KHAI THÁC SỬ DỤNG WEB THEO MỐI QUAN TÂM CỦA NGƯỜI DÙNG

Văn Thị Thiên Trang
Khoa Công nghệ thông tin, Trường Đại học Kinh tế - Tài chính TP. HCM, Việt Nam,
trangvtt@uef.edu.vn

***Tóm tắt**: Với sự phát triển bùng nổ của dữ liệu thu thập được từ World Wide Web, việc khám phá và phân tích nguồn thông tin hữu ích từ World Wide Web đã trở thành nhu cầu cấp thiết. Khai thác web là áp dụng các công nghệ khai thác dữ liệu vào kho dữ liệu web đồ sộ. Khai thác sử dụng web là nhiệm vụ phát hiện các hoạt động của người sử dụng khi họ đang truy cập và điều hướng thông qua web. Mục đích là để hiểu biết các chuyển hướng của du khách để để nâng cao chất lượng của dịch vụ thương mại điện tử, để cá nhân hoá các cổng thông tin web hoặc để cải thiện cấu trúc web. Bài báo này trình bày khảo sát bài toán khai thác sử dụng web và đưa ra qui trình khai thác sử dụng web dựa trên mối quan tâm của người dùng.*

***Từ khóa**. Khai thác sử dụng web, mẫu truy cập web, mẫu tuần tự phổ biến, ràng buộc.*

1. Giới thiệu

Khai thác web liên quan đến một loạt các ứng dụng nhằm phát hiện và chiết xuất thông tin ẩn trong dữ liệu lưu trữ trên web. Mục đích quan trọng của khai thác web là cung cấp cơ chế giúp cho việc truy cập dữ liệu hiệu quả và đầy đủ hơn. Hơn nữa, là để khám phá những thông tin được bắt nguồn từ các hoạt động của người sử dụng, được lưu trữ trong các tập tin log. Về cơ bản, khai thác web được chia làm ba loại:

(1) Khai thác nội dung trang web (web content mining): rút trích những tri thức từ nội dung của các trang web.
(2) Khai thác cấu trúc web (web structure mining): mô tả cấu trúc quan hệ của các trang web và khai phá dựa trên các siêu liên kết giữa các trang có liên quan.
(3) Khai thác sử dụng web (web usage mining): khai thác các mẫu truy cập của người dùng (khai thác web log để biết xu hướng truy cập trang web của từng đối tượng người dùng tại các thời điểm khác nhau.

Khai thác sử dụng web là nhiệm vụ phát hiện các hoạt động của người sử dụng khi họ đang truy cập và điều hướng thông qua web. Mục đích là để hiểu biết các chuyển hướng của du khách để để nâng cao chất lượng của dịch vụ thương mại điện tử (e-commerce), để cá nhân hoá các cổng thông tin web hoặc để cải thiện cấu trúc web.

Phần lớn các trang web có thể được truy cập hàng nghìn lần mỗi ngày, đặc biệt là những trang web thương mại. Vấn đề là làm cách nào để thu thập những thông tin này nhằm phân tích xem người dùng duyệt gì, cần gì để có thể đưa ra những chiến lược quan trọng cho mô hình thương mại của các doanh nghiệp hiện tại. Các thông tin này thường được lưu trữ trong web log vì khi người dùng mở các trang web khác nhau thì dấu vết cũng như hành vi duyệt web của người sử dụng tự động lưu vào file log. Chính vì vậy, khai thác tri thức từ web log sẽ giúp các tổ chức trong việc đưa ra các quyết định kinh doanh, cải tiến, thiết kế trang web đạt đến một đỉnh cao mới trong lĩnh vực thương mại điện tử.

2. Khai thác sử dụng web theo mối quan tâm người dùng

Khai thác sử dụng web chính là khai thác mẫu truy cập web (hay còn gọi khai thác web log) là một dạng ứng dụng của khai thác mẫu tuần tự (đề xuất cầu tiên bởi Agrawal và Srikant, 1995) [1]. Khai thác mẫu tuần tự là đi tìm những chuỗi con phổ biến trong cơ sở dữ liệu chuỗi, trong đó mỗi chuỗi là một dãy

các sự kiện và mỗi sự kiện là một tập các item. Trong khai thác mẫu truy cập web, chuỗi dữ liệu là tập các trang web mà người dùng đã truy cập. Do đó, các sự kiện có thứ tự trong mỗi chuỗi dữ liệu của cơ sở dữ liệu (CSDL) truy cập web là một item đơn chứ không phải là một tập các item, giả sử rằng một người dùng chỉ có thể truy cập một trang web tại một thời điểm. Nếu xét ràng buộc thời gian *window-time* thì trong một khoảng thời gian cụ thể, người dùng có thể duyệt một tập hợp nhiều trang, khi đó CSDL truy cập web lúc này có dạng chung của CSDL chuỗi itemset. Hầu hết các nghiên cứu khai thác mẫu truy cập web hiện nay đều coi việc truy cập là một trang web tại một thời điểm. Ví dụ, cho tập sự kiện $E = \{a, b, c, d, e, f\}$, mỗi sự kiện đại diện cho một trang web mà khách hàng đã truy cập trong một ứng dụng thương mại điện tử. Một CSDL gồm các chuỗi truy cập web của 4 khách hàng sẽ có 4 bản ghi: [T1, ⟨*abdac*⟩]; [T2, ⟨*eadbcac*⟩]; [T3, ⟨*babfaec*⟩]; [T4, ⟨*abfad*⟩]. Khai thác mẫu truy cập web trên CSDL này có thể cho ra chuỗi phổ biến là *abac* với số lần xuất hiện 3/4, nghĩa là 70% trong số những người vào trang web của sản phẩm *a : http://www.company.com/producta.htm* cũng sẽ vào trang web của sản phẩm *b http://www.company.com/productb.htm*, và sau đó sẽ quay lại trang của sản phẩm *a* trước khi đến trang web của sản phẩm *c*. Dựa trên quy luật này, quản lý cửa hàng có thể thay thế giá quảng cáo của sản phẩm *a* trên trang web (*a* là sản phẩm xuất hiện nhiều lần trong chuỗi truy cập) để tăng doanh thu của các sản phẩm khác.

Các mẫu truy cập web khai thác được từ web log là những tri thức hữu ích và thú vị trong thực tiễn, được ứng dụng khá nhiều như giúp cải tiến thiết kế cho các trang web, phân tích hiệu suất hệ thống, hiểu được các hoạt động và tương tác người dùng, xây dựng các trang web thích nghi. Mặc đù đã có nhiều thuật toán được đề xuất cho khai thác mẫu truy cập web [3], [2], [7], [5], [4], nhưng tất cả đều khai thác với một độ đo duy nhất là độ phổ biến. Như đã đề cập từ đầu, khai thác với độ phổ biến vẫn đối diện với các thách thức về cả hiệu quả và hiệu suất thực hiện. Nếu chúng ta chỉ tập trung vào những mẫu có độ thú vị cao với người dùng thì sẽ rút ngắn thời gian khai thác. Trong ngữ cảnh bài toán khai thác mẫu truy cập web, chúng tôi đề xuất ràng buộc chuỗi con, là ràng buộc yêu cầu mẫu khám phá được phải chứa một trong số các chuỗi con gồm các trang web được truy cập theo thứ tự do người dùng chỉ ra trước. Chẳng hạn, nhà phân tích muốn tìm những mẫu truy cập có chứa chuỗi truy cập bắt đầu từ trang web du lịch, đến trang khách sạn và tiếp đến là trang đặt chuyến bay, như vậy phải tìm các mẫu có chứa chuỗi *{tourism → hotel → booking flight}*. Ở đây, mối quan tâm của người dùng được đại diện bởi ràng buộc chuỗi con, là hàm Boolean *C(p)* trên các mẫu, có thể biểu diễn dưới dạng $C(p) \equiv (\exists u \in U$ sao cho $p \supseteq u)$, với U là tập các chuỗi ràng buộc cho trước.

Như vậy, trong bài báo này, chúng tôi trình bày các bước giải quyết bài toán khai thác sử dụng web, trong đó kỹ thuật khai thác dữ liệu chính được áp dụng là khai thác mẫu truy cập web với ràng buộc chuỗi con đại diện cho các trang web theo thứ tự người dùng quan tâm truy cập.

3. Qui trình khai thác sử dụng web

Tổng thể quá trình khai thác sử dụng web có thể được chia thành các bước sau: thu thập dữ liệu, tiền xử lý dữ liệu, khai thác mẫu, phân tích đánh giá mẫu.

3.1. Thu thập và tiền xử lý dữ liệu

Một công việc quan trọng trong bất kỳ ứng dụng khai thác dữ liệu là việc tạo ra dữ liệu phù hợp để các giải thuật khai phá dữ liệu và thống kê có thể được áp dụng. Điều này đặc biệt quan trọng trong việc khai phá sử dụng web do các đặc điểm dữ liệu clickstream và mối quan hệ của nó với các dữ liệu khác có liên quan thu thập từ nhiều nguồn và qua nhiều kênh khác nhau. Quá trình chuẩn bị dữ liệu thường là bước tốn nhiều thời gian, công sức tính toán nhất và thường đòi hỏi việc sử dụng các thuật toán đặc biệt và công nghệ thường ít được sử dụng trong các lĩnh vực khác. Quá trình có thể bao gồm tiền xử lý các dữ liệu ban đầu,

tích hợp dữ liệu từ nhiều nguồn và chuyển đổi dữ liệu thành dạng phù hợp. Dữ liệu này sẽ là đầu vào cho các giải thuật khai phá. Tất cả các công việc trên được gọi chung là chuẩn bị dữ liệu.

3.1.1. Thu thập dữ liệu

Quá trình thu thập dữ liệu diễn ra như sau:

(1) Khách hàng gõ URL vào trình duyệt.

(2) Một yêu cầu sẽ được gửi đến Web server.

(3) Web server nhận yêu cầu vào tạo ra một dòng lưu trữ trong file log cho yêu cầu đó. (Tên trang, địa chỉ IP, thông tin trình duyệt, ngày, giờ).

(4) Web server gửi trang web yêu cầu về cho khách hàng.

Định dạng của tập tin Web log: mặc dù web log được lưu tại máy chủ nhưng tuỳ theo nhà cung cấp dịch vụ web, máy chủ có thể được cài đặc hệ điều hành, Web server và các phần mềm quản lý web khác nhau. Do đó, cấu trúc của các sseb log cũng khác nhau, có các định dạng: NCSA common, NCSA combined, định dạng mở rộng W3C format và định dạng IIS.

Tập tin web log ghi lại các thông tin cơ bản về các yêu cầu của người dùng đối với một website. Thông tin này được ghi lại dưới dạng: *host/ip user [date:time] "method url" status bytes "ReferenceUrl" "agent",* **Hình 1** minh họa một đoạn thông tin trong file web log gồm 3 lượt truy cập. Trong đó:

- *host/ip*: địa chỉ *host/ip* của máy tính truy cập vào trang web.
- *user*: số định danh người dùng (biểu thị định danh bị giấu đi).
- *[date: time]:* thời gian truy cập.
- *method*: phương thức yêu cầu của người sử dụng web (GET; POST).
- *url*: đường dẫn của trang web được truy cập.
- *status:* tình trạng của yêu cầu (200).
- *byte:* số lượng byte dữ liệu đã yêu cầu.
- *RefernceUrl:* địa chỉ trang web trước mà từ đó dẫn đến địa chỉ hiện tại.
- *agent:* thông tin về hệ điều hành, trình duyệt của máy người sử dụng, chẳng hạn như tên người dùng, ngày, giờ, loại yêu cầu, mã trạng thái HTTP, và số lượng các byte được gửi bởi máy chủ.

```
66.249.79.19 - - [15/Jun/2018:04:06:53 +0700] "GET /toyota-corolla-altis-2015.html HTTP/1.0" 200 6794 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
37.140.141.38 - - [15/Jun/2018:04:14:06 +0700] "GET /robots.txt HTTP/1.0" 200 865 "-" "Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)"
5.45.254.225 - - [15/Jun/2018:04:14:09 +0700] "GET /san-pham/yaris/Page-2.html HTTP/1.0" 200 6931 "-" "Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)"
```

Hình 1. *Ví dụ về nội dung của một file web log.*

Trong số những thông tin được ghi lại trong tập tin log, có những thông tin không cần thiết cho việc khai thác. Do đó, cần có bước tiền xử lý loại bỏ những thông tin dư thừa để thu được CSDL chuỗi dùng cho việc khai thác mẫu truy cập web có dạng mẫu tuần tự [4].

3.1.2. Tiền xử lý dữ liệu

Trong giai đoạn tiền xử lý, dữ liệu truy cập của người sử dụng web được làm sạch và phân hoạch thành một tập hợp các chuỗi truy cập của người dùng. Tập hợp này đại diện cho các hoạt động của mỗi người dùng trong những lần truy cập trang web.

Khai thác thói quen sử dụng web tiến hành trên dữ liệu được tạo ra bằng cách xem xét các session hoặc các thói quen, chúng được lưu trữ trên web log được rút trích từ các Web server.

Các bước tiền xử lý dữ liệu gồm: làm sạch dữ liệu; xác định người dùng và các lần truy cập (User & Session Identification).

Tổng hợp và làm sạch dữ liệu (Data fusion and cleaning): Làm sạch dữ liệu bao gồm các công việc như loại bỏ các thành phần không cần thiết với mục đích phân tích như tập tin style, đồ họa hoặc âm thanh. Quá trình làm sạch cũng có thể bao gồm việc loại bỏ ít nhất là một số các trường dữ liệu (ví dụ: số byte được chuyển giao hoặc phiên bản của

giao thức HTTP được sử dụng) mà có thể không cung cấp thông tin hữu ích trong việc khai thác phân tích hoặc dữ liệu. Làm sạch dữ liệu cũng đòi hỏi việc loại bỏ thông tin thu thập không cần thiết. Điều này không phải là không phổ biến cho một tập tin log có chứa một tỷ lệ phần trăm (đôi khi cao đến 50%) kết quả từ các công cụ tìm kiếm hoặc thu thập thông tin khác. Ta có thể xác định và loại bỏ nó bằng cách duy trì một danh sách các trình thu thập được biết đến.

Xác định các lượt xem (Pageview Identification): Việc xác định các trang được xem phụ thuộc nhiều vào cấu trúc bên của trang web cũng như về nội dung trang. Về mặt ý niệm mỗi lần xem trang có thể được xem như là tập của các đối tượng web hoặc tài nguyên tương ứng cho một "sự kiện người dùng" cụ thể, ví dụ: nhấp chuột vào một liên kết, xem một trang sản phẩm, thêm sản phẩm vào giỏ hàng. Trong trang web thương mại điện tử, xem trang có thể tương ứng với các tiêu chí khác nhau dựa trên sản phẩm, chẳng hạn như quan điểm về sản phẩm, đăng ký, thay đổi giỏ mua hàng, mua hàng… Trong trường hợp này, xác định các lần xem trang có thể yêu cầu thêm môt tiêu chí nào đó mà từ đó những người sử dụng khác nhau có thể được phân loại.

Chứng thực xác định phiên người dùng (Sessionization) là quá trình phân mảnh các hoạt động của từng người sử dụng thành các session, mỗi session đại diện cho một lần đến trang web. Website mà không có cơ chế xác thực bổ sung từ người sử dụng thì phải dựa vào kinh nghiệm để định ra session. Mục tiêu của một heuristic sessionization là tái xây dựng từ các dữ liệu clickstream thành chuỗi các hành động bởi một người dùng trong một lần truy cập cho trang web.

3.2. Khai thác mẫu

Khai thác mẫu truy cập web (còn gọi là khai thác thói quen sử dụng web, khai thác web log) là một ứng dụng quan trọng của khai thác mẫu tuần tự, có liên quan đến việc tìm kiếm các mẫu điều hướng của người dùng trên hệ thống World Wide Web bằng cách rút trích những tri thức từ các truy cập web được ghi lại trong các tập tin log, ở đó các sự kiện có thứ tự trong mỗi chuỗi của CSDL là các trang web mà người dùng đã truy cập.

Giả sử rằng một người dùng web chỉ có thể truy cập một trang web tại một thời điểm bất kỳ, CSDL chuỗi truy cập web có dạng đặc biệt, đó là mỗi sự kiện trong chuỗi chỉ có một item. Bài toán khai thác mẫu tuần tự khi giới hạn trong phạm vi khai thác mẫu truy cập web với một CSDL *WD*, *minSup* và tập các sự kiện $E = \{e_1, e_2, \ldots, e_m\}$ đại diện cho địa chỉ các trang web với đặc điểm như sau:

Các mẫu trong một web log bao gồm các trang web kế tiếp nhau mà người dùng đã xem (tương ứng là các item trong chuỗi). Tại một thời điểm, không thể có hai trang cùng được truy cập bởi một người dùng, do đó chuỗi chỉ gồm các 1-itemset (tức các itemset chỉ có gồm một item đơn). Ví dụ, chuỗi ⟨BCAB⟩ là chuỗi gồm 4 1-itemset khác với chuỗi dạng tổng quát ⟨B(CA)B⟩ là chuỗi gồm 3 itemset (mỗi itemset gồm 1 hoặc nhiều item).

Cũng mang đặc điểm của dữ liệu chuỗi nói chung, thứ tự các trang web được duyệt trong chuỗi truy cập cũng đóng vai trò quan trọng. Ngoài ra, mỗi sự kiện hay item có thể xuất hiện lặp lại, biểu diễn cho việc quay lui duyệt lại trang web hoặc thao tác *"refresh"* trang. Ví dụ, hai chuỗi ⟨ABA⟩, ⟨AABC⟩ có A là sự kiện xuất hiện lặp lại.

Dữ liệu web log là những tập dữ liệu thưa, tức là thường có nhiều item nhưng chỉ có một số ít item xuất hiện lặp lại trong chuỗi duyệt web tương ứng của một người dùng.

Khám phá những thông tin ẩn từ dữ liệu web log được gọi là khai thác hành vi sử dụng web. Mục đích của việc khám phá các mẫu tuần tự phổ biến trong dữ liệu web log là để có được thông tin về các hành vi truy cập của người sử dụng với mục đích dự đoán và tìm nạp trước các trang web mà người dùng có khả năng truy cập.

3.2.1. Các thuật toán khai thác mẫu

Vì mẫu truy cập web là một trường hợp riêng của mẫu tuần tự nên tất cả các thuật toán khai thác mẫu tuần tự đều áp dụng được cho khai thác mẫu truy cập web.

Vì cấu trúc của mẫu truy cập web đơn giản hơn cấu trúc mẫu tuần tự nên ngoài những phương pháp khai thác mẫu tuần tự chung, có những phương pháp khai thác riêng dành cho loại dữ liệu này.

Pei và đồng sự (2000) đã đề xuất cấu trúc cây để lưu thông tin mẫu truy cập web, gọi tắt là cây-*WAP*, để biểu diễn cho thông tin về sự xuất hiện của mẫu trong CSDL và thuật toán *WAP-Mine* để khai thác tất cả các mẫu phổ biến tìm được từ cây-WAP này [2]. Mỗi nút trên cây được gán nhãn là một item và độ phổ biến của item đó, và mỗi nhánh trên cây biểu diễn cho một mẫu truy cập web. Các mẫu có chung tiền tố sẽ có chung một đoạn đường đi trên cây. Các nút có cùng nhãn sẽ được liên kết với nhau bằng cách chia sẻ chung nhãn liên kết trong một hàng đợi. Có một bảng để giữ danh sách các phần tử đầu hàng đợi.

Để dựng cây *WAP*, thuật toán *WAP-Mine* duyệt CSDL đầu vào hai lần, một lần để tìm các item phổ biến và một lần để đưa các chuỗi dữ liệu đầu vào đã bỏ đi các item không phổ biến vào cây. Sau đó, thuật toán tiến hành khai thác các mẫu phổ biến từ cây *WAP*. Ý tưởng chính của thuật toán là tìm kiếm có điều kiện. Trước hết, nó tìm các mẫu hậu tố có điều kiện, sau đó dựng lại cây WAP điều kiện trung gian sử dụng các mẫu tìm được ở bước trước. *WAP-Mine* không tạo ra tập ứng viên khổng lồ như Apriori nhưng phải dựng rất nhiều cây *WAP* trung gian trong suốt quá trình khai thác, do đó nó vẫn tiêu tốn khá nhiều thời gian và bộ nhớ.

Một số nghiên cứu cải tiến từ cây *WAP* bao gồm cây *PLWAP* (Lu & Ezeife, 2003 [3]), *FLWAP-tree* (Tang, Turkia, & Gallivan, 2007 [5]) và cây *AWAPT* (Vijayalakshmi, Mohan, & Suresh, 2010 [4]). Để tránh lặp lại đệ quy việc tái cấu trúc các cây *WAP* trung gian, *PLWAP* gán thêm mã vị trí nhị phân cho các nút trên cây để có thể xác định nhanh hậu tố của bất kỳ mẫu tiền tố phổ biến nào. *FLWAP* và *AWAPT* đều là các phiên bản cải tiến của cây *PLWAP*. Tất cả thuật toán dùng cây *WAP* hoặc cây tương tự cây *WAP* đều khác biệt so với các thuật toán dựa trên Apriori và vượt trội hơn. Chúng không tạo ra khối lượng ứng viên đồ sộ, duyệt CSDL nhiều lần và đếm độ phổ biến cũng dễ dàng hơn. Chúng sử dụng cấu trúc cây-liên kết để đại diện cho CSDL đầu vào định dạng ngang. Tuy nhiên, chúng vẫn có những nhược điểm, phương pháp cây *WAP* phải lưu trữ các mẫu trung gian khi xây dựng lại nhiều cây *WAP* trung gian, *PLWAP* phải gia tăng kích thước của các nút trên cây và *FLWAP* chiếm dụng bộ nhớ cao như *PLWAP*.

Nhìn chung các thuật toán theo hướng tiếp cận dùng cây *WAP* tối ưu về thời gian và bộ nhớ hơn so với các phương pháp Apriori, song lại không hiệu quả bằng các phương pháp định dạng CSDL theo chiều dọc, do đó không còn thu hút nghiên cứu trong thời gian gần đây.

3.2.2. Thuật toán khai thác mẫu theo mối quan tâm của người dùng

Trong ngữ cảnh bài toán khai thác mẫu truy cập web, ràng buộc chuỗi con đại diện cho mối quan tâm của người dùng, là ràng buộc yêu cầu mẫu khám phá được phải chứa một trong số các chuỗi con gồm các trang web được truy cập theo thứ tự do người dùng chỉ ra trước.

Một số định nghĩa:

Định nghĩa 1 (Chuỗi truy cập web). Cho tập $E = \{e_1, e_2, ..., e_m\}$ gồm m sự kiện còn gọi là các item (mỗi item đại diện cho một trang web mà khách hàng đã truy cập). Một chuỗi truy cập web có dạng $s = \langle e_1\ e_2\ ...\ e_n\rangle$ ($e_i \in E$ với mỗi $1 \le i \le n$) là dãy item có thứ tự, có thể xuất hiện lặp lại, và n được gọi là độ dài của chuỗi. Một chuỗi có độ dài n được gọi là *n-sequence*, kí hiệu $|s| = n$. Ví dụ, chuỗi ⟨ABCBAC⟩ là *6-sequence*.

Định nghĩa 2 (Chuỗi con, vị trí xuất hiện). Chuỗi $S_a = \langle a_1\ a_2\ \dots\ a_n\rangle$ được gọi là chuỗi con của chuỗi $S_b = \langle b_1\ b_2\ \dots\ b_m\rangle$, và S_b là chuỗi cha của S_a, ký hiệu $S_a \subseteq S_b$ hay $S_b \supseteq S_a$, nếu tồn tại các số nguyên $1 \le j_1 < j_2 < \dots < j_n \le m$ sao cho $a_1 = b_{j1}$, $a_2 = b_{j2}$, ..., $a_n = b_{jn}$. Khi đó, j_n được gọi là vị trí xuất hiện của chuỗi S_a trong S_b, kí hiệu $pos_{Sa} = j_n$. Ở đây, vị trí của chuỗi S_a được lấy theo vị trí xuất hiện của item cuối cùng a_n của nó trong chuỗi S_b. Một chuỗi con được gọi là ***mẫu***. Ví dụ, chuỗi

⟨ABC⟩ là chuỗi con của chuỗi ⟨ABACAC⟩ với vị trí xuất hiện là {4, 6} (giả sử vị trí đầu tiên tính là 1). Vị trí {4} là ⟨**AB**A**C**AC⟩ và {6} là ⟨**AB**ACA**C**⟩. Tuy nhiên, chuỗi ⟨CAB⟩ không phải là chuỗi con của ⟨ABACAC⟩.

Định nghĩa 3 (mẫu thỏa ràng buộc). Cho một ràng buộc $u \in U$, mẫu p được gọi là *mẫu thỏa ràng buộc-u* nếu u là một chuỗi con của p, tức $p \supseteq u$.

Định nghĩa 4 (Cơ sở dữ liệu chuỗi truy cập web). Cơ sở dữ liệu chuỗi truy cập web *WD* dùng để khai thác là một tập hợp các chuỗi web log đầu vào. Mỗi chuỗi có một chỉ số định danh duy nhất. CSDL truy cập web này được lấy từ các tập tin log gốc qua bước tiền xử lý và chuẩn hóa dữ liệu. Chuỗi đầu vào s được coi là *chứa* mẫu p nếu p là chuỗi con của s hay có thể nói *p có mặt* trong s.

Phát biểu bài toán: Cho CSDL chuỗi truy cập web *WD*, tập mẫu ràng buộc $U = \{u_1, u_2 ... u_n\}$ và ngưỡng phổ biến tối thiểu *minSup* do người dùng chỉ ra. Bài toán khai thác mẫu truy cập web với ràng buộc chuỗi con là đi tìm tất cả mẫu phổ biến trong CSDL mà có chứa bất kỳ mẫu nào của tập U dưới dạng chuỗi con.

$FCP = \{p | sup(p) \geq minSup \wedge \exists k: 1 \leq k \leq n, p \supseteq u_k\}$.

Thuật toán đề xuất cấu trúc cây mẫu truy cập web theo tiền tố, gọi tắt là cây *PreWAP*, để lưu trữ các mẫu ứng viên. Mỗi nút trên cây *PreWAP* gồm hai phần: nhãn và *DBVP*, trong đó nhãn của nút là một mẫu truy cập web và *DBVP* là đại diện thông tin về sự xuất hiện của mẫu trong CSDL.

Mở rộng mẫu: mẫu mới được tạo ra bằng cách mở rộng một k-mẫu ($k > 0$) phổ biến với một item phổ biến. Item được thêm vào cuối mẫu. Như vậy, mở rộng mẫu truy cập web là dạng mở rộng sequence của mẫu tuần tự nói chung.

Cho $\alpha = \langle a_1 a_2 ... a_n \rangle$ là một mẫu phổ biến và là e một item phổ biến. Gọi SID_α, SID_e, lần lượt là ID của các chuỗi trong *WD* có chứa α, e và pos_α, pos_e là vị trí xuất hiện tương ứng của α, e trong các chuỗi đầu vào đó. Mở rộng mẫu α bằng item e, ta được mẫu mới $\alpha' = \langle a_1 a_2 ... a_n e \rangle$ với $SID_{\alpha'} = SID_e$, $pos_{\alpha'} = pos_e$ nếu $(SID_\alpha = SID_e) \wedge (pos_\alpha < pos_e)$, ngược lại $SID_{\alpha'} = 0$, $pos_{\alpha'} = \varnothing$. Ta có α là tiền tố của α'.

Cách xây dựng cây PreWAP [8]: bắt đầu từ nút gốc của cây tại mức *0*, nút gốc được gán nhãn là một chuỗi rỗng ⟨⟩. Tại mức k bất kỳ, mỗi nút được gán nhãn là một mẫu tuần tự độ dài k. Các nút ở mức *(k+1)* kế tiếp được xây dựng đệ quy bằng cách mở rộng mẫu độ dài k ở mức trước đó.

Trước hết chúng tôi đề xuất ba mệnh đề làm cơ sở cho các kỹ thuật được áp dụng trong thuật toán nhằm tăng hiệu quả về thời gian và không gian lưu trữ trong quá trình khai thác mẫu. Đặt $u = \langle E_1 E_2 ... E_n \rangle \in U$ là một chuỗi con ràng buộc, s là một chuỗi dữ liệu đầu vào trong CSDL, F_1 là tập các *atom* với *atom* là các mẫu-*1*.

Mệnh đề 1. *Lấy p là mẫu thỏa ràng buộc-u. Nếu* $(u \not\subset s)$ *thì* $(p \not\subset s)$.

Chứng minh: Giả sử $(u \not\subset s)$ và $p \subseteq s$ (1). Vì *p là mẫu thỏa ràng buộc-u* nên theo Định nghĩa 3 ta có $p \supseteq u$ (2). Từ (1) và (2) suy ra $u \subseteq s$. Điều này trái với giả thiết, do đó ta có điều phải chứng minh.

Mệnh đề 2. *Lấy* $X \in F_1$ *là một atom, ST(X) là cây con có gốc tại nút X của cây PreWAP và* $p \in ST(X)$ *là mẫu thỏa ràng buộc-u. Nếu* $(p \subseteq s)$ *thì* $(X \subseteq s) \wedge (u \subseteq s) \wedge (pos_X \leq pos_{E_1})$.

Chứng minh:

Vì $p \in ST(X) \Rightarrow X$ là tiền tố của p (tính chất của cây *PreWAP*) $\Rightarrow X \subseteq p$ (3). Theo giả thiết $p \subseteq s$ (4). Từ (3) và (4), ta có $X \subseteq s$.

Vì *p là mẫu thỏa ràng buộc-u* $\Rightarrow p \supseteq u$ hay $u \subseteq p$ (Định nghĩa 3) và $p \subseteq s$ (giả thiết) $\Rightarrow u \subseteq s$.

Vì $p \in ST(X)$ nên p có dạng $p = \langle XA_1A_2 ... A_m \rangle$. Theo giả thiết, p là *mẫu thỏa ràng buộc-u* $\Rightarrow p \supseteq u = \langle E_1 E_2 ... E_n \rangle$. Có 2 trường hợp xảy ra: $E_1 = X \Rightarrow pos_X = pos_{E_1}$ hoặc $E_1 = A_i$ $(1 \leq i < m) \Rightarrow pos_X < pos_{E_1}$.

Vậy ta có mệnh đề 2 được chứng minh.

Mệnh đề 3. *Lấy ST(p) là cây con gốc tại nút p của cây PreWAP. Nếu p là mẫu thỏa ràng buộc-u thì q cũng là mẫu thỏa ràng buộc-u với mọi q ∈ ST(p).*

Chứng minh: Theo giả thiết, *p là mẫu thỏa ràng buộc-u* $\Rightarrow u \subseteq p$. Vì $q \in ST(p)$ nên *p* là tiền tố của *q* (theo tính chất của cây *PreWAP*) $\Rightarrow p \subseteq q$. Vậy ta có $u \subseteq q \Rightarrow$ *q là mẫu thỏa ràng buộc-u.*

Dựa vào các mệnh đề trên, thuật toán *EMWAPC* ý tưởng chính như sau: tiến trình khai thác xuất phát từ mỗi cây con có gốc tại mỗi *atom* trong F_1, *EMWAPC* tỉa không gian tìm kiếm của cây *PreWAP* ngay từ đầu trước khi thực hiện mở rộng mẫu nhờ Mệnh đề 1 và Mệnh đề 2. Sau đó, trong quá trình mở rộng mẫu để tạo mẫu ứng viên mới, thay vì phải kiểm tra ràng buộc cho mỗi ứng viên mới như *MWAPC*, *EMWAPC* có thể bỏ qua bước kiểm tra này cho một số lượng lớn ứng viên dựa trên Mệnh đề 3. Chi tiết của thuật toán *EMWAPC* được mô tả dưới đây:

Bảng 1. Thuật toán khai thác mẫu truy cập web dựa trên ràng buộc

```
Thuật toán EMWAPC
Đầu vào: WD, minSup, tập ràng buộc U = {u1, u2... un}
Đầu ra: FCP (tập các mẫu truy cập web thỏa minSup và U).
1. FCP = ∅;
2. Duyệt WD để tìm F1 và DBVP của chúng;
3. Tìm U' = {ui ∈ U | sup(ui) ≥ minSup} bằng
   cách tính DBVPui, ∀ui ∈ U;
4. F1* = Gọi EARLY-PRUNING(F1, U', minSup);
5. For each node r in in F1* do
6.     If (r.label thỏa 1 ràng buộc u ∈ U')
   then
7.         FCP = FCP ∪ {r.label};
8.         Gọi EXTENSION (r, F1, minSup);
9.     If (r.label không thỏa mọi ràng buộc ue∈
   U') then
10.            Gọi EXTENSION-CHECK (r, F1,
   minSup, U');
Thủ tục EXTENSION (r, I, minSup)
11.    Lấy I1 = {e ∈ I| sup(đặt pe = Pattern -
   Extension(p, e)) ≥ minSup};
12.    For each item e in I1 do
13.        FCP = FCP ∪ {pe.label};
14.        Gọi EXTENSION(pe, I1, minSup);
Thủ tục EXTENSION-CHECK(r, I, minSup, U')
15.    Lấy I1 = {e ∈ I| sup(đặt pe = Pattern -
   Extension(p, e)) ≥ minSup};
16.    For each item e in I1 do
17.        If (pe.label thỏa 1 ràng buộc u ∈
   U') then
18.            FCP = FCP ∪ {pe.label};
19.            Gọi EXTENSION (pe, I1,
   minSup);
20.        If (pe.label không thỏa mọi ràng
   buộc ue∈ U') then
21.            Gọi EXTENSION-CHECK (pe, I1,
   minSup, U');
```

Đầu tiên, thuật toán duyệt CSDL để tìm tập F_1 cùng với các *DBVP* tương ứng (dòng 2). Kế tiếp, xác định độ phổ biến của các chuỗi con ràng buộc trong tập *U* bằng cách sử dụng *DBVP* mà không cần truy cập CSDL (dòng 3). Nhờ đó, *EMWAPC* không tốn thời gian quét qua một lượng lớn các chuỗi dữ liệu để xác định có chứa $u \in U$ hay không như thuật toán *MWAPC*. Có thể tìm $DBVP_u$ bằng cách dùng phép mở rộng mẫu với lần lượt các item trong *u*. Ở đây, vị trí xuất hiện của *u* trong các chuỗi đầu vào được đại diện bởi vị trí của item đầu tiên của *u* (thay vì item cuối cùng như thông thường) để phục vụ cho chiến lược tỉa. Lưu ý rằng, nếu $\exists e \in u$ nhưng $e \notin F_1$ hoặc $sup(u) < minSup$ thì xóa *u* khỏi tập *U*. Sau đó, thuật toán gọi thủ tục EARLY-PRUNING để tỉa không gian tìm kiếm ngay từ đầu nhờ áp dụng Mệnh đề 1 và Mệnh đề 2 (dòng 4). Tiếp theo, thuật toán thực hiện mở rộng mẫu để tìm mẫu mới theo cách tương tự như *MWAPC*. Tuy nhiên, *EMWAPC* không cần kiểm tra ràng buộc cho toàn bộ mẫu dược tạo ra. Dựa trên Mệnh đề 3, nếu nút gốc của một cây con là mẫu thỏa ràng buộc thì chúng ta chỉ cần thực hiện mở rộng mẫu và đưa vào tập kết quả mà không cần kiểm tra ràng buộc bằng thủ tục EXTENSION (dòng 5 - 8); nghĩa là thuật toán bỏ qua bước kiểm tra ràng buộc cho tất cả các mẫu trên cây con đó. Ngược lại, nếu không thỏa ràng buộc nào thì tiếp tục mở rộng và kiểm tra ràng buộc trên mẫu mới nhờ thủ tục EXTENSION-CHECK (dòng 9, 10).

Tương tự, nếu mẫu mới tạo lại thỏa ràng buộc thì bỏ qua bước kiểm tra ràng buộc cho tất cả nút con cháu của nó. Quá trình này lặp đệ quy cho đến khi không còn tạo ra ứng viên phổ biến.

Thủ tục EARLY-PRUNING: Kỹ thuật tỉa không gian tìm kiếm ngay từ đầu dựa trên Mệnh đề 1 và Mệnh đề 2. Mệnh đề 1 phát biểu rằng nếu chuỗi con ràng buộc *u* không có mặt trong chuỗi đầu vào *s* thì các mẫu *p* thỏa ràng buộc-*u* cần tìm ($p \in FCP$) cũng sẽ không có mặt trong *s*. Do đó, có thể loại bỏ các chuỗi đầu vào không tham gia vào độ phổ biến của mẫu *p* ngay từ đầu dựa vào DBV_u. Vì *p* được tạo ra từ các *atom* $X \in F_1$ nên thực hiện loại bỏ ngay trên DBV_X. Nếu bit thứ *k* của DBV_u là *'0'* thì ta cho bit thứ *k* của DBV_X là *'1'* bằng cách $DBV_X = (DBV_u$ *AND* $DBV_X)$. Hơn nữa, từ Mệnh đề 2 có thể suy luận rằng những chuỗi đầu vào đóng góp vào độ phổ biến của mẫu *p* là những chuỗi có chứa cả *X* và *u*, trong đó *X* đứng trước *u*. Vì vậy, sau khi thực hiện phép *AND*, nếu kết quả là bit *'1'* và *X* xuất hiện sau *u* thì ta xóa những vị trí không hợp lệ đó trong dãy vị trí của $DBVP_X$. Do $|U'| \geq 1$ và mẫu chỉ cần thỏa một trong số các ràng buộc của tập *U'*, gọi *d* là đại diện của tập *U'*, khi đó $DBVP_d$ được xác định như sau:

$$DBVP_d \begin{cases} DBV_d = OR(DBV_{u_i}) \text{ (sử dụng phép OR bit)} \\ pos_d = MAX(pos_{u_i}), \quad \text{với } u_i \in U', 1 \leq i \leq |U'| \end{cases}$$

Tóm lại, nhờ áp dụng các mệnh đề, thuật toán *EMWAPC* có thể tỉa không gian tìm kiếm ngay từ đầu quá trình khai thác và bỏ qua bước kiểm tra ràng buộc cho một số lượng lớn ứng viên.

3.3. Phân tích đánh giá mẫu

Sử dụng các phương pháp khai thác dữ liệu trong các lĩnh vực khác nhau như khai thác mẫu, luật tuần tự, luật kết hợp, phân tích, thống kê, phân tích đường dẫn, phân lớp v.v… để khám phá ra các mẫu người dùng.

Trong giai đoạn cuối của quá trình, các mẫu phát hiện và thống kê được tiếp tục xử lý, lọc và có thể được sử dụng làm đầu vào cho các ứng dụng như công cụ trực quan, phân tích web và các công cụ tạo báo cáo. Phân tích mô hình, thống kê, tìm kiếm tri thức và tác nhân thông minh. Phân tích tính khả thi, truy vấn dữ liệu hướng tới sự tiêu dùng của con người.

4. Kết luận

Bài báo trình bày tổng quan về quy trình khai thác sử dụng web từ thu thập tiền xử lý dữ liệu, khai thác mẫu và phân tích tích đánh giá mẫu. Trong đó, ứng dụng thuật toán khai thác mẫu truy cập web dựa trên ràng buộc chuỗi con vào khai thác hành vi sử dụng web. Việc sử dụng ràng buộc chuỗi truy cập giúp quá trình khai thác sử dụng web hiệu quả hơn về tập kết quả thu được theo mối quan tâm của người dùng và thời gian khai thác được thu gọn.

Tài liệu tham khảo

[1] R. Agrawal, R.. Srikant (1995), "Mining sequential patterns", *Proceedings of the 11th International Conference on Data Engineering*, pp. 3-14.

[2] J. Pei, J. Han, B. Mortazavi-asl, & H. Zhu (2000), "Mining Access Patterns Efficiently from Web Logs", *In PAKDD*, LNCS, vol. 1805, pp. 396–407.

[3] Y. Lu, & C. I. Ezeife (2003), "Position Coded Pre-order Linked WAP-Tree for Web Log Sequential Pattern Mining", *In PAKDD*, LNCS (LNAI), vol. 2637, pp. 337–349.

[4] X. Y. Li (2013), "Data preprocessing in web usage mining", In *The 19th International Conference on Industrial Engineering and Engineering Management*, pp. 257-266, Springer, Berlin, Heidelberg.

[5] P. Tang, M. P. Turkia, & K. A. Gallivan (2007), " Mining web access patterns with first-occurrence linked WAP-trees", In SEDE' 07, pp. 247-252.

[6] J. Pei, J. Han, B. Mortazavi-asl, & H. Zhu (2000), "Mining Access Patterns Efficiently from Web Logs", *In PAKDD*, LNCS, vol. 1805, pp. 396–407.

[7] A. Rajimol, & G. Raju (2012), "Web access pattern mining–a survey", *Data Engineering and Management*, Lecture Notes in Computer Science, vol 6411. Springer, Berlin, Heidelberg, pp. 24–31.

[8] V. Trang, A Yoshitaka, & L. Bac (2018), "Mining web access patterns with supper-pattern constraints", *Applied Intelligence*, vol. 48(11), pp. 3902-3914.