}(B)=4$, $sup_{SDB}(C)=1$, $sup_{SDB}(D)=4$.

A(3)	
SID	Itemset
1	1,4
2	1,4
3	
4	2

B(4)	
SID	Itemset
1	2,6
2	2,5
3	2
4	3

C(1)	
SID	Itemset
1	3
2	
3	
4	

D(4)	
SID	Itemset
1	5
2	3
3	1
4	1

Hình 2.2 CSDL định dạng dọc

Trong thuật toán SPADE, để mở rộng một chuỗi có chiều dài k sang 1 chuỗi có chiều dài k+1 ta thực hiện phép kết chuỗi dựa trên thông tin của các sự kiện. Ví dụ như phép kết được thực hiện trong hình 2.3, ta mở rộng chuỗi A sang chuỗi AA bằng cách kết thông tin vị trí theo các định danh chuỗi tương ứng.

A(3)	
SID	Itemset
1	**1**,4
2	**1**,4
3	
4	2

∩

A(3)	
SID	Itemset
1	1,4
2	1,4
3	
4	2

=

AA(2)	
SID	Itemset
1	4
2	4

Hình 2.3 : Phép kết mở rộng chiều dài chuỗi

Dựa vào việc kết thông tin của các sự kiện lại với nhau thuật toán đã phát sinh thành công một chuỗi tuần tự mới. Sau đó, thuật toán sẽ tính độ hỗ trợ của chuỗi mới và so sánh với ngưỡng hỗ trợ nhỏ nhất để xem chuỗi mới phát sinh có phải là một mẫu tuần tự hay không. Sau tất cả các thao tác kết và kiểm tra, kết quả thu được sẽ là tất cả các mẫu tuần tự có trong CSDL.

2.2.2 Thuật toán

Thuật toán SPADE, được mô tả trong hình 2.4, nhận tham số đầu vào là CSDL tuần tự SDB và ngưỡng hỗ trợ nhỏ nhất minsup. Đầu tiên, thuật toán sẽ chuyển CSDL dạng ngang sang CSDL dạng dọc V(SDB) rồi định nghĩa tập các mẫu tuần tự F1, là tập các mẫu tuần tự có độ dài bằng 1 hay chuỗi 1-sequence phổ biến. Sau đó, SPADE gọi thủ tục ENUMERATE với tham số là tập F1.

SPADE(SDB,minsup)

1. Scan SDB to create V(SDB) and identify F_1 the list of frequent items
2. **ENUMERATE**(F_1)

ENUMERATE(an equivalence class F)

1. **FOR** each pattern $A_i \in F$
2. Output Ai
3. $T_i := \varnothing$.
4. **FOR** each pattern $A_j \in F$, with $j \geq i$
5. R=MergePattern(A_i , A_j)
6. **FOR** each pattern $r \in R$
7. **IF** sup(R) ≥ minsup **THEN**
8. $T_i := T_i \cup \{R\}$;
9. **ENUMERATE**(Ti)

Hình 2.4 Thuật toán SPADE

Thủ tục ENUMERATE nhận tham số là lớp tương đương F. Một lớp tương đương kích thước k được định nghĩa là tập tất cả các mẫu tuần tự chứa k sự kiện và có chung tiền tố chứa k-1 sự kiện trước đó. Thủ tục xuất ra tất cả các mẫu tuần tự A_i chứa trong tập F. Sau đó, thuật toán thực hiện phép kết tất cả các mẫu tuần tự chứa trong tập F với nhau để tạo chuỗi mới, rồi tính độ hỗ trợ của chuỗi mới và thực hiện so sánh với ngưỡng hỗ trợ nhỏ nhất để tìm ra mẫu tuần tự mới. Sau cùng là đưa mẫu

tuần tự mới vào lớp tương đương T_i để thực hiện đệ quy, bằng cách gọi thủ tục ENUMERATE với lớp tương đương T_i. Kết quả của thuật toán là xuất ra tất cả các mẫu tuần tự thỏa ngưỡng hỗ trợ nhỏ nhất được lưu trữ trong các lớp tương đương.

Ví dụ : Với CSDL đã cho trong hình 2.2 và minsup=3, sử dụng thuật toán SPADE để khai thác mẫu tuần tự, chúng ta có tập 1-sequence phổ biến như hình 2.5

A(3)	
SID	Itemset
1	1,4
2	1,4
3	
4	2

B(4)	
SID	Itemset
1	2,6
2	2,5
3	2
4	3

D(4)	
SID	Itemset
1	5
2	3
3	1
4	1

Hình 2.5 : Bước khai thác tập 1-sequence phổ biến

Với mỗi sự kiện trong hình 2.5 chúng ta tiến hành mở rộng để thu được chuỗi 2-sequence phổ biến như sau :

AA(2)	
SID	Itemset
1	4
2	4
3	
4	

AB(3)	
SID	Itemset
1	2,6
2	2,5
3	
4	3

AD(2)	
SID	Itemset
1	5
2	3
3	
4	

BA(2)	
SID	Itemset
1	4
2	4
3	
4	

BB(2)	
SID	Itemset
1	6
2	5
3	
4	

BD(2)	
SID	Itemset
1	5
2	3
3	
4	

DA(2)	
SID	Itemset
1	
2	4
3	
4	2

DB(4)	
SID	Itemset
1	6
2	5
3	2
4	3

DD(0)	
SID	Itemset
1	
2	
3	
4	

Hình 2.6 : Bước khai thác tập 2-sequence phổ biến

Sau bước khai thác 2-sequence chúng ta tiếp tục mở rộng mẫu AB và DB để được tập 3-sequence như sau :

ABB(2)	
SID	Itemset
1	6
2	5
3	
4	

DBB(0)	
SID	Itemset
1	
2	
3	
4	

Hình 2.7 : Bước khai thác tập 3-sequence phổ biến

Sau bước phát sinh chuỗi 3-sequence, các mẫu ứng viên được phát sinh không thỏa minsup=3 nên thuật toán dừng. Tập mẫu phổ biến gồm các mẫu : A, B, D, AB, DB.

2.2.3 Ưu điểm và khuyết điểm

Ưu điểm của thuật toán SPADE là việc sử dụng định dạng dữ liệu dạng dọc cho phép tối ưu việc phát sinh mẫu ứng viên và tính toán độ hỗ trợ của mẫu mà không tốn chi phí truy xuất CSDL. Thuật toán chỉ cần duyệt CSDL 3 lần để khởi tạo CSDL định dạng dọc, sau đó mọi thao tác tính toán độ hỗ trợ đều dựa vào thông tin lưu trữ trong các mẫu ứng viên. Do không tốn nhiều chi phí truy xuất CSDL nên hiệu quả đạt được của thuật toán SPADE cao hơn so với thuật toán sử dụng định dạng dữ liệu ngang.

Khuyết điểm của thuật toán SPADE là sử dụng chiến thuật phát sinh ứng viên và kiểm tra độ hỗ trợ của từng ứng viên so với ngưỡng hỗ trợ nhỏ nhất. Với CSDL lớn, số lượng ứng viên cần phát sinh rất nhiều. Trong các ứng viên được phát sinh có những ứng viên không xuất hiện trong CSDL hoặc có độ hỗ trợ nhỏ hơn ngưỡng hỗ trợ nhỏ nhất. Từ đó dẫn đến việc tốn chi phí cho thao tác lưu trữ và tính toán trên các mẫu ứng viên đó.

2.3 Cấu trúc CMAP và thuật toán CM-SPADE

2.3.1 Cấu trúc CMAP

Cấu trúc CMAP là một cấu trúc ánh xạ, do Philippe Fournier-Viger đã đề xuất năm 2014, để khắc phục khuyến điểm của thuật toán SPADE, trong việc giảm số lượng ứng viên thừa phát sinh, trong quá trình khai thác mẫu tuần tự.

Dữ liệu được lưu trữ trong cấu trúc CMAP là ánh xạ mỗi sự kiện k thuộc *I* vào tập các sự kiện mà các sự kiện ấy có thể kết hoặc mở rộng với sự kiện k để tạo ra mẫu phổ biến. Ví dụ như bảng 2.1 mô tả cấu trúc CMAP cho CSDL trong hình 2.2 với minsup=2. Các sự kiện A,B,D thuộc tập mở rộng phổ biến, trong cấu trúc CMAP, của sự kiện A thì các mẫu AA, AB, AD là các mẫu phổ biến.

Sự kiện	Tập mở rộng phổ biến
A	{A, B, D}
B	{A, B, D}
C	
D	{A, B}

Bảng 2.1 : Cấu trúc CMAP với minsup = 2

Tác giả Philippe Fournier-Viger đã đưa ra tính chất làm cơ sở cho cấu trúc CMAP như sau:

Tính chất 1: Cho một mẫu phổ biến A và sự kiện k. Nếu tồn tại sự kiện j thuộc A và k không thuộc tập mở rộng phổ biến của j trong bảng CMAP thì mẫu ứng viên được tạo bằng cách mở rộng mẫu A với sự kiện k sẽ là mẫu không phổ biến. Chứng minh : Nếu k không thuộc tập ánh xạ của j trong CMAP thì việc mở rộng j với sự kiện k sẽ tạo ra mẫu không phổ biến. Do đó, nếu mẫu A, có chứa sự kiện j, được mở rộng với sự kiện k cũng sẽ tạo ra mẫu không phổ biến [7].

Với tính chất trên, việc kết hợp cấu trúc CMAP vào thuật toán SPADE sẽ làm giảm số lượng ứng viên dư thừa của thuật toán SPADE ban đầu.

2.3.2 Thuật toán CM-SPADE

Thuật toán CM-SPADE là sự kết hợp cấu trúc CMAP và thuật toán SPADE. Trong thuật toán CM-SPADE, thủ tục ENUMERATE của thuật toán SPADE được thay đổi như sau : Xét mẫu r được tạo bằng cách kết 2 mẫu A_i=P $\cup$ x và A_j =P $\cup$ y, với P là tiền tố của A_i và A_j. Giả sử y được thêm vào A_i để mở rộng thành r. Nếu sự kiện y không thuộc tập mở rộng phổ biến của sự kiện x trong cấu trúc CMAP, thì mẫu r là mẫu không phổ biến và ta có thể loại bỏ mẫu r mà không cần thực hiện thao tác kết. Chiến lượt này đúng dựa trên tính chất 1. Ví dụ như trong hình 2.5 chúng ta cần mở rộng sự kiện D trong tập 1-sequence. Dựa vào tập mở rộng phổ biến của sự kiện D trong bảng 2.1 chúng ta thấy chỉ có 2 sự kiện là A và B nên chúng ta chỉ mở rộng thành mẫu DA và DB, tránh việc tạo thành mẫu DD do ta biết trước mẫu này không phổ biến.

2.3.3 Ưu điểm và khuyết điểm

Ưu điểm của thuật toán CM-SPADE là tận dụng được những lợi thế của thuật toán SPADE gốc. Việc bổ sung thêm bước kiểm tra, dựa trên cấu trúc CMAP, cho phép loại bỏ nhanh các ứng viên không phổ biến và tránh việc tốn chi phí cho thao tác kết chuỗi dư thừa. Từ đó, tăng nhanh tốc độ xữ lý cũng như giảm bộ nhớ hao phí khi chạy thuật toán.

Khuyết điểm của thuật toán CM-SPADE là sử dụng chiến lượt phát sinh và kiểm thử của thuật toán SPADE. Với chiến lượt này rất nhiều mẫu ứng viên dư thừa được phát sinh dẫn đến việc tốn rất nhiều tài nguyên cho thao tác lưu trữ và tính toán. Mặc dù, với việc bổ sung cấu trúc CMAP cho phép loại bỏ đi rất nhiều ứng viên nhưng số lượng ứng viên dư thừa được tạo ra vẫn rất lớn. Trong chương 3, đồ án chỉ ra một số trường hợp hạn chế của thuật toán CM-SPADE để từ đó cải tiến thuật toán này nhằm loại bỏ các mặt hạn chế còn lại của thuật toán CM-SPADE.

CHƯƠNG 3: KỸ THUẬT CẮT NHÁNH DỰA TRÊN BẢNG TRẠNG THÁI

3.1 Ý tưởng

Chiến lượt khai thác mẫu tuần tự của thuật toán SPADE và CM-SPADE là phát sinh mẫu ứng viên và kiểm tra độ hỗ trợ của mẫu ứng viên so với ngưỡng hỗ trợ nhỏ nhất. Nếu CSDL cần xét có kích thước lớn thì số lượng ứng viên phát sinh là rất lớn. Trong số các ứng viên phát sinh, có những ứng viên không tồn tại trong CSDL hoặc có độ hỗ trợ nhỏ hơn ngưỡng hỗ trợ nhỏ nhất. Việc tốn không gian lưu trữ và chi phí tính toán cho các mẫu ứng viên dư thừa như vậy là hạn chế chung của phương pháp SPADE và CM-SPADE.

Nội dung chính của đồ án là khắc phục nhược điểm của 2 thuật toán SPADE và CM-SPADE bằng cách ngăn chặn việc phát sinh các ứng viên dư thừa trong quá trình phát sinh mẫu dựa vào thông tin bảng trạng thái.

Bảng trạng thái là cấu trúc lưu trữ thông tin của mẫu tuần tự giống như cấu trúc lưu trữ sự kiện của SPADE, bao gồm danh sách các <sid,pos(s)> của mẫu tuần tự trong CSDL. Điểm khác biệt duy nhất của bảng trạng thái là với mỗi định danh sid, bảng trạng thái chỉ lưu duy nhất một vị trí xuất hiện mẫu chứ không lưu danh sách nhiều vị trí như trong cấu trúc lưu trữ thông tin của SPADE.

A(3)	
SID	Itemset
1	1,4
2	1,4
3	
4	2

Cấu trúc lưu trữ của SPADE

A(3)	
SID	Itemset
1	1
2	1
3	
4	2

Bảng trạng thái

Hình 3.1 : Sự khác nhau giữa bảng trạng thái và cấu trúc lưu trữ của SPADE là bảng trạng thái chỉ lưu duy nhất 1 vị trí/tập sự kiện ứng với mỗi sid.

Cấu trúc bảng trạng thái gồm tên sự kiện hoặc tên mẫu tuần tự mà bảng trạng thái mô tả và danh sách <sid,pos(s)> của mẫu tuần tự đó. Trong đó, vị trí của mẫu

được thể hiện trong bảng trạng thái là vị trí xuất hiện của sự kiện cuối cùng trong mẫu. Ví dụ như mẫu AD có thông tin lưu trữ của sự kiện A tại sid=1 là <1,1> và <1,4>, thông tin lưu trữ của sự kiện D tại sid=1 là <1,5>. Khi đó thông tin lưu trữ trong bảng trạng thái của mẫu AD tại sid=1 sẽ là vị trí của sự kiện sau cùng, sự kiện D, là <1,5>.

Dựa vào thông tin bảng trạng thái và thông tin lưu trữ sự kiện của cấu trúc SPADE, tác giả đồ án rút ra các nhận xét sau :

Nhận xét 1 : Một mẫu ứng viên có thể được kiểm tra là phổ biến hoặc không phổ biến dựa vào thông tin của bảng trạng thái và thông tin lưu trữ sự kiện của cấu trúc SPADE mà không cần thực hiện thao tác kết. Ví dụ, với độ hỗ trợ bằng 2, ta có thông tin của bảng trạng thái của sự kiện A là {<1,2>,<2,1>,<4,2>}. Ta lấy ra thông tin bảng trạng thái của sự kiện B, sao cho mỗi vị trí lấy ra là vị trí lớn nhất ứng với mỗi định danh chuỗi sid, là {<1,6>,<2,5>,<4,3>}. Do thông tin vị trí trong bảng trạng thái của B ở mọi định danh sid đều lớn hơn vị trí trong bảng trạng thái của A nên ta kết luận mẫu AB là phổ biến.

Nhận xét 2 : Chi phí hao tốn cho thao tác kiểm tra độ phổ biến của mẫu sẽ thấp hơn chi phí kết chuỗi. Với mỗi một thao tác kết 2 sự kiện x và y, tại định danh sid, ta phải duyệt tất cả các vị trí trong mảng vị trí của sự kiện y để tìm ra vị trí nhỏ nhất trong mảng vị trí của y và vị trí ấy lớn hơn vị trí nhỏ nhất trong mảng vị trí của sự kiện x. Ví dụ ta có x=<sid,{3,4}> và y=<sid,{1,2,5,6}>, vị trí nhỏ nhất trong mảng vị trí của x là <sid,3>, vị trí nhỏ nhất trong mảng vị trí của y có thể kết được là <sid,5>. Kết quả thao tác kết là xy=<sid,{5,6}>. Trong khi đó, thao tác kiểm tra dựa trên bảng trạng thái chỉ lấy ra vị trí lớn nhất trong danh sách các vị trí ở cùng sid. Nếu mảng vị trí đã được sắp xếp tăng dần thì vị trí lớn nhất chính là vị trí cuối cùng của mảng. Khi đó, thao tác truy xuất sẽ nhanh hơn so với duyệt từng phần tử. Bên cạnh đó, khi chúng ta xây dựng CSDL dạng dọc, chúng ta có thể sắp xếp mảng các vị trí của chuỗi tăng dần dựa trên quá trình truy xuất CSDL ban đầu.

Từ nhận xét 1 và 2, các mẫu dư thừa trong quá trình khai thác sẽ được phát hiện dựa trên bảng trạng thái mà không cần đến thao tác kết chuỗi.

Trong thuật toán SPADE, các mẫu tuần tự được khai thác bằng thủ tục ENUMERATE. Tham số của thủ tục ENUMERATE là lớp tương đương F. Trong lớp tương đương F là các mẫu tuần tự có cùng tiền tố với nhau. Do đó, việc phát sinh mẫu trong thuật toán là việc mở rộng những chuỗi có cùng k-1 tiền tố và mở rộng ở sự kiện thứ k. Thao tác này đòi hỏi phải lưu trữ tòan bộ các mẫu trong cùng một lớp tương đương,sau đó mới có thể mở rộng từng mẫu trong lớp tương đương đó. Từ đó dẫn đến việc tốn nhiều vùng nhớ để lưu trữ mẫu trong quá trình khai thác.

Đồ án đề xuất một góc nhìn khác của việc khai thác mẫu tuần tự mà không cần lưu trữ thông tin của tất cả các mẫu trong cùng lớp tương đương. Đó là phương pháp khai thác dựa trên thông tin trạng thái và thông tin lưu trữ sự kiện của thuật toán SPADE.

Kết quả đạt được trong bước đầu tiên của thuật toán SPADE là CSDL định dạng dọc và tập các mẫu phổ biến độ dài 1, hay còn gọi là mẫu 1-sequence phổ biến. Sau bước phát sinh cấu trúc CMAP của thuật toán CM-SPADE, kết quả thu được là tập các sự kiện và tập mở rộng phổ biến của từng sự kiện đó. Từ tính chất 1 đồ án đưa ra nhận xét.

Nhận xét 3: Tất cả các mẫu tuần tự khởi đầu bằng sự kiện x đều là các mẫu được mở rộng từ những sự kiện chứa trong tập mở rộng phổ biến của sự kiện x, lưu trong cấu trúc CMAP.

Từ nhận xét 3, việc khai tác mẫu tuần tự tương đương với việc giảm số lượng sự kiện trong tập mở rộng phổ biến của sự kiện x ứng với mỗi mẫu được mở rộng từ x. Ví dụ : Trong cấu trúc CMAP trong bảng 2.1, sự kiện A có tập mở rộng phổ biến là {A,B,D}. Với minsup=2, tập mẫu tuần tự có độ dài 2 và khởi đầu bằng sự kiện A bao gồm là {AA,AB,AD}. Dựa trên thuật toán CM-SPADE đã cho, tập các sự kiện

có thể mở rộng của mẫu AA là {A,B,D}. Với mẫu AB, tập các sự kiện có thể mở rộng vẫn là {A,B,D}. Với tập AD, tập các sự kiện có thể mở rộng là tập {A,B}. Các tập sự kiện có thể mở rộng của các mẫu AA,AB,AD đều là tập con của tập sự kiện có thể mở rộng của mẫu A. Dựa trên thông tin lưu tữ của bảng trạng thái ứng với từng mẫu và thông tin lưu trữ của các sự kiện trong CSDL dọc, ta sẽ loại bỏ dần các sự kiện trong tập mở rộng của các mẫu AA,AB,AD để sao cho khi mở rộng các mẫu AA,AB,AD với các sự kiện còn lại thì sẽ tạo ra mẫu tuần tự phổ biến.

3.2 Thuật toán

Thuật toán khai thác mẫu tuần tự dựa trên thông tin của bảng trạng thái nhận tham số đầu vào là CSDL tuần tự và ngưỡng hỗ trợ nhỏ nhất. Đầu tiên thuật toán chuyển CSDL dạng ngang sang CSDL dạng dọc V(SDB) và xác định tập mẫu tuần tự phổ biến có chiều dài 1, hay còn gọi là 1-sequence phổ biến, là tập F_1.

STATE-SPADE(SDB,minsup)

1. Scan SDB to create V(SDB) and identify F_1 the list of frequent items
2. Init CMAP
3. **FOR EACH** pattern A **IN** F_1
4. Output A
5. Init state table of A
6. **ENUMERATE**(A,StateTable(A),CMAP(A))

Hình 3.2 : Thuật toán SPADE mở rộng

Sau đó thuật toán khởi tạo cấu trúc CMAP cho các sự kiện trong tập F_1. Với mỗi mẫu tuần tự A trong tập F1, thuật toán xuất ra mẫu A. Sau đó tiến hành khởi tạo bảng trạng thái cho mẫu A. Thao tác khởi tạo này bao gồm việc sao chép thông tin của mẫu A trong CSDL dạng dọc và giữ lại vị trí nhỏ nhất trong mỗi định danh chuỗi. Sau cùng là gọi thủ tục ENUMERATE để tìm ra tất cả các mẫu tuần tự khởi đầu bằng mẫu A.

Thủ tục ENUMERATE nhận tham số đầu vào là tên của bảng trạng thái, hay mẫu tuần tự mà bảng trạng thái mô tả. Tham số thứ 2 là bảng trạng thái của mẫu tuần tự ở tham số 1. Tham số thứ 3 là danh sách các sự kiện mà mẫu tuần tự có thể mở rộng thành công. Với tham số ban đầu là mẫu tuần tự trong F1, nên mẫu tuần tự đó cũng là sự kiện phổ biến trong CSDL hay mẫu 1-sequence phổ biến. Vì thế, thông tin về tập mở rộng thành công của tham số ban đầu này được lưu trong cấu trúc CMAP. Ví dụ : Với CSDL như hình 2.2 và minsup=2, mẫu đang xét trong F1 là A, khi đó tham số của thủ tuc ENUMERATE bao gồm : Ký tự A, bảng trạng thái của A là {<1,1>,<2,1>,<4,2>} và tập mở rộng thành công của sự kiện A lưu trong bảng 2.1 là {A,B,D}.

ENUMERATE(X , StateTable(X), ListItem(X))

1. **FOR EACH** pattern B **IN** ListItem(X)
2. Output XB
3. **UPDATE_STATE**(StateTable(X) , info(B))
4. **UPDATE_ITEMSET**(B , StateTable(XB) ,ListItem(X)) => ListItem(XB)
5. **ENUMERATE**(XB,StateTable(XB),ListItem(XB))

Hình 3.3 : Thủ tục ENUMERATE

Đầu tiên, thủ tục ENUMERATE duyệt tất cả các sự kiện có trong mảng sự kiện ở tham số thứ 3. Ký hiệu ListItem(X) có ý nghĩa rằng các sự kiện trong danh sách này đều có thể mở rộng thành công với mẫu tuần tự X, hay khi ta mở rộng mẫu tuần tự X với từng phần tử trong ListItem(X) đều sẽ tạo ra mẫu tuần tự phổ biến. Với mỗi sự kiện B trong mảng ListItem(X), thuật toán xuất ra mẫu X mở rộng với sự kiện B, là XB. Bước kế tiếp là gọi thủ tục UPDATE_STATE để cập nhật bảng trạng thái của mẫu X thành bảng trạng thái của mẫu mới XB dựa trên thông tin trạng thái của mẫu X lưu trong tham số thứ 2 và thông tin của sự kiện B lưu trong CSDL định dạng dọc. Bước kế tiếp là gọi thủ tục UPDATE_ITEMSET để cập nhật lại mảng các sự kiện có thể mở rộng của mẫu tuần tự X thành mảng sự kiện có thể

mở rộng của mẫu mới XB. Sau cùng là gọi đệ quy trên mẫu mới XB với các tham số tương ứng của XB là bảng trạng thái của XB và mảng các sự kiện mở rộng thành công của XB.

Thủ tục UPDATE_STATE nhận tham số đầu vào là bảng trạng thái của mẫu tuần tự A và thông tin của sự kiện B lưu trong CSDL định dạng dọc. Bảng chất của thủ tục này là thao tác kết thông tin của mẫu A và thông tin của sự kiện B. Vì thông tin trong bảng trạng thái cũng là thông tin lưu trong CSDL định dạng dọc nhưng chỉ lưu có duy nhất 1 vị trí ứng với mỗi định danh chuỗi (sid).

```
UPDATE_STATE(StateTable(A), info(B) )
1. FOR EACH SID IN StateTable(A) which is in info(B)
2.       posA=pos( StateTable(A) , SID )
3.       FOR EACH posB IN pos_list(info(B) , SID)
4.              IF  posB > posA THEN
5.                     Set( StateTable(AB) , SID , posB)
6. RETURN StateTable(AB)
```

Hình 3.4 : Thủ tục UPDATE_STATE

Thủ tục sẽ duyệt từng định danh chuỗi SID vừa có trong bảng trạng thái của mẫu tuần tự A vừa có trong cấu trúc lưu thông tin của sự kiện B. Sau đó lấy ra thông tin vị trí duy nhất tại SID trong bảng trạng thái của mẫu A rồi so sánh vị trí đó trong bảng trạng thái với từng vị trí trong danh sách vị trí tương ứng của sự kiện B để tìm ra vị trí mới, khởi tạo cho bảng trạng thái của mẫu AB. Vị trí mới phải thuộc mảng vị trí của sự kiện B và có giá trị lớn hơn vị trí tương ứng lưu trong bảng trạng thái của mẫu tuần tự A. Ký hiệu pos_list(info(B) , SID) chỉ danh sách vị trí đã sắp xếp tăng dần trong cấu trúc lưu trữ thông tin của sự kiện B tại định danh chuỗi là SID. Kết quả của thủ tục là bảng trạng thái của mẫu mới AB.

Thủ tục UPDATE_ITEMSET được dùng để loại bỏ các sự kiện dư thừa trong mảng mở rộng. Đây cũng là phương pháp cắt nhánh được đề xuất trong đồ án

này. Thủ tục nhận tham số là bảng trạng thái của mẫu XB, sự kiện cuối cùng trong mẫu XB là sự kiện B và mảng các sự kiện ArrItem được dùng để mở rộng. Mục đích chính của thủ tục là loại bỏ những sự kiện trong mảng sự kiện ArrayItem mà việc mở rộng mẫu XB với sự kiện ấy sẽ tạo ra mẫu không phổ biến.

```
UPDATE_ITEMSET( B, StateTable(XB) , ArrItem )
1.  FOR EACH item IN ArrayItem
2.      IF item NOT-PASS CMAP-step THEN
3.          REMOVE item FROM ArrItem
4.          CONTINUE
5.      TMP=0;
6.      FOR EACH SID IN StateTable(XB) which is in info(item)
7.          posXB=pos( StateTable(XB) , SID )
8.          IF posXB < MAX(pos_list(info(item) , SID)) THEN
9.              TMP=TMP+1
10.     IF TMP<minsup THEN
11.         REMOVE item FROM ArrItem
12. RETURN ArrItem
```

Hình 3.5 : Thủ tục UPDATE_ITEMSET

Thủ tục UPDATE_ITEMSET thực hiện 2 bước kiểm tra. Bước 1 là kiểm tra dựa trên cấu trúc CMAP được đề xuất bởi Philippe Fournier-Viger, bằng cách kiểm sự kiện *item* có thuộc tập mở rộng phổ biến của sự kiện cuối cùng trong mẫu XB là sự kiện B hay không. Bước 2 là kiểm tra dựa trên bảng trạng thái dùng cho những sự kiện đã vượt qua bước 1. Ở bước kiểm tra thứ 2, thủ tục sẽ lấy ra vị trí lớn nhất trong mảng vị trí của sự kiện *item* lưu trong CSDL định dạng dọc và so sánh với vị trí tương ứng SID trong bảng trạng thái của mẫu XB. Nếu số lượng tìm được, vị trí của sự kiện *item* lớn hơn vị trí tương ứng SID của mẫu XB, nhỏ hơn minsup thì ta sẽ loại bỏ sự kiện *item* khỏi mảng sự kiện ArrayItem. Kết quả của thủ tục

UPDATE_ITEMSET là mảng ArrayItem gồm các sự kiện mà khi mở rộng với mẫu XB sẽ tạo ra mẫu phổ biến.

3.3 Ưu điểm và khuyết điểm

Ưu điểm của phương pháp này là cho phép phát hiện và loại bỏ các mẫu ứng viên không phổ biến trước khi chúng được phát sinh. Bằng việc kiểm tra thông tin trạng thái cho phép xác định được những mẫu tuần tự không phổ biến, từ đó tránh thao tác phát sinh mẫu dư thừa. Bên cạnh đó, việc chỉ lưu trữ thông tin trạng thái của các mẫu phổ biến, thuật toán đã tiết kiệm rất nhiều không gian vùng nhớ từ việc không tốn vùng nhớ lưu trữ những mẫu ứng viên không phổ biến. Và với việc chi phí của thao tác kiểm tra thấp hơn so với thao tác kết, cho phép thuật toán chạy nhanh hơn thuật toán gốc trên những bộ dữ liệu mà số lượng mẫu dư thừa được phát sinh lớn.

Khuyết điểm của thuật toán là với những mẫu tuần tự phổ biến thuật toán phải sử dụng cả 2 thao tác là kiểm tra và thao tác kết chuỗi. Với những mẫu tuần tự có nhiều sự kiện nối tiếp nhau sẽ đòi hỏi tốn nhiều chi phí để phát sinh hơn thuật toán gốc. Từ đó, đặt ra yêu cầu về việc xác định tính chất dữ liệu để có thể sử dụng thuật toán một cách hiệu quả.

CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

4.1 Tổng quan

Nội dung chương 4 trình bày kết quả thực nghiệm của thuật toán mà đồ án đề xuất, đồng thời so sánh kết quả với thuật toán SPADE của Zaki và thuật toán CM-SPADE của Philippe Fournier-Viger về mặt thời gian chạy và bộ nhớ sử dụng. Tất cả mã lệnh được viết bằng ngôn ngữ Java, chạy trên hệ điều hành Fedora 24, trên bộ vi xử lý 3.2GHz Intel Core I3, RAM 4GB. Phiên bản hệ điều hành Fedora 24 có thể tìm thấy tại địa chỉ web *https://getfedora.org*

4.2 Tập dữ liệu

Đồ án sử dụng 3 bộ dữ liệu thực tế để kiểm nghiệm kết quả, đó là: Leviathan, BMSWebView1 and Kosarak10k. 3 bộ dữ liệu này được đề xuất trong bài báo của Philippe Fournier-Viger khi trình bày về thuật toán CM-SPADE[7], và có thể tải về tại địa chỉ web http://www.philippe-fournier-viger.com/spmf . Bảng 4.1 mô tả đặc điểm của 3 bộ dữ liệu thực nghiệm.

Dataset	Sequence count	Distinct item	Avg. seq. length	type of data
Leviathan	5834	9025	33.81	book
BMS	59601	497	2.51	web click stream
Kosarak10k	10000	10094	8.14	web click stream

Bảng 4.1 : Đặc điểm của các bộ dữ liệu

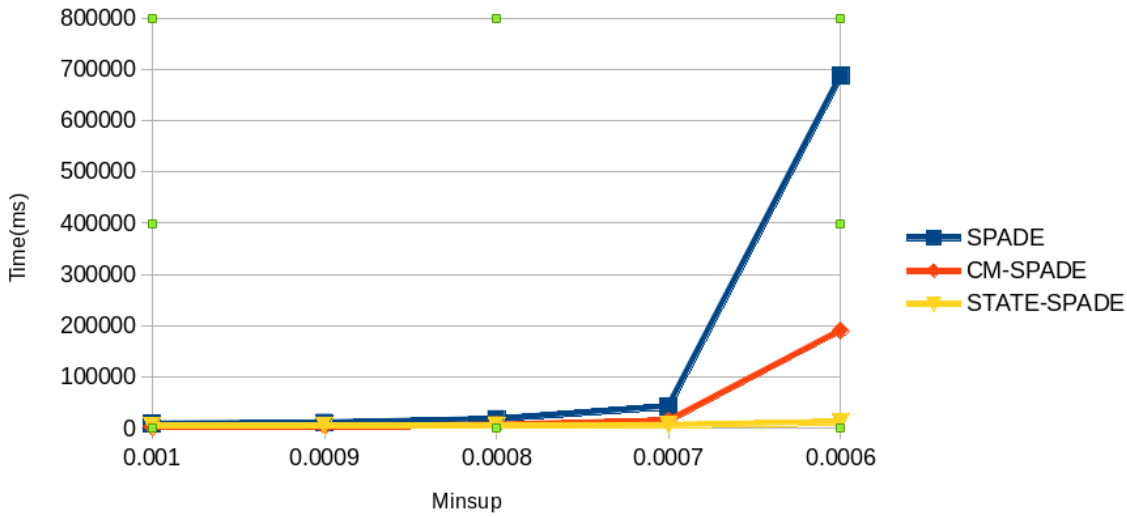
Quá trình thực nghiệm được thực hiện bằng cách chạy lần lượt từng thuật toán trong 3 thuật toán SPADE, CM-SPADE và thuật toán đề xuất trên từng bộ dữ liệu với ngưỡng hỗ trợ giảm dần cho đến khi có sự khác biệt rõ ràng giữa các thuật toán hoặc thời gian chạy quá lâu đến mức không thể thực hiện được nữa hoặc đến khi tràn bộ nhớ. Với mỗi lần thực nghiệm trên từng bộ dữ liệu, kết quả thu được bao gồm thời gian chạy chương trình, số mẫu tuần tự khai thác được, lượng bộ nhớ tiêu

hao. Sau đó sẽ so sánh kết quả của các thuật toán với nhau về mặt thời gian chạy và bộ nhớ sử dụng.

4.3 So sánh về thời gian chạy

Thời gian chạy của 3 thuật toán SPADE, CM-SPADE và thuật toán đề xuất được thể hiện bằng đồ thị trong hình 4.1, hình 4.2 và hình 4.3.

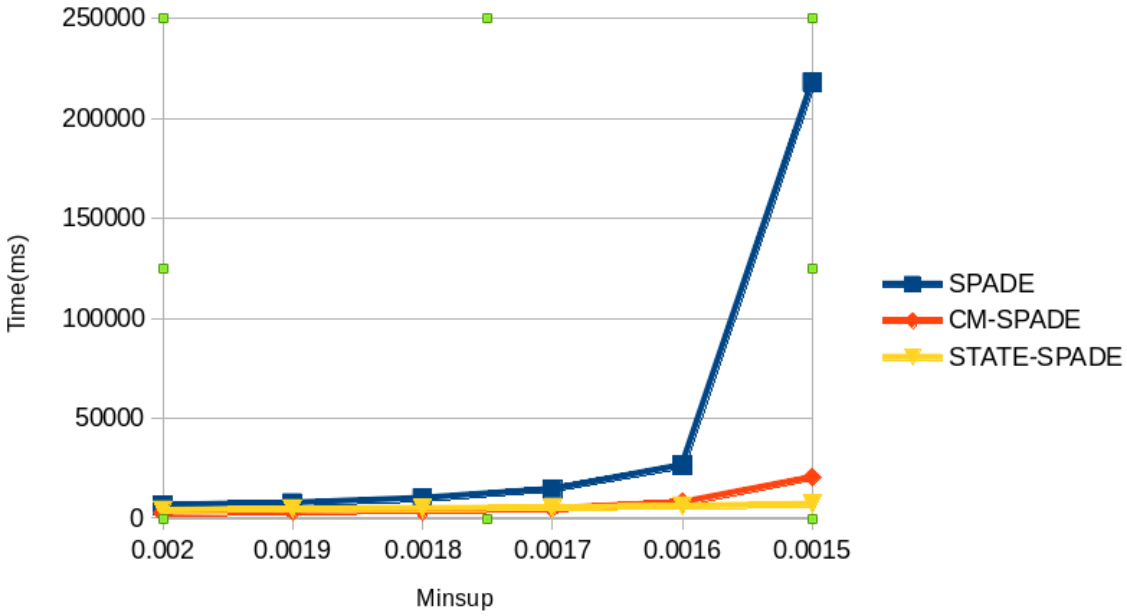
Từ 3 đồ thị trong hình 4.1, hình 4.2 và hình 4.3, thấy được rằng: Thời gian chạy của thuật toán đề nhỏ hơn thời gian chạy của thuật toán SPADE và CM-SPADE khi ta hạ minsup xuống rất nhỏ. Bởi vì thuật toán được đề xuất để loại bỏ số lượng mẫu ứng viên dư thừa, nên khi hạ minsup xuống thì số lượng ứng viên dư thừa tăng, khi đó thuật toán đề xuất mới thể hiện rõ hiệu quả.



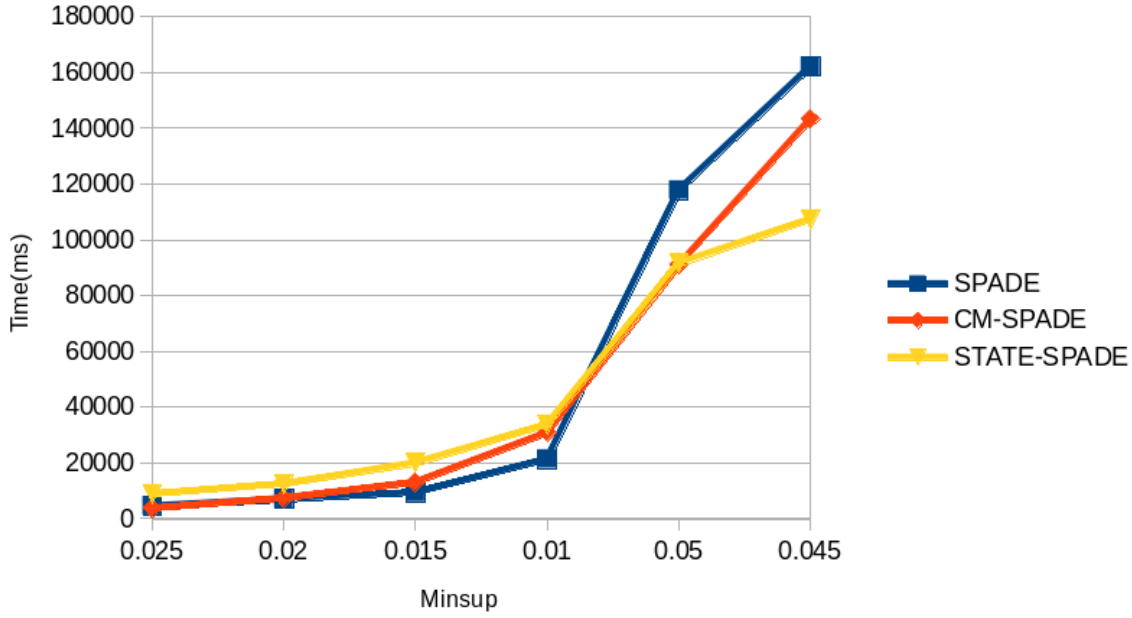
Hình 4.1 : Thời gian thực thi thuật toán trên bộ dữ liệu BMSWebView1

Với ngưỡng minsup cao thì số lượng mẫu dư thừa ít, nên chi phí thực hiện các thao tác của thuật toán đề xuất là cao hơn so với thuật toán SPADE và CM-SPADE, nên thời gian thực thi của thuật toán đề xuất không hiệu quả bằng 2 thuật toán SPADE và CM-SPADE.

Với ngưỡng minsup thấp thì số lượng mẫu dư thừa nhiều, khi đó chi phí của thuật toán đề xuất thấp hơn so với chi phí xử lý của thuật toán SPADE và CM-SPADE, nên thời gian thực thi của thuật toán đề xuất sẽ hiệu quả hơn.



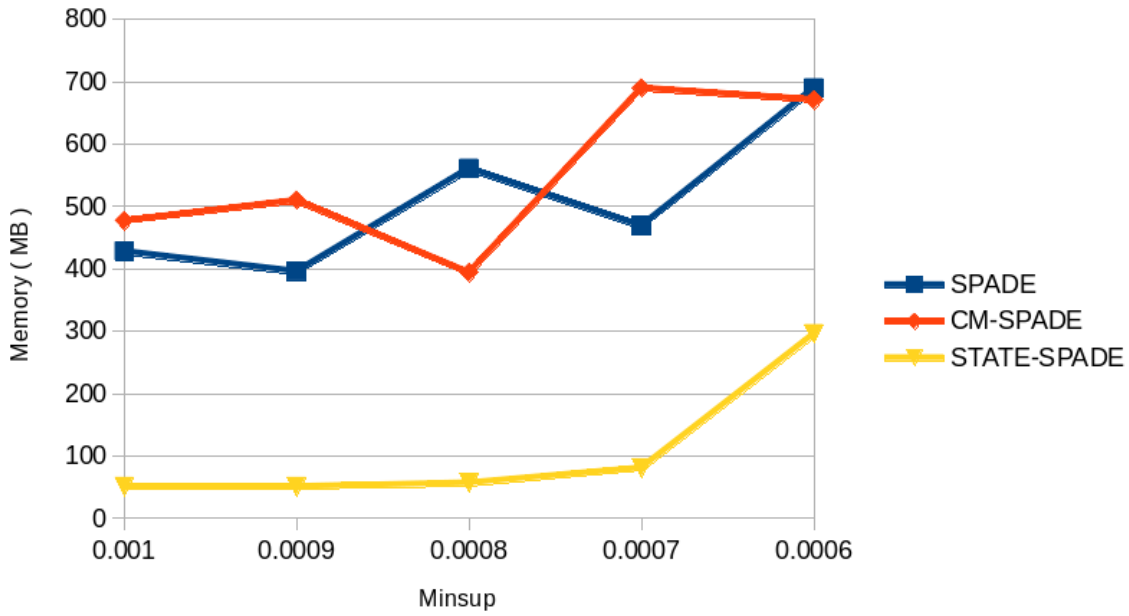
Hình 4.2 : Thời gian thực thi thuật toán trên bộ dữ liệu Kosarak10k



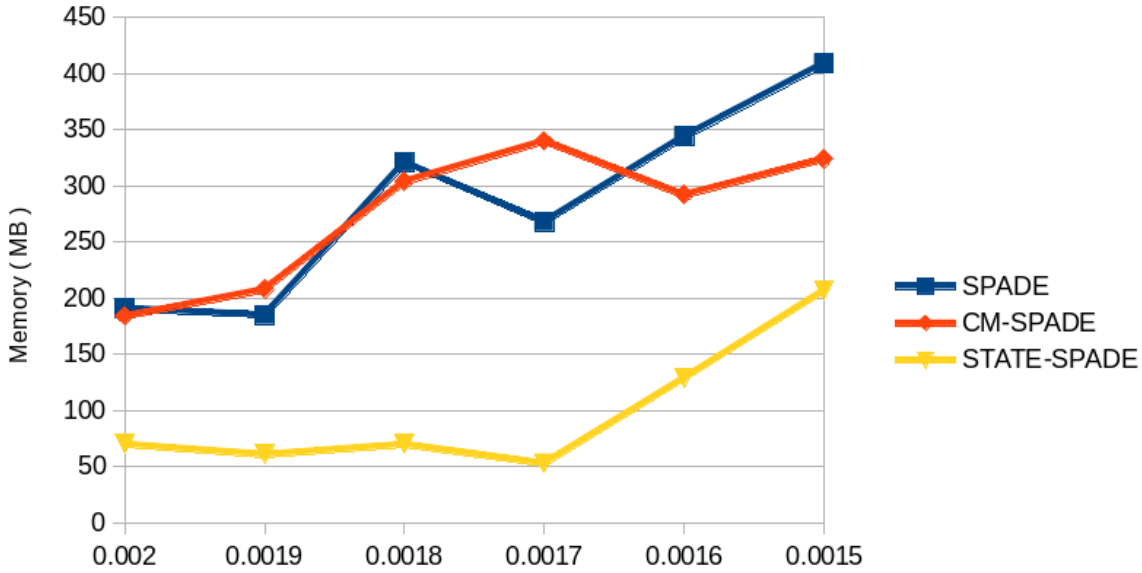
Hình 4.3: Thời gian thực thi thuật toán trên bộ dữ liệu Leviathan

4.4 So sánh về bộ nhớ sử dụng

Kết quả so sánh bộ nhớ tiêu hao của 3 thuật toán được thể hiện trong đồ thị hình 4.4, hình 4.5, hình 4.6.

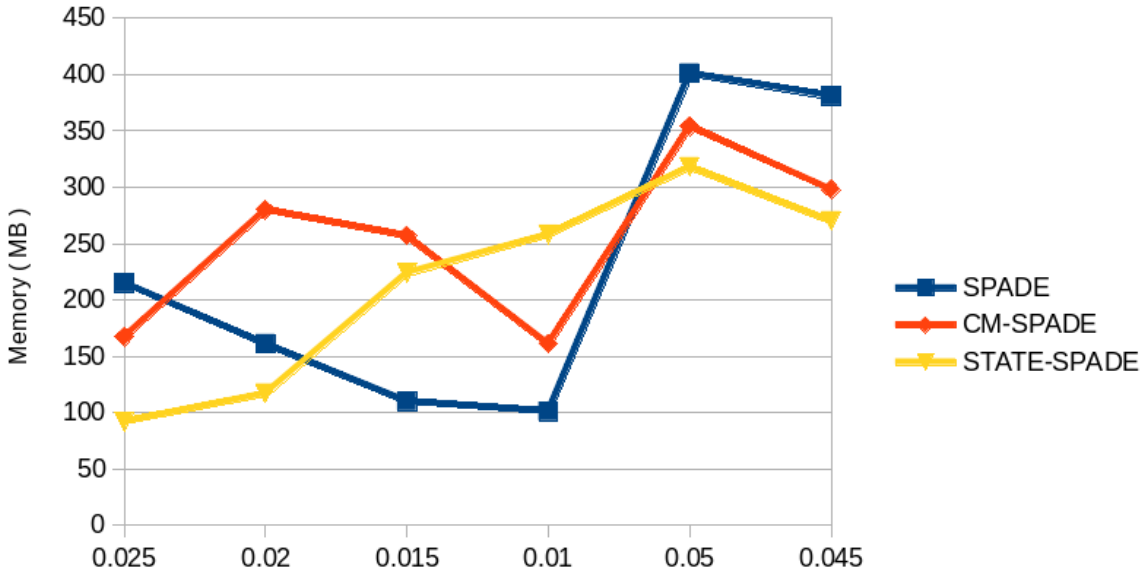


Hình 4.4 : Bộ nhớ sử dụng khi chạy trên bộ dữ liệu BMSWebView1



Hình 4.5 : Bộ nhớ sử dụng khi chạy trên bộ dữ liệu Kosarak10k

Trong đồ thị hình 4.4 và 4.5 ta thấy rằng: thuật toán đề xuất sử dụng ít vùng nhớ hơn trong quá trình khai thác mẫu tuần tự trên tập dữ liệu BMSWebView1 và Kosarak10k. Nguyên nhân của việc sử dụng hiệu quả bộ nhớ bởi vì cả 2 bộ dữ liệu BMSWebView1 và Kosarak10k là dữ liệu thưa nên khi khai thác, số lượng ứng viên thừa phát sinh nhiều hơn rất nhiều, so với số thao tác cần xử lý của thuật toán đề xuất, nên bộ nhớ sử dụng của thuật toán đề xuất ít hơn nhiều so với thuật toán SPADE và CM-SPADE.



Hình 4.6 : Bộ nhớ sử dụng khi chạy trên bộ dữ liệu Leviathan

Trong khi đó, với bộ dữ liệu Leviathan trong hình 4.6, lượng bộ nhớ sử dụng của thuật toán đề xuất không chênh lệch rõ ràng so với 2 thuật toán SPADE và CM-SPADE. Nguyên nhân của việc này là vì dữ liệu chứa bên trong bộ dữ liệu Leviathan phân bố dày đặt hơn so với bộ dữ liệu BMSWebView1 và Kosarak10k, do đó với mỗi mẫu tuần tự phổ biến được phát sinh trong thuật toán đề xuất sẽ tốn nhiều chi phí hơn, do thuật toán đề xuất sử dụng đến 2 thao tác kiểm tra và kết chuỗi.

4.5 Kết luận

Từ kiết qủa thực nghiệm cho thấy rằng, thuật toán đề xuất cho kết quả tốt hơn 2 thuật toán SPADE của Zaki và CM-SPADE của Philippe Fournier-Viger trên 3 bộ dữ liệu Leviathan, BMSWebView1, Kosarak10k với ngưỡng minsup thấp. Bên cạnh đó, phân bố dữ liệu của 2 bộ dữ liệu BMSWebView1 và Kosarak10k là thưa hơn so với bộ dữ liệu Leviathan, đồng thời kết quả thực nghiệm trên 2 bộ BMSWebView1 và Kosarak10k cũng dễ nhận thấy hơn so với kết quả trên bộ dữ liệu Leviathan. Từ đó có thể kết luận rằng phân bố dữ liệu cũng ảnh hưởng đến hiệu quả của thuật toán đề xuất.

CHƯƠNG 5 : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Đồ án đã trình bày được những vấn đề như sau :

- Trình bày khái quát về khái niệm dữ liệu chuỗi cũng như những đặc điểm đặc trưng của dữ liệu chuỗi; các ví dụ về dữ liệu chuỗi bao gồm chuỗi văn bản, chuỗi sự kiện và chuỗi sinh học; trình bày về 2 hướng khai thác dữ liệu chuỗi phổ biến hiện nay là khai thác mẫu tuần tự và khai thác luật tuần tự.
- Trình bày chi tiết nội dung của phương pháp khai thác chuỗi tuần tự phổ biến SPADE do Zaki đề xuất năm 2001. Đồ án chỉ ra những ưu điểm và khuyết điểm của phương pháp này, đồng thời trình bày về phương pháp cải tiến của Philippe Fournier-Viger năm 2014, cho phép loại bớt số ứng viên phát sinh trong thuật toán SPADE.
- Trình bày ý tưởng và nội dung chi tiết của phương pháp khai thác dựa trên bảng thông tin trạng thái bao gồm khái niệm của bảng trạng thái, đặc điểm của mẫu tuần tự mà đồ án khai thác và bổ sung thêm bước loại bỏ ứng viên dư thừa dựa trên thông tin trạng thái và thông tin lưu trữ trong CSDL dọc.
- Trình bày kết quả thực nghiệm của phương pháp đề xuất so với 2 phương pháp SPADE và CM-SPADE. Từ kết quả cho thấy thuật toán đề xuất cho kết quả tốt hơn so với 2 thuật toán ban đầu trên 3 tập dữ liệu thực nghiệm.

5.2 Hướng phát triển

Bên cạnh những ưu điểm thu được, phương pháp đề xuất của đồ án còn rất nhiều hạn chế có thể cải tiến trong tương lai. Những hướng cải tiến bao gồm :

- Phương pháp nói đến trong đồ án chỉ đề xuất để giải bài toán với số lượng sự kiện trong một tập sự kiện trong chuỗi là hạn chế. Ví dụ như số sự kiện xảy

ra đồng thời bằng 1 hay trong một tập sự kiện chỉ có 1 sư kiện. Từ đó có thể thay đổi một số bước để phương pháp này có thể áp dụng cho dạng bài toán mà số sự kiện diễn ra đồng thời nhiều hơn hay trong 1 tập sự kiện trong chuỗi có nhiều hơn 1 sự kiện.

- Ta có thể thay đổi một số bước của thuật toán để có thể giải quyết được bài toán khai thác mẫu tuần tự đóng, dựa trên việc kiểm tra độ hỗ trợ của những chuỗi có khởi đầu bằng sự kiện x. Sau đó kết hợp với thuật toán ClaSP để khai thác mẫu tuần tự đóng.
- Có thể cải tiến phương pháp kiểm tra mẫu có phải phổ biến hay không dựa trên thông tin đã có và khác với phương pháp mà đồ án đề xuất là phương pháp so sánh từng cặp vị trí ứng với mỗi định danh chuỗi.
- Từ việc kỹ thuật xử lý song song đang ngày càng phát triển, ta có thể áp dụng kỹ thuật song song vào thuật toán để tăng nhanh hiệu quả xữ lý dữ liệu.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Agrawal, R. and Srikant, R., "Mining Sequential Patterns," in Proc. Int. Conf.on Data Engineering, Taipei, Taiwan, March 6-10, 1995, pp. 3-14.

[2] Pei, J. et al., "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach," IEEE Trans. Knowledge and Data Engineering, vol. 16, no. 10, pp.1-17, 2001

[3] Zaki, M. J., "SPADE: An Efficient Algorithm for Mining Frequent," Machine Learning, vol. 40, pp. 31-60, 2001.

[4] Agrawal, R. and Srikant, R., "Mining Sequential Patterns: Generalizations and Performance Improvements," in Proceeding EDBT '96 Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology, London, UK, 1996, pp. 3-17.

[5] Ayres, J., Flannick, J., Gehrke, J., and Yiu, T., "Sequential Pattern Mining Using a Bitmap Representation," in Proc. 8th ACM SIGKDD Int. Conf. On Knowledge Discovery and Data Mining (KDD 2002), Edmonton, Alberta, July 23-26, 2002, pp. 429-435.

[6] Gomariz, A., Campos, M., Marin, R., Goethals, B.: ClaSP: An Efficient Algorithm for Mining Frequent Closed Sequences. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013, Part I. LNCS, vol. 7818, pp. 50–61. Springer, Heidelberg (2013)

[7] Fournier-Viger, P., Gomariz, A., Campos, M., Thomas, R., “Fast Vertical Mining of Sequential Patterns Using Co-occurrence Infomation” PAKDD 2014. Part 1, LNAI 8443, pp. 40-52, 2014

[8] Lê Hoài Bắc “Báo cáo tổng kết kết quả đề tài KHCN cấp đại học quốc gia : Nâng cao hiệu quả của các thuật toán khai thác dữ liệu”, tại đại học khoa học tự nhiên, Tp. Hồ Chí Minh. 2011.