DƯƠNG QUỐC THẮNG

LUẬN VĂN THẠC SĨ

2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

DƯƠNG QUỐC THẮNG

ỨNG DỤNG KHAI THÁC MẪU CHUỖI ĐỂ KHAI THÁC HÀNH VI SỬ DỤNG WEB

LUẬN VĂN THẠC SĨ

Chuyên ngành : Công nghệ thông tin

Mã số ngành: 60480201

TP. HỒ CHÍ MINH, tháng 02 năm 2016

DƯƠNG QUỐC THẮNG

LUẬN VĂN THẠC SĨ

2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

DƯƠNG QUỐC THẮNG

ỨNG DỤNG KHAI THÁC MẪU CHUỖI ĐỂ KHAI THÁC HÀNH VI SỬ DỤNG WEB

LUẬN VĂN THẠC SĨ

Chuyên ngành : Công nghệ thông tin

Mã số ngành: 60480201

CÁN BỘ HƯỚNG DẪN KHOA HỌC: PGS.TS. VÕ ĐÌNH BẢY

TP. HỒ CHÍ MINH, tháng 02 năm 2016

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

Cán bộ hướng dẫn khoa học : **PGS.TS. VÕ ĐÌNH BẢY**
(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Luận văn Thạc sĩ được bảo vệ tại Trường Đại học Công nghệ TP. HCM ngày 20 tháng 03 năm 2016

Thành phần Hội đồng đánh giá Luận văn Thạc sĩ gồm:
(Ghi rõ họ, tên, học hàm, học vị của Hội đồng chấm bảo vệ Luận văn Thạc sĩ)

TT	**Họ và tên**	**Chức danh Hội đồng**
1	PGS. TSKH Nguyễn Xuân Huy	Chủ tịch
2	PGS. TS Vũ Đức Lung	Phản biện 1
3	TS. Cao Tùng Anh	Phản biện 2
4	TS. Hồ Đắc Nghĩa	Ủy viên
5	TS. Vũ Thanh Hiền	Ủy viên, Thư ký

Xác nhận của Chủ tịch Hội đồng đánh giá Luận sau khi Luận văn đã được sửa chữa (nếu có).

Chủ tịch Hội đồng đánh giá LV

TRƯỜNG ĐH CÔNG NGHỆ TP. HCM
PHÒNG QLKH – ĐTSĐH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

TP. HCM, ngày … tháng … năm .....

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ tên học viên: .Dương Quốc Thắng — Giới tính: Nam

Ngày, tháng, năm sinh: 15/03/1982 — Nơi sinh:Tiền Giang

Chuyên ngành: Công nghệ thông tin — MSHV: 1441860024

I- Tên đề tài:

ỨNG DỤNG KHAI THÁC MẪU CHUỖI ĐỂ KHAI THÁC HÀNH VI SỬ DỤNG WEB

II- Nhiệm vụ và nội dung:

- Cơ sở lý thuyết khai thác mẫu chuỗi và khai thác luật.
- Khai thác mẫu chuỗi đươc đề xuất dựa theo thuật toán sự kết hợp của bit vectơ động cho khai thác chuỗi phổ biến đóng và tìm hiểu chi tiết khai thác luật.
- Viết ứng dụng vào thuật toán đã đươc tìm hiểu.

III- Ngày giao nhiệm vụ: 15/0*7/2015*

IV- Ngày hoàn thành nhiệm vụ: 15/02/2016

V- Cán bộ hướng dẫn: PGS.TS. VÕ ĐÌNH BẢY

CÁN BỘ HƯỚNG DẪN — **KHOA QUẢN LÝ CHUYÊN NGÀNH**

(Họ tên và chữ ký) — (Họ tên và chữ ký)

LỜI CAM ĐOAN

Tôi xin cam đoan đề tài **"Ứng dụng khai thác mẫu chuỗi để khai thác hành vi sử dụng Web"** là công trình nghiên cứu của riêng tôi. Các số liệu, kết quả nêu trong Luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tôi xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện Luận văn này đã được cảm ơn và các thông tin trích dẫn trong Luận văn đã được chỉ rõ nguồn gốc.

Học viên thực hiện Luận văn

(Ký và ghi rõ họ tên)

LỜI CÁM ƠN

Để có được kết quả như ngày hôm nay, tôi luôn ghi nhớ công ơn của các thầy cô, bạn bè, đồng nghiệp và gia đình, những người đã dạy bảo và ủng hộ tôi trong suốt quá trình học tập.

Trước hết, tôi muốn gửi lời cảm ơn đến Viện đào tạo sau đại học đã quan tâm tổ chức chỉ đạo và trực tiếp giảng dạy khoá cao học của chúng tôi. Đặc biệt, tôi xin gửi lời cảm ơn sâu sắc đến thầy hướng dẫn PGS.TS. Võ Đình Bảy, người đã tận tình chỉ bảo và góp ý về mặt chuyên môn cho tôi trong suốt quá trình làm luận văn.

Cũng qua đây, tôi xin gửi lời cảm ơn đến ban lãnh đạo Trường Đai Học Công Nghệ TP.HCM – HUTECH đã tạo mọi điều kiện thuận lợi cho tôi trong thời gian hoàn thành các môn học cũng như trong suốt quá trình làm luận văn tốt nghiệp.

Trong suốt quá trình làm luận văn, bản thân tôi đã cố gắng tập trung tìm hiểu, nghiên cứu và tham khảo thêm nhiều tài liệu liên quan. Tuy nhiên, do bản thân mới bắt đầu trên con đường nghiên cứu khoa học, chắc chắn bản luận văn vẫn còn nhiều thiếu sót. Tôi rất mong được nhận sự chỉ bảo của các Thầy Cô giáo và các góp ý của bạn bè, đồng nghiệp để luận văn được hoàn thiện hơn.

TpHCM, tháng 03 năm 2016

Dương Quốc Thắng

TÓM TẮT

Sự phát triển nhanh chóng của công nghệ thông tin đã ảnh hưởng rất lớn đến nhiều lĩnh vực. Trong số đó, có thể kể đến sự bùng nổ của công nghệ World Wide Web, do những lợi ích của nó mang lại nên nhu cầu của nó ngày càng phổ biến. Phần lớn các trang Web có thể được truy cập hàng ngàn 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 cho chiến lược quan trọng trong mô hình thương mại của các doanh nghiệp hiện tại. Các thông tin này thường được lưu trữ trong Web log. Chính vì vậy, khai thác tri thức từ Web log để quyết định đúng đắn và đáp ứng kịp thời 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.

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 chuỗi 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.

Kỹ thuật khai thác dữ liệu thông thường được đề xuất là không hiệu quả vì chúng cần phải được tái thực hiện mỗi lần thay đổi truy cập và cũng đòi hỏi nhiều lần quét cơ sở dữ liệu. Khai thác mẫu chuỗi là quá trình áp dụng các kỹ thuật khai thác dữ liệu vào một cơ sở dữ liệu cho các mục đích phát hiện các mối quan hệ tương quan tồn tại giữa một danh sách có thứ tự các sự kiện. Nhiệm vụ khám phá mẫu chuỗi phổ biến là một thách thức bởi vì các thuật toán cần xử lý một số tổ hợp của các trình tự.

Trong luận văn này, các thuật toán khai thác mẫu chuỗi phổ biến được thực hiện. Từ đó trích xuất luật và điều này được thử nghiệm trên dữ liệu nhật ký Web. Các kết quả thực nghiệm chứng minh cho tính hiệu quả được đưa ra trong luận văn này.

ABSTRACT

The rapid development of information technology has a great influence to many areas. Among them, it is possible to observe the explosion of the World Wide Web technology. Since the benefits of it, its demand increasingly popular. Most Web sites can be accessed thousands of times each day. The problem is how to collect this information in order to analyze what users saw, or searched to be able to valued strategic business models for existing enterprises. Such data is normally stored in the Web log. Hence, mining knowledge from Web logs for proper decisions and instance responses will serve these organizations in making business decisions, improvements, and design Web pages to achieve a new pinnacle in e-commerce.

Discover hidden information from the Web log data is called mining Web usage behavior. The purpose of the discovery of common patterns in the data string Web log is to get information about the access behavior of users for the purpose of predicting and prefetching of Web sites that the user has the ability access.

Data mining techniques are generally ineffective proposal because they need to be re-done each time changing access and also requires a lot of database scans. Exploitation is the process chain template to apply data mining techniques into a database for the purpose of detecting the correlation relationship exists between an ordered list of events. Tasks explore popular chain form is a challenge because the algorithm needs to handle a number of combinations of sequences.

In this thesis, the algorithms exploit popular chain pattern is done. From this extract and this law is tested on Web log data. The experimental results demonstrate the effectiveness is given in this thesis.

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT, KÝ HIỆU

Từ viết tắt	Tiếng Anh	Nghĩa tiếng Việt
CSDL	Database(s)	Cơ sở dữ liệu
Sfs	Frequent Sequence	Chuỗi phổ biến
Sfcs	Closed Frequent Sequence	Chuỗi phổ biến đóng
I	Items	Tập các item
minsup	Minimum support	Độ phổ biến tối thiểu
minconf	Minimum confidence	Độ tin cậy tối thiểu
minsup_count	Minimum support count	Độ phổ biến tối thiểu (tính theo số đếm)

DANH MỤC CÁC BẢNG

DANH MỤC CÁC BIỂU ĐỒ, ĐỒ THỊ, SƠ ĐỒ, HÌNH ẢNH

MỞ ĐẦU

1. Lý do chọn đề tài

Ngày nay, các ứng dụng về công nghệ thông tin đều phát tiển trên nền Web cùng với sự bùng nổ của công nghệ, truyền thông, v.v… Công nghệ Web sẽ toàn cầu hóa hầu hết trong các lĩnh vực đời sống: kinh doanh - thương mại, y tế, khoa học, giáo dục, v.v… Chính vì thế, việc sử dụng các trang Web, số lượng duyệt Web, số lần giao dịch và truy cập vào các ứng dụng Web ngày càng gia tăng dẫn đến tình trạng khó khăn cho các nhà cung cấp và phát triển dịch vụ Web: nghẽn mạng, tốn nhiều không gian, chiếm nhiều bộ nhớ server, chi phí cao; mất nhiều thời gian sử dụng Web của người dùng vì thông tin bị trùng lắp, dư thừa, v.v… làm mất đi thói quen sử dụng những trang Web mặc dù đã nhiều lần truy cập trước đây.

Vấn đề đặt ra, làm sao giải quyết các vấn đề này nhằm giảm chi phí, tốn kém cho các nhà cung cấp dịch vụ; tối ưu các hóa tiện ích của Web, quảng bá tốt hơn nhằm tăng doanh số, doanh thu cho các tổ chức, cá nhân sử dụng dịch vụ Web. Đặt biệt là thể hiện tính tiện dụng cao, phù hợp với sở thích, thói quen sử dụng Web của người dùng. Vì vậy chọn đề tài "**Ứng dụng khai thác mẫu chuỗi để khai thác hành vi sử dụng web**".

2. Mục tiêu đề tài

Nghiên cứu cơ sở lý thuyết các kỹ thuật khai thác dữ liệu, kỹ thuật thu thập thông tin người dùng truy cập trên Web. Cụ thể là khai thác dữ liệu mẫu chuỗi (sequence database) và xây dựng công cụ hỗ trợ trong việc khai thác hành vi sử dụng Web của người dùng dựa trên thông tin của Web log đối với những trang Web thương mại điện tử.

3. Phạm vi nghiên cứu đề tài

Vì tầm quan trọng của một số ứng dụng khai thác mẫu chuỗi duyệt web, nhiều thuật toán đã được đề xuất trong lĩnh vực khai thác mẫu chuỗi trong thập kỷ qua; hầu hết các thuật toán đều tập trung cải tiến để hỗ trợ tìm kiếm các chuỗi cô đọng hơn như chuỗi đóng, chuỗi cực đại, chuỗi tăng cường, chuỗi phân cấp, chuỗi tuần tự, chuỗi tuần hoàn, chuỗi có thứ tự bộ phận, chuỗi chuỗi sinh học xấp xỉ.

Luận văn này tập trung nghiên cứu giải pháp cho sự kết hợp của bit vectơ động cho khai thác chuỗi phổ biến đóng. Luận văn khảo sát các thuật toán đã có bằng cách đưa ra một nguyên tắc phân loại để phân lớp các thuật toán khai thác mẫu chuỗi dựa trên các đặc trưng quan trọng chủ yếu của các kỹ thuật. Việc phân lớp này nhằm mục đích làm rõ bài toán khai thác mẫu chuỗi, thực trạng hiện tại của các giải pháp đã có và hướng nghiên cứu trong lĩnh vực này. Luận văn cũng đưa ra phân tích kết quả thực hiện của nhiều kỹ thuật chủ chốt, đặc biệt là kỹ thuật khai thác mẫu chuỗi và thảo luận các khía cạnh về mặt lý thuyết của lĩnh vực này, sau đó ứng dụng các kết quả đã chứng minh vào khai thác hành vi sử dụng Web.

Dựa trên một số công trình nghiên cứu trong lĩnh vực khai thác mẫu chuỗi đã công bố trong những năm gần đây, từ đó luận văn trình bày:

- Phương pháp khai thác mẫu chuỗi từ dữ liệu chuỗi. Sự kết hợp của bit vectơ động cho khai thác chuỗi phổ biến đóng.
- Luật: Ý nghĩa luật, phát biểu bài toán và các hướng tiếp cận thuật toán khai thác luật..
- Web log: Ý nghĩa Web log, cách thu thập thông tin, phương pháp tiền xử lý và phân tích Web log thành cơ sở dữ liệu thực nghiệm, từ item đơn thành itemset theo từng Session của người dùng. Ứng dụng thuật toán khai thác mẫu chuỗi và luật vào khai thác Web log nhằm đưa ra hành vi người sử dụng.
- Xây dựng tập cơ sở dữ liệu thực nghiệm, so sánh các kết quả đạt được và đánh giá hiệu quả của ứng dụng.

4. Bố cục đề tài

Chương 1: Giới thiệu tổng quan

Chương 2: Cơ sở lý thuyết khai thác mẫu chuỗi và khai thác luật

Chương 3: Ứng dụng luật vào khai thác hành vi sử dụng Web

Chương 4: Thực nghiệm, kết luận và hướng phát triển

Luận văn trình bày trong 4 chương. Chương một trình bày tổng quan về CSDL chuỗi, khái quát về lĩnh vực khai thác mẫu và luật trên CSDL chuỗi. Chương

này cung cấp một cái nhìn chung nhất về lĩnh vực khai thác dữ liệu trên CSDL chuỗi.

Chương hai trình bày bài toán về khai thác mẫu chuỗi. Trong đó, luận văn mô tả chi tiết thuật toán kết hợp của bit vectơ động cho khai thác chuỗi phổ biến đóng, là thuật toán được chọn cho khai thác mẫu chuỗi. Cuối cùng trình bày cơ sở lý thuyết về khai thác luật .

Chương ba trình bày tổng quan về khai thác Web, lý do vì sao chọn khai thác sử dụng Web. Sau đó, ứng dụng luật đã nghiên cứu vào khai thác hành vi sử dụng Web.

Chương bốn trình bày những kết quả thực nghiệm, kết luận của luận văn và hướng phát triển trong tương lai.

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN

1.1. Giới thiệu về khai thác dữ liệu (data mining)

1.1.1. Tại sao phải khai thác dữ liệu

Ngày nay lượng thông tin được lưu trữ trên các thiết bị điện tử (đĩa cứng, CD-ROM, băng từ, v.v…) không ngừng tăng lên. Sự tích lũy dữ liệu này xảy ra với một tốc độ bùng nổ. Người ta ước đoán rằng lượng thông tin trên toàn cầu tăng gấp đôi sau khoảng hai năm và theo đó số lượng cũng như kích cỡ của các cơ sở dữ liệu (CSDL) cũng tăng lên một cách nhanh chóng. Lượng dữ liệu đang ngày càng tăng lên khiến cho chúng ta bị ngập trong khối dữ liệu khổng lồ đó. Câu hỏi đặt ra là liệu chúng ta có thể khai thác được gì từ những dữ liệu thực sự có giá trị thì lại nằm trong chính khối dữ liệu đó?

Data Mining ra đời như một hướng giải quyết hữu hiệu cho câu hỏi vừa đặt ra ở trên. Khá nhiều định nghĩa về Data Mining, tuy nhiên có thể tạm hiểu rằng Data Mining như là một *công nghệ tri thức* giúp khai thác những thông tin hữu ích từ những kho dữ liệu được tích trữ trong suốt quá trình hoạt động của một công ty, tổ chức nào đó. Do vậy, khai phá dữ liệu (Data mining) ra đời để giúp ta chắt lọc được những thông tin có giá trị từ những khối dữ liệu thô khổng lồ ta nhận được.

1.1.2. Khai thác dữ liệu là gì ?

Khai thác dữ liệu được định nghĩa như là một quá trình chắt lọc hay khai thác tri thức từ một lượng lớn dữ liệu. Một ví dụ hay được sử dụng là là việc khai thác vàng từ đá và cát, data mining được ví như công việc "Đãi cát tìm vàng" trong một tập hợp lớn các dữ liệu cho trước. Thuật ngữ khai thác dữ liệu ám chỉ việc tìm kiếm một tập hợp nhỏ có giá trị từ một số lượng lớn các dữ liệu thô. Có nhiều thuật ngữ hiện được dùng cũng có nghĩa tương tự với từ khai thác dữ liệu như khai thác tri thức (knowledge mining), chắt lọc tri thức (knowledge extraction) , phân tích dữ liệu/mẫu (data/patern analysis) , khảo cổ dữ liệu (data archaeoloogy) , nạo vét dữ liệu (datadredging) ,v.v…

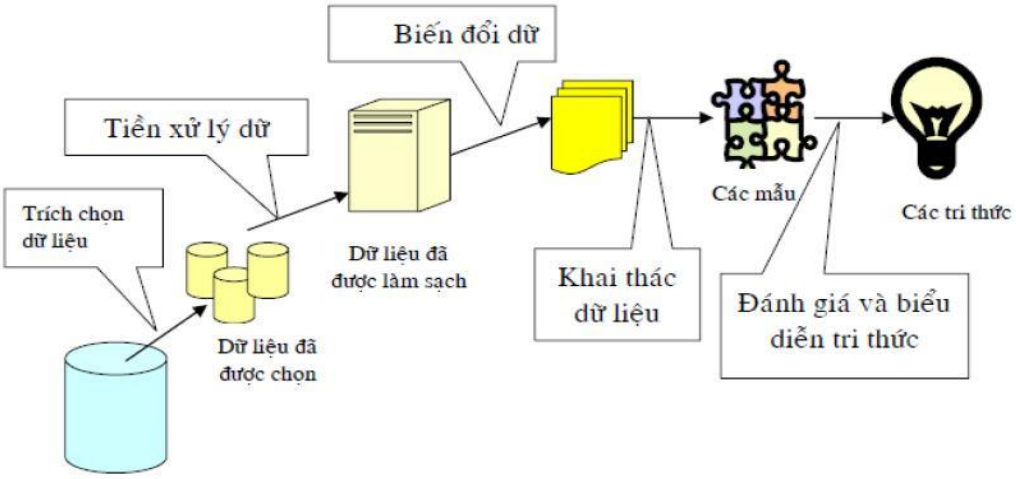
Khai thác dữ liệu được gọi là khám phá tri thức trong cơ sở dữ liệu. Đây là tiến trình của khám phá những mẫu hay tri thức có ích từ nguồn dữ liệu, như là cơ

sở dữ liệu, văn bản, ảnh, Web, v.v… Các mẫu phải có giá trị, có khả năng hữu ích và dễ hiểu. Khai thác dữ liệu là một lĩnh vực đa ngành liên quan đến máy học, thống kê, cơ sở dữ liệu, trí tuệ nhân tạo, thu thập thông tin, và mô phỏng trực quan.

Định nghĩa: Khai thác dữ liệu là một tập hợp các kỹ thuật được sử dụng để tự động khai thác và tìm ra các mối quan hệ lẫn nhau của dữ liệu trong một tập hợp dữ liệu khổng lồ và phức tạp, đồng thời cũng tìm ra các mẫu tiềm ẩn trong tập dữ liệu đó.

1.1.3. Quy trình phát hiện tri thức và khai thác dữ liệu

Khai thác dữ liệu là một bước trong bảy bước của quá trình KDD (Knowleadge Discovery in Database) và KDD được xem như các quá trình khác nhau theo thứ tự sau:



Hình 1. 1 - Quy trình phát hiện tri thức và khai thác dữ liệu [1],[2]

Bắt đầu của quá trình là kho dữ liệu thô và kết thúc với tri thức được chiết xuất ra. Về lý thuyết thì có vẽ rất đơn giản nhưng thực sự đây là một quá trình rất khó khăn gặp phải rất nhiều vướng mắc như: quản lý các tập dữ liệu, phải lặp đi lặp lại toàn bộ quá trình, v.v…

❖ Gom dữ liệu (Gathering): tập hợp dữ liệu là bước đầu tiên trong quá trình khai thác dữ liệu. Đây là bước được khai thác trong một cơ sở dữ liệu, một kho dữ liệu và thậm chí các dữ liệu từ các nguồn ứng dụng Web.

❖ Trích lọc dữ liệu (Selection**):** ở giai đoạn này dữ liệu được lựa chọn hoặc

phân chia theo một số tiêu chuẩn nào đó, trích chọn dữ liệu từ những kho dữ liệu và sau đó chuyển đổi về dạng thích hợp cho quá trình khai thác tri thức. Quá trình này bao gồm cả việc xử lý với dữ liệu nhiễu (noisy data), dữ liệu không đầy đủ (incomplete data), v.v… Ví dụ chọn ra những thông tin duyệt web của người dùng được lưu tại Web log mà những thông tin đó cho biết được người dùng đã xem một sản phẩm cụ thể nào chứ không phải là người dùng đã thao tác khác (xem hoặc lưu hình ảnh; truy cập trang không tồn tại; v.v…)

❖ Làm sạch, tiền xử lý và chuẩn bị trước dữ liệu (Cleaning, Pre-processing and Preparation): giai đoan thứ ba này là giai đoạn hay bị sao lãng, nhưng thực tế nó là một bước rất quan trọng trong quá trình khai thác dữ liệu. Một số lỗi thường mắc phải trong khi gom dữ liệu là tính không đủ chặt chẽ, logic. Vì vậy, dữ liệu thường chứa các giá trị vô nghĩa và không có khả năng kết nối dữ liệu. Ví dụ: trong Weblog, cần loại bỏ các liên kết mà người dùng truy cập mà không tồn tại.

Giai đoạn này sẽ tiến hành xử lý những dạng dữ liệu không chặt chẽ nói trên. Những dữ liệu dạng này được xem như thông tin dư thừa, không có giá trị. Bởi vậy, đây là một quá trình rất quan trọng vì dữ liệu này nếu không được "làm sạch - tiền xử lý - chuẩn bị trước" thì sẽ gây nên những kết quả sai lệch nghiêm trọng.

❖ Chuyển đổi dữ liệu (Transformation): tiếp theo là giai đoạn chuyển đổi dữ liệu, dữ liệu đưa ra có thể sử dụng và điều khiển được bởi việc tổ chức lại nó. Dữ liệu đã được chuyển đổi phù hợp với mục đích khai thác.

❖ Khai thác dữ liệu (Data mining): đây là giai đoạn quan trọng và tốn nhiều chi phí nhất của quá trình khai thác tri thức. Xác định nhiệm vụ khai thác dữ liệu và lựa chọn kỹ thuật khai thác để thực hiện khai thác, phát sinh tập mẫu. Các mẫu này là nguồn tri thức thô. Trong giai đoạn này, có thể cần sự tương tác của người dùng để điều chỉnh và rút ra các thông tin cần thiết nhất. Các tri thức nhận được có thể được lưu lại và sử dụng lại.

❖ Diễn giải và đánh giá kết quả mẫu (Interpretation / Evaluation of Result): đây là giai đoạn cuối trong quá trình khai thác dữ liệu. Ở giai đoạn này, các mẫu dữ liệu được chiết xuất ra bởi phần mềm khai thác dữ liệu. Không phải bất cứ mẫu dữ liệu

nào cũng đều hữu ích, đôi khi nó còn bị sai lệch. Vì vậy, cần phải ưu tiên những tiêu chuẩn đánh giá để chiết xuất ra các tri thức (Knowledge).

Quá trình khai thác tri thức không chỉ là một quá trình tuần tự từ bước đầu tiên đến bước cuối cùng mà là một quá trình lặp và quay lui lại các bước đã qua. Trong khai thác tri thức thì có thể cần có sự tương tác của con người để điều chỉnh rút trích dữ liệu cần thiết.

1.1.4. Các kỹ thuật khai thác dữ liệu

Data mining được chia nhỏ thành một số hướng chính như sau:

Mô tả khái niệm (concept description): thiên về mô tả, tổng hợp và tóm tắt khái niệm. Ví dụ: tóm tắt văn bản.

Luật kết hợp (association rules): là dạng luật biểu diễn tri thứ ở dạng khá đơn giản. Ví dụ: "60 % nam giới vào siêu thị nếu mua bia thì có tới 80% trong số họ sẽ mua thêm thịt bò khô". Luật kết hợp được ứng dụng nhiều trong lĩnh vực kính doanh, y học …

Phân lớp và dự đoán (classification & prediction): xếp một đối tượng vào một trong những lớp đã biết trước. Ví dụ: phân lớp vùng địa lý theo dữ liệu thời tiết. Hướng tiếp cận này thường sử dụng một số kỹ thuật của machine learning như cây quyết định (decision tree), mạng nơron nhân tạo (neural network)… còn gọi phân lớp là học có giám sát (học có thầy).

Phân cụm (clustering): xếp các đối tượng theo từng cụm (số lượng cũng như tên của cụm chưa được biết trước. Người ta còn gọi phân cụm là học không giám sát (học không thầy).

Phân tích độ lệch (deviation analysis)*:* là kỹ thuật so sánh giá trị hiện tại với giá trị bình thường đã xác định trước để kiểm tra sự bất bình thường. Phân tích độ lệch là công cụ hữu dụng cho các ứng dụng bảo mật, trong đó nó cảnh báo người quản trị có sự thay đổi đột ngột trong việc sử dụng tài nguyên của một người dùng nào đó.

Khai thác chuỗi (sequential mining / temporal patterns): tương tự như khai thác luật kết hợp nhưng có thêm tính thứ tự và tính thời gian; khai thác các mẫu phổ

biến liên quan đến thời gian hoặc các sự kiện khác. Một luật mô tả mẫu tuần tự có dạng X→Y phản ánh sự xuất hiện của biến cố X sẽ dẫn đến việc xuất hiện biến cố Y kế tiếp. Khai thác tuần tự được sử dụng trong việc dự báo và chăm sóc khách hàng. Hướng tiếp cận này được ứng dụng nhiều trong lĩnh vực tài chính và thị trường chứng khoán vì nó có tính dự báo cáo.

Mô hình phụ thuộc (dependent modeling): mục đích là để mô tả các phụ thuộc quan trọng giữa các phần tử trong tập dữ liệu. Phụ thuộc vào ý nghĩa mà giá trị của một phần tử có thể được dự báo với vài giá trị của các phần tử đã biết trước (ví dụ: A→B, CF=0.93). Tập các phụ thuộc có quan hệ với nhau hình thành một đồ thị phụ thuộc.

Phân tích không gian phụ thuộc (spatial dependency analysis): khai thác các mẫu của dữ liệu từng phần trong hệ thống thông tin địa lý, các hệ sinh thái. Ví dụ "giá cây bonsai tại vị trí X thường giảm quý 1 và tăng vào quý 4 của năm".

Khai thác mẫu duyệt đường đi (discovering path traversal patterns): thỉnh thoảng các phụ thuộc giữa các phần tử có thể phù hợp với mô hình sử dụng đồ thị.

Một trong những ứng dụng tiêu biểu là khai thác các mẫu duyệt đường đi trong việc truy xuất web. Biết được các mẫu có thể giúp thiết kế các ứng dụng web tốt hơn.

1.1.5. Ứng dụng của khai thác dữ liệu

Khai thác dữ liệu tuy là một hướng tiếp cận mới nhưng thu hút được rất nhiều sự quan tâm của các nhà nghiên cứu và phát triển nhờ vào những ứng dụng thực tiễn của nó. Chúng ta có thể liệt kê ra đây một số ứng dụng điển hình:

Tài chính và thị trường chứng khoán: Phân tích tình hình tài chính và dự báo giá của các loại cổ phiếu trong thị trường chứng khoán. Danh mục giá, lãi suất, phát hiện gian lận.

Phân tích dữ liệu và ra quyết định: Phân tích dữ liệu từ tập thô để tìm tập phổ biến từ đó suy ra những quy luật cần thiết để hỗ trợ quá trình ra quyết định.

Khoa học xã hội: Phân tích dữ liệu nhân khẩu, dự báo kết quả bầu cử.

Thiên văn học: Phân tích ảnh vệ tinh.

Luật: Kiểm tra gian lận thuế, v.v…

Thị trường: Dự báo thị trường, xác định loại khách hàng và hàng hóa, các mẫu phổ biến có triển vọng.

Kỹ thuật: Khai thác mẫu mạch tích hợp, dự báo các khả năng lỗi của hệ thống thiết bị.

Nông nghiệp: Phân loại bệnh của cây trồng.

Xuất bản: Khai khá profile của đọc giả để định hướng xuất bản.

Điều trị y học và chăm sóc y tế: Một số thông tin về chuẩn đoán bệnh được lưu trong các hệ database bệnh án. Phân tích mối liên hệ giữa triệu chứng bệnh, chuẩn đoán và phương pháp điều trị.

Trong lĩnh vực mạng máy tính và truyền thông: Khai thác dữ liệu cũng được ứng dụng rộng rãi trong các hệ thống phát hiện xâm nhập, các hệ thống thu thập thông tin và đặt biệt một số ứng dụng khai thác tri thức còn được sử dụng trong một số phần mềm phân tích các hệ thống mạng viễn thông.

1. 2. Tổng quan về cơ sở dữ liệu chuỗi

1.2.1. Các khái niệm về chuỗi dữ liệu

Cho tập I = {i_1, i_2, …, i_m} gồm m phần tử còn gọi là các item. Một itemset là tập không có thứ tự khác rỗng, gồm các item. Itemset i ký hiệu là (i_1, i_2, …, i_k) với mỗi i_j là một item. Itemset có lực lượng là k ký hiệu là k-itemset. Không mất tính tổng quát, giả sử các item trong itemset được sắp theo thứ tự tăng dần.

Một chuỗi (sequence) là một danh sách có thứ tự những itemset. Chuỗi s được ký hiệu là s_1 s_2… s_n hoặc $s_1 \rightarrow s_2 \rightarrow \ldots \rightarrow s_n$ với mỗi s_i là một itemset, n là số lượng itemset. Kích thước của chuỗi bằng số lượng itemset có trong chuỗi. Chiều dài của chuỗi là tổng số item có trong chuỗi, ký hiệu là $= \sum_{J=1}^{n} S_i$. Chuỗi có chiều dài còn được gọi là k-sequence. Ví dụ, s = (B)(AC) là một 3-sequence có kích thước là 2.

Chuỗi $\beta = b_1 b_2 \ldots b_m$ được gọi là chuỗi con của chuỗi $\alpha = a_1 a_2 \ldots a_n$ hay α là chuỗi cha của β, ký hiệu $\beta \subseteq \alpha$, nếu tồn tại những số nguyên $1 \leq j_1 < j_2 < \ldots < j_n \leq m$ sao cho $b_1 \subseteq a_{j1}$, $b_2 \subseteq a_{j2}$, …, $b_m \subseteq a_{jm}$. Ví dụ chuỗi (B)(AC) là chuỗi con của

(AB)(E)(ACD) ; nhưng (AB)(E) không phải là chuỗi con của chuỗi (ABE) và ngược lại.

Cơ sở dữ liệu chuỗi (sequence database): Cơ sở dữ liệu chuỗi là một tập hợp các bộ dữ liệu có dạng (sid, s), trong đó sid là định danh của chuỗi và s là chuỗi các itemset.

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

Độ hỗ trợ (support): Cho CSDL chuỗi D, mỗi chuỗi có một chỉ số định danh duy nhất. Độ hỗ trợ tuyệt đối của một mẫu tuần tự f là tổng số chuỗi trong D có chứa f, ký hiệu sup_D ()f = |{ $S_i \in D$ |f $\subseteq S_i$ }|. Độ hỗ trợ tương đối của f là tỉ lệ phần trăm chuỗi trong D chứa f. Ở đây, mức hỗ trợ tuyệt đối sẽ được sử dụng chuyển đổi qua lại, ký hiệu là sup(f).

Mẫu tuần tự (sequence pattern): Cho trước ngưỡng hỗ trợ tối thiểu (minsup) xác định bởi người dùng, minsup $\in (0,1]$ Một mẫu f được coi là phổ biến nếu độ hỗ trợ của nó lớn hơn hoặc bằng minsup: sup(f) $\geq$ minsup, khi đó f được gọi là mẫu tuần tự phổ biến.

Ví dụ: Cho CSDL như bảng 1.1 có tập các item phân biệt là {A, B,C} và minsup tuyệt đối là 2. Xét chuỗi s_1 = (AB)(B)(B)(AB)(B)(AC), chuỗi s_1 có 6 itemset là: (AB), (B), (B), (AB), (B), (AC) và có 9 item. Vậy s_1 có kích thước là 6 và có độ dài là 9. Trong chuỗi s_1, item A xuất hiện ba lần nhưng nếu tính độ hỗ trợ thì độ hỗ trợ của item A chỉ được tính là 1 đối với chuỗi s_1 đó. Chuỗi p = (AB)(C) là một chuỗi con của chuỗi s_1, vì vậy chuỗi con p còn được gọi là mẫu. Trong CSDL, chỉ có chuỗi s_1, s_2 và s5 có chứa mẫu p, vậy độ hỗ trợ của mẫu p là 3. Vì sup(p) > minsup nên p là một mẫu chuỗi

Bảng 1. 1 - CSDL Chuỗi

SID	Chuỗi dữ liệu
1	(AB)BB(AB)B(AC)
2	(AB)(BC)(BC)
3	(B)(AB)

4	BB(BC)
5	(AB)(AB)(AB)A(BC)

Khai thác mẫu chuỗi (sequence pattern mining)**:** Cho trước CSDL chuỗi và ngưỡng minsup. Khai thác mẫu tuần tự là đi tìm tập đầy đủ tất cả các mẫu tuần tự có trong CSDL.

Tiền tố, hậu tố, tiền tố không hoàn toàn:

Cho hai chuỗi dữ liệu $\alpha = a_1 a_2 \dots a_n$ và $\beta = b_1 b_2 \dots b_m$ $(m \le n)$, (trong đó *ai*, *bi* là các itemset). βđược gọi là tiền tố của α nếu và chỉ nếu $b_i = a_i$ với mọi $1 \le I \le m$. Sau khi loại bỏ phần tiền tố β trên chuỗi α, phần chuỗi còn lại được gọi là hậu tố của α. Chuỗi β được gọi là tiền tố không hoàn toàn của chuỗi α nếu và chỉ nếu $b_i = a_i$ với mọi $1 \le i \le m-1$, $b_m \subset a_m$ và tất cả các item trong tập $(a_m - b_m)$ đều là những item đứng sau các item trong *bm* xét theo thứ tự từ điển.

Từ định nghĩa trên, ta thấy rằng nếu một chuỗi có kích thước *k* sẽ có *(k-1)* tiền tố. Ví dụ, chuỗi A(BC)D có 2 tiền tố là A và A(BC). Do đó, (BC)D là hậu tố đối với tiền tố A và D là hậu tố đối với tiền tố A(BC). Hai chuỗi AB và (BC) không phải là tiền tố của chuỗi đã cho; tuy nhiên chuỗi AB) là một tiền tố không hoàn toàn.

1.2.2. Đặc điểm dữ liệu chuỗi

Dữ liệu chuỗi có một số đặc điểm riêng biệt so với các loại dữ liệu khác. Do đó, khai thác dữ liệu chuỗi đặt ra nhiều cơ hội và thách thức, thu hút nhiều quan tâm nghiên cứu. Dữ liệu chuỗi có đặc điểm như sau:

Kích thước chuỗi có thể rất dài. Trong cùng một CSDL, kích thước của mỗi chuỗi là khác nhau thậm chí có sự chênh lệch lớn. Ví dụ các chuỗi gen có độ dài tối thiểu là vài trăm nhưng độ dài tối đa lên đến hàng trăm nghìn.

Một mẫu là một chuỗi con, nghĩa là các thành phần trong chuỗi con phải liên tục kề nhau trong chuỗi cha ban đầu. Một mẫu cũng có thể là một tập hợp con của chuỗi, các thành phần của mẫu không liên tục trong chuỗi cha.

Vị trí tuyệt đối của các thành phần trong chuỗi thường không quan trọng. Chẳng hạn, khi cần kiểm tra một chuỗi dữ liệu có chứa một mẫu hay không thì

không cần quan tâm mẫu đó xuất hiện trong chuỗi ở vị trí tuyệt đối nào.

Mối quan hệ về thứ tự/vị trí giữa các thành phần trong chuỗi đóng vai trò quan trọng. Ví dụ chuỗi XY hoàn toàn khác với chuỗi YX. Hơn nữa, khoảng cách giữa hai thành phần trong chuỗi cũng có ý nghĩa. Mối quan hệ về thứ tự/vị trí giữa các thành phần trong chuỗi là đặc điểm duy nhất chỉ có ở dữ liệu chuỗi. Đây chính là điểm khác biệt cơ bản của dữ liệu chuỗi so với các loại dữ liệu khác.

1.2.3. Một số ví dụ về dữ liệu chuỗi

Chuỗi dữ liệu sinh học: DNA, RNA và Protein

Chuỗi dữ liệu sinh học giúp chúng ta hiểu rõ về cấu trúc và chức năng của các loại tế bào khác nhau, hỗ trợ cho việc chẩn đoán và chữa bệnh. Có ba loại chuỗi sinh học là chuỗi deoxyribonucleic acid (DNA), chuỗi amino acid (hay còn gọi là Protein) và ribonucleic acid (RNA). Hình 1.2 và 1.3 minh họa một phần của chuỗi DNA và một phần của chuỗi protein.

```
GAATTCTCTGTAACACTAAGCTCTCTTCCTCAAAACCAGAGGTAGAT
AGAATGTAATAATTTACAGAATTTCTAGACTTCAACGATCTGATTTTT
```

Hình 1. 2 - Một phân đoạn chuỗi AND [25]

```
SSQIRQNYSTEVEAAVNRLVNLYLRASYTYLSLGFYFDRDDVALEGVCH
EFREAEEKREGAERLLKMQNQRGGRALFQDLQKPSQDEWGTTPDAMK
```

Hình 1. 3 - Một phân đoạn chuỗi Protein [25]

Một số bài toán phân tích dữ liệu trên chuỗi sinh học thường gặp là: Phân tích cấu trúc và chức năng của protein từ chuỗi protein, xác định đặc điểm của mẫu trong họ các chuỗi DNA, RNA hay protein, so sánh họ các chuỗi với nhau, v.v…

Chuỗi sự kiện:

Chuỗi lịch sử bán hàng, chuỗi lịch sử mua sắm của khách hàng, chuỗi vết hệ thống, chuỗi truy cập Web, v.v… Chiếm phần lớn các loại chuỗi là chuỗi sự kiện. Từ những chuỗi như vậy, có thể hiểu được cách thức các đối tượng hoạt động như thế nào, từ đó rút ra cách tốt nhất để giải quyết chúng. Sau đây là một số ví dụ về chuỗi sự kiện.

Chuỗi truy cập Web là một chuỗi các cặp gồm định danh người dùng và sự kiện. Một sự kiện là một yêu cầu về một tài nguyên web chẳng hạn như một trang web hay một dịch vụ. Với mỗi trang được yêu cầu, một số thông tin truyền thống là có sẵn, ví dụ như kiểu và nội dung của trang, lượng thời gian người dùng đã tiêu tốn trên trang, lượng thời gian người dùng đã tiêu tốn trên trang đó. Các sự kiện của một mẫu truy cập web được liệt kê dưới dạng danh sách tăng dần theo thời gian.

Hình 1.4 minh họa một chuỗi truy cập web, trong đó *a, b, c, d, e* là các sự kiện và *100, 200, 300* và *400* là các định danh của người dùng.

⟨100, a⟩, ⟨100, b⟩, ⟨200, a⟩, ⟨300, b⟩⟨400, a⟩, ⟨100, a⟩, ⟨400, b⟩, ⟨300, a⟩, ⟨100, c⟩, ⟨200, c⟩, ⟨400, a⟩, ⟨400, e⟩

Chuỗi vết hệ thống cũng tương tự như chuỗi truy cập web. Chúng là chuỗi các bản ghi ghi nhận các hoạt động được thực hiện bởi người dùng hoặc tiến trình, trên những loại dữ liệu và tài nguyên khác nhau trong một hay nhiều hệ thống.

Chuỗi lịch sử mua sắm của khách hàng là những chuỗi các bộ, trong đó mỗi bộ gồm định danh khách hàng, địa điểm, thời gian và tập các mặt hàng đã mua… Hình 1.5 minh họa chuỗi các lần mua sắm của một khách hàng có định danh là 200

⟨200, 25/08/15, 10am, CentralStation, {Beef, Chicken,, Milk}⟩,
⟨200, 25/08/15, 11am, CentralStation, {Burger, Pepsi, Banana}⟩,

Hình 1. 5 - Chuỗi các lần mua sắm của một khách hàng [25]

Chuỗi lịch sử bán hàng là chuỗi các bộ, mỗi bộ gồm định danh cửa hàng, thời gian, tổng số các mặt hàng đã bán và doanh thu tương ứng tại thời gian đó và một số thông tin khác.

⟨97100, 05/06, {⟨Apple : $85K⟩, ⟨Bread : $100K⟩, ⟨Cereal : $150K⟩, …}⟩,
⟨90089, 05/06, {⟨Apple : $65K⟩, ⟨Bread : $105K⟩, ⟨Diaper : $20K⟩, …}⟩,

Hình 1. 6 - Chuỗi lịch sử bán hàng của các cửa hàng

1.2.4. Các kỹ thuật khai thác dữ liệu chuỗi

Khai thác dữ liệu chuỗi đóng vai trò rất quan trọng trong đời sống hằng ngày. Ứng dụng vào các lĩnh vực khai thác thói quen sử dụng Web, phân tích thói quen

mua sắm khách hàng, chuẩn đoán bệnh, truy vết hệ thống, v.v… Các thuật toán liên quan đã được đề xuất nhằm giải quyết nhu cầu này. Xuất phát từ thuật toán ApioriAll đến nay đã có nhiều nghiên cứu khai thác cơ sở dữ liệu chuỗi tuần tự.

Khai thác dữ liệu có thể được thực hiện trong ba bước chính:

(1) Thu thập dữ liệu và tiền xử lý,

(2) Khai thác dữ liệu

(3) Rút trích ra thông tin hữu ích theo nhu cầu.

Quá trình khai thác dữ liệu được lặp nhiều lần để đạt được các kết quả khả quan cuối cùng. Dữ liệu thường được lưu trữ trong các bảng quan hệ, bảng tính, hoặc các tập tin dạng bảng.

Khai thác dữ liệu phụ thuộc vào loại tri thức mà hệ thống khai thác tri thức và khai thác dữ liệu tìm kiếm. Mỗi nhiệm vụ khai thác dữ liệu có đặc tính riêng của nó và thực hiện theo các bước trong quá trình khai thác tri thức. Sau đây là các nhiệm vụ khai thác dữ liệu thường được sử dụng phổ biến trong ứng dụng khai thác dữ liệu chuỗi [26].

Khai thác chuỗi con phổ biến hay còn gọi là khai thác mẫu tuần tự (mining frequent subsequence hoặc mining sequential pattern).

Khai thác mẫu tuần tự là khai thác các mẫu phổ biến liên quan đến thời gian hoặc các sự kiện khác, với yêu cầu là các mẫu phổ biến là những chuỗi con trong CSDL chuỗi mà sự xuất hiện của chúng lớn hơn ngưỡng hỗ trợ do người dùng chỉ ra.

Phân lớp các chuỗi *(classification)*

Khai thác có hay không một phần tử thuộc về một trong các lớp đã biết trước. Vấn đề là phải xác định các lớp như thế nào. Trong thực tế, các lớp thường được xác định dựa trên giá trị của trường nào đó trong mẫu tin hoặc dẫn xuất của các giá trị khác nhau trong các trường.

Phân cụm các chuỗi (cluster identification)

Sắp xếp các đối tượng theo từng cụm. Ngược với lớp, số lượng và tên của cụm chưa được biết trước. Khi xác định các cụm, các độ đo khoảng cách được sử

dụng để tính toán sao cho mức độ tương tự giữa các đối tượng trong cùng một cụm là lớn nhất và mức độ tương tự giữa các đối tượng nằm trong các cụm khác nhau là nhỏ nhất.

Khai thác luật (mining rules)

Khai thác luật là quá trình tìm kiếm những mối quan hệ theo thời gian giữa các sự kiện tuần tự. Một luật mô tả mẫu tuần tự có dạng X→Y phản ánh sự xuất hiện của biến cố X sẽ dẫn đến sự xuất hiện của biến cố Y kế tiếp

1. 3. Khai thác luật trên cơ sở dữ liệu chuỗi

Trên CSDL chuỗi, đã có nhiều nghiên cứu trên các loại luật khác nhau: luật tuần tự (sequential rules), luật thú vị (interesting rules), luật phân lớp tuần tự (sequential classification rules), luật tuần hoàn (recurrent rules).

Luật tuần tự [4], [5] mở rộng khả năng sử dụng và tăng cường ý nghĩa của mẫu chuỗi. Một luật khai thác được sẽ biểu diễn ràng buộc là: trong một chuỗi sự kiện, những sự kiện xảy ra trước sẽ được theo sau bởi một loạt sự kiện khác. Ngoài ra, độ thú vị của một luật được đo bởi cả hai yếu tố là độ hỗ trợ và độ tin cậy. Độ tin cậy là một độ đo hữu ích, đặc biệt là khi ngưỡng hỗ trợ tối thiểu có giá trị thấp. Do đó, luật tuần tự đặc biệt hữu dụng cho việc dò và lọc ra những dị thường mà vi phạm các ràng buộc. Chính vì vậy, chúng được ứng dụng rộng rãi trong việc dò lỗi, phát hiện xâm nhập và bẫy lỗi… Bên cạnh đó, các nghiên cứu cũng cho thấy rằng luật tuần tự cũng rất có ích trong lĩnh vực y dược và công nghệ phần mềm.

Luật thú vị [7] tương tự như luật tuần tự, nó cũng biểu diễn ràng buộc giữa các loạt sự kiện theo thời gian. Độ thú vị của luật cũng được đo bởi hai yếu tố là độ hỗ trợ và độ tin cậy, tuy nhiên bổ sung thêm một độ đo mới, đó là độ tăng cường (improvement). Phân lớp là một kỹ thuật khai thác dữ liệu quan trọng, cho phép gán nhãn tự động cho một đối tượng dữ liệu vào một phân lớp đúng.

Luật phân lớp [5] là luật biểu thị dưới dạng X→c, với X là một chuỗi và c là nhãn của một lớp. Luật X→c là một luật phân lớp tuần tự trong CSDL chuỗi khi và chỉ khi tồn tại một chuỗi dữ liệu trong CSDL là chuỗi cha của X và chuỗi dữ liệu đó thuộc phân lớp có nhãn là c.

Luật tuần hoàn [22] là luật được tạo ra từ hai loại mẫu: mẫu tuần tự (frequent sequential pattern) và phân đoạn phổ biến (frequent episode). Phân đoạn (episode) được định nghĩa là mẫu gồm các sự kiện xuất hiện tương đối gần nhau trong chuỗi, tức là các sự kiện xuất hiện trong một giới hạn thời gian (time window). Luật tuần tự được sinh từ mẫu tuần tự. Một luật tuần tự X→Y phát biểu rằng khi một chuỗi trong CSDL là chuỗi cha của mẫu X thì nó cũng là chuỗi cha của mẫu gồm mẫu X nối với mẫu Y. Luật sinh từ phân đoạn gọi là luật phân đoạn. Một luật phân đoạn X→Y phát biểu rằng: khi một phân đoạn là chuỗi cha của mẫu X thì nó cũng là chuỗi cha của mẫu X nối với Y. Luật tuần hoàn khái quát cả hai loại luật – tuần tự và phân đoạn: khái quát luật tuần tự ở chỗ phần tiền kiện X và hậu kiện Y được xét có thể lấy từ cùng một chuỗi hoặc nhiều chuỗi khác nhau; đồng thời, luật tuần hoàn khái quát luật phân đoạn bằng cách cho phép phần tiền kiện X và hậu kiện Y được tách riêng bởi một số sự kiện tùy ý trong CSDL chuỗi.

Trong các loại luật trên, luật tuần tự là cơ bản nhất, các loại luật còn lại đều là dạng biến đổi của luật tuần tự bằng cách bổ sung thêm hoặc loại bỏ đi một số thông tin hoặc ràng buộc. Do đó, luận văn tập trung nghiên cứu trên bài toán khai thác luật tuần tự.

Khai thác luật là việc tìm kiếm những mối quan hệ theo thời gian giữa các sự kiện tuần tự. Một luật biểu diễn dưới dạng X→Y, nghĩa là nếu X có mặt trong một chuỗi bất kỳ của CSDL thì với một độ tin cậy cao có thể khẳng định Y cũng xuất hiện trong chuỗi đó theo sau X.

Tuy nhiên, nếu khai thác luật dựa trên những độ đo khác chẳng hạn như độ đo *lift* [22] hoặc độ đo *conviction* [23] thì cách tiếp cận giải quyết bài toán của David Lo và đồng sự không còn phù hợp. Hiện nay, chỉ có duy nhất một phương pháp cơ bản để khai thác tập luật tuần tự đầy đủ do Spiliopoulou đề xuất [6]. Từ những mô tả của phương pháp này, Lo cùng đồng sự đã khái quát thành thuật toán Full [5]. Đặc điểm cơ bản của thuật toán Full là dựa trên phương pháp vét cạn, do đó tốn nhiều chi phí tính toán và thực nghiệm trên dữ liệu Web log.

1. 4. Giới thiệu về khai thác Web (Web mining)

1.4.1. Nhu cầu

Sự phát triển nhanh chóng của mạng Internet đã sinh ra một khối lượng khổng lồ các dữ liệu dạng siêu văn bản (dữ liệu Web). Cùng với sự thay đổi và phát triển hàng ngày, hàng giờ về nội dung cũng như số lượng của các trang Web trên Internet thì vấn đề tìm kiếm thôn g tin đối với người sử dụng lại ngày càng khó khăn. Có thể nói nhu cầu tìm kiếm thông tin trên môt CSDL phi cấu trúc đã được phát triển chủ yếu cùng với sự phát triển của Internet. Thực vậy với Internet con người đã làm quen với các trang Web cùng với vô vàn các thông tin. Trong những năm gần đây Intrnet đã trở thành một trong những kênh về khoa học, thông tin kinh tế, thương mại và quảng cáo. Một trong những lý do cho sự phát triển này là sự thấp về giá cả tiêu tốn khi công khai một trang Web trên Internet. So sánh với những dịch vụ khác như mua bản hay quảng cáo trên một tờ báo hay tạp chí, thì một trang Web "đòi hỏi" rẻ hơn rất nhiều và cập nhật nhanh chóng hơn tới hàng triệu người dùng khắp mọi nơi trên thế giới. Có thể nói trang Web như là cuốn từ điển Bách khoa toàn thư. Thông tin trên các trang Web đa dạng về mặt nội dung cũng như hình thức. Có thể nói Internet như một xã hội ảo, nó bao gồm các thông tin về mọi mặt của đời sống kinh tế, xã hội được trình bày dưới dạng văn bản, hình ảnh, âm thanh, v.v…

Tuy nhiên cùng với sự đa dạng và số lượng lớn thông tin như vậy đã nảy sinh vấn đề quá tải thông tin.

Mặt khác, giả sử chúng ta có các trang Web về các vấn đề tin học, thể thao, kinh tế - xã hội và xây dựng, v.v… Căn cứ vào nội dung của các tài liệu mà khách hàng xem hoặc download về, sau khi phân lớp chúng ta sẽ biết khách hàng hay tập trung vào nội dung gì trên trang Web của chúng ta, từ đó chúng ta sẽ bổ sung thêm nhiều các tài liệu về các nội dung mà khách hàng quan tâm và ngược lại. Còn về phía khách hàng sau khi phân tích chúng ta cũng biết được khách hàng hay tập trung về vấn đề gì, để từ đó có thể đưa ra những hỗ trợ thêm cho khách hàng đó. Từ những nhu cầu thực tế trên, khai thác hành vi sử dụng của người dùng Web vẫn là

bài toán hay và cần phát triển nghiên cứu hiện nay.

1.4.2. Khó khăn [24]

Hệ thống phục vụ World Wide Web như là một hệ thống trung tâm rất lớn phân bố rộng cung cấp thông tin trên mọi lĩnh vực khoa học, xã hội, thương mại, văn hóa, v.v… Web là một nguồn tài nguyên giàu có cho khai thác dữ liệu. Những quan sát sau đây cho thấy Web đã đưa ra sự thách thức lớn cho công nghệ khai thác dữ liệu.

Web dường như quá lớn để tổ chức thành một kho dữ liệu phục vụ data mining

Các CSDL truyền thống thì có kích thước không lớn lắm và thường được lưu trữ ở một nơi, trong khi đó kích thước Web rất lớn, tới hàng terabyte và thay đổi liên tục, không những thế còn phân tán trên rất nhiều máy tính khắp nơi trên thế giới. Một vài nghiên cứu về kích thước của Web đã đưa ra các số liệu như sau: Hiện nay trên Internet có khoảng hơn một tỷ các trang Web được cung cấp cho người sử dụng, giả sử kích thước trung bình của mỗi trang là 5-10Kb thì tổng kích thước của nó ít nhất là khoảng 10 terabyte. Còn tỷ lệ tăng của các trang Web thì thật sự gây ấn tượng. Hai năm gần đây số các trang Web tăng gấp đôi và còng tiếp tục tăng trong hai năm tới. Nhiều tổ chức và xã hội đặt hầu hết những thông tin công cộng của họ lên Web. Như vậy việc xây dựng một kho dữ liệu (datawarehouse) để lưu trữ, sao chép hay tích hợp các dữ liệu trên Web là gần như không thể.

Độ phức tạp của trang Web lớn hơn rất nhiều so với những tài liệu truyền thống khác

Các dữ liệu trong các CSDL truyền thống thì thường là loại dữ liệu đồng nhất (về ngôn ngữ, định dạng,…), còn dữ liệu Web thì hoàn toàn không đồng nhất. Ví dụ về ngôn ngữ dữ liệu Web bao gồm rất nhiều loại ngôn ngữ khác nhau (Cả ngôn ngữ diễn tả nội dung lẫn ngôn ngữ lập trình), nhiều loại định dạng khác nhau (Text, HTML, PDF, hình ảnh âm thanh, v.v…), nhiều loại từ vựng khác nhau (địa chỉ Email, các liên kết (link), các mã nén (zipcode), số điện thoại)

Nói cách khác, trang Web thiếu một cấu trúc thống nhất. Chúng được coi như một thư viện kỹ thuật số rộng lớn, tuy nhiên con số khổng lồ các tài liệu trong

thư viện thì không được sắp xếp tuân theo một tiêu chuẩn đặc biệt nào, không theo phạm trù, tiêu đề, tác giả, số trang hay nội dung,... Điều này là một thử thách rất lớn cho việc tìm kiếm thông tin cần thiết trong một thư viện như thế.

Web là một nguồn tài nguyên thông tin có độ thay đổi cao

Web không chỉ có thay đổi về độ lớn mà thông tin trong chính các trang Web cũng được cập nhật liên tục. Theo kết quả nghiên cứu, hơn 500.000 trang Web trong hơn 4 tháng thì 23% các trang thay đổi hàng ngày, và khoảng hơn 10 ngày thì 50% các trang trong tên miền đó biến mất, nghĩa là địa chỉ URL của nó không còn tồn tại nữa. Tin tức, thị trường chứng khoán, các công ty quản cáo và trung tâm phục vụ Web phổ biến cập nhật trang Web của họ. Thêm vào đó sự kết nối thông tin và sự truy cập bản ghi cũng được cập nhật.

Web phục vụ một cộng đồng người dùng rộng lớn và đa dạng

Internet hiện nay nối với khoảng 50 trạm làm việc, và cộng đồng người dùng vẫn đang nhanh chóng lan rộng. Mỗi người dùng có một kiến thức, mối quan tâm, sở thích khác nhau. Nhưng hầu hết người dùng không có kiến thức tốt về cấu trúc mạng thông tin, hoặc không có ý thức cho những tìm kiếm, rất dễ bị "lạc" khi đang "mò mẫm" trong "bóng tối" của mạng hoặc sẽ chán khi tìm kiếm mà chỉ nhận những mảng thông tin không mấy hữu ích

Chỉ một phần rất nhỏ của thông tin trên Web là thực sự hữu ích

Theo thống kê, 99% của thông tin Web là vô ích với 99% người dùng Web. Trong khi những phần Web không được quan tâm lại bị búi vào kết quả nhận được trong khi tìm kiếm. Vậy thì ta cần phải khai thác Web như thế nào để nhận được trang web chất lượng cao nhất theo tiêu chuẩn của người dùng?

Như vậy chúng ta có thể thấy các điểm khác nhau giữa việc tìm kiếm trong một CSDL truyền thống với việc tìm kiếm trên Internet. Những thách thức trên đã đẩy mạnh việc nghiên cứu khai thác và sử dụng tài nguyên trên Internet

1.4.3. Thuận lợi [24]

Bên cạnh những thử thách trên, còn một số lợi thế của trang Web cung cấp cho công việc khai thác Web.

Web bao gồm không chỉ có các trang mà còn có cả các hyperlink trỏ từ trang này tới trang khác. Khi một tác giả tạo một hyperlink từ trang của ông ta tới một trang A có nghĩa là A là trang có hữu ích với vấn đề đang bàn luận. Nếu trang A càng nhiều Hyperlink từ trang khác trỏ đến chứng tỏ trang A quan trọng. Vì vậy số lượng lớn các thông tin liên kết trang sẽ cung cấp một lượng thông tin giàu có về mối liên quan, chất lượng, và cấu trúc của nội dung trang Web, và vì thế là một nguồn tài nguyên lớn cho khai thác Web.

Một máy chủ Web thường đăng ký một bản ghi đầu vào (Weblog entry) cho mọi lần truy cập trang Web. Nó bao gồm địa chỉ URL, địa chỉ IP, timestamp. Dữ liệu Weblog cung cấp lượng thông tin giàu có về những trang Web động. Với những thông tin về địa chỉ URL, địa chỉ IP, v.v… một cách hiển thị đa chiều có thể được cấu trúc nên dựa trên CSDL Weblog. Thực hiện phân tích OLAP đa chiều có thể đưa ra N người dùng cao nhất, N trang Web truy cập nhiều nhất, và khoảng thời gian nhiều người truy cập nhất, xu hướng truy cập Web.

1. 5. Các hình thức khai thác Web (Web mining)

Chuỗi truy cập Web là một chuỗi các cặp gồm định danh người dùng và sự kiện. Một sự kiện là một yêu cầu về một tài nguyên Web chẳng hạn như một trang hay một dịch vụ. Với mỗi trang được yêu cầu, một số thông tin truyền thống là có sẵn, ví dụ như kiểu và nội dung của trang, lượng thời gian người dùng đã viếng thăm trên trang đó. Các sự kiện của một mẫu truy cập Web được liệt kê dưới dạng danh sách tăng dần theo thời gian.

Khai thác Web nhằm mục đích khám phá thông tin hữu ích hoặc tri thức từ cấu trúc liên kết Web, nội dung trang, và sử dụng dữ liệu. Mặc dù khai thác Web sử dụng nhiều kỹ thuật khai thác dữ liệu nhưng nó không phải là hoàn toàn là một ứng dụng kỹ thuật khai thác dữ liệu truyền thống do không đồng nhất và bản chất bán cấu trúc hoặc không có cấu trúc của dữ liệu Web. Dựa trên các loại chính của dữ liệu được sử dụng trong quá trình khai thác, nhiệm vụ khai thác Web có thể được phân loại thành ba loại:

(1) Khai thác nội dung trang Web (Web content mining)

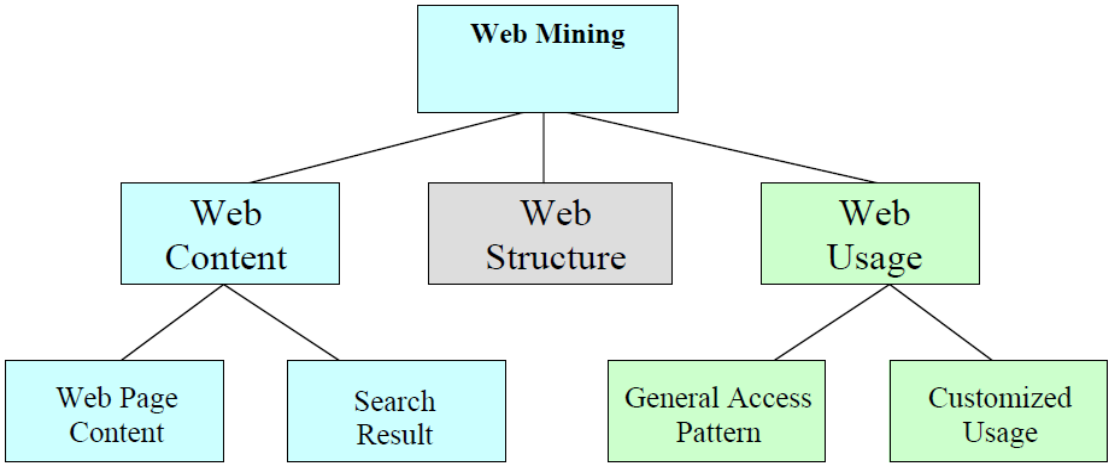
- Nội dung trang Web (Web page content): nghĩa là sẽ sử dụng chỉ các từ trong văn bản mà không tính đến các liên kết giữa các văn bản. Đây chính là khai thác dữ liệu Text (Text mining)
- Tìm kiếm theo kết quả (search result): trong các máy tìm kiếm, sau khi đã tìm ra những trang Web thoả mãn yêu cầu người dùng, còn một công việc không kém phần quan trọng, đó là phải sắp xếp kết quả theo thứ tự độ gần nhau với nội dung cần tìm kiếm. Đây cũng chính là khai thác nội dung trang Web.

(2) Khai thác cấu trúc Web (Web structure mining): Khai thác dựa trên các siêu liên kết giữa các văn bản có liên quan

(3) Khai thác sử dụng Web (Web usage mining)

- Phân tích mẫu truy cập tổng quát (general access partern tracking): phân tích các Web log để khám phá ra các mẫu truy cập của người dùng trong trang Web.
- Phân tích sử dụng của khách hàng (customize usage tracking): phân tích các mẫu truy cập của người dùng tại mỗi thời điểm để biết xu hướng truy cập trang Web của từng đối tượng người dùng tại mỗi thời điểm khác nhau

Quá trình khai thác Web cũng tương tự như quá trình khai thác dữ liệu. Sự khác biệt thường là trong việc thu thập dữ liệu. Trong khai thác dữ liệu truyền thống, dữ liệu thường đã được thu thập và lưu trữ trong một kho dữ liệu. Đối với khai thác Web, thu thập dữ liệu có thể là một nhiệm vụ quan trọng, đặc biệt là khai thác cấu trúc và khai thác nội dung Web, trong đó bao gồm việc thu thập dữ liệu của một số lượng lớn các trang Web.



Hình 1. 7 - Các hình thức khai thác Web

1. 6. Tổng kết chương

Chương mở đầu đã giới thiệu tổng quan về khai thác dữ liệu. Với những ưu điểm của khai khác dữ liệu nó mang đến những ứng dụng phổ biến và rộng rãi trong nhiều lĩnh vực. Để chứng minh điều đó, trong chương cũng đã giới thiệu tổng quan về khai thác mẫu chuỗi phổ biến và trích xuất luật từ cơ sở dữ liệu này. Bên cạnh đó cũng đã giới thiệu về khai thác Web, một loại khai thác dữ liệu mới hiện nay. Cụ thể là Web log, một dạng cơ sở dữ liệu dùng để lưu trữ các thông tin duyệt Web của người dùng. Trong các chương tiếp theo sẽ đi sâu vào tìm hiểu các kỹ thuật khai thác tập phổ biến, khai thác luật cũng như phương pháp khai thác Web log và ứng dụng luật vào khai thác hành vi sử dụng Web, đồng thời đưa ra kết quả thực nghiệm đã được kiểm chứng.

CHƯƠNG 2: KHAI THÁC MẪU CHUỖI VÀ KHAI THÁC LUẬT

2.1. Khai thác mẫu chuỗi

2.1.1. Giới thiệu

Khai thác mẫu chuỗi là khám phá các chuỗi con phổ biến (được gọi là mẫu) trong một cơ sở dữ liệu (CSDL) chuỗi. Một CSDL chuỗi lưu trữ một số bản ghi (record), trong đó tất cả các bản ghi là những chuỗi các sự kiện có thứ tự, có thể gắn với một khái niệm thời gian cụ thể hoặc không.

Ví dụ, CSDL chuỗi giao dịch gồm các giao dịch bán lẻ cho khách hàng, hoặc các chuỗi mua sắm của khách hàng, tương ứng với một khách hàng là một chuỗi các mặt hàng mà khách hàng đó đã mua mỗi tuần/tháng. Các chuỗi mua sắm này có thể biểu diễn dưới dạng bản ghi gồm các trường [ID của giao dịch/ ID khách hàng, <chuỗi các sự kiện có thứ tự>], với mỗi sự kiện là một tập hợp các mặt hàng như bánh mì, đường, trà, sữa, v.v… Giả sử chỉ xét trên hai khách hàng, CSDL mua sắm tuần tự sẽ là [T1, (bread, milk), (bread, milk, sugar), (milk), (tea, sugar)]; [T2, (bread), (sugar, tea)]. Trong đó, khách hàng thứ nhất có ID giao dịch là T1, mỗi tuần trong tháng đều mua hàng; còn khách hàng thứ hai đại diện bởi T2, chỉ mua hàng 2 lần/tháng. Như vậy, một khách hàng có thể mua một hoặc nhiều mặt hàng ở mỗi lần giao dịch, do đó, mỗi bản ghi trong CSDL chuỗi có thể có độ dài khác nhau và mỗi sự kiện trong một chuỗi có thể có một hay nhiều item.

Khai thác mẫu chuỗi là bài toán quan trọng được ứng dụng rộng rãi, bao gồm: phân tích thói quen mua sắm của khách hàng, mẫu truy cập web, các thí nghiệm khoa học, chẩn đoán bệnh, các thảm họa thiên nhiên ….

Thuật toán khai thác mẫu tuần tự trên CSDL chuỗi là đi tìm những mẫu xuất hiện lặp lại (được gọi là chuỗi phổ biến) để tìm kiếm mối liên quan giữa các item khác nhau, hoặc giữa các sự kiện tiềm ẩn trong dữ liệu phục vụ cho các mục đích như các chiến dịch tiếp thị, tái tổ chức kinh doanh, dự báo và lập kế hoạch.

Việc sử dụng các trang web trên toàn thế giới trong các lĩnh vực như thương mại điện tử doanh nghiệp, các dịch vụ web, v.v… ngày càng gia tăng, do đó khai

thác việc sử dụng web là một trong những lĩnh vực ứng dụng phổ biến nhất của khai thác mẫu tuần tự. Các ứng dụng điển hình của khai thác thói quen sử dụng web thường rơi vào các lĩnh vực mô hình người dung như cá nhân hóa nội dung trang web, tái tổ chức trang web, thương mại điện tử, trí tuệ kinh doanh.

2.1.2. Định nghĩa bài toán

Phần này trình bày định nghĩa hình thức của bài toán khai thác mẫu tuần tự và ứng dụng của nó vào khai thác web log. Cho:

i. CSDL tuần tự gồm một tập hợp các bản ghi (còn được gọi là chuỗi).
ii. Ngưỡng hỗ trợ tối thiểu minSup ξ
iii. Tập các item hay sự kiện phân biệt $I = \{i_1, i_2, \ldots, i_k\}$

Bài toán khai thác mẫu tuần tự là đi tìm tập hợp tất cả các chuỗi phổ biến S có trong CSDL chuỗi D tạo bởi các item I tại minSup ξ.

Ví dụ, trong lĩnh vực khai thác sử dụng web, các item trong tập I có thể biểu diễn danh sách các trang web (như các trang a, b, c, d, e, f) hoặc danh sách các sản phẩm (như TV, radio) đã được bán tại một trang web thương mại điện tử.

Itemset là một tập khác rỗng gồm các item không có thứ tự (ví dụ, (abe)) là những trang web được truy cập tại cùng một thời điểm.

Một chuỗi (sequence) là một dãy có thứ tự từ điển các itemset, ví dụ S = a(be)c(ad). Thứ tự từ điển là một thứ tự tuyến tính toàn bộ được định nghĩa như sau: Giả sử một itemset p gồm các item phân biệt $t = \{ i_1, i_2, \ldots, i_k\}$, và một itemset t' khác gồm các item phân biệt là $t' = \{j_1, j_2, \ldots, j_l\}$, với $i_1 \leq i_2 \leq i_k$ và $j_1 \leq j_2 \leq j_k$, sao cho $\leq$ thể hiện mối quan hệ "xuất hiện trước", khi đó ta có $t < t'$ nếu thỏa một trong các điều kiện sau:

(1) Với một số nguyên h, $0 \leq h \leq \min\{k, l\}$, ta có $i_r = j_r$ với $r < h$ và $i_h < j_h$ hoặc

(2) $k < l$, và $i_1 = j_1, i_2 = j_2, \ldots, i_k = j_k$

Ví dụ cho trường hợp (1) là abc < abec và af < bf; với trường hợp (2) tương tự như quan hệ tập con đúng, trong đó t là một tập con đúng của t', chẳng hạn như ab < abc.

Một itemset được tạo từ các item trong tập I, kí hiệu là (i_1, i_2,…, i_k), với i_j là một item hay một sự kiện. Một chuỗi S được coi là một chuỗi các itemset hay chuỗi các sự kiện ($e_1e_2e_3…e_q$), với e_i là một itemset (ví dụ, (be) là một itemset trong chuỗi a(be)c(ad)); itemset cũng có thể chỉ có 1 item được gọi là 1-itemset. Một chuỗi có k item được kí hiệu là chuỗi-k. Một item chỉ có thể xuất hiện một lần trong một itemset nhưng có thể xuất hiện nhiều lần trong các itemset khác nhau của một chuỗi.

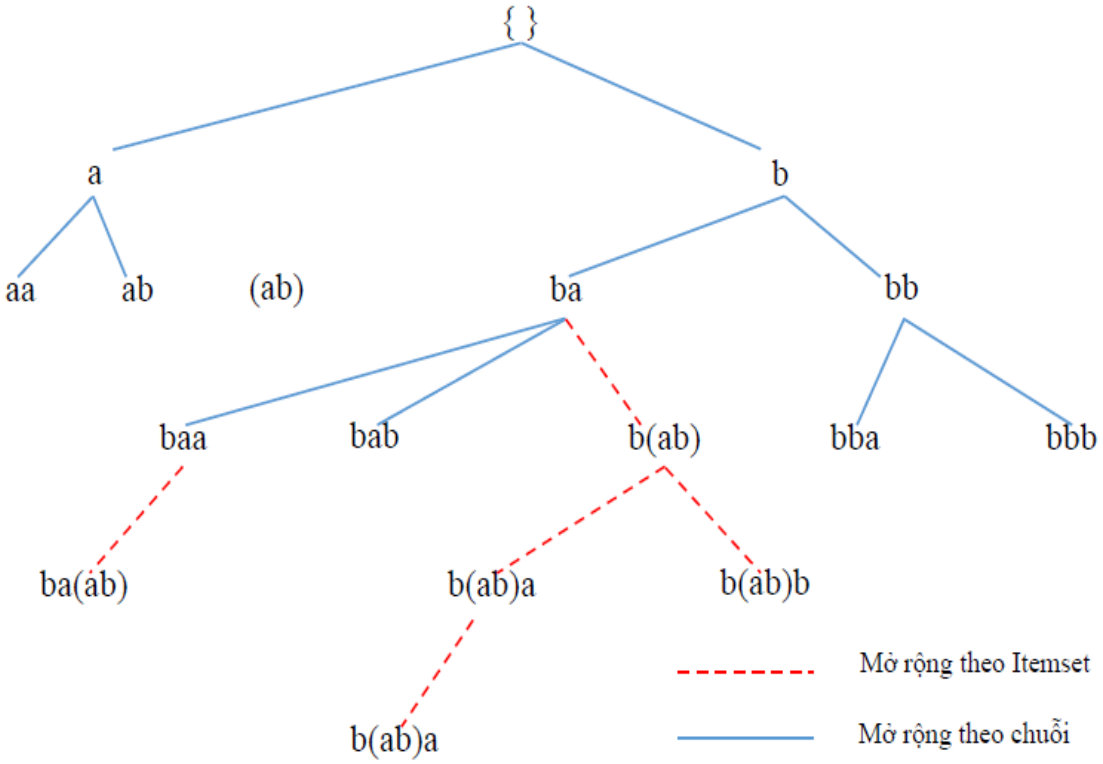
Một chuỗi $\alpha = e_{i1}e_{i2}e_{i3},…,e_{im}$ là một chuỗi con của chuỗi $\beta = e_1e_2e_3…e_n$, kí hiệu là $\alpha \leq \beta$ nếu tồn tại các số nguyên $1 \leq i_1 < i_2 < … < i_m$ và tất cả các sự kiện $e_{ij} \in \alpha$, $e_i \in \beta$ và $i_1 \leq 1$ và $i_m \leq n$ sao cho $e_{ij} \subseteq e_i$.

Mẫu tuần tự cực đại (maximal): một mẫu tuần tự là cực đại nếu nó không là chuỗi con của bất kỳ mẫu tuần tự nào.

Thứ tự từ điển giữa các chuỗi có thể định nghĩa như sau. Giả sử có một thứ tự từ điển $\leq$ giữa các item trên tập I trong CSDL truy cập tuần tự, kí hiệu là $\leq_I$. Nếu một item i xuất hiện trước item j, kí hiệu là $i \leq_I j$; thứ tự này cũng mở rộng áp dụng cho chuỗi và chuỗi con bởi định nghĩa $Sa \preccurlyeq Sb$ nếu Sa là một chuỗi con của Sb. Xét các chuỗi được bố trí trên một cây chuỗi T (được coi là cây từ điển) biểu diễn ở hình 1 như sau: Nút gốc của cây được gán nhãn {}. Một cáchđệ quy, nếu n là một nút trên cây T thì các nút con của n là tất cả các nút n’ sao cho $n \leq n'$ và $\forall m \in T$: $n' \leq m \Rightarrow n \leq m$, mỗi chuỗi trên cây có thể mở rộng bằng cách thêm vào một itemset mới chỉ gồm 1 item hoặc thêm một item vào itemset cuối trong chuỗi. Trường hợp đầu tiên được gọi là mở rộng chuỗi theo chuỗi (sequence-extended sequence), còn trường hợp thứ hai là mở rộng chuỗi theo itemset (itemset-extended sequence) – trường hợp này không thể áp dụng cho khai thác dữ liệu web log vì trong chuỗi dữ liệu web log, mỗi itemset chỉ có duy nhất một item.

Độ phổ biến hay độ hỗ trợ (support) của một chuỗi (hoặc chuỗi con) S, kí hiệu là $\partial(S)$ được tính bằng số lượng chuỗi trong CSDL D có S là chuỗi con chia cho tổng số lượng chuỗi trong D, còn dộ hỗ trợ tuyệt đối (absolute support hoặc support count) của một chuỗi (hoặc chuỗi con) S là tổng số lượng chuỗi trong D mà

có S là chuỗi con.



Hình 2.1.1 - Cây từ điển biểu diễn các chuỗi, với đường nét đứt là mở rộng theo chuỗi và nét liền là mở rộng theo itemset

Chuỗi phổ biến đóng (closed frequent sequence): một chuỗi được gọi là phổ biến nếu độ hỗ trợ của nó không nhỏ hơn ngưỡng hỗ trợ tối thiểu (minimum support) do người dùng chỉ ra, kí hiệu là minSup hay kí hiệu bởi kí tự ξ. Một chuỗi phổ biến S_α được gọi là chuỗi phổ biến đóng nếu không tồn tại chuỗi cha đúng nào của S_α có cùng độ hỗ trợ với S_α, tức là $\nexists\beta$ sao cho $S_\alpha \preccurlyeq S_\beta$ và $\partial(S_\alpha) = \partial(S_\beta)$; ngược lại ta nói chuỗi S_αbị phủ (absorbed) bởi S_β. Ví dụ, giả sử rằng chuỗi phổ biến S_β = beadc là chuỗi cha duy nhất của chuỗi phổ biến S_α= bea, nếu $\partial(S_\alpha) = \partial(S_\beta)$ thì S_α là không phải là chuỗi phổ biến đóng, ngược lại nếu $\partial(S_\alpha) > \partial(S_\beta)$ thì S_α là chuỗi phổ biến đóng. Lưu ý rằng $\partial(S_\alpha)$ không thể lớn hơn $\partial(S_\beta)$ vì $S_\alpha \preccurlyeq S_\beta$.

2.1.3. Cách tổ chức dữ liệu

Dạng biểu diễn ngang: Dữ liệu được tổ chức theo chiều ngang, mỗi hang đại diện cho dãy sự kiện (event) tương ứng với đối tượng (object).

Dạng biểu diễn dọc: Dữ liệu được tổ chức theo chiều dọc, mỗi hàng đại diện cho dãy đối tượng tương ứng với sự kiện.

Ví dụ: Cho CSDL chuỗi:

Đối tượng	Chuỗi sự kiện
1	A, B, C
2	A, D, E, F
3	B, E

CSDL trên có thể biểu diễn theo 2 cách sau:

Biểu diễn ngang

Đối tượng	Sự kiện
1	A, B, C
2	A, D, E, F
3	B, E

Biểu diễn dọc

Sự kiện	Đối tượng
A	1, 2
B	1, 3
C	1
D	2
E	2, 3
F	2

Trong hai cách tổ chức dữ liệu theo chiều dọc và theo chiều ngang, thao tác đếm độ hỗ trợ cho một sự kiện ở CSDL được tổ chức theo chiều dọc đơn giản và nhanh hơn. Bởi vì theo cách tổ chức này, có thể lấy được ngay các đối tượng ứng với sự kiện mà không phải duyệt toàn bộ CSDL. Hơn nữa, đối với CSDL lớn, việc tổ chức theo chiều dọc mang tính cô đọng, giúp thực thi nhanh hơn và cho phép lặp lại việc tìm các mẫu tuần tự một cách dễ dàng. Tuy nhiên, dữ liệu gốc ban đầu thường được tổ chức theo chiều ngang, nếu muốn tổ chức theo chiều dọc phải có bước tiền xử lý để chuyển đổi.

2.1.4. Các dạng bài toán tiếp cận

Các thuật toán khai thác chuỗi mà mỗi itemset là một item được thảo luận theo CSDL Bài toán 1 (bảng 2.1.1), còn những thuật toán khai thác chuỗi mà có mỗi itemset gồm nhiều item được thảo luận theo CSDL Bài toán 2 (bảng 2.1.2)

Bài toán 1:

Giả sử D (minh họa ở bảng 2.1.1) là một CSDL chuỗi truy cập web được rút

trích từ một số web log, và được sắp xếp theo định danh của người dùng (User, IDs) và các giao dịch tương ứng (TIDs), với mỗi giao dịch t_i là một chuỗi các trang mà người dùng đã xem, và kí hiệu C_1={a, b, c, d, e} là tập các chuỗi ứng viên độ dài 1, cho trước ngưỡng minSup=60%, công việc cần làm là tìm tập L gồm tất cả các chuỗi phổ biến (còn gọi là mẫu tuần tự phổ biến).

Bảng 2.1.1 - CSDL chuỗi D, mỗi itemset chỉ là một item

UserID	TID	Chuỗi truy cập
100	t1	bcbae
200	t2	bacbae
300	t3	abe
400	t4	abebd
500	t5	abad

Bài toán 2:

Cho CSDL chuỗi như bảng 2.1.2, dữ liệu này dùng cho những thuật toán mà xử lý trường hợp itemset trong mỗi chuỗi dữ liệu có nhiều item chứ không phải 1 item như bảng 1, bài toán đặt ra là tìm tất cả các mẫu phổ biến với ngưỡng minSup=25%.

Bảng 2.1.2 - CSDL chuỗi D, mỗi itemset gồm nhiều item

ID chuỗi mỗi khách hàng	CSDL chuỗi
1	ae
2	(fg)a(fbkc)
3	ahcd
4	a(abcd)e
5	e

2.1.5. Các thuật toán khai thác mẫu chuỗi

2.1.5.1.Các kỹ thuật dựa trên Apriori

Apriori dựa vào 2 nguyên tắc cơ bản để tìm tập chuỗi phổ biến:

(1) Mọi tập con của tập phổ biến đều phổ biến.

(2) Mọi tập cha của tập không phổ biến đều không phổ biến.

Các thuật toán dựa vào nguyên lý Apriori để sinh và kiểm tra các mẫu ứng viên, trong đó nếu một mẫu được kiểm tra không thỏa mãn ngưỡng tối thiểu thì các mẫu chứa nó cũng sẽ bị loại. Có một số thuật toán theo phương pháp tiếp cận dựa trên Apriori như AprioriAll, GSP, SPAM, v.v... và các biến thể của chúng

2.1.5.2.Các kỹ thuật phát triển mẫu chuỗi

Ra đời sau các phương pháp dựa trên Apriori vào giữa thập niên 90, các phương pháp phát triển mẫu ra đời đầu những năm 2000, là giải pháp cho vấn đề phát sinh và kiểm tra mẫu. Ý tưởng chính là để tránh phát sinh các ứng viên cùng lúc và để tập trung tìm kiếm trên phần CSDL giới hạn đã được chia nhỏ từ CSDL ban đầu. Hầu hết các thuật toán phát triển mẫu đều bắt đầu bằng việc xây dựng một dạng biểu diễn của CSDL cần khai thác, sau đó đề xuất cách chia nhỏ không gian tìm kiếm, và tạo ra một số ít chuỗi ứng viên có thể nhờ phát triển trên các chuỗi phổ biến đã khai thác được ở bước trước, và áp dụng tính chất Apriori khi duyệt không gian tìm kiếm đệ quy. Các thuật toán đầu tiên khởi đầu bằng các CSDL chiếu, ví dụ như FreeSpan [12], PrefixSpan [13].

Một chuỗi con α’ của α được gọi là chiếu của α theo tiền tố β nếu và chỉ nếu (1) α’ có tiền tố β, và (2) không tồn tại chuỗi cha đúng α’’ nào của α’ sao cho α’’ là chuỗi con của α và α’’ có tiền tố β. PrefixSpan dựa trên việc xây dựng mẫu đệ quy bằng cách phát triển trên tiền tố, đồng thời, giới hạn tìm kiếm đối với CSDL chiếu. Nhờ đó, không gian tìm kiếm giảm đi tại mỗi bước. PrefixSpan vẫn được coi là một tiêu chuẩn và là một trong những thuật toán khai thác mẫu tuần tự nhanh nhất sát cánh với SPADE [16]. Thuật toán khác, WAP-mine [12] là thuật toán đầu tiên trong số các thuật toán phát triển mẫu sử dụng cấu trúc cây vật lý để biểu diễn CSDL chuỗi cùng với độ hỗ trợ, sau đó khai thác cây này thay vì phải duyệt toàn bộ CSDL chuỗi ở mỗi bước.

2.1.5.3.Các kỹ thuật loại trừ sớm

Trong tất cả nghiên cứu đã có, các thuật toán dựa trên kỹ thuật loại trừ sớm là hướng tiếp cận mới đối với khai thác mẫu tuần tự. Những thuật toán này sử dụng

một loại quy nạp vị trí để loại trừ các chuỗi ứng viên rất sớm trong quá trình khai thác và tránh phải đếm độ hỗ trợ càng nhiều càng tốt. Cách giải quyết của quá trình khai thác là phát triển mẫu đơn giản. Ý tưởng quy nạp vị trí như sau: Nếu vị trí sau cùng của một item nhỏ hơn vị trí của tiền tố hiện thời (suốt quá trình khai thác) thì item không thể xuất hiện sau tiền tố đó trong cùng một chuỗi dữ liệu.

Các thuật toán này thường sử dụng một bảng để dò các vị trí sau cùng của mỗi item trong chuỗi và sử dụng thông tin này cho việc loại trừ sớm chuỗi ứng viên; vì vị trí sau cùng của item là chìa khóa để xác định liệu có thể item thêm vào một chuỗi tiền tố độ dài *k* cho trước hay không, vì vậy tránh phải đếm độ hỗ trợ và tạo ra chuỗi ứng viên không phổ biến. So sánh LAPIN [16] với PrefixSpan [13], và LAPIN là một trong số thuật toán đầy triển vọng cho loại thuật toán này. Đã có nhiều nỗ lực khác nhau nghiên cứu thuật toán theo dạng này, đi đầu là LAPIN-SPAM [15] là giải pháp cho thách thức tràn các thao tác xử lý bit của SPAM; và để đáp ứng thách thức giảm việc sử dụng bộ nhớ đi một nửa thì đã xuất hiện các thuật toán LAPIN LCI, LAPIN Suffix [16] với phương pháp phát triển hậu tố; và LAPIN-WEB [18] dùng cho khai thác web log. Hơn nữa, cần phải làm khảo sát kiểm tra liệu việc xây dựng bảng tối ưu có tạo ra trì hoãn khởi động hay không. Thuật toán LAPIN khác so với LAPIN-SPAM ở chỗ nó không sử dụng bitmap để biểu diễn CSDL, trong thuật toán HVSM, đã nhấn mạnh tầm quan trọng của biểu diễn bit và bổ sung thêm phương pháp duyệt CSDL *theo chiều ngang trước sau theo chiều dọc sau,* sử dụng cấu trúc dữ liệu cây đặc biệt, trong đó mỗi nút lấy các nút anh em phổ biến cũng như các nút con ucar nó trong quá trình kết nối để mở rộng itemset, vì vậy tránh phải đếm độ hỗ trợ; tuy nhiên sự thực hiện của HVSM không vượt qua được SPAM [16]. Mặt khác, với thuật toán DISC-all ,bên cạnh thứ tự từ điển thông thường trên các item, thuật toán còn sử dụng thứ tự thời gian và dạng biến đổi của CSDL chiếu để chia nhỏ không gian tìm kiếm, phụ thuộc vào thứ tự của các chuỗi phổ biến độ dài 1.

2.1.5.4.Các thuật toán lai

Một số thuật toán kết hợp một số đặc trưng trong số ba phân loại đã đề xuất

trên. Ví dụ, PLWAP kết hợp phép chiếu cây và đặc trưng phát triển mẫu từ phân loại phát triển mẫu và đặc trưng mã-vị trí từ phân loại loại trừ sớm. Tất cả những đặc trưng này là đặc điểm chủ chốt của loại này, vì vậy chúng ta coi PLWAP là thuật toán lai giữa phát triển mẫu và loại trừ sớm. Có những đặc trưng không thể kết hợp cùng vào một kỹ thuật như "duyệt CSDL nhiều lần" và "chiếu cây", vì "chiếu cây" được dùng như là một dạng biến đổi của CSDL trong bộ nhớ và tránh phải đếm độ hỗ trợ, nó không thể kết hợp với "duyệt CSDL nhiều lần".

2.1.6. Khai thác mẫu chuỗi đóng

2.1.6.1. Mục tiêu khai thác mẫu chuỗi đóng

Thông qua một số kỹ thuật khai thác mẫu tuần tự ở trên, có rất nhiều các kỹ thuật hiệu quả và các biến thể của chúng được đề xuất, cung cấp cho chúng ta các giải pháp hiệu quả trong nhiều trường hợp. Tuy nhiên các kỹ thuật này vẫn còn một số hạn chế mà đến nay vẫn đang là những trở ngại và thách thức trong quá trình thực hiện khai thác mẫu chuỗi:

(1) Các thuật toán theo hướng tiếp cận dựa trên Apriori, với cơ sở dữ liệu chuỗi lớn nó sẽ tạo ra một lượng rất lớn các ứng viên. Mặt khác, trong quá trình khai thác kỹ thuật này đòi hỏi phải duyệt cơ sở dữ liệu gốc nhiều lần.

(2) Trong hướng tiếp cận phát triển mẫu, chi phí chủ yếu của các kỹ thuật này là việc xây dựng cơ sở dữ liệu quy chiếu. Trong trường hợp xấu nhất, chúng ta phải xây dựng cơ sở dữ liệu quy chiếu cho mọi mẫu, với số lượng mẫu lớn thì chi phí này là không tầm thường.

(3) Trong quá trình khai thác có thể tạo ra lượng lớn các mẫu chuỗi phổ biến, nhưng người sử dụng chỉ quan tâm một lượng nhỏ các mẫu có ích trong đó phù hợp với mục tiêu sử dụng, việc biểu diễn toàn bộ các mẫu gây khó khăn cho việc sử dụng.

(4) Khó khăn trong quá trình khai thác các mẫu chuỗi dài bởi vì các mẫu này phải được phát triển từ một lượng lớn các mẫu ngắn, vì thế số lượng các ứng viên được sinh ra sẽ là cấp số nhân.

Khai thác mẫu chuỗi đã được tập chung nghiên cứu mạnh mẽ trong những

năm gần đây. Với các mục đích ứng dụng khác nhau tạo ra một lượng lớn sự đa dạng các thuật toán khai thác mẫu tuần tự, nhiều phần mở rộng của các định nghĩa ban đầu được đề xuất cho các mục đích đặc biệt như khai thác mẫu tuần tự với những rành buộc, mẫu tuần tự đóng, đa chiều, gia tăng, v.v…

Các thuật toán khai phá mẫu tuần tự được phát triển cho đến nay có hiệu suất tốt trong cơ sở dữ liệu có chứa các mẫu phổ biến ngắn. Tuy nhiên, khi khai thác các mẫu tuần tự phổ biến dài, hoặc khi sử dụng ngưỡng hỗ trợ rất thấp, hiệu suất của các thuật toán trên thường giảm đáng kể.

Ví dụ, cơ sở dữ liệu chỉ chứa một mẫu chuỗi phổ biến <(a1),(a2),(a3),..(a100),> , nó sẽ tạo ra 2100 - 1 mẫu phổ biến nếu ngưỡng tối thiểu là 1, mặc dù tất cả các mẫu phổ biến này ngoại trừ mẫu dài nhất là không cần thiết.

Xét CSDL minh họa trong bảng 2.1.3, mẫu chuỗi phổ biến <(DVD Rec) (Video Soft)> không phải là mẫu chuỗi đóng vì nó được chứa trong mẫu dãy s_2 và có cùng độ hỗ trợ (50%). Mặt khác, mẫu dãy <(Camcorder, MiniDV)> là mẫu dãy đóng vì nó được chứa trong mẫu dãy s_1 nhưng có độ hỗ trợ là 75%, khác với độ hỗ trợ của s_1 là 50%.

Bảng 2.1.3 - Các dãy dữ liệu của 4 khách hàng mua trong 4 ngày

Cust	June 04, 2004	June 05, 2004	June 06, 2004	June 07, 2004
C1	Camcorder, MiniDV	Digital Camera	MemCard	USB Key
C2	Camcorder, MiniDV	DVD Rec, DVD-R		Video Soft
C3	DVD Rec, DVD-R	MemCard	Video Soft	USB Key
C4		Camcorder, MiniDV	Laptop	DVD Rec, DVD-R

2.1.6.2.Ý nghĩa khai thác mẫu chuỗi đóng

Khai thác mẫu chuỗi đóng (*mining closed sequentail pattern*) là một trong những chủ đề nghiên cứu thiết thực và quan trọng hiện nay của lĩnh vực khai thác dữ liệu. Mục đích của nó là tìm ra các chuỗi đóng phổ biến trong cơ sở dữ liệu với số lượng giao dịch lớn, để tìm ra các mối quan hệ giữa các mẫu dữ liệu này. Việc tìm kiếm chuỗi tuần tự khá đơn giản nhưng thông tin mà nó mang lại có ý nghĩa

quan trọng, phục vụ cho việc phát sinh luật để hỗ trợ cho quá trình quyết định, quản lý và có tính định hướng. Đặc biệt, phục vụ cho việc nghiên cứu chính đã đặt ra của luận này.

Khai thác mẫu chuỗi đóng giúp giảm không gian lưu trữ và thời gian khai thác luật cho giai đoạn kế tiếp, các thuật toán khai thác thác chuỗi tuần tự phổ biến đóng hiệu quả đã được đề xuất: A-Close sử dụng phương pháp tìm kiếm breadth-first đề tìm các tập phổ biến; CLOSET và CLOSET+[10] sử dụng cây tập mẫu phổ biến để nén các mẫu dữ liệu, ngoài ra còn có rất nhiều thuật toán khác như CHARM [21].

2.1.6.3. Định nghĩa bài toán

Cho tập các item hay sự kiện phân biệt $I = \{i_1, i_2, ..., i_k\}$. Một chuỗi S được xem là một chuỗi các item hay là chuỗi các sự kiện, ký hiệu $e_1,e_2,e_3...e_m$, với e_i là một item, $e_i \in I$ *sao cho* $1 \leq i \leq m$. Một trong các itemset hay sự kiện khác nhau item có thể xuất hiện nhiều lần của một chuỗi. Số lượng các itemtrong một chuỗi gọi là độ dài của chuỗi, và một chuỗi có độ dài *k* được gọi là một *k – sequence.* Ví dụ: S = AABCCA là một *6-sequence*

Một chuỗi $S_a = a_1a_2.....a_n$ là một chuỗi con của chuỗi $S_b = b_1b_2....b_m$, nếu tồn tại các số nguyên $1 \leq i1 < i2 < ... < i_n \leq im$, $a_1 = b_{i1}$, $a_2 = b_{i2}$, $a_n = b_{in}$. Nếu chuỗi S_a chứa trong chuỗi S_b , S_a được gọi là chuỗi con của chuỗi S_b và chuỗi S_b là chuỗi cha của S_a, ký hiệu $S_a \subseteq S_b$. Đầu vào CSDL chuỗi *SDB* là các chuỗi (*sid, S*) với sid là chuỗi định danh, *S* là chuỗi đầu vào. Số lượng của chuỗi trong SDB được gọi là kích thước cơ bản của SDB, ký hiệu |*SDB*|. Một chuỗi (*sid, S*) được cho là chứa một chuỗi S_α nếu *S* là một chuỗi cha của S_α , $S_\alpha \sqsubseteq S$. Độ hỗ trợ tuyệt đối của chuỗi S_α trong CSDL chuỗi SDB là tổng số chuỗi trong SDB chứa S_α , ký hiệu sup^{SDB} (S_α), và độ hỗ trợ tương đối là tỷ lệ phần trăm của các chuỗi trong SDB có chứa S_α, (gọi là $sup^{SDB}(S_\alpha)/|SDB|$). Cho ngưỡng hỗ trợ *min_sup*, chuỗi S_α được xem là phổ biến trên *SDB* nếu $sup^{SDB}(S_\alpha) \geq min\_sup$. Nếu chuỗi S_α là phổ biến và không tồn tại chuỗi cha đúng nào của S_α có cùng độ hỗ trợ với S_α tức là không tồn tại S_β sao cho $S_\alpha \subset S_\beta$ và $sup^{SDB}(S_\alpha) = sup^{SDB}(S_\beta)$, ta gọi S_α là chuỗi phổ biến đóng. Vấn đề của khai thác

chuỗi phổ biến đóng là tìm ra tập hoàn chỉnh chuỗi phổ biến đóng cho cho CSDL chuỗi SDB, cho một ngưỡng hỗ trợ tối thiểu, min_sup.

Bảng 2.1.4 - CSDL chuỗi SDB

Chuỗi định danh	Chuỗi
1	C A A B C
2	A B C B
3	C A B C
4	A B B C A

Ví dụ: Bảng 2.1.4, tổng cộng có 3 item và 4 chuỗi (tức là |SDB| = 4), giả sử *min_sup*=2. Tập hoàn chỉnh của mẫu phổ biến đóng S_{fcs} = {*AA:2, ABB:2, ABC:4, CA:3, CABC:2, CB:3*} bao gồm 6 chuỗi trong khi đó nếu khai thác chuỗi phổ biến bình thường thì có 17 chuỗi, đó là: S_{fs} = {*A:4, AA:2, AB:4, ABB:2, ABC:4, AC:4, B:4, BB:2, BC:4, C:4, CA:3, CAB:2, CABC:2, CAC:2, CB:3, CBC:2, CC:2*}. Rõ ràng, S_{fcs} nhỏ gọn hơn S_{fs}, ngoài ra nếu một chuỗi S_α có cùng độ hỗ trợ với chuỗi cha S_β đúng nghĩa của nó thì S_α bị phủ (*absorbed*) bởi S_β.Ví dụ: chuỗi phổ biến CBC:2 được phủ bởi chuỗi CABC:2, bởi vì (CBC ⊂ *CABC*) và ($sup^{SDB}(CBC) = sup^{SDB}(CABC) = 2$).

2.1.6.4.Thuật toán CloSpan

Thuật toán Clospan (Closed sequemtial pattern mining) [20] với việc phát hiện các mẫu tuần tự đóng, tránh được một số lượng lớn các lần gọi đệ quy. Đầu tiên tạo ra tập các chuỗi ứng viên đóng được lưu trữ trong một cấu trúc cây hash-chỉ mục và sau đó cắt tỉa nó. Nó sử dụng một số phương pháp cắt tỉa như *CommomPrefix* và *Backward Sub-Pattern* tỉa không gian tìm kiếm. Bởi vì CloSpan cần phải duy trì sự tập hợp của các chuỗi ứng viên đóng, nó sẽ tiêu tốn nhiều bộ nhớ và dẫn đến một không gian tìm kiếm lớn cho việc kiểm tra mô hình đóng khi có nhiều chuỗi phổ biến đóng. Kết quả là, nó không tốt đối với số các số lượng chuỗi phổ biến đóng.

Thuật toán CloSpan dựa trên việc phát hiện các mẫu tuần tự có độ dài 2, ví dụ như “A luôn xảy ra trước hoặc sau B”. Xét CSDL trong hình 2.3, chúng ta biết

rằng <(DVD Rec) (Video Soft)> là một mẫu thường xuyên. Các tác giả của thuật toán CloSpan đề xuất các phương pháp liên quan để chứng minh rằng <(DVD-R)> luôn luôn xảy ra trước <(Video Soft)>. Dựa vào quan sát này, CloSpan có thể chỉ ra rằng <(Rec DVD, DVD-R) (Video Soft)> là mẫu thường xuyên mà không cần bất kỳ lần quét CSDL nào nữa.

2.1.6.5.Thuật toán BIDE

Thuật toán BIDE (BI-Directional Extension) [19] là mở rộng của thuật toán CloSpan. Đầu tiên, thông qua một phần mở rộng chuỗi mới, được gọi là BI-Directional Extension, thuật toán sử dụng cả hai phương pháp:

(1) Mẫu tiền tố và kiểm tra thuộc tính đóng để phát triển.

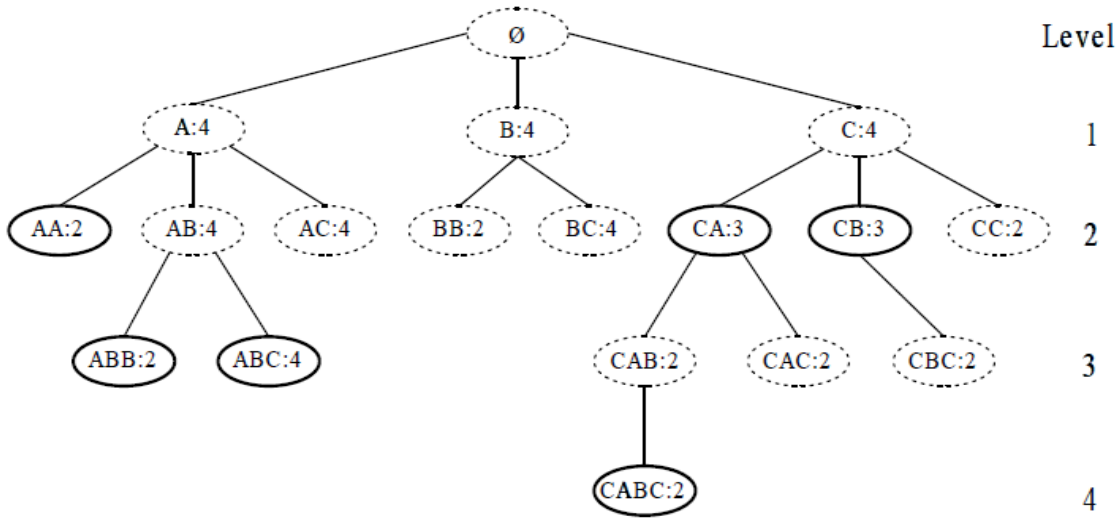
(2) Để lược bớt không gian tìm kiếm sâu hơn so với phương pháp tiếp cận trước, thuật toán đề nghị một phương pháp cắt tỉa gọi là *BackScan*. Ý tưởng chính của phương pháp này là để tránh mở rộng chuỗi bằng cách phát hiện trước phần mở rộng đã được chứa trong một chuỗi đã có nhằm tăng tốc độ khai thác mà vẫn giữ đính đúng đắn của việc khai thác chuỗi phổ biến đóng.

Liệt kê chuỗi phổ biến về mặt khái niệm, không gian tìm kiếm của khai thác chuỗi tạo thành một cấu trúc cây và được định nghĩa đệ quy như sau: nút gốc ở mức *0* của cây chứa chuỗi rỗng được gán nhãn ∅. Các nút ở mức *L* của cây sẽ được gán nhãn là các chuỗi có kích thước *L*. Nút con ở mức *(L+1)* được tạo bằng cách mở rộng chuỗi ở mức L để được chuỗi độ dài *(L+1)*. Bằng cách loại bỏ các chuỗi không phổ biến, các nút còn lại trong cây từ điển chuỗi phổ biến, tạo thành tập chuỗi phổ biến. Trong hình 2.3, mỗi nút là một chuỗi phổ biến và độ hỗ trợ của nó, và nút đứt nét là chứa những chuỗi không đóng.

BIDE duyệt cây theo chiều sâu (DFS–depth first search), trong Hình 2.1.2, chuỗi phổ biến sẽ được khai thác theo một trình tự, như vậy: *A*: 4, *AA*: 2, *AB*: 4, *ABB*: 2, *ABC*: 4, *AC*: 4, *B*: 4, *BB*: 2, *BC*: 4, *C:* 4, *CA*:3, *CAB*: 2, *CABC*: 2, *CAC*: 2, *CB*: 3, *CBC*: 2, *CC*: 2.

Một nút nào đó trong cây có thể được coi như là một chuỗi tiền tố mà từ đó các tập con của nó có thể được tạo ra bằng cách thêm một item trong *I*. Trong

BIDE, chỉ sử dụng phương pháp chiếu giả (*PseudoProjection*) để tìm ra tập hợp các item đối với một tiền tố nhất định và sử dụng chúng để phát triển các tiền tố tương ứng



Hình 2.1.2 - Cây từ điển chuỗi phổ biến

Nhiều nghiên cứu đã xây dựng mà kín khai thác mô hình có sức mạnh ý nghĩa tương tự như của tất cả khai thác mô hình phổ biến chưa dẫn đến kết quả nhỏ gọn hơn thiết lập và hiệu quả tốt hơn đáng kể. nghiên cứu của chúng tôi cho thấy rằng điều này thường là đúng khi số lượng các mô hình phổ biến là tốn rất lớn, trong trường hợp số lượng các mô hình phổ biến đóng cũng có khả năng rất lớn. Các thuật toán khai thác mô hình đóng phát triển trước đó dựa trên các thiết lập lịch sử của mô hình phổ biến đóng (hoặc ứng viên) để kiểm tra xem một mô hình phổ biến mới được tìm thấy là đóng hoặc nếu nó có thể làm mất hiệu lực một số đã được khai thác đóng các ứng cử viên. Bởi vì bộ đã khai thác mô hình phổ biến đóng không ngừng tăng lên trong quá trình khai thác, không chỉ nó sẽ tiêu thụ bộ nhớ hơn, nhưng cũng dẫn đến sự kém hiệu quả do không gian tìm kiếm phát triển để kiểm tra mô hình đóng.

BIDE là một thuật toán mới để khai thác chuỗi phổ biến đóng. Nó tránh được lời nguyền ofthe ứng cử viên bảo trì và thử nghiệm mô hình, mà không gian tìm kiếm sâu hơn và kiểm tra việc đóng mô hình một cách hiệu quả hơn trong khi tiêu thụ ít bộ nhớ nhiều trái ngược với mô hình đóng phát triển trước đó các thuật toán

khai thác mỏ. Nó không cần phải duy trì sự tập các mô hình đóng lịch sử, do đó nó quy mô rất tốt trong số các mẫu phổ biến đóng. BIDE áp dụng một chiều sâu để tìm kiếm đầu tiên nghiêm ngặt và có thể xuất ra các mô hình phổ biến đóng trong một thời trang trực tuyến. Một bộ đầy đủ của các thí nghiệm trên nhiều bộ dữ liệu thực tế với tính năng phân phối khác nhau đã cho thấy hiệu quả của việc thiết kế thuật toán: BIDE tiêu thụ hàng tầm cỡ bộ nhớ ít hơn trong khi có thể được hơn một thứ tự cường độ nhanh hơn so với thuật toán CloSpan. Nó cũng có khả năng mở rộng tuyến tính về số lượng các trình tự trong cơ sở dữ liệu. Nhiều nghiên cứu đã chỉ ra rằng hạn chế là rất cần thiết cho nhiều ứng dụng khai thác mô hình tuần tự.

2.1.6.6.Kết hợp của bit vectơ động cho khai thác chuỗi phổ biến đóng [3]

a) Giới thiệu

Các thuật toán khai thác tự cố gắng để khai thác tất cả các chuỗi phổ biến càng tốt. Các thuật toán này tạo ra kết quả dự phòng, tăng thêm không gian lưu trữ cần thiết và thời gian chạy, đặc biệt là đối với cơ sở dữ liệu chuỗi lớn. Trong những năm gần đây, nhiều nghiên cứu đã chứng minh rằng khai thác chuỗi phổ biến đóng có hiệu quả hơn khai thác tất cả các chuỗi phổ biến. Các thông tin mong muốn có thể được chiết xuất hoàn toàn từ các chuỗi phổ biến đóng. Hầu hết các thuật toán để khai thác chuỗi phổ biến đóng sử dụng một mô hình ứng tỉa bớt và kiểm tra. Đề xuất một thuật toán gọi CloFS-DBV sử dụng vectơ bit động. Phương pháp khác nhau được sử dụng để giảm sử dụng bộ nhớ và thời gian chạy. Kết quả thí nghiệm cho thấy rằng CloFS-DBV là hiệu quả hơn so với BIDE và các thuật toán CloSpan về thời gian thực hiện và sử dụng bộ nhớ.

Khai thác mô hình tuần tự là một vấn đề cơ bản trong khám phá tri thức và khai thác dữ liệu với ứng dụng rộng rãi, bao gồm cả những người trong việc phân tích hành vi khách hàng mua hàng, các mẫu truy cập Web , các thí nghiệm khoa học, điều trị bệnh, phòng chống thiên tai, và sự hình thành protein. Khai thác mô hình tuần tự bao gồm hai giai đoạn chính: khai thác mô hình phổ biến và khai thác mỏ quy tắc. Nhiều nghiên cứu đã sửa đổi các thuật toán AprioriAll [9] cho khai thác mẫu tuần tự phổ biến. Không giống như việc khai thác chung của chuỗi phổ biến,

việc khai thác chuỗi phổ biến đóng chưa được nghiên cứu rộng rãi. Mặc dù một số thuật toán đã được đề xuất, chẳng hạn như CloSpan[34], CLOSET+[35], và BIDE[36], hiệu suất của họ là kém cho cơ sở dữ liệu lớn. BIDE phát hiện các trình tự phổ biến, những không đóng cửa, và tỉa bớt, thay vì sử dụng các mô hình bảo trì và kiểm tra.

Gần đây, nhiều tác giả đã đề xuất kỹ thuật mà dữ liệu có trong một định dạng thẳng đứng [36], sử dụng cơ sở dữ liệu chiếu[37], sử dụng cấu trúc dữ liệu bit vector [38], tất cả đều đã được chứng minh là có hiệu quả. Tuy nhiên, không gian lưu trữ và thời gian thực hiện có thể được tiếp tục giảm trong quá trình khai thác khoáng sản cho cơ sở dữ liệu chuỗi lớn.

Thuật toán CloFS-DBV, trong đó sử dụng một định dạng dữ liệu và nén dữ liệu theo chiều dọc, và chia không gian tìm kiếm để giảm không gian lưu trữ cần thiết và thời gian thực hiện đối với khai thác các trình tự phổ biến đóng.

b) Định nghĩa vấn đề

Hãy xem xét một cơ sở dữ liệu liên tục với một tập hợp các sự kiện riêng biệt $I=\{i_1, i_2, i_3, \cdots, i_n\}$, nơi i_j is là một sự kiện (hay một mục), nơi $1 \leq j \leq n$. Một tập hợp các sự kiện có thứ tự được gọi là một tập phổ biến. Mỗi tập phổ biến được đặt trong dấu ngoặc đơn, ví dụ (ABC). Để đơn giản hóa các ký hiệu, cho tập phổ biến có chứa chỉ một mục duy nhất, các dấu ngoặc đơn được bỏ qua, ví dụ như B. Một chuỗi $S = \{e_1 , e_2, e_3 ,\ldots, e_m\}$ là một danh sách có thứ tự các sự kiện, nơi e_j ($1 \leq j \leq m$) là một tập phổ biến. Giả sử rằng ℓ là số các sự kiện trong một chuỗi. Một chuỗi với chiều dài ℓ được gọi là một chuỗi ℓ. Ví dụ, AB(AE)CB là một chuỗi 6 sequence. A sequence $S_a = a1, a2, \cdots, a_m$ là được chứa trong một chuỗi $S_b = b_1 , b_2 ,\cdots , b_n$; nếu có tồn tại số nguyên $1 \leq i_1 < i_2 < ... < i_m \leq n$ như vậy $a_i = b_{i1} , a_2 = b_{i2} , \cdots , a_m = b_{im}$. Nếu chuỗi S_a được chứa trong chuỗi S_b, S_a được gọi là một dãy con của S_b và S_b được gọi là một supersequence của S_a, ký hiệu là $S_a \subseteq S_b$. Một cơ sở dữ liệu trình tự được ký hiệu là $D = \{s_1 , s_2 , s_3 , \cdots, s_{|D|}\}$, $|D|$ là số chuỗi trong D và s_i ($1 \leq i \leq |D|$) là một giao dịch theo hình thức ID; Trình tự, nơi các ID thuộc tính được

sử dụng để mô tả các thông tin của s_i tương ứng với thông tin giao dịch theo thời gian.

Sự hỗ trợ tuyệt đối (support) của một chuỗi S_a trong cơ sở dữ liệu trình tự D được tính toán là số lần xuất hiện của S_a trong các giao dịch của D, ký hiệu là $sup^D(S_a)$. Sự hỗ trợ của một trình tự được đưa ra trong chuỗi ký hiệu: support. Ví dụ, một chuỗi AB với sự hỗ trợ 3 được biểu diễn như là AB: 3.

Cho một minSup ngưỡng hỗ trợ tối thiểu, một chuỗi S_a là một chuỗi phổ biến trên D nếu $sup^D(S_a) \geq$ minSup. Nếu chuỗi S_a là phổ biến và có tồn tại không thích hợp supersequence S_b của S_a với sự hỗ trợ tương, S_a được gọi là một chuỗi phổ biến đóng, tức là, có không tồn tại S_b như vậy mà $S_a \subseteq S_B$ và $sup^D(S_a) = sup^D(S_b)$. Vấn đề khai thác chuỗi phổ biến đóng là để tìm một bộ hoàn chỉnh các phổ biến đóng trình tự do cho một cơ sở dữ liệu D chuỗi đầu vào và một ngưỡng hỗ trợ tối thiểu cho minSup.

Ví dụ : Hãy xem xét các cơ sở dữ liệu trình tự trong Table 1. Các cơ sở dữ liệu có năm mặt hàng độc đáo I = { A, B , C , D , E} Ví dụ và bốn giao dịch, tức là, |Dj| =4. Giả sử rằng các ngưỡng hỗ trợ tối thiểu là minSup = 2 (50%). Nếu tất cả các trình tự thường xuyên của D được khai thác với minSup định, 32 chuỗi sau thu được: S_{FS}={A:4, AA: 4, AB: 3, AC: 4, (AC): 2, AAB: 2, AAC : 2, A (AC): 2, ABA: 3, ABB: 3, ABC: 3, A (BC): 3, ACA: 2, ACB: 2, ABAB: 2, AB (BC): 2, A (BC) A: 2, A (BC) B: 2, B: 3, BA: 3, BB: 3, BC: 3, (BC): 3, BAB: 2, B (BC): 2, (BC) A: 2, (BC) B: 2, C: 4, CA: 3, CB: 2, CC: 2, CAC: 2}. Ngược lại, khai thác phổn biến đóng các chuỗi sản lượng S_{FCS} = {AA: 4, AC: 4, AAC: 2, A (AC): 2, ABA: 3, ABB: 3, ABC: 3, A (BC): 3, ABAB: 2, AB (BC): 2, A (BC) A: 2, A (BC) B: 2, CA: 3, CAC: 2}, trong đó chỉ có 14 chuỗi.

Bảng 2.1.5 - **Table 1**

Ví dụ cơ sở dữ liệu sequence D.

ID	Sequence
1	CAA(AC)
2	AB(ABC)B

3	A(BC)ABCE
4	AB(BC)AD

Chuỗi phổ biến đóng S_{FCS} là như vậy, nhỏ gọn hơn so trình tự phổn biến S_{FS}. Điều này là do dãy chuỗi S_a với sự hỗ trợ tương tự như của siêu chuỗi S_b bị hấp thụ bởi S_b mà không ảnh hưởng đến kết quả khai thác. Ví dụ, trình tự (BC)A: 2 được hấp thụ bởi chuỗi A(BC)A: 2 vì (BC)A DA(BC)A và $sup^D((BC)A) = sup^D(A(BC)A) = 2$.

Lúc đầu, các chuỗi phổn biến với chiều dài 1 được khai thác từ một cơ sở dữ liệu chuỗi. Sau đó, những trình tự này phổ biến sẽ kết hợp (hoặc mở rộng) với nhau để tạo thành ứng cử viên mới có chiều dài 2. Quá trình này được lặp đi lặp lại cho đến khi không phổ biến tạo ra các chuỗi mới. Nói chung, các chuỗi với độ dài k được sử dụng để tạo ra các chuỗi với độ dài k + 1. Bên cạnh đó các tỉa bớt và kiểm tra các chuỗi phổ biến đóng được áp dụng trong mỗi quá trình.

Định nghĩa 1: (chuỗi con của một chuỗi). Gọi S là một dãy. $sub_{ij}(S)(i \leq j)$ được định nghĩa là một chuỗi có độ dài (i – j + 1) từ vị trí i đến vị trí j của S. Ví dụ, $sub_{1,3}$(BABC) là BAB và $sub_{4,4}$(BABC) là C.

Định nghĩa 2. (mở rộng một chuỗi từ một 1-sequence). Hãy cho α và β là hai phổ biến 1-sequence. $\{t_\alpha , p_\alpha\}$và $\{t_\beta , p_\beta\}$ là những giao dịch và vị trí của chuỗi α và β, tương ứng. Có hai hình thức gia hạn tự

Phần mở rộng tập phổ biến: $<(\alpha\beta)>\{t_\beta , p_\beta \}$, nếu $(\alpha < \beta) \wedge (t_\alpha = t_\beta) \wedge (p_\alpha = p_\beta)$ (2.1)

Mở rộng chuỗi : $<\alpha\beta> \{t_\beta , p_\beta \}$, nếu $(\alpha < \beta) \wedge (t_\alpha = t_\beta)$ (2.2)

Định nghĩa 3. (mở rộng một chuỗi từ một k-sequence). Hãy cho α và β là hai phổ biến k-sequence (k > 1), $u = sub_{k,k}(\alpha)$, và $v = sub_{k,k}(\beta)$. $\{t_\alpha , p_\alpha\}$và $\{t_\beta , p_\beta\}$ là những giao dịch và vị trí của chuỗi a và β, tương ứng. Có hai hình thức gia hạn tự

Mở rộng tập phổ biến: $\alpha + {}_i\beta = sub_{1,k-1}(\alpha)(uv)\{t_\beta , p_\beta\}$

Nếu $(u < v) \wedge (t_\alpha = t_\beta) \wedge (p_\alpha = p_\beta) \wedge (sub_{1,k-1}(\alpha) = sub_{1,k-1}(\beta))$ (3.1)

Mở rộng chuỗi: $\alpha + {}_s\beta = \alpha v\{t_\beta , p_\beta\}$

Nếu $(t_\alpha = t_\beta) \wedge (p_\alpha = p_\beta) \wedge (sub_{1,k-1}(\alpha) = sub_{1,k-1}(\beta))$ (3.2)

Định nghĩa 4. Hãy cho S= $e_1e_2 \cdots e_n$. Một mục e' có thể được thêm vào một phần mở rộng mô hình của S trong một trong ba vị trí

S' =$e_1e_2..e_ne' \wedge (sup^D(S') = sup^D(S))$ (4.1)

$\exists i(1 \leq i < n)$ mà S' = $e_1e_2..e_ie'..e_n \wedge (sup^D(S') = sup^D(S))$ (4.2)

S' = $e'e_1e_2..e_n \wedge (sup^D(S') = sup^D(S))$ (4.3)

Trong (DBV data structure), mục e' xuất hiện sau khi e_n, vì vậy mục e' được gọi là một tiền tố mở rộng và S' được gọi là một tiền tố chuỗi mở rộng(forward-extension). Ví dụ, chuỗi AC: 4 là một tiền tố có phần mở rộng của chuỗi A: 4 vì dãy C được mở rộng sau khi chuỗi A và hỗ trợ của họ là 4. (CloFS-DBVPattern data structure) và (CloFS-DBV algorithm), mục e' xuất hiện trước e_n, vì vậy mục e' được gọi là hậu tố - phần mở rộng và S' được gọi là một trình tự ngược mở rộng(backward- extension).

Ví dụ, chuỗi CAC: 2 là ngược mở rộng của chuỗi CC: 2 vì dãy A được mở rộng ở giữa dãy CC và hỗ trợ là 2.

Định nghĩa 5. Cho S= $e_1e_2 \cdots e_n$. Các vị trí bắt đầu của chuỗi S là vị trí của sự xuất hiện đầu tiên của e_1 tập phổ biến. Ví dụ, chuỗi AB(ABC)CB, vị trí bắt đầu của chuỗi (ABC) là 3, và của chuỗi ABB là 1.

c) Công việc có liên quan

Khai thác chuỗi phổ biến được đề xuất lần đầu vào năm 1995 bởi Agrawal và Srikant với thuật toán AprioriAll của họ, mà là dựa trên tài sản Apriori. Agrawal và Srikant sau đó mở rộng các vấn đề khai thác một cách tổng quát với các thuật toán GSP[11]. Kể từ đó, nhiều thuật toán khai thác chuỗi thường xuyên đã được đề xuất để nâng cao hiệu quả khai thác. Các thuật toán sử dụng phương pháp tiếp cận khác nhau để tổ chức dữ liệu và lưu trữ thông tin khai thác. thuật toán điển hình bao gồm SPADE[17], PrefixSpan[13], SPAM[16] và LAPIN-SPAM[15]. Các thuật toán Spam tổ chức dữ liệu trong một định dạng bitmap dọc và sử dụng một cấu trúc cây từ điển để lưu trữ thông tin khai thác. PrefixSpan sử dụng phép chiếu cơ sở dữ liệu cho phần mở rộng chuỗi để giảm không gian tìm kiếm, với các dữ liệu được trình bày theo chiều ngang. Các thuật toán Lapin-Spam sử dụng một danh sách để lưu trữ

các vị trí cuối cùng của các mặt hàng và một tập hợp các vị trí ranh giới của tiền tố để giảm phạm vi của không gian tìm kiếm.

Các thuật toán khác nhau đã tìm hiểu để khai thác trình tự thường xuyên không dự phòng để giảm không gian lưu trữ cần thiết và thời gian chạy cho các quy tắc khai thác mỏ. Khai thác phổ biến đóng các thuật toán khai thác tập phổ biến bao gồm A-CLOSE[39], CLOSET[40], CHAR[21], và CLOSET+[35]. Hầu hết các thuật toán giữ gìn khai thác tập phổ biến để kiểm tra chuỗi khép kín thường xuyên, đòi hỏi nhiều bộ nhớ. CLOSET+ sử dụng một hai cấp băm - cấu trúc chỉ mục và một cấu trúc cây để lưu trữ các tập phổ biến để giảm bộ nhớ không gian và thời gian cần thiết để thử nghiệm tập phổ biến đóng cửa. CloSpan[20] sử dụng một phương pháp mô hình duy trì và kiểm tra và kết hợp một cấu trúc băm-index với một cấu trúc cây để lưu trữ các chuỗi. Đây thuật toán tỉa mô hình sử dụng các kỹ thuật như Tiền tố chung và ngược Sub-Pattern để giảm không gian tìm kiếm. Clasp[41] thuật toán sử dụng một chiến lược định dạng cơ sở dữ liệu theo chiều dọc, như thực hiện bởi các thuật toán Spade, và một heuristic để tỉa trình tự không khép kín, như thực hiện bởi các thuật toán CloSpan. Tuy nhiên, thuật toán duy trì các ứng cử viên trước để kiểm tra việc đóng cửa các trình tự và loại bỏ chúng sau này. Việc duy trì các ứng viên làm tăng mức tiêu thụ bộ nhớ, và số lượng thí sinh kiểm tra tăng lên cùng với số lượng chuỗi phổ biến đóng tạo ra.

d) Thuật toán tìm hiểu

Mô tả thuật toán CloFS-DBV, trong đó sử dụng một vector bit năng động (DBV) cấu trúc kết hợp với thông tin vị trí trong cấu trúc của giao dịch CloFS-DBVPattern mỏ thường xuyên trình tự đóng lại.

DBV data structure

Các thuật toán khai thác chuỗi dựa trên một định dạng dữ liệu theo chiều dọc đã được chứng minh là có hiệu quả hơn những người dựa trên một định dạng dữ liệu ngang. Thuật toán điển hình mà sử dụng một định dạng dọc bao gồm SPADE[17], DISC-all [42], HVSM[36] và MSGPs[43]. Các thuật toán quét các cơ sở dữ liệu một lần duy nhất và tính toán sự hỗ trợ của các trình tự một cách nhanh

chóng. Tuy nhiên, bất lợi là họ tiêu thụ nhiều bộ nhớ để lưu trữ các thông tin bổ sung. BitTableFI[44]và Index-BitTableFI [45] đã giải quyết được vấn đề này bằng cách nén dữ liệu bằng cách sử dụng một bảng bit (BitTable).

Hạn chế chính của cấu trúc vector bit là một kích thước cố định, mà phụ thuộc vào số lượng giao dịch trong một cơ sở dữ liệu chuỗi. '1' chỉ ra rằng các mục xuất hiện trong các giao dịch và '0' chỉ khác. Trong thực tế, thường có rất nhiều '0' bit trong một vector bit, tức là, các mục trong cơ sở dữ liệu trình tự ngẫu nhiên thường xuất hiện trong cơ sở dữ liệu chuỗi. Ngoài ra, trong quá trình mở rộng của chuỗi (sử dụng bitwise AND) của '0' bit sẽ xuất hiện nhiều hơn. Do đó làm tăng bộ nhớ và xử lý yêu cầu thời gian. Để khắc phục vấn đề này, kiến trúc vector bit động được sử dụng[46]. Gọi A và B là hai vectơ bit. p_1 và p_2 là xác suất của các bit '1' trong hai vectơ bit A và B, tương ứng. Giả sử k là xác suất của '0' bit sau khi gia nhập A và B để có được AB bởi quá trình mở rộng của chuỗi. Do đó, xác suất của các bit '1' trong vector bit AB là $\min(p_1 , p_2) - k$, nơi $\min(p_1 , p_2)$ là giá trị tối thiểu của p_1 và p_2. Rõ ràng, xác suất của '1' trong AB sẽ giảm trong tương phản xác suất của '0' trong tăng đó. Hơn nữa, khoảng cách giữa p_1 và p_2 sẽ lớn hơn một cách nhanh chóng sau một vài mở rộng chuỗi.

Giả sử có 16 giao dịch trong một cơ sở dữ liệu chuỗi. Một item i tồn tại trong giao dịch 7, 9, 10, 11, và 13. Các vector bit cho các mục i cần 16 byte, như thể hiện trong Table 2. non-zero byte đầu tiên xuất hiện ở chỉ số 7. DBV chỉ các cửa hàng bắt đầu từ chỉ số và chuỗi các byte bắt đầu từ non-zero byte đầu tiên cho đến khi không zero byte cuối cùng, như thể hiện trong Table 3. Chỉ có 8 byte được yêu cầu để lưu trữ các thông tin bằng cách sử dụng cấu trúc DBV.

Bảng 2.1.6 - **Table 2**

Ví dụ về bit vector 16-byte.

0	0	0	0	0	0	1	0	1	1	1	0	1	0	0	0

Mỗi DBV bao gồm hai phần:

(1) Bắt đầu bit: vị trí của sự xuất hiện đầu tiên của '1'

(2) Bit vector: chuỗi các bit bắt đầu từ non-zero byte đầu tiên cho đến khi

không zero byte cuối cùng. Cơ cấu DBV được sử dụng để lưu trữ các giao dịch trong một định dạng thẳng đứng. Hỗ trợ trình tự có thể dễ dàng được tính toán bằng cách đếm số '1' bit.

Ví dụ. Xem xét cơ sở dữ liệu D trong Table 1. Trình tự A tồn tại trong giao dịch 1, 2, 3, và 4, do đó các bit bắt đầu là 1, và các vector bit là 1111. Các vector bit có bốn '1' bit, do đó, hỗ trợ của dãy A là 4. Trình tự B tồn tại trong các giao dịch 2, 3 và 4, do đó các bit bắt đầu là 2, và do đó các vector bit là 111. các vector bit có ba '1' bit, do đó, sự hỗ trợ của dãy B là 3. Bảng 4 cho thấy sự chuyển đổi của cơ sở dữ liệu D trong bảng 1 để định dạng DBV.

CloFS-DBVPattern data structure

Cơ cấu CloFS-DBVPattern kết hợp một cấu trúc DBV với một đại diện của thông tin trình tự. Mỗi CloFS-DBVPattern bao gồm hai phần:

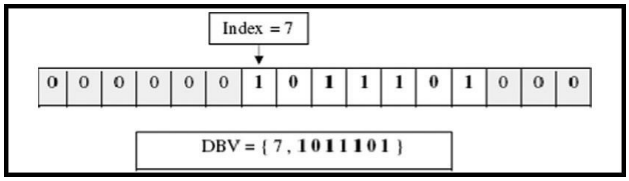
(1) Sequence: thông tin sequence

(2) BlockInfo: DBV và danh sách các vị trí xuất hiện trong các chuỗi giao dịch. vị trí danh sách của mỗi giao dịch được đại diện trong các hình thức startPos: (danh sách vị trí), startPos là sự xuất hiện đầu tiên của chuỗi trong mỗi giao dịch.

Ví dụ: Trong cơ sở dữ liệu D (Table 1), dãy A tồn tại trong giao dịch 1, 2, 3 và 4. Đối với các giao dịch đầu tiên, trình tự A sẽ xuất hiện tại các vị trí {2, 3, 4}. Các vị trí bắt đầu là 2, và vì thế 2: { 2, 3 , 4} được lưu trữ. Đối với các giao dịch thứ hai, dãy A sẽ xuất hiện tại các vị trí {1 , 3 }. Các vị trí bắt đầu là 1, và do đó 1: {1 , 3} được lưu trữ. Đối với các giao dịch thứ ba, thứ tự A sẽ xuất hiện tại các vị trí {1; 3}. Các vị trí bắt đầu là 1, và do đó 1: {1 , 3 } được lưu trữ. Tương tự như vậy, đối với các giao dịch cuối cùng, dãy A sẽ xuất hiện tại các vị trí {1 , 4} và vị trí bắt đầu là 1, do đó 1: {1 , 4 } được lưu trữ. Table 5 trình bày các CloFS-DBVPattern cho dãy A trong Table 1.

Bảng 2.1.7 - **Table 3**

Chuyển đổi các bit vector trong Table 2 về DBV.



Bảng 2.1.8 - **Table 4**

Chuyển đổi cơ sở dữ liệu của D trong bảng 1 để định dạng DBV.

Item	ID	Bit vector		Start bit	Bit vector	Value
A	1,2,3,4	1 1 1 1	Chuyển đổi để DBV	1	1 1 1 1	15
B	2, 3, 4	0 1 1 1		2	1 1 1	7
C	1, 2, 3, 4	1 1 1 1		1	1 1 1 1	15
D	4	0 0 0 1		4	1	1
E	3	0 0 1 0		3	1	1

Bảng 2.1.9 - **Table 5**

CloFS-DBV Pattern cho dãy A trong Table 1.

Sequence	A			
Start bit	1			
Value	15			
Index	4	3	2	1
Positions	1:{1,4}	1:{1,3}	1:{1,3}	2:{2,3,4}

Cây CloFS-DBV được sử dụng để lưu trữ CloFS-DBVPattern. Cây CloFS-DBV là một phần mở rộng của cây tiền tố. Các cây tiền tố có thể được xây dựng trong các cách sau đây. Các nút gốc của cây là ở cấp cao nhất và NULL dán nhãn. Đệ quy, mỗi nút X ở mức k trong cây có thể được mở rộng bằng cách thêm một mục để có được một X’ nút con ở cấp k + 1. Các con của nút X được tạo ra và sắp xếp theo thứ tự từ điển. Bằng cách sử dụng các cây tiền tố, thế hệ của các quy tắc tự trở nên hiệu quả hơn. Thuật toán điển hình để xây dựng một cây tiền tố bao gồm CloGen[47], IMSR_PreTree[48]. Trong cây CloFS-DBV, mỗi nút là một CloFS-DBVPattern: một sequence, một DBV, và một danh sách các vị trí của các sequence trong mỗi giao dịch. Mỗi nút trong cây được mở rộng trong hai hình thức: mở rộng

chuỗi(sequence extension) và phần mở rộng tập phổ biến(itemset extension). Hình 2.1.3 cho thấy ứng cử viên cho các cơ sở dữ liệu trong Table 1 thu được bằng cách sử dụng các thuật toán CloFS-DBV.

Thuật toán CloFS-DBV

Dự luật 1. (kiểm tra sequence đóng). Nếu tồn tại một chuỗi S_b đó là một quay về -mở rộng (forward-extension) hoặc ngược mở rộng(backward-extension) của chuỗi S_a, chuỗi S_a không phải là đóng, và S_a có thể được hấp thụ một cách an toàn bởi S_b. Xét ví dụ trên, giả sử rằng S_a = CC: 2 và S_b = CAC: 2. Sau đó, CC: 2 sẽ được hấp thụ bởi CAC vì CC ⊆ CAC và $sup^D(CC) = sup^D(CAC) = 2$.

Dự luật 2. (cắt tỉa một tiền tố). Hãy xem xét một tiền tố $S_p = e_1e_2 \cdots e_n$. Nếu tồn tại một mục e trước khi vị trí bắt đầu của tiền tố S_p trong mỗi giao dịch có chứa S_p trong chuỗi cơ sở dữ liệu D, phần mở rộng có thể được cắt tỉa bởi tiền tố Sp. Ví dụ, hãy xem xét các cơ sở dữ liệu D trong Table 1. Có được không cần phải mở rộng tiền tố B vì có tồn tại một dãy A xảy ra trước B trong mỗi giao dịch có chứa tiền tố B. Nếu chúng ta mở rộng tiền tố B, kết quả thu được sẽ được hấp thu do để mở rộng tiền tố A đã có chứa B và có sự hỗ trợ tương tự.

Các thuật toán CloFS-DBV bao gồm bốn giai đoạn chính:

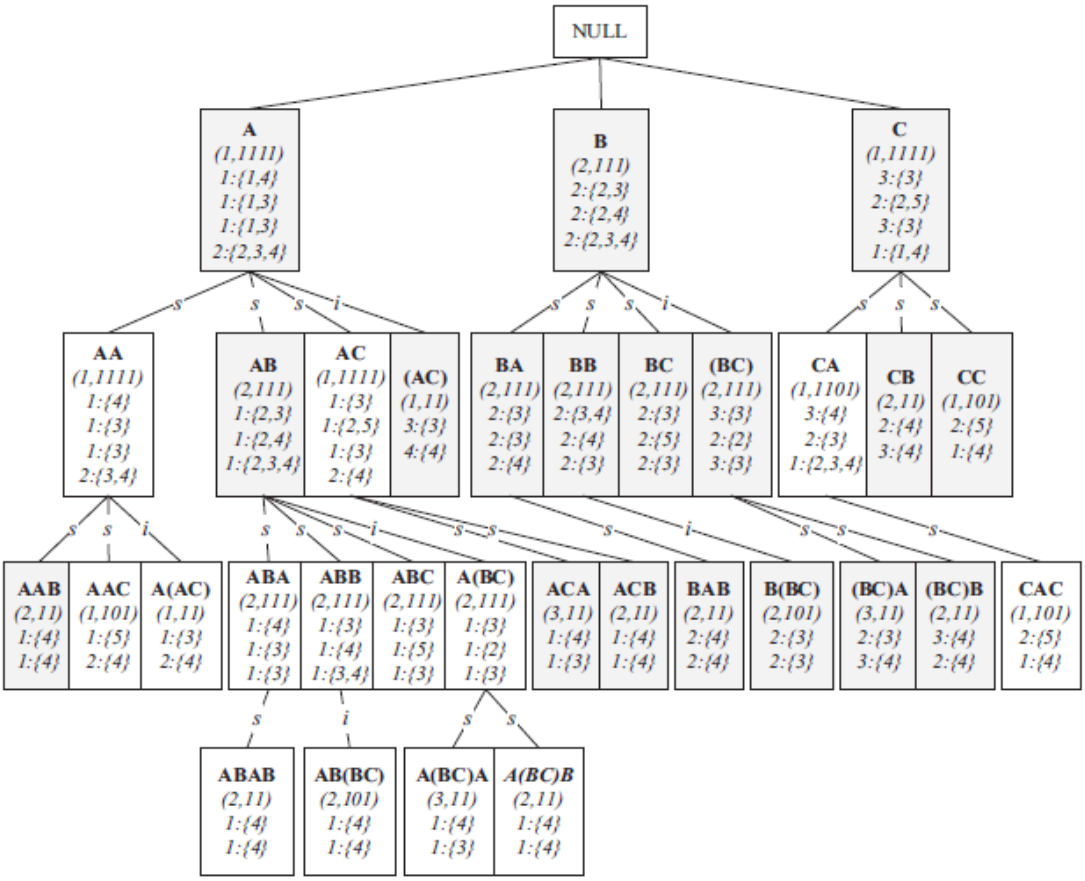
(1) chuyển đổi cơ sở dữ liệu sequence về cấu trúc CloFS-DBVPattern,

(2) kiểm tra các chuỗi phổ biến đóng,

(3) cắt tỉa đầu của chuỗi tiền tố.

(4) mở rộng của chuỗi. Kể từ CloFS-DBV sử dụng cấu trúc CloFS-DBVPattern, nó có thể kiểm tra backward-extension và forward-extension nhanh chóng. Đối với mỗi giao dịch, CloFS-DBV chỉ xem xét khi vị trí bắt đầu hoặc vị trí cuối cùng của dãy. Do đó, nếu dãy có N giao dịch, các CloFS-DBV chỉ mất N để kiểm tra mỗi ứng cử viên. Ngược lại, Bide toán có hiệu quả hơn CloSpan trong hầu hết các trường hợp sử dụng một cơ sở dữ liệu địa phương để kiểm tra backward-extension và sử dụng một cơ sở dữ liệu địa phương dự để kiểm tra forward-extension, tức là nó phải quét từng khoản mục trên mỗi giao dịch trong cơ sở dữ liệu này. K là độ dài chuỗi, và N là số thứ tự giao dịch. Vì vậy, đòi hỏi phải chờ k x

N để kiểm tra mỗi ứng cử viên



Hình 2.1.3: CloFS-DBV cây cho cơ sở dữ liệu trong Table 1. Che bóng hình chữ nhật đại diện cho ứng cử viên mà không đóng. Hình chữ nhật không có chụp đại diện cho chuỗi phổ biến đóng. Đường biểu tượng s cho thấy phần mở rộng chuỗi. Đường với biểu tượng i chỉ ra phần mở rộng tập phổ biến.

Table 6 cho thấy các mã giả của đề xuất thuật toán CloFS-DBV. Các thuật toán đầu tiên quét cơ sở dữ liệu D để tìm 1 chuỗi phổ biến và lưu chúng vào các f_{CS1} như CloFS-DBVPattern (dòng 2). Sau đó, các mục trong f_{CS1} đều được sắp xếp trong thứ tự tăng dần (dòng 3), để giảm các bước trong giai đoạn mở rộng của các tập phổ biến. Dòng 6 các thuật toán thực hiện việc mở rộng chuỗi theo các nút con của FCS. root.

Table 7 cho thấy thuật toán DBV-Pattern-Extension gọi bằng thuật toán CloFS-DBV. Việc mở rộng chuỗi trong hai hình thức: mở rộng chuỗi (dòng 5) và

phần mở rộng tập phổ biến (dòng 8). Trước khi mở rộng chuỗi, các bài kiểm tra thuật toán và loại bỏ các tiền tố đó không thể mở rộng chuỗi phổ biến đóng sử dụng Proposition 2 (dòng 3). Quá trình thực hiện đệ quy (dòng 12) cho đến khi không có phổ biến đóng được tạo ra. Dòng 14 sử dụng Proposition 1 để kiểm tra S_p tiền tố. Nếu S_p không phải là một chuỗi phổ biến đóng, nó sẽ được thiết lập để NULL.

Bảng 2.1.10 - **Table 6**

Thuật toán CloFS-DBV

Phương pháp : CloFS-DBV (D, minSup)
Đầu vào: Một cơ sở dữ liệu trình tự D và một minSup ngưỡng hỗ trợ
Đầu ra: Một bộ đầy đủ các trình tự thường xuyên FCS khép kín
1. Hãy FCS.root = NULL;
2. Cho f cs_1 = {$i_{CloFS\text{-}DBVPattern(i)}$| i ∈ I sup(i) minSup};
3. Sắp xếp (f cs_1) tăng theo thứ tự của mục i;
4. Thêm f cs_1 đến nút con của FCS.root;
5. For (mỗi nút subnode con trong FCS.root) do
6. Gọi DBV-Pattern-Extension (subnode, minSup);
7. End For

Bảng 2.1.11 - **Table 7**

Thuật toán DBV-Pattern-Extension(DBV-Pattern-Extension algorithm)

Thuật toán DBV-Pattern-Extension.

Phương pháp: DBV- Pattern -Extension (root, minSup)
Đầu vào: Một gốc của tiền tố và một minSup
Đầu ra: Một bộ thường xuyên chuỗi gốc đóng
1. Hãy để nút list_node = nút con của root;
2. For (mỗi S_p trong list_node) do
3. If (S_p không tỉa) then
4. For (mỗi S_a trong list_node) do
5. If (sup(Hãy S_{pa}=Sequence-Extension (S_p, S_a)≥ minSup) then

6.	Thêm S_{pa} tới nút con của S_p;
7.	End if
8.	If (sup(Hãy S_{pa} = Itemset-Extension (S_p, S_a) ≥ minsup) then
9.	Thêm S_{pa} tới nút con của S_p;
10.	End if
11.	End for
12.	Gọi DBV- Pattern -Extension (S_p, minSup)
13.	End if
14.	If (S_p không phải là một chuỗi phổ biến đóng) then
15.	Hãy S_p = NULL;
16.	End if

Ví dụ này cho thấy phần mở rộng sequence cho các thuật toán CloFS-DBV với cơ sở dữ liệu trình tự D trong Table 1 và 2 minSup = 2(50%). Sau khi dòng 2 (Table 6) được thực thi, ba thường xuyên 1 chuỗi được lưu trữ, tức là, f_{CS1} = {A: 4 , B: 3 , C: 4} (Table 8).

Bảng 2.1.12 - **Table 8**

Sequences A, B, và C trong cơ sở dữ liệu mẫu sau khi chuyển đổi để CloFS-DBV-Pattern

Sequence	A				B			C			
tart bit	1				2			15			
Value	15				7			4			
Index	4	3	2	1	4	3	2	4	3	2	1
Positions	1:{1,4}	1:{1,3}	1:{1,3}	2:{2,3,4}	2;{2,3}	2:{2,4}	2;{2,3,4}	3:{3}	2;{2,5}	3:{3}	1:{1,4}

Trong ví dụ này, tiền tố A không phải là một chuỗi đóng sau khi quá trình backward-extension và tiền tố B có thể được lược bớt khi quá trình tiền tố tỉa. Các thuật toán thực hiện mở rộng chuỗi để tạo ra phổ biến đóng 2 chuỗi mới. Bắt đầu với tiền tố A, thu được mở rộng với các trình tự A, B, và C trong các hình thức gia hạn liên tục (Table 9a) và phần mở rộng tập phổ biến (Bảng 9b).

Bảng 2.1.13 - **Table 9**

Ví dụ về (a) mở rộng chuỗi(sequence extension) và (b) phần mở rộng tập phổ biến (itemset extension) cho tiền tố A

Sequence	AA				Sequence	(AB)			
Start bit	1				Start bit	2			
Value	15				Value	1			
Index	4	3	2	1	Index	4	3	2	1
Positions A	1:{1,4}	1:{1,3}	1:{1,3}	2:{2,3,4}	Positions A	1:{1,4}	1:{1,3}	1:{1,3}	2:{2,3,4}
Positions A	1:{1,4}	1:{1,3}	1:{1,3}	2:{2,3,4}	Positions B	2:{2,3}	2:{2,4}	2:{2,3,4}	
Positions AA	1:{4}	1:{3}	1:{3}	2:{3,4}	Positions (AB)	∅	∅	3:{3}	∅
Sequence	AB				Sequence	(AC)			
Start bit	2				Start bit	1			
Value	7				Value	3			
Index	4	3	2	1	Index	4	3	2	1
Positions A	1:{1,4}	1:{1,3}	1:{1,3}	2:{2,3,4}	Positions A	1:{1,4}	1:{1,3}	1:{1,3}	2:{2,3,4}
Positions B	2:{2,3}	2:{2,4}	2:{2,3,4}		Positions C	3:{3}	2:{2,5}	3:{3}	1:{1,4}
Positions AB	1:{2,3}	1:{2,4}	1:{2,3,4}	∅	Positions (AC)	∅	∅	3:{3}	4:{4}
Sequence	AC								
Start bit	1								
Value	15								
Index	4	3	2	1					
Positions A	1:{1,4}	1:{1,3}	1:{1,3}	2:{2,3,4}					
Positions C	3:{3}	2:{2,5}	3:{3}	1:{1,4}					
Positions AC	1:{3}	1:{2,5}	1:{3}	2:{4}					
(a) Sequence-extension					(b) Itemset-extension				

Bảng 2.1.14 -**Table 10**

Định nghĩa của các thông số cho cơ sở dữ liệu tiêu chuẩn từ IBM.

C	Số lượng trung bình của một tập mỗi dãy
T	Số lượng trung bình của các mặt hàng trên mỗi tập phổ biến
S	Số lượng trung bình của một tập trong các trình tự tối đa
I	Số lượng trung bình của các mục trong trình tự tối đa
N	Số mục riêng biệt
D	Số trình tự

Vị trí 3 và 4 của itemset (AB) là trống rỗng, vì vậy các bit tương ứng với những vị trí được đặt là '0' và tập phổ biến này được lấy ra ($sup^D((AB)) = 1 <$ minSup). Quá trình mở rộng tiếp tục cho đến khi không có ứng cử viên được tạo ra. Các kết quả thu được được thể hiện trong Hình. 2.4.

2.1.7. Nhận xét

Các thuật toán CloFS-DBV, trong đó sử dụng DBV và thông tin giao phổ biến đóng các sequence. Các thuật toán CloFS-DBV được chia thành hai giai đoạn chính:

(1) cơ sở dữ liệu sequence ban đầu được chuyển đổi thành một mẫu định dạng dữ liệu dọc gọi là CloFS-DBV, nơi mà mỗi sequence CloS-DBV lưu vị trí của chuỗi phổ biến trong cơ sở dữ liệu

(2) chuỗi phổ biến đóng được tạo ra và thử nghiệm, và các tiền tố được tỉa sớm. Các thuật toán CloFS-DBV quét cơ sở dữ liệu một lần duy nhất và tính toán hỗ trợ dựa trên DBV để tạo ra các mẫu mới. Do việc sử dụng nó trong một cấu trúc nén, thuật toán CloFS-DBV là hiệu quả hơn so với chờ và thuật toán CloSpan về cách sử dụng bộ nhớ và thời gian chạy.

Các thuật toán CloFS-DBV có một vài hạn chế đó sẽ được giải quyết trong tương lai. Phổ biến đóng liên chuỗi sẽ được khai thác để giảm số lượng các mô hình dự phòng. Dựa vào khai thác thường xuyên đóng cửa liên trình tự, thế hệ của luật này sẽ được làm nhỏ gọn và hiệu quả hơn. Ngoài ra, khai thác chuỗi biến tối đại đã được đề xuất trong những năm gần đây. Cấu trúc dữ liệu DBV sẽ được áp dụng cho việc khai thác có hiệu quả các sequence.

2.2. Khai thác luật

2.2.1. Định nghĩa luật

Luật tuần tự biểu diễn mối quan hệ giữa hai loạt sự kiện xảy ra tuần tự, biểu thị dưới dạng X→Y (sup, conf), trong đó X là loạt sự kiện xảy ra trước, Y là loạt sự kiện xảy ra sau, sup là giá trị độ hỗ trợ và conf là giá trị độ tin cậy của luật [4].

Từ mẫu chuỗi đã có, luật tuần tự được xây dựng bằng cách tách mẫu chuỗi ra làm hai phần: phần tiền tố X và phần hậu tố Y (nối tiền tố với hậu tố: X++Y, ta được mẫu chuỗi như ban đầu). Độ hỗ trợ và độ tin cậy của luật được xác định như sau:

Độ hỗ trợ: sup = sup(X++Y) ×100%

Độ tin cậy: conf = sup(X++Y)/sup(X)×100%

Độ hỗ trợ của một luật bằng số chuỗi trong CSDL có chứa mẫu chuỗi tạo nên luật. Như vậy độ hỗ trợ của luật bằng độ hỗ trợ của mẫu chuỗi sinh ra luật.

Độ tin cậy của một luật r bằng với khả năng chuỗi trong CSDL có chứa tiền kiện của luật dẫn đến chứa hậu kiện của luật. Một luật có độ hỗ trợ cao hơn minSup

thì luật đó được coi là phổ biến. Tương tự, nếu luật có độ tin cậy cao hơn ngưỡng tin cậy tối thiểu (minimum confidence), kí hiệu là minConf, thì được coi là đáng tin cậy.

Với mỗi mẫu chuỗi kích thước k, có thể tạo ra (k-1) luật vì mẫu chuỗi kích thước k sẽ có (k-1) tiền tố. Ví dụ, với mẫu chuỗi A(BC)D có kích thước là 3, có thể tạo ra 2 luật là A→(BC)D, A(BC)→D.

2.2.2. Phát biểu bài toán khai thác luật

Khai thác luật là đi tìm ra những luật thỏa mãn tối thiểu ngưỡng minSup và minConf cho trước. Quá trình này gồm hai giai đoạn:

Giai đoạn 1: Tìm tất cả các mẫu chuỗi từ CSDL, tức đi tìm tất cả các mẫu f sao cho sup(f) ≥ minSup.

Giai đoạn 2: Sinh luật tin cậy từ các mẫu chuỗi tìm được ở giai đoạn 1, tức là tìm tất cả các luật r thỏa sup(r) ≥ minConf.

Bảng 2.2.1 - CSDL Chuỗi

SID	Chuỗi dữ liệu
1	(AB)BB(AB)B(AC)
2	(AB)(BC)(BC)
3	B(AB)
4	BB(BC)
5	(AB)(AB)(AB)A(BC)

Ví dụ: Cho CSDL chuỗi D như bảng 2.2.1, độ hỗ trợ tối thiểu minSup= 50%, và độ tin cậy tối thiểu minConf=70%. Nếu khai thác luật tuần tự từ CSDL này ta được kết quả như sau:

Tìm tập mẫu chuỗi gồm các mẫu có độ hỗ trợ ≥ minSup, tức support 50%×5≈ 3. Tập mẫu chuỗi tìm được như bảng 2.2.2.

Bảng 2.2.2 - Tập mẫu chuỗi

Kích thước	Mẫu tuần tự: Độ hỗ trợ
1	A: 4, B: 5, C: 4, (AB): 4, (BC) : 3

2	AB: 3, AC: 3, (AB)B: 3, (AC)C: 3, BA : 3, BB: 5, BC: 4, B(AB): 3, B(BC) : 3
3	ABB: 3, ABC: 3, BBB: 4, BBC : 4, (AB)BB : 3, (AB)BC: 3, BB(BC) : 3

Với tập mẫu chuỗi tìm được, ta có tập luật như bảng 2.2.3 (chỉ sinh luật từ những mẫu có kích thước lớn hơn 1).

Bảng 2.2.3 - Tập luật sinh từ tập mẫu chuỗi

Mẫu tuần tự	Luật conf=sup(X++Y)/sup(X)×100%	Độ tin cậy conf ≥ minConf?
AB : 3	A → B , 3/4 x 100% = 75%	Có
AC: 3	A→C, 3/4×100% = 75%	Có
(AB)B: 3	(AB)→B, 3/4×100% = 75%	Có
(AB)C: 3	(AB)→C, 3/4×100% = 75%	Có
BA: 3	B→A, 3/5×100% = 60%	Không
BB: 5	B→B, 5/5×100% = 100%	Có
BC: 4	B→C, 4/5×100% = 80%	Có
B(AB): 3	B→(AB), 3/5×100% = 60%	Không
B(BC): 3	B →(BC), 3/5×100% = 60%	Không
ABB: 3	A→BB, 3/4×100% = 75% AB→B, 3/3×100% = 100%	Có Có
ABC: 3	A→BC, 3/4×100% = 75% AB→C, 3/3×100% = 100%	Có Có
BBB: 4	B→BB, 4/5×100% = 80% BB→B, 4/5×100% = 80%	Có Có
BBC: 4	B→BC, 4/5×100% = 80% BB→C, 4/5×100% = 80%	Có Có
(AB)BB: 3	(AB)→BB, 3/4×100% = 75% (AB)B→B, 3/3×100% = 100%	Có Có
(AB)BC: 3	(AB)→BC, 3/4×100% = 75% (AB)B→C, 3/3×100% = 100%	Có Có

BB(BC): 3	B→ B(BC), 3/5×100% = 60% BB→ (BC), 3/5×100% = 60%	Không Không

Vậy, với CSDL đã cho, có thể khai thác được 18 luật thỏa minSup và minConf.

2.2.3. Ý nghĩa của luật

Luật biểu diễn mối quan hệ giữa các mẫu theo thời gian. Có thể coi luật là mở rộng tự nhiên của chuỗi mẫu, tương tự như luật kết hợp là mở rộng tự nhiên của tập phổ biến [24]. Một luật tuần tự biểu thị dưới dạng X→Y, nghĩa là trong các chuỗi dữ liệu, nếu mẫu X xuất hiện thì mẫu Y cũng xuất hiện theo sau mẫu X với độ tin cậy cao. So với các mẫu tuần tự, các luật giúp ta hiểu tốt hơn về thứ tự thời gian thể hiện trong CSDL chuỗi. Ví dụ, một người mua đĩa phim Star Wars phần 4 sẽ mua tiếp phần 5 và phần 6.

Như vậy mẫu mua hàng (4, 5, 6) là mẫu thể hiện hoạt động mua. Tuy nhiên, trong thực tế một cửa hàng bán đĩa có hàng trăm khách hàng với sở thích khác nhau. Do đó, mẫu (4, 5, 6) có xu hướng xuất hiện với độ hỗ trợ thấp. Khai thác với độ hỗ trợ thấp vẫn trả về các mẫu, tuy nhiên sẽ có nhiều mẫu sai và không thích hợp. Nếu như sử dụng luật, có thể loại bỏ đi các mẫu sai như vậy bằng cách đưa ra khái niệm độ tin cậy cho tập mẫu. Chỉ có những luật thỏa ngưỡng hỗ trợ và ngưỡng tin cậy mới được khai thác.

Như vậy, thông qua luật tuần tự chúng ta có thể biết được loạt sự kiện nào thường sẽ xảy ra sau loạt sự kiện trước đó. Luật tuần tự tuy khá đơn giản nhưng những thông tin mà luật mang lại có nhiều ý nghĩa quan trọng, hỗ trợ không nhỏ cho quá trình ra quyết định, quản lý và có tính định hướng. Luật tuần tự rất hữu ích trong nhiều lĩnh vực: y dược, thương mại, công nghệ phần mềm. Một số ví dụ về ý nghĩa của luật tuần tự trong các lĩnh vực ứng dụng:

Phân tích thị trường: Nếu một khách hàng mua xe hơi thì sau đó khách hàng này sẽ mua bảo hiểm. Quy luật này rất hữu ích cho việc thiết kế chiến lược quảng cáo sản phẩm đối với khách hàng.

Y dược: Nếu một bệnh nhân bị sốt và giảm mức thrombosite, sau đó xuất

hiện những đốm đỏ trên da thì có khả năng bệnh nhân này mắc bệnh sốt xuất huyết. Luật tuần tự này giúp dự đoán bệnh để có hướng điều trị thích hợp cho bệnh nhân.

2.2.4. Khai thác luật từ tập mẫu chuỗi

Trong lĩnh vực khai thác luật, các nghiên cứu tập trung trên bài toán khai thác. Trong khi đó, chỉ có duy nhất một phương pháp cơ bản để khai thác tập luật do Spiliopoulou đề xuất. Từ những mô tả của phương pháp này, L_o cùng đồng sự đã khái quát thành thuật toán Full [5].

Spiliopoulou [6] đã đề xuất việc tạo ra tập luật đầy đủ (tất cả các luật thỏa ngưỡng hỗ trợ và tin cậy) từ tập đầy đủ các mẫu chuỗi (tất cả các mẫu chuỗi). Việc tạo ra tập luật đầy đủ tiêu tốn rất nhiều thời gian và bộ nhớ. Số lượng mẫu con phổ biến tỉ lệ lũy thừa với độ dài cực đại của mẫu cha. Cụ thể, nếu một mẫu chuỗi kích thước k là phổ biến thì 2^k mẫu con của nó cũng phổ biến. Mà mỗi mẫu chuỗi kích thước k có thể tạo ra (k-1) luật (tùy thuộc vào ngưỡng tin cậy tối thiểu). Do đó, số lượng luật sẽ gia tăng đồ sộ theo kích thước của mẫu. Thuật toán Full được trình bày trong hình 2.2.1 [5].

Trước hết, thuật toán tìm tập tất cả các mẫu chuỗi, là những mẫu chuỗi có độ hỗ trợ thỏa ngưỡng minSup. Với mỗi mẫu chuỗi trong tập chuỗi tìm được, thuật toán tiến hành sinh tất cả các luật có thể có ứng với chuỗi đó. Cụ thể:

Với mỗi mẫu chuỗi có kích thước k, có thể tạo ra (k-1) luật. Mỗi luật có dạng pre→post, trong đó pre là tiền tố của mẫu f và pre++post = f.

Do đó, với mỗi mẫu chuỗi, thuật toán xét lần lượt từng tiền tố. Với mỗi mẫu tiền tố, thuật toán phải duyệt toàn bộ tập mẫu chuỗi để tìm độ hỗ trợ của mẫu tiền tố, từ đó tính độ tin cậy có thể có nếu sinh luật ứng theo tiền tố này. Nếu độ tin cậy thỏa ngưỡng minConf thì xuất ra luật đó.

Nếu gọi n là số lượng mẫu của tập các mẫu chuỗi, k là kích thước trung bình của mẫu, thì độ phức tạp của thuật toán này là $O(n^2 \times k)$.

Thuật toán Full

Đầu vào: CSDL chuỗi, *minSup, minConf*

Kết quả: Tất cả các luật có ý nghĩa

Phương pháp thực hiện:

1. Tìm tập tất cả các mẫu chuỗi *Freq*, gồm các mẫu có độ hỗ trợ ≥ *minSup*
2. Với mỗi mẫu chuỗi *f* ∈ *Freq* thực hiện
3. Với mỗi tiền tố *pre* của *f* thực hiện
4. Duyệt tập mẫu chuỗi để tìm *sup(f)*
5. Đặt *post = px*, sao cho *pre++px = f*
6. Đặt *r = pre→post, sup = sup(f)* và *conf = sup(f)/sup(pre)*
7. Nếu *conf* ≥ *minConf* thì:
8. Xuất luật *r(sup, conf)*

Hình 2.2.1 - Thuật toán Full [4]

Ví dụ: Cho CSDL chuỗi như bảng 2.2.1, với minsup = 50% và minconf = 70%, kết quả tập luật như sau:

Bảng 2.2.4 - Tập luật có độ tin cậy ≥ minConf

Số luật	Luật : Độ tin cậy
1	A → B : 75%
2	A→C : 75%
3	(AB)→B : 75%
4	(AB)→C : 75%
5	B→B : 100%
6	B→C : 80%
7	A→BB : 75%
8	AB→B : 100%
9	A→BC : 75%
10	AB→C : 100%
11	B→BB : 80%

12	BB→B : 80%
13	B→BC : 80%
14	BB→C : 80%
15	(AB)→BB : 75%
16	(AB)B→B :100%
17	(AB)→BC : 75%
18	(AB)B→C : 100%

2.3. Tổng kết chương

Thứ nhất trình bày cơ sở lý thuyết về khai thác chuỗi phổ biến, các hướng tiếp cận về khai thác chuỗi. Đồng thời, đưa ra các dẫn chứng để chứng minh vì sao cần phải khai thác chuỗi phổ biến. Trình bày chi tiết về thuật toán khai thác chuỗi phổ biến đóng kết hợp của bit vectơ động cho khai thác phổ biến đóng.

Thứ hai trình bày các định nghĩa, bài toán và ý nghĩa của khai thác luật. Đồng thời, đưa ra các hướng tiếp cận, dẫn dắt đến việc cần phải sử dụng một thuật toán để khai thác tập luật và từ tập mẫu chuỗi.

CHƯƠNG 3: ỨNG DỤNG LUẬT TUẦN TỰ TRONG KHAI THÁC HÀNH VI SỬ DỤNG WEB

3. 1. Giới thiệu

Sử mở rộng của World Wide Web đã dẫn đến một số lượng lớn dữ liệu tổng hợp nhằm phục vụ cho lợi ích của người sử dụng Web. Do đó, việc áp dụng các kỹ thuật khai thác dữ liệu trên Web hiện nay là trọng tâm của các nhà nghiên cứu. Một số phương pháp khai thác dữ liệu được sử dụng để phát hiện các thông tin ẩn trong trang web. Tuy nhiên, khai thác Web không chỉ có nghĩa là áp dụng các kỹ thuật khai thác dữ liệu để lưu trữ dữ liệu Web. Các thuật toán đã được đề xuất và sửa đổi để phú hợp hơn với nhu cầu của Web. Hơn nữa, không chỉ có các thuật toán khai thác dữ liệu, mà còn trí tuệ nhân tạo, phục hồi thông tin và kỹ thuật xử lý ngôn ngữ tự nhiên cũng có thể được sử dụng một cách hiệu quả. Như vậy, khai thác Web đã được phát triển thành một lĩnh vực nghiên cứu tự trị.

3. 2. Các hướng tiếp cận

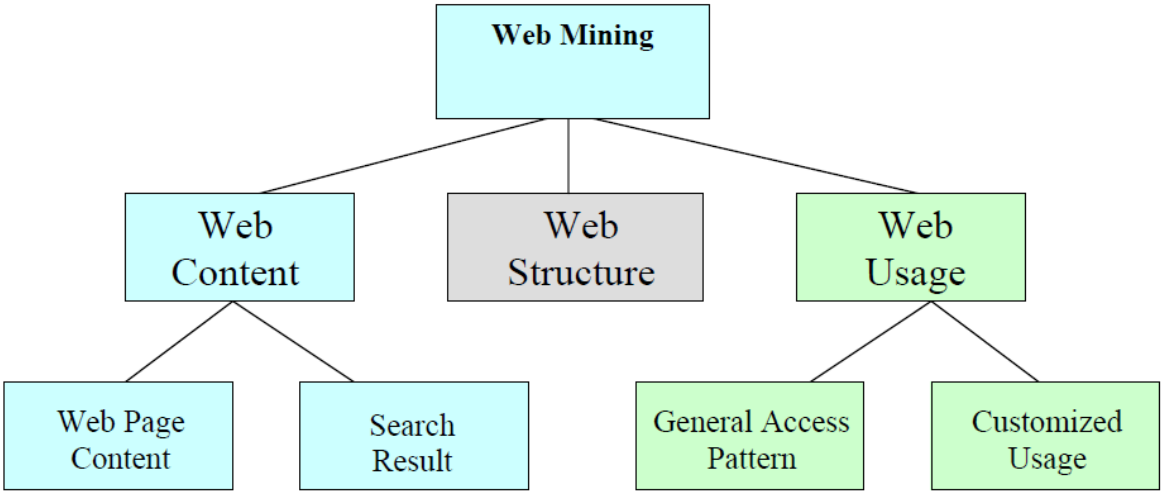
Khai thác Web liên quan đến một loạt các ứng dụng nhằn phát hiện và chiết xuất thông tin ẩn trong dữ liệu lưu trữ trên Web. Một mục đích quan trọng của khai thác Web là cung cấp một cơ chế để làm cho truy cập dữ liệu hiệu quả và đầy đủ hơn. Hơn nữa, là để khám phá những thông tin được bắt nguồn từ các hoạt động của người sử dụng, được lưu trữ trong các tập tin log. ví dụ cho tiên đoán Web [4] bộ nhớ đệm. Như vậy, khai thác Web được chia thành ba loại: Khai thác nội dung Web (Web Content Mining), khai thác cấu trúc Web (Web Structure Mining) và khai thác sử dụng Web (Web Usage Mining)

Khai thác nội dung Web (Web Content Mining): khai thác dữ liệu từ nội dung của trang web dựa trên mô hình kết hợp**.**

Khai thác cấu trúc Web (Web Structure Mining): mô tả cấu trúc quan hệ của các trang web và sử dụng để trích xuất thông tin từ các cấu trúc siêu liên kết.

Khai thác sử dụng web (Web Usage Mining): một quá trình khai thác thông tin hữu ích từ các bản ghi máy chủ. Khi người dùng sử dụng internet và 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.



Hình 3. 1 - Các hình thức khai thác Web

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 [1],[2],[27],[28].

Có ba loại tập tin log có thể được sử dụng để khai thác sử dụng Web. Web log có thể nằm ở server, hoặc client hoặc trên một proxy server, ở mỗi nơi đều có ưu và nhược điểm riêng cho việc tìm mẫu thích hợp của người dùng và phiên truy cập [1],[2],[27],[28]. Web log được ghi trên server phản ánh việc truy cập vào một trang web bởi nhiều người dùng, do đó dữ liệu web log được ghi trên server sẽ và tốt cho việc khai thác thói quen sử dụng web của nhiều người dùng; tuy nhiên, dữ liệu web log trên server có thể không hoàn toàn đáng tin cậy do việc lưu trữ, chẳng hạn như cached page views không được ghi lại trong server log. Bộ thu thập dữ liệu phía client đòi hỏi phải cài đặt một trạm (agent) điều khiển từ xa hoặc sử dụng một trình duyệt thay đổi để thu thập dữ liệu của từng người dùng đơn lẻ, do đó loại trừ được các vấn đề caching và nhận dạng session, và rất hữu ích cho các ứng dụng cá nhân hóa nội dung web. Mặt khác, một proxy server có thể đưa ra những yêu cầu HTTP thực sự từ nhiều client đến nhiều web servers, vì vậy web log lưu trên

proxy server mô tả thói quen duyệt web của một nhóm người dùng nặc danh cùng chia sẻ một server chung [30]. Vì vậy, hầu hết các thuật toán chỉ dựa vào các dữ liệu phía máy chủ. Một số thuật toán khai thác dữ liệu thường được sử dụng để khai thác sử dụng Web là luật kết hợp, luật tuần tự và phân nhóm [31].

Một số kỹ thuật khai thác mẫu tuần tự mà có thể áp dụng vào khai thác thói quen sử dụng web đã được giới thiệu vào giữa thập niên 90. Trước đây, các khảo sát nghiên cứu đã xem xét các phương pháp khai thác khác nhau áp dụng cho khai thác web log [1],[2],[27],[28],[30], nhưng chúng đều thiếu ba điểm quan trọng:

(i) thất bại với việc tập trung trên các mẫu tuần tự như là một giải pháp đầy đủ;

(ii) cung cấp phân loại

(iii) cung cấp một khảo sát kỹ lưỡng cho các lý thuyết và kỹ thuật dùng trong khai thác mẫu chuỗi.

3. 3. Ứng dụng của khai thác sử dụng Web

Khai thác sử dụng web đã trở thành một trong những lĩnh vực quan trọng trong máy tính và thông tin khoa học. Cách sử dụng các phương pháp khai thác trong dữ liệu Web log để trích xuất các hành vi của người sử dụng được sử dụng trong các ứng dụng khác nhau như:

- Tìm ra những khách hàng tiềm năng trong thương mại điện tử.
- Chính phủ điện tử (e-Gov), giáo dục điện tử (e-Learning).
- Xác định những quảng cáo tiềm năng.
- Nâng cao chất lượng truyền tải của các dịch vụ thông tin Internet đến người dùng cuối.
- Cải tiến hiệu suất hệ thống phục vụ của các máy chủ Web.
- Cá nhân dịch vụ Web thông qua việc phân tích các đặc tính cá nhân người dùng.
- Cải tiến thiết kế Web thông qua việc phân tích thói quen duyệt Web và phân tích các mẫu nội dung trang truy cập của người dùng.
- Phát hiện gian lận và xâm nhập bất hợp lệ trong dịch vụ thương mại điện tử và các dịch vụ Web khác.

- Thông qua việc phân tích chuỗi truy cập của người dùng để có thể dự báo những hành vi của người dùng trong quá tình tìm kiếm thông tin.

3.4. Khai thác sử dụng Web

Với sự tăng trưởng và phát triển không ngừng của thương mại điện tử, dịch vụ Web và hệ thống thông tin dựa trên nền tảng web, khối lượng của các clickstream và dữ liệu người dùng được các tổ chức Web thu thập bởi trong hoạt động hàng ngày đạt tỷ lệ cao. Việc phân tích dữ liệu đó có thể giúp các tổ chức Web xác định giá trị thời gian sống của khách hàng, qua đó thiết kế chiến lược tiếp thị trên các sản phẩm và dịch vụ, đánh giá hiệu quả của chiến dịch khuyến mãi, tối ưu hóa chức năng của các ứng dụng trên nền Web, cung cấp nội dung tốt hơn cho khách hàng, và tìm thấy một bố cục hợp lý hơn cho trang web của mình. Dạng phân tích này liên quan đến phát hiện các mô hình có ý nghĩa và các mối quan hệ từ một lượng lớn dữ liệu thường được lưu trữ trên web và các file log của ứng dụng cũng như trong các nguồn dữ liệu liên quan khác.

Khai thác sử dụng Web (*Web usage mining*) đề cập đến việc phát hiện và phân tích tự động các mẫu clickstream, dữ liệu liên quan xuất ra kết quả về sự tương tác của người dùng với nguồn tài nguyên Web trên một hoặc nhiều trang web. Mục tiêu là để thu thập, mô hình và phân tích các hành vi của người dùng tương tác với một trang web. Các mẫu thu thập được trong quá trình này thường là tập các trang web, thành phần hoặc tài nguyên mà một nhóm người dùng hay truy cập với cùng một mục đích chung.

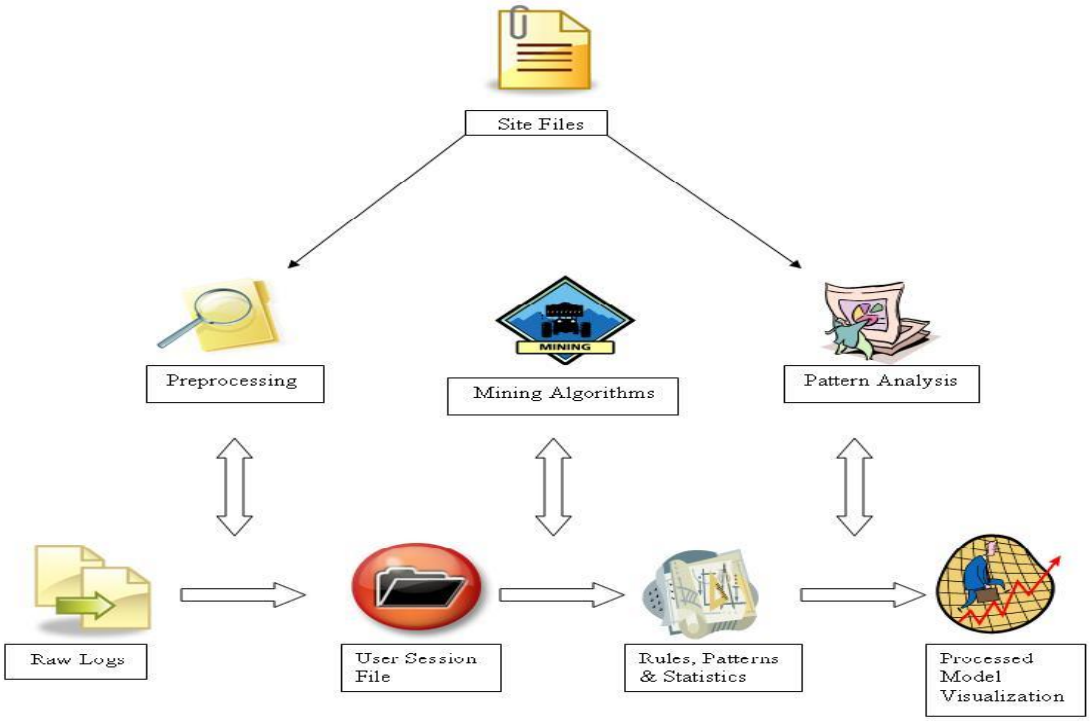
Khai thác sử dụng Web là việc xử lý các để lấy ra các thông tin hữu ích trong hồ sơ truy cập web. Thông thường các web server thường ghi lại và tích lũy các dữ liệu về tương tác của người dùng mỗi khi nó nhận được một yêu cầu truy cập. Việc phân tích các hồ sơ truy cập web của các website khác nhau sẽ dự đoán các tương tác của người dùng khi họ tương tác với website cũng như tìm hiểu cấu trúc của website, từ đó cải thiện các thiết kế của các hệ thống liên quan. Có hai xu hướng chính trong lĩnh vực này:

Phân tích các mẫu truy cập: phân tích các hồ sơ web để biết được các mẫu và

các xu hướng truy cập. Cách phân tích này có thể giúp cấu trúc lại các site trong các phân nhóm hiệu quả hơn hay xác định các vị trí quảng cáo hiệu quả nhất, cũng như gắn các quảng cáo sản phẩm nhất định cho những người dùng nhất định để đạt hiệu quả cao nhất.

Phân tích các xu hướng cá nhân: mục đích là để chuyên biệt hóa các website cho các lớp đối tượng người dùng. Các thông tin được hiển thị, độ sâu của cấu trúc trang và định dạng của các tài nguyên, tất cả đều có thể chuyên biệt hóa một cách tự động cho mỗi người dùng theo thời gian dựa trên các mẫu truy cập của họ.

Tổng thể quá trình khai thác sử dụng web có thể được chia thành ba giai đoạn liên thuộc nhau: *thu thập dữ liệu và tiền xử lý, khai thác mẫu* và *phân tích mẫu*. Quá trình tổng thể được mô tả như sau:



Hình 3. 2 - Kiến trúc tổng quát của khai thác dữ liệu theo sử dụng Web [27]

Khai thác thói quen sử dụng Web (còn gọi là khai thác mẫu truy cập Web – Web log mining) 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, ở đó các sự kiện có thứ tự trong mỗi chuỗi dữ liệu của CSDL chuỗi là một item đơn

chứ không phải là một tập các item, giả sử rằng một người dùng web chỉ có thể truy cập một trang web tại một thời điểm bất kỳ. Nếu có ràng buộc thời gian window-time đối với việc truy cập thì trong một khoảng thời gian cụ thể, người dùng web có thể duyệt một tập hợp nhiều trang web, khi đó CSDL truy cập web lúc này có dạng chung của CSDL chuỗi itemset. Hầu hết các giải pháp khai thác thói quen sử dụng web hiện nay đều coi việc truy cập của người dùng là chỉ truy cập một trang Web tại một thời điểm, vì vậy với trường hợp này, CSDL chuỗi có dạng đặc biệt, đó là mỗi sự kiện trong chuỗi chỉ có một item. Ví dụ, cho tập sự kiện *E={a, b, c, d, e, f}*, mỗi sự kiện đại diện cho một trang web mà người dùng đã truy cập trong một ứng dụng thương mại điện tử. Một CSDL gồm các chuỗi truy cập web của 4 người dùng sẽ có 4 bản ghi: [T1, *abdac*]; [T2, *eadbcac*]; [T3, *babfaec*]; [T4, *abfac*]. Khai thác mẫu truy cập web trên CSDL web này có thể cho ra chuỗi phổ biến là *abac*, nghĩa là trên 90% trong số những người vào trang web của sản phẩm *a hostname/producta.htm* cũng sẽ vào trang web của sản phẩm *b hostname/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, v.v…

Trong lĩnh vực khai thác sử dụng Web, một CSDL chuỗi các giao dịch đại diện bởi các truy cập web, với mỗi giao dịch *T* có một định danh duy nhất (Transaction id (TID) hoặc Session id (SID)). Bài toán khai thác mẫu tuần tự khi giới hạn trong phạm vi khai thác mẫu tuần tự web log với một CSDL *D*, *minSup* và tập các sự kiện *E = {a, b, c, d…}* đại diện cho địa chỉ các trang web với đặc điểm như sau:

(1) 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 *bcabdac* khác

với chuỗi dạng tổng quát (*ba*)(*ab*)*d*(*ac*).

(2) Trong khai thác sử dụng web, thứ tự đóng vai trò quan trọng, cũng như thứ tự các trang web được duyệt trong chuỗi các giao dịch 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, aab* có *a* là sự kiện xuất hiện lặp lại).

(3) 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, vì vậy các thuật toán thực hiện trên tập dữ liệu thưa. Tuy nhiên, kết quả thực nghiệm lại cho thấy rằng thuật toán LAPIN Suffix chỉ thực hiện tốt hơn thuật toán PrefixSpan đối với các tập dữ liệu lớn và tại một mức hỗ trợ cao hơn, mà tại mức hỗ trợ đó nhiều mẫu phổ biến không bao giờ xuất hiện.

3. 5. Thu thập và tiền xử lý dữ liệu

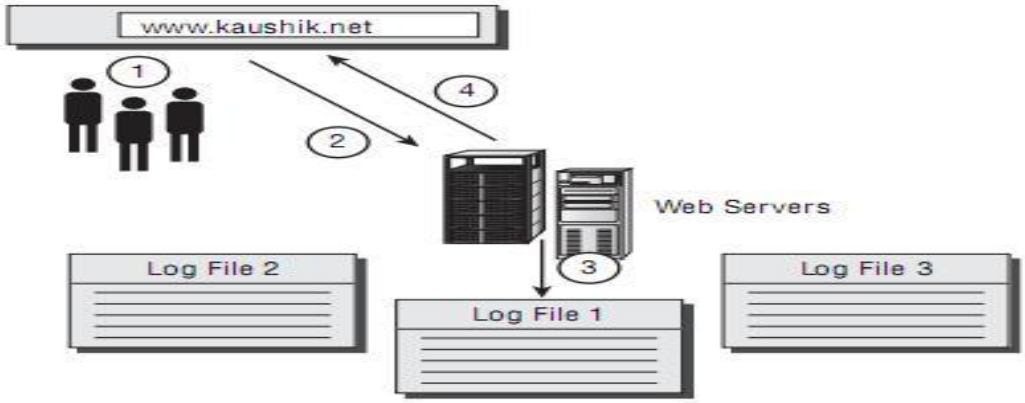
Một công việc quan trọng trong bất kỳ ứng dụng khai thác dữ liệu là việc tạo ra dữ liệu phù hợp để các giải thuật khai phá dữ liệu và thống kê có thể được áp dụng. Điều này đặc biệt quan trọng trong việc khai phá sử dụng web do các đặc điểm dữ liệu clickstream và mối quan hệ của nó với các dữ liệu khác 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.

Phần lớn các nghiên cứu và thực tiễn trong chuẩn bị dữ liệu thường tập trung vào tiền xử lý và tích hợp các nguồn dữ liệu để phân tích khác nhau. Việc chuẩn bị dữ liệu trình bày một số cách thức duy nhất đã dẫn đến một loạt các thuật toán và heuristic cho công việc tiền xử lý như tổng hợp và làm sạch dữ liệu, xác định người dùng và các giao dịch, xác định các trang (pageview identification). Việc áp dụng

các kỹ thuật khai phá dữ liệu có thành công hay không lại phụ thuộc phụ vào tính chính xác của dữ liệu trong quá trình tiền xử lý.

3.5.1 Thu thập dữ liệu

Web log là nguồn thu thập dữ liệu thông thường cho việc truy cập web. Lúc đầu nó chỉ được dùng để thu thập lỗi nhưng sau này đã được phát triển để thu thập nhiều thông tin hơn từ thông tin kỹ thuật đến thông tin nghiệp vụ.



Hình 3. 3 - Thu thập dữ liệu bằng Web log

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 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 Web log cũng khác nhau, có các định dạng: NCSA common, NCSA combined, định dạng mở rộng W3C format and định dạng IIS.

NCSA common [32]: ghi lại các thông tin cơ bản về các yêu cầu của người dùng, 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.

```
172.21.13.45 - Fred [08/Feb/2010:16:20:14 -0800] "GET /home.htm
HTTP/1.0" 200 3401
```

Hình 3. 4 - Định dạng tập tin log NCSA

Định dạng mở rộng W3C: lưu trữ thông tin nhiều hơn so với định dạng NCSA [32]. Định dạng bản ghi này có thể được tùy chỉnh, quản trị viên có thể thêm hoặc loại bỏ các lĩnh vực tùy thuộc vào thông tin gì họ muốn ghi lại. Những trường này là tách bằng dấu cách và thời gian được ghi nhận là GMT

```
#software: Microsoft Internet Information Services6.0
#version: 1.0
#Date: 2002-05-02 17:42:15
#Fields: date time c-ip cs-username s-ip s-port cs-method cs-uri-stem cs-uri-
query sc-status cs(User-Agent)
2002-05-02 17:42:15 172.22.255.255 - 172.30.255.255 80 GET
/images/p.jpg – 200 Mozilla/4.0+(
compatible;MSIE+5.5;+Windows+2000+Server)
```

Hình 3.5 - Định dạng tập tin log W3C

Định dạng IIS: cố định (không có thể được tùy chỉnh) các định dạng [32]. Các lĩnh vực được phân cách bằng dấu phẩy, làm cho định dạng dễ đọc hơn.

Thời gian được ghi nhận là giờ cục bộ. Các định dạng tập tin IIS log ghi lại các dữ liệu sau: địa chỉ IP, tên người dùng, ngày, giờ, dịch vụ và dụ, tên máy chủ, địa chỉ IP máy chủ, thời gian thực hiện, số byte khách hàng gửi, số bytes máy chủ gửi, mã trạng thái dịch vụ, mã trạng thái cửa sổ , loại yêu cầu, đường dẫn, các thông số.

```
192.168.114.201, -, 05/15/11, 7:55:20, W3SVC2, SERVER, 172.21.13.45,
4502, 163, 3223, 200, 0, GET,/index.htm, -,
```

Hình 3. 6 - Định dạng tập tin log IIS

Lợi ích khi dùng Web log

- Web log là nguồn thu thập dữ liệu đơn giản nhất. Mọi Web server đều có

khả năng tạo ra các file log.

- Web log có khả năng lưu trữ các truy xuất và hành vi của các máy tìm kiếm trên Website của mình. Các search engine robot không thực thi JavaScript tags, vì vậy nếu muốn phân tích truy cập từ Google, Microsoft Network, Yahoo Search ta phải dùng Web log.

Những vấn đề liên quan :

- Web log được tạo chủ yếu để thu thập những thông tin kỹ thuật (thông tin lỗi, truy cập, loại trình duyệt). Chúng không có tùy chọn để thu thập thông tin nghiệp vụ hay bán hàng.

- Nếu thông tin nghiệp vụ và bán hàng cần được thu thập thì phải có sự cộng tác của lực lượng IT và sự phụ thuộc vào thời gian biểu của họ.

- Web log được tạo để lưu trữ những truy suất đến server. Khi phân tích log file, cần phải loại bỏ những thông tin không cần thiết như hình ảnh, lỗi trang, css…

- Page caching từ các nhà cung cấp và proxy server làm cho Web server không thể lưu trữ toàn bộ thông tin truy cập (tỉ lệ khoảng 10%). Với page caching thông thường, một số trang Web như trang chủ, trang sản phẩm được lưu trữ tại các ISP hoặc proxy server. Vì thế khi người dùng gửi yêu cầu đến trang chủ thì ISP sẽ trả lời yêu cầu chứ không phải Webserver. Vì thế chúng ta không lưu trữ được thông tin này trên server.

- Thông tin được cung cấp có thể không đây đủ, không chi tiết.

- Không có thông tin về nội dung các trang đã được thăm.

- Có quá nhiều sự ghi lại các lần truy cập Web của người dùng, trong đó có những thông tin không cần thiết cho việc khai thác.

- Đặc biệt là việc lọc các chuỗi giao tác: các chuỗi giao tác với tên file mở rộng như gif, jpg, png, bmp,jpeg,…các trang yêu cầu tạo ra bởi các tác nhân tự động và các chương trình gián điệp.

- Ước lượng thời gian thăm trang: thời gian dùng để thăm một trang là một độ đo tốt cho vấn đề xác định mức độ quan tâm của người dùng đối với trang Web đó, nó cung cấp một sự đánh giá ngầm định đối với trang Web đó.

- Khoảng thời gian thăm trang: đó là khoảng thời gian giữa hai yêu cầu trang khác nhau liên tiếp.

Nội dung của một dạng Web log

Một file Web log là một tập các sự ghi lại những yêu cầu người dùng đối với một Website

```
66.249.79.45 - - [15/Jun/2015:04:02:55 +0700] "GET
/feed/rss.html HTTP/1.0" 200 429 "-" "Mozilla/5.0 (compatible;
Googlebot/2.1; +http://www.google.com/bot.html)"
66.249.79.19 - - [15/Jun/2015:04:06:53 +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/2015:04:14:06 +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/2015:04:14:09 +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 3. 7 - Một phần nội dung Web log

Thông tin này được ghi lại dưới dạng: ***host/ip user [date:time] "method url" status bytes "ReferenceUrl" "agent"***, nó được biểu diễn từ trái sang phải.

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 (12:03:24 p.m. vàoAug 30, 2009 tại 5 giờ sau giờ chuẩn Greenwich Mean Time (GMT)).

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.

ReferenceUrl: 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

Không thể ghi lại cached page views trên một log. Vì vậy, để tăng nguồn dữ liệu sử dụng Web của người dùng, cần sử dụng thêm một số kỹ thuật để tiền xử lý Web log. Trong hầu hết trường hợp, các nghiên cứu đều giả sử rằng thông tin duyệt web của người dùng được ghi lại đầy đủ trên Web server log, đã được tiền xử lý để thu được CSDL giao dịch dùng cho việc khai thác chuỗi.

3.5.2. Tiền xử lý dữ liệu

Trong giai đoạn tiền xử lý, dữ liệu giao tác của người sử dụng Web được làm sạch và phân hoạch thành một tập hợp các giao dịch 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:

(i) Làm sạch dữ liệu (Data Cleaning);

(ii) Xác định người dùng dựa trên địa chỉ IP và các lần truy cập

Tổng hợp và làm sạch dữ liệu (Data fusion and cleaning)

Trong các trang Web quy mô lớn, rất bình thường khi các nội dung phục vụ cho người sử dụng đến từ nhiều máy chủ Web hoặc ứng dụng. Trong một số trường hợp, nhiều máy chủ với nội dung dự phòng được sử dụng để giảm tải trên bất kỳ máy chủ nào đó. Hợp nhất dữ liệu đề cập đến việc sáp nhập các file log từ các máy chủ Web hoặc ứng dụng. Điều này có thể yêu cầu sự đồng bộ hóa toàn cục trên các máy chủ. Khai phá sử dụng Web phân tích hành vi người dùng thông qua các tập tin log từ nhiều trang Web.

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. Đối với Web tĩnh, mỗi tập tin HTML là ánh xạ một-một với một lần xem trang. Đối với các trang Web động, một lần xem có thể đại diện cho một sự kết hợp của các thành phần tĩnh và nội dung được tạo ra bởi máy chủ ứng dụng dựa trên một tập hợp các thông số.

Ngoài ra, ta có thể xem xét lần truy cập ở mức độ kết hợp cao hơn, nơi mà mỗi lần xem trang đại diện cho một tập hợp các trang liên quan đến một chủ đề nào đó. 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.

Ký hiệu tập các session bằng R, đại diện cho các hoạt động thực sự của người dùng trên trang Web. Một hàm heuristic sessionization h để ánh xạ R thành một tập hợp các session, ký hiệu là Ch. Dựa trên ý tưởng về heuristic, h*, ta có Ch* = R. Nói cách khác, dùng heuristic có thể xác định gần chính xác hành động của người dùng trong một session. Sessionization heuristic chia thành hai loại cơ bản: hướng thời gian hoặc hướng cấu trúc. Giải pháp hướng thời gian dùng thời gian để xác định các session liên tiếp nhau, trong khi giải pháp cấu trúc theo định hướng sử dụng hoặc cấu trúc trang Web tĩnh hoặc liên kết cơ cấu ngầm được lưu trữ trong trường "referer" của file log.

Sau đây là một ví dụ, xét:

- h1: Tổng thời gian cho một session không được vượt quá một ngưỡng θ. Với t0 là thời điểm mà yêu cầu đầu tiên của người dùng được gửi đến server thuộc session S, một yêu cầu tại thời điểm t của người đó đến server là thuộc về S khi và chỉ khi $t - t0 \leq \theta$.

- h2: Tổng số thời gian trên một trang không được vượt quá một ngưỡng δ. Với t1 là thời điểm mà yêu cầu của người dùng được gửi đến server thuộc session S một yêu cầu tại thời điểm t của người đó đến server là thuộc về S khi và chỉ khi $t2 - t1 \leq \delta$.

- h-ref: Một yêu cầu q được thêm vào session S nếu referrer cho q trước đó đã thuộc S. Nếu không, q được xem là khởi đầu của một session mới. Lưu ý rằng với heuristic này, một yêu cầu q có khả năng có thể thuộc nhiều hơn một session đang mở, bởi q có thể được truy cập trước đó trong nhiều session. Trong trường hợp này, thông tin bổ sung có thể được sử dụng để lựa chọn. Ví dụ, q có thể được thêm vào session gần đây nhất thỏa các điều kiện nêu trên.

User 1

Time	IP	URL	Ref
0:01	1.2.3.4	A	-
0:09	1.2.3.4	B	A
0:19	1.2.3.4	C	A
0:25	1.2.3.4	E	C
1:15	1.2.3.4	A	-
1:26	1.2.3.4	F	C
1:30	1.2.3.4	B	A
1:36	1.2.3.4	D	B

Session 1

0:01	1.2.3.4	A	-
0:09	1.2.3.4	B	A
0:19	1.2.3.4	C	A
0:25	1.2.3.4	E	C

Session 2

1:15	1.2.3.4	A	-
1:26	1.2.3.4	F	C
1:30	1.2.3.4	B	A
1:36	1.2.3.4	D	B

(a) Xác định phiên người dùng với định hướng thời gian

User 1

Time	IP	URL	Ref
0:01	1.2.3.4	A	-
0:09	1.2.3.4	B	A
0:19	1.2.3.4	C	A
0:25	1.2.3.4	E	C
1:15	1.2.3.4	A	-
1:26	1.2.3.4	F	C
1:30	1.2.3.4	B	A
1:36	1.2.3.4	D	B

Session 1

0:01	1.2.3.4	A	-
0:09	1.2.3.4	B	A
0:19	1.2.3.4	C	A
0:25	1.2.3.4	E	C
1:26	1.2.3.4	F	C

Session 2

1:15	1.2.3.4	A	-
1:30	1.2.3.4	B	A
1:36	1.2.3.4	D	B

(b) Xác định phiên người dùng bằng liên kết

Hình 3. 8 - Định ra các session từ log file [33]

Trong hình trên, hàm heuristic h1, với θ = 30 phút đã được sử dụng để phân chia log hoạt động người dùng (từ ví dụ của hình trên) thành hai session riêng biệt. Nếu áp dụng h2 với một ngưỡng 10 phút, hồ sơ người dùng sẽ được xem như là ba session, cụ thể là A → B → C →E, A và F → B → D. Trong khi đó, hình dưới mô tả một ví dụ của việc sử dụng h-ref heuristic. Trong trường hợp này, một khi các yêu cầu cho F (với t=1:26) tới, có hai session mở, cụ thể là A → B → C → E và A. Tuy nhiên, F được thêm vào đầu tiên bởi vì C là referrer của nó thuộc session 1. Yêu cầu cho B (t=1:30) có khả năng có thể thuộc về cả hai session mở vì referrer của nó là A đều thuộc session 1 và 2. Trong trường hợp này, nó được thêm vào session 2, vì nó là session gần đây nhất là mở ra.

3.5.3. Thuật toán làm sạch dữ liệu (Data Cleaning)

Web log của một Website luôn tồn tại các thông tin không cần thiết (như 3.5.2.), do đó cần phải loại bỏ các thông tin này để có được các thông tin có ích cho các giai đoạn kế tiếp.

Thuật toán Data Cleaning

Đầu vào: tập tin Web log ban đầu

Đầu ra: CSDL có ý nghĩa

Phương pháp thực hiện:

1. Đặt D{} = tập tin Web log ban đầu
2. Đặt Table{} = các dòng tập tin Web log
3. Với mỗi *item* trong D thực hiện
4. Nếu khác l_i.StartsWith("#Software") và khác l_i.StartsWith("#Version") và khác l_i.StartsWith("#Date")
5. Nến l_i.StartsWith("#Fields: ") thì
6. Thêm Title Table
7. Ngược lại
8. Nếu url.ext#{css,js,jpg,jpeg,png,bmp,txt,ico,rss} tức là tồn tại phần mở rộng ".apsx" và *method # "GET"* và *status = 200* thì
9. Thêm vào Table
10. Nếu Table > 0 thì
11. *Lưu* Table xuống CSDL Web log

Hình 3. 9 - Thuật toán làm sạch dữ liệu Data Cleaning

3.5.4. Thuật toán xác định người dùng dựa vào IP (User IP)

Một chuỗi sự kiện trong Web log được xem là một giao dịch, một IP máy của người sử dụng được xem là một user. Các bước xác định người sử dụng là xác định địa chỉ IP của họ. Nếu chuỗi giao dịch có địa chỉ IP mới, thì đó là người dùng mới. Nếu địa chỉ IP là giống nhau nhưng phiên bản trình duyệt hoặc hệ điều hành khác nhau sau thì đó cũng là một người sử dụng mới, ngược lại tôi xác định đó là cùng một người sử dụng. Xác định người dùng nhằm tạo ra tập các danh sách IP theo định danh riêng biệt.

Bảng 3. 1 - Tập IP người sử dụng

IP	Time	Url
203.113.152.18	9/4/2015 4:53:16 AM	/products/A
203.113.152.18	9/4/2015 5:50:10 AM	/products/B
203.113.152.18	9/4/2015 4:54:34 PM	/products/D
203.165.15.192	9/4/2015 4:56:02 AM	/products/B

Vấn đề phát sinh, nếu một người truy cập một Website nhiều lần, mỗi lần truy cập sử dụng phiên bản trình duyệt hoặc hệ điều hành khác nhau như vậy sẽ tạo ra nhiều chuỗi giao dịch mới.

Do đó, cần phải xác định phiên người dùng (session), phiên người dùng (session) có thể được xem là một tập hợp các trang Web truy cập bởi cùng một người dùng trong khoảng thời gian một lần truy cập vào một trang Web cụ thể.

Phương pháp nhận dạng phiên là định hướng theo thời gian dựa trên tổng thời gian phiên với 30 phút là thời gian chờ mặc định (tiemout) của Cooley [33].

Phương pháp thứ hai phụ thuộc vào thời gian nghỉ trang được tính toán với sự khác biệt giữa hai mốc thời gian. Nếu nó vượt quá 10 phút thì được giả định như phiên mới được lưu CSDL.

Nếu người dùng truy cập lần đầu (có nghĩa là chưa tồn tại trong tập IP người sử dụng) thì xem đây là một phiên sử dụng mới, ngược lại nếu đã tồn tại người dùng này và thời gian truy cập giữa các lần lớn hơn thời gian mặc định thì xác định đó cũng là phiên sử dụng mới, trường hợp còn lại xem là cùng một phiên sử dụng.

Bảng 3. 2 - Tập phiên sử dụng của người truy cập

IP	Url
203.113.152.18	/products/A,products/B,/products/D
203.165.15.192	/products/B, /products/D

Các nghiên cứu trước đây [27],[28] và [30] về khai thác Web log, cụ thể là trong việc xác định phiên sử dụng Web đều cho kết quả là tập các item đơn. Luận văn đề xuất hướng phát triển xác định phiên người dùng là tập các itemset, có nghĩa là tạo ra một tập với định danh là một người sử dụng với các phiên sử dụng

của chính người sử dụng đó, một phiên sử dụng là một tập các trang Web với thời gian giữa các trang không lớn hơn 10 phút, ngược lại thi tạo một phiên sử dụng khác người đó.

Chi tiết thuật toán lưu session vào CSDL đề xuất như sau:

Thuật toán lưu session vào CSDL

Đầu vào: website

Đầu ra: CSDL

Phương pháp thực hiện: mysession(hostname)

1. Nếu thuộc tên miền của Web và không là trang default thì
2. Đặt ListOfTable{} = danh sách thông tin web (Ip, date, User-Agent, url)
3. Thêm các thông tin Web vào ListOfTable
4. Đặt ListOfTable{} = GetMyListObjects() // gọi tập thông tin dạng bản của Web
5. Nếu ListOfTable{} > 0
6. Nếu link ListStringString{} cuối phải khác link hiện tại
7. Thêm các thông tin Web vào ListString
8. Ngược lại
9. Thêm các thông tin Web vào ListString
10. EasyTimer.SetTimeout // 10 phút SetTimeout
11. Lưu ListOfTable vào CSDL

Hình 3. 10 - Thuật toán lưu session vào CSDL

Chi tiết thuật toán User IP đề xuất như sau:

Thuật toán User IP

Đầu vào: Web log D đã làm sạch dữ liệu

Đầu ra: CSDL giao dịch

Phương pháp thực hiện:

1. Đặt ListOfTable={}

2. ListOfUserIP= trong CSDL thực hiện với IP GROUP BY RefUrl
3. ListofRefUrIP= trong CSDL thực hiện ORDER BY IP, datetime, url tăng
4. Với mỗi ListOfUserIP trong CSDL thực hiện
5. Với mỗi ListofRefUrIP trong CSDL thực hiện
6. Nếu ListOfUserIP.IP = ListofRefUrIP.IP thì
7. Thêm mẫu chuỗi RefUrlIP
8. Đặt IP trong ListOfUserIP
9. Thêm IP và chuỗi refUrlIP vào ListOfTable
10. ListOfTable CSDL giao dịch

Hình 3.11 - Thuật toán xác định người dùng dựa trên User IP

Bảng 3. 3 - Tập xác định người dùng dựa IP đề xuất của luận văn

UserID	Url
1	/products/A, products/B, /products/C,/products/A, /products/D
2	/products/B, /products/D
3	/products/C

3. 6. Khai thác và phân tích đánh giá mẫu chuỗi

Trong giai đoạn khai thác mẫu, thống kê và các tác vụ tính toán được thực hiện để tìm ra những mô hình trong đó, từ đó phản ánh những hành vi điển hình của người sử dụng cũng như các thống kê tóm lược về việc truy cập của họ.

Sử dụng các phương pháp khai thác dữ liệu trong các lĩnh vực khác nhau như 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.

Phân tích đường dẫn: hầu hết các đường dẫn thường được viếng thăm được bố trí theo đồ thị vật lý trang Web. Mỗi nút là một trang, mỗi cạnh là đường liên kết của các trang đó. Thông qua việc phân tích đường dẫn trong quá tình truy cập của người dùng ta có thể phân bệt được mối quan hệ trong việc truy cập của người giữa các đường liên quan.

Ví dụ:

- 70% các khách hàng truy cập vào /company/ptoduct2 đều xuất phát từ

/company thông qua /company/new, /company/products và /company/product1.

- 80% khách hàng truy cập vào Website bắt đầu từ */company/products.*
- 65% khách hàng rời khỏi Website sau khi thăm 4 hoặc ít hơn 4 trang.

Luật tuần tự: sự tương quan giữa các tham chiếu đến các file khác nhau có trên dịch vụ nhờ việc sử dụng luật tuần tự, nó giúp cho việc phát triển chiến lược kinh doanh phú hợp, xây dựng và tổ chức một cách tốt nhất không gian Web của doanh nghiệp, v.v…

Ví dụ: 40% khách hàng truy cập vào Web có đường dẫn */company/product1* cũng sẽ truy cập vào */company/product2.*

Chuỗi các mẫu: các mẫu thu được giữa các giao tác và chuỗi thời gian. Thể hiện một tập các phần tử được theo sau bởi phần tử khác trong thứ tự thời gian lưu hành tập thao tác. Quá trình truy cập của khách hàng được ghi lại trên từng giai đoạn thời gian.

Ví dụ: 60% khách hàng đặt hàng trực tuyến ở */company/product1* thì cũng đặt hàng trực tuyến ở */company/product4* trong *15* ngày.

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.

3. 7. Tổng kết chương

Chương này trình này tổng quát về khai thác Web, vì sao luận văn phải khai thác theo hướng sử dụng Web, đồng thời đã giới thiệu các dạng và cấu trúc của một Web log, các bước khai thác sử dụng Web, cách thu thập dữ liệu, thuật toán tiền xử lý dữ. Bên cạnh đó, luận văn đề xuất thuật toán xác định người dung dựa vào IP và truy cập theo dạng khai thác dữ liệu.

CHƯƠNG 4: THỰC NGHIỆM, KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Thực nghiệm

4.1. 1. Mục tiêu

Dữ liệu chuỗi nói chung và dữ liệu Web log nói riêng là loại dữ liệu phổ biến trong nhiều lĩnh vực ứng dụng. Mục tiêu của chương này là kiểm tra lại tính đúng của lý thuyết bằng dữ liệu thực tế. Để làm được điều này sẽ tiến hành cài đặt các thuật toán nêu trên và tiến hành chạy trên CSDL Web log thực tế của một Cty thương mại tại Việt Nam.

Việc thực nghiệm được tiến hành trên máy tính Intel(R), Core(TM) i7 3537U, CPU 2.00 GHz, cài đặt trên ngôn ngữ lập trình Visual C#.net, sử dụng Visual Studio.Net 2012.

4.1.2. Thực nghiệm và đánh giá

Web log của một www.thiepcuoi.info, doanh nghiệp chuyên cung ứng các sản phẩm may mặc và quà tặng tại Việt Nam, Web log được thu tập trong đầu tháng 4/2015 – 7/2015, có 474298 chuỗi sự kiện.

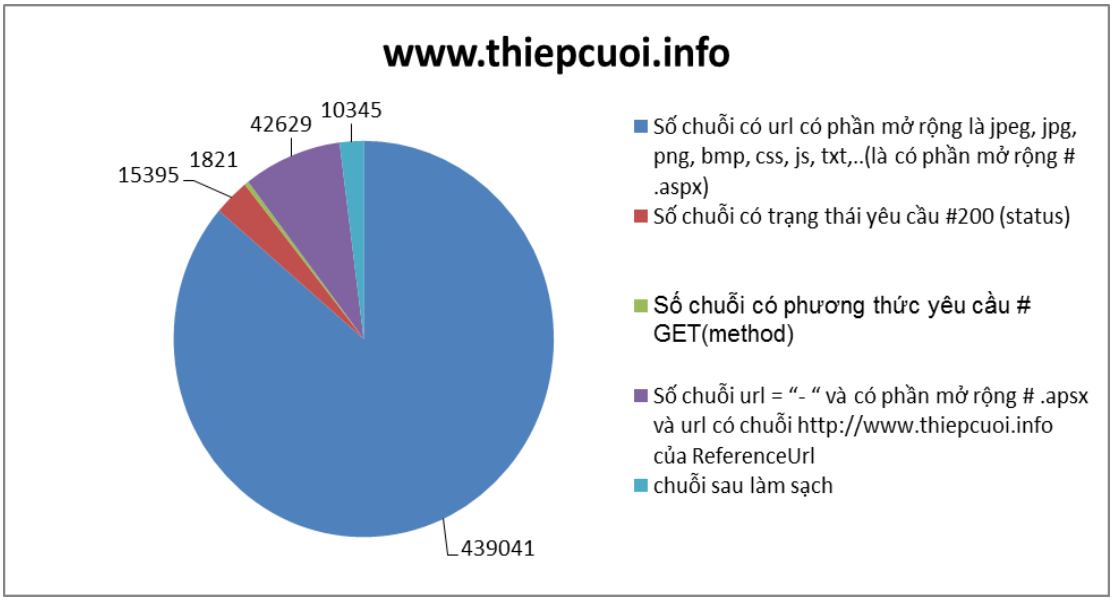
4.1.2.1.Giai đoạn tiền xử lý dữ liệu

Làm sạch dữ liệu

Bảng 4. 1 - Số chuỗi sự kiện của Web log www.thiepcuoi.info

Số chuỗi ban đầu	Số chuỗi có url có phần mở rộng là jpeg, jpg, png, bmp, css, js, txt,..(là có phần mở rộng # .aspx)	Số chuỗi có trạng thái yêu cầu #200 (status)	Số chuỗi có phương thức yêu cầu # GET(method)	Số chuỗi url = "- " và có phần mở rộng # .apsx và url có chuỗi http://www.thiepcuoi.info của ReferenceUrl	chuỗi sau làm sạch
474298	439041	15395	1821	42629	10345

Hình 4. 1 - Biểu đồ Web log của www.thiepcuoi.info sau khi làm sạch



Xác định người dùng

Bảng 4. 2 - Kết quả sau khi xác định người dùng với Web log www.thiepcuoi.info

Số chuỗi ban đầu	Tổng số người dùng	Số chuỗi sau khi tiền xử lý
474298	1699	10345

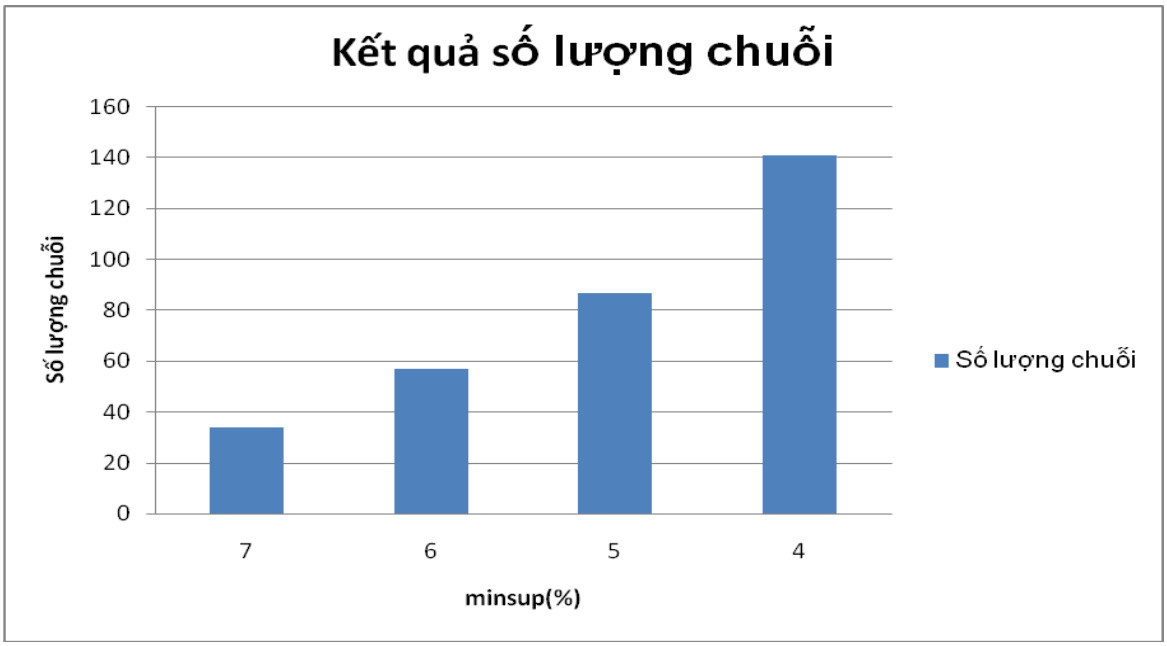
Như vậy, sau khi qua giai đoạn tiền xử lý, Web log của bathiphauniform.com đã được chuyển thành dạng CSDL giao dịch, theo cấu trúc <*SID, Transaction*>, với *SID* là IP định danh người dùng, *Transaction* là một dãy mẫu chuỗi. Do đó, CSDL giao dịch hiện tại có 1699 chuỗi sự kiện.

4.1.2.2.Giai đoạn khai thác và phân tích mẫu

➢ Sử dụng thuật toán khai thác sự kết hợp của bit vectơ động và thông tin giao dịch cho khai thác chuỗi khép kín thường xuyên có hiệu quả để khai thác mẫu phổ biến

Bảng 4. 3 - Kết quả sử dụng kết hợp của bit vectơ động v cho khai thác chuỗi phổ biến đóng trên Web log www.thiepcuoi.info với minConf = 50%

Web log	Độ hỗ trợ (%)	Số mẫu chuỗi phổ biến
thiepcuoi.info	7(119)	34
	6(102)	57
	5(85)	87
	4(68)	141

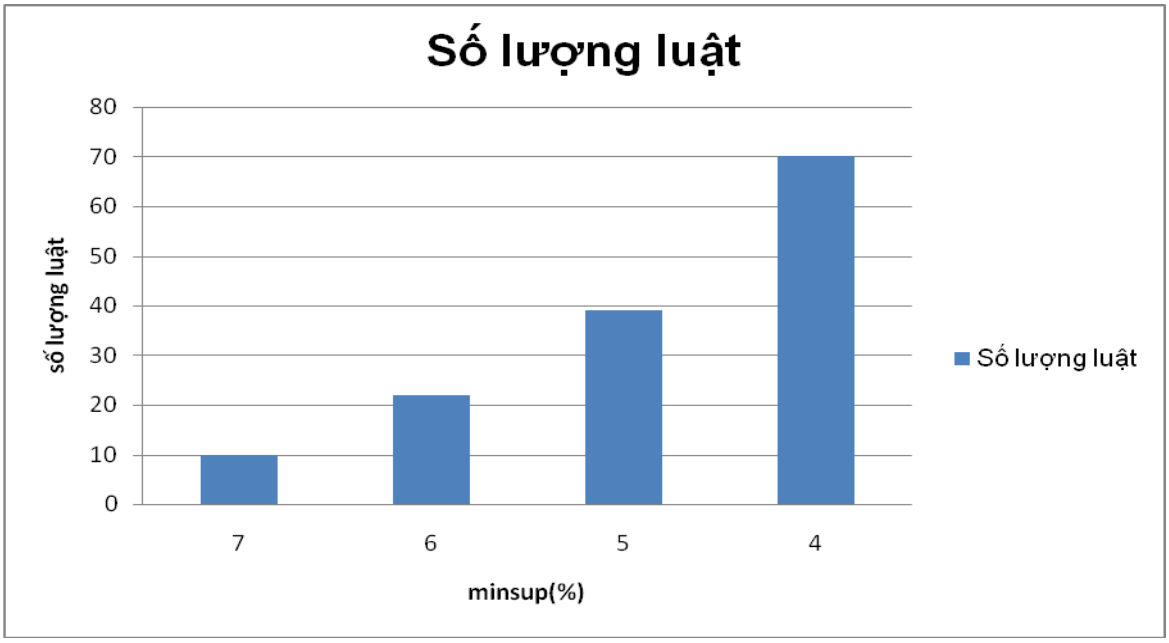


Hình 4. 2 - Sử dụng thuật toán kết hợp của bit vectơ động cho khai thác chuỗi phổ biến đóng trên Web log www.thiepcuoi.info với minConf = 50%

- Khai thác luật trên Web log

Bảng 4.4 - Số lượng luật thực hiện trên Web log www.thiepcuoi.info (minConf = 50%)

minSup %	Số lượng mẫu chuỗi phổ biến	Số lượng luật
7(119)	34	10
6(102)	57	22
5(85)	87	39
4(68)	141	70



Hình 4.3 – Số lượng luật với dụng thuật toán khai thác sự kết hợp của bit vectơ động và thông tin giao dịch cho khai thác chuỗi khép kín thường xuyên

➢ Danh sách luật khi sử dụng thuật toán khai thác kết hợp của bit vectơ động cho khai thác chuỗi phổ biến đóng với minsup = 0.07 và minConf=50%.

Bảng 4. 5 - Danh sách các luật khi minsup = 0.07 và minConf = 50% của Web log www.thiepcuoi.info

R	**Luật**	**(sup, conf)**
1	http://www.thiepcuoi.info/thiep-cuoi-co-dien-pci-5.aspx => http://www.thiepcuoi.info/thiep-cuoi-hien-dai-pci-6.aspx	(S= 197, C=0.5)
2	http://www.thiepcuoi.info/thiep-cuoi-co-dien-pci-5.aspx, http://www.thiepcuoi.info/thiep-cuoi-hien-dai-pci-6.aspx => http://www.thiepcuoi.info/thiep-cuoi-phong-thu-pci-7.aspx	(S= 130, C=0.66)
3	http://www.thiepcuoi.info/thiep-cuoi-nghe-thuat-pci-3.aspx, http://www.thiepcuoi.info/thiep-cuoi-co-dien-pci-5.aspx => http://www.thiepcuoi.info/thiep-cuoi-hien-dai-pci-6.aspx	(S= 121, C=0.69)
4	http://www.thiepcuoi.info/thiep-cuoi-nghe-thuat-pci-3.aspx => http://www.thiepcuoi.info/thiep-cuoi-hien-dai-pci-6.aspx	(S= 206, C=0.50)
5	http://www.thiepcuoi.info/thiep-cuoi-vintage-pci-4.aspx, http://www.thiepcuoi.info/thiep-cuoi-co-dien-pci-5.aspx => http://www.thiepcuoi.info/thiep-cuoi-hien-dai-pci-6.aspx	(S= 114, C=0.59)

6	http://www.thiepcuoi.info/thiep-cuoi-vintage-pci-4.aspx ,http://www.thiepcuoi.info/thiep-cuoi-co-dien-pci-5.aspx => http://www.thiepcuoi.info/thiep-cuoi-hien-dai-pci-6.aspx	(S= 127, C=0.52)
7	http://www.thiepcuoi.info/thiep-cuoi-vintage-pci-4.aspx, http://www.thiepcuoi.info/thiep-cuoi-co-dien-pci-5.aspx => http://www.thiepcuoi.info/thiep-cuoi-hien-dai-pci-6.aspx	(S= 142, C=0.58)
8	http://www.thiepcuoi.info/thiep-cuoi-vintage-pci-4.aspx, http://www.thiepcuoi.info/thiep-cuoi-co-dien-pci-5.aspx => http://www.thiepcuoi.info/thiep-cuoi-hien-dai-pci-6.aspx	(S= 125, C=0.51)
9	http://www.thiepcuoi.info/thiep-cuoi-vintage-pci-4.aspx, http://www.thiepcuoi.info/thiep-cuoi-hien-dai-pci-6.aspx => http://www.thiepcuoi.info/thiep-cuoi-phong-thu-pci-7.aspx	(S= 145, C=0.59)
10	http://www.thiepcuoi.info/thiep-cuoi-vintage-pci-4.aspx, http://www.thiepcuoi.info/thiep-cuoi-phong-thu-pci-7.aspx => http://www.thiepcuoi.info/thiep-cuoi-in-hoa-pci-8.aspx	(S= 120, C=0.56)

4.1.2.3. Nhận xét

Dựa vào bảng 4.5, ta thấy luật thứ 3 thể hiện *http://www.thiepcuoi.info/ thiep-cuoi-nghe-thuat-pci-3.aspx,http://www.thiepcuoi.info/thiep-cuoi-co-dien-pci-5.aspx => http://www.thiepcuoi.info/thiep-cuoi-hien-dai-pci-6.aspx*, vì có độ tin cậy 69%, như vậy nếu người sử dụng truy cập vào Website xem sản phẩm *thiep-cuoi-nghe-thuat-pci-3.aspx* đến *thiep-cuoi-co-dien-pci-5.aspx* và sẽ tìm xem sản phẩm *thiep-cuoi-hien-dai-pci-6.aspx* và các luật khác tương tự. Qua đó, Website này có thể lập kế hoạch kinh doanh, báo cáo, thống kê... Phát triển những sản phẩm của mình, đồng thời mô hình hoá trong việc thiết kế Web site để tạo thuận tiện cho người duyệt Web.

4.1.3. Tổng kết thực nghiệm

Thực nghiệm đã trình bày các kết quả thực nghiệm của các vấn đề mà luận văn đã đề cập: khai thác mẫu phổ biến, khai thác luật và ứng dụng của các thuật toán trên CSDL Weblog của một Website. Dựa vào các kết quả phân tích được từ Web log, Website này có thể định hướng phát triển các sản phẩm đặc nổi bật, lập kế hoạch kinh doanh, và phát triển Website theo hướng tiên dụng cho khách hàng hơn.

4. 2. Kết luận

Luận văn đã tìm hiểu cơ sở lý thuyết về khai thác mẫu chuỗi, khai thác luật

và ứng luật trên CSDL chuỗi. Bên cạnh đó trong luận văn cũng giới thiệu về khai thác Web (Web Mining) và đi sâu theo hướng khai thác sử dụng Web. Mục đích của luận văn là đưa ra phương pháp hiệu quả để khai thác và ứng dụng luật trên CSDL Web log.

Luận văn đã giải quyết những vấn đề sau:

Chương một đã trình bày tổng quan về CSDL chuỗi. Đồng thời trình bày khái quát về lĩnh vực khai thác mẫu chuỗi và luật trên CSDL chuỗi. Chương này cung cấp một cái nhìn chung nhất về lĩnh vực khai thác dữ liệu trên CSDL chuỗi.

Chương hai gồm hai phần, thứ nhất trình bày bài toán về khai thác mẫu chuỗi. Trong đó, luận văn mô tả chi tiết thuật toán sự kết hợp của bit vectơ cho khai thác chuỗi phổ biến đóng, là thuật toán được chọn cho khai thác mẫu chuỗi. Thứ hai trình bày cơ sở lý thuyết về khai thác luật.

Chương ba luận văn trình bày tổng quan về khai thác Web, lý do vì sao chọn khai thác sử dụng Web. Sau đó, ứng dụng luật đã nghiên cứu vào khai thác hành vi sử dụng Web.

Chương bốn gồm có ba phần, thứ nhất trình bày thực nghiệm và phân tích kết quả trên một Web log thực tế của một doanh nghiệp tại Việt Nam. . Thứ hai kết luận. Thứ ba nêu hướng phát triển.

Tuy nhiên, trong báo cáo luận văn chỉ mới nghiên cứu về mặt lý thuyết, chưa vận dụng vào các ứng dụng thực tế để thấy tính ứng dụng và hiệu quả của luật trong khai thác sử dụng Web

4. 3. Hướng phát triển

Khai thác luật rất hữu ích trong việc khám phá những tri thức tiềm ẩn trong các nguồn dữ liệu ở dạng tuần tự. Tuy nhiên tình trạng bùng nổ thông tin hiện nay, khối lượng dữ liệu ngày càng trở nên đồ sộ.

Khai thác phân tán có thể đưa ra cách xử lý mở rộng cho các CSDL lớn và chuỗi dữ liệu dài. Trong lĩnh vực khai thác thói quen sử dụng Web, có thể áp dụng khai thác phân tán để khai thác Web log bị phân tán trên nhiều server.

Ngữ nghĩa bao hàm, điều này đòi hỏi một mô hình mà có thể suy luận ra các

mối quan hệ ngữ nghĩa từ tri thức lĩnh vực thu được thông qua bản chất hay chú thích trên các chuỗi và tránh phải đếm độ hỗ trợ nếu chuỗi ứng viên không bao giờ xuất hiện về mặt ngữ nghĩa. Một số luận án tiến sĩ đã đề xuất mô hình xử lý cho bài toán khai thác thói quen sử dụng Web, trong đó tích hợp nội dung, cấu trúc và cách xử lý vào quá trình khai thác để khai thác có ràng buộc trực tiếp hơn nhưng không sử dụng tri thức lĩnh vực trong quá trình khai thác.

Đi sâu vào tính ứng dụng của bài toán khai thác luật trên CSDL Web log.

TÀI LIỆU THAM KHẢO

[1] Olatz Arbelaitz , Ibai Gurrutxaga, Aizea Lojo, Javier Muguerza, Jesús Maria Pérez, Iñigo Perona, (2013) . "Web usage and content mining to extract knowledge for modelling the users of the Bidasoa Turismo Web site and to adapt it", Expert Systems with Applications, 40(18), 7478–7491 .

[2] C.J. Carmona , S. Ramírez-Gallego , F. Torres , E. Bernal , M.J. del Jesus , S. García, (2012) . "Web usage mining to improve the design of an e-commerce Web site: OrOliveSur.com", Expert Systems with Applications, 39(12), 11243–11249 .

[3] Minh-Thai Tran , Bac Le , Bay Vo , (2012) . "Combination of dynamic bit vectors and transaction information for mining frequent closed sequences efficiently", Engineering Applications of Artificial Intelligence, 39(12), 11243–11249 .

[4] Elena Baralis, Silvia Chiusano, Riccardo Dutto, (2008) . "Applying Sequential Rules to Protein Localization Prediction", Computer and Mathematics with Applications, 55(5), 867–878.

[5] David Lo, Siau-Cheng Khoo, Limsoon Wong, (2009). "Non-Redundant Sequential Rules-Theory and Algorithm, Information Systems", Information Systems, 34(4-5), 438-453.

[6] Myra Spiliopoulou, (1999) . "Managing Interesting Rules in Sequence Mining", Proceedings of European Conference on Principles of Data Mining and Knowledge Discovery, 554–560.

[7] Heikki Mannila, Hannu Toivonen, A. Inkeri Verkamo , (1997) . "Discovery of frequent episodes in event sequences", Data Mining and Knowledge Discovery, 1 (3), 259–289.

[8] Sergey Brin , Rajeev Motwani , Jeffrey D. Ullman , Shalom Tsur, (1997) . "Dynamic Itemset Counting and Implication Rules for Market Basket Data", Newsletter ACM SIGMOD Record , 26(2), 255-264.

[9] Rakesh Agrawal , Ramakrishnan Srikant, (1995) . "Mining sequential patterns", Proceedings of IEEE International Conference on Data Engineering, 3–14.

[10] Claudio Lucchese, Salvatore Orlando, Raffaele Perego , (2003) . “CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets” , Proceeding KDD '03 Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 236-245.

[11] Ramakrishnan Srikant, Rakesh Agrawal, (1996) . “Mining sequential patterns: Generalizations and performance improvements” , Proceeding EDBT '96 Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology, 3-17.

[12] Jiawei Han, Jian Pei, Behzad Mortazavi-Asl, Qiming Chen, Umeshwar Dayal, Mei-Chun Hsu, (2000) . “Freespan: Frequent pattern-projected sequential pattern mining”, Proceeding KDD '00 Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, 355– 359.

[13] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, Mei-Chun Hsu, (2004) . “Mining Sequential Patterns by Pattern-Growth: ThePrefixSpan Approach”, Journal IEEE Transactions on Knowledge and Data Engineering, 16(11) , 1424–1440 .

[16] Zhenglu Yang, University of Tokyo, Kitsuregawa M., (2004) . “LAPIN-SPAM: An Improved Algorithmfor Mining Sequential Pattern, ICDE Workshops”, Data Engineering Workshops, 2005. 21st International Conference on , 1222.

[17] Mohammed J. Zaki , (2001) . “SPADE: An efficient algorithm for mining frequentsequences, Machine Learning”, Journal Machine Learning , 42(1-2), 31–60.

[18] Zhenglu Yang, (2008) . “LAPIN-WEB: Fast Algorithms for Sequentail Pattern Mining “, ICDE Workshops.

[19] Jianyong Wang, Jiawei Han, (2007) . “BIDE: efficient mining of frequent closed sequences”, Proceeding ICDE '04 Proceedings of the 20th International Conference on Data Engineering, 79–90.

[21] Mohammed J.Zaki, Ching-Jui Hsiao, (2002) . “CHARM: An Efficient Algorithm for Closed Itemset Mining”, 457–473

[22] Berry, M.J., Linoff, G.S. , (1997) . “Data Mining Techniques for Marketing, Sales and Customer Support”, John Wiley & Sons.

[23] Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, Shalom Tsur , (1997) . “Dynamic itemset counting and implication rules for market basket data”, Proceeding SIGMOD '97 Proceedings of the 1997 ACM SIGMOD international conference on Management of data, 255-264.

[24] Joshila Grace, V. Maheswari, Dhinaharan Nagamalai, (2011) . “Analysis of web logs and web user in web mining”, IJNSA ,

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

[26] Ramakrishnan Srikant, Rakesh Agrawal, (1996) . “Mining sequential patterns: Generalizations and performance improvements” , Proceeding EDBT '96 Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology, 3-17.

[27] Magdalini Eirinaki, Michalis Vazirgiannis, (2003) . “Web mining for web personalization” , Journal ACM Transactions on Internet Technology (TOIT), 3(1), 1-27.

[28] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Hua Zhu , (2000) . “Mining access patterns efficiently from web logs”, Proceeding PADKK '00 Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications, 396-407.

[29] Renáta Iváncsy, István Vajk, (2006). “Frequent Pattern Mining in Web Log Data”, Acta Polytechnica Hungarica, 3(1), 77-90.

[30] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, Pang-Ning Tan, (2000) .“Web usage mining: Discovery and applications of usage patterns from web data”, Newsletter ACM SIGKDD Explorations Newsletter, 1(2), 12-23.

[31] Robert Cooley , Bamshad Mobasher , Jaideep Srivastava, (1999) . “Data preparation for mining world wide web browsing patterns”, Knowledge and Information Systems, 1(1), 5-32.

[32] MSDN Library. [online] available at https://msdn.microsoft.com/en-us/library/ms525807(v=vs.90).aspx , 2011

[33] Ashwin G. Raiyani, Prof. Sheetal S. Pandya. (2013) . "Discovering User Identification Mining Technique For Preprocessed Web Log Data", journal of information, knowledge and research in computer engineering, 2(2), 477-482.

[34] Xifeng Yan, Jiawei Han, Ramin Afshar , (2003). "CloSpan: Mining Closed Sequential Patterns in Large Datasets ", Proceedings of the SIAM International Conference on Data Mining, 166-177.

[35] Jianyong Wang, Jiawei Han, Jian Pei , (2003). "CLOSET+: searching for the best strategies for mining frequent closed itemsets", Proceeding KDD '03 Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 236-245.

[36] Shijie Song, Huaping Hu, Shiyao Jin , (2005) . "HVSM:a new sequential pattern mining algorithm using bitmap representation", Proceeding ADMA'05 Proceedings of the First international conference on Advanced Data Mining and Applications, 455-463.

[37] Jian Pei , Jiawei Han , Behzad Mortazavi-asl , Helen Pinto , Qiming Chen , Umeshwar Dayal , Mei-chun Hsu, (2001) . "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth", Proceedings of the International Conference on Data Engineering, 215—224.

[38] Wei Songa, Bingru Yangb, Zhangyan Xuc, (2008) ."Index-BitTableFI: an improved algorithm for mining frequent itemsets", Knowledge-Based Systems, 21(6) , 507–513.

[39] Nicolas Pasquier , Yves Bastide, Rafik Taouil, Lotfi Lakhal, (1999) . "Discovering frequent closed itemsets for association rules", Proceeding ICDT '99 Proceedings of the 7th International Conference on Database Theory , 398-416.

[40] Jian Pei , Jiawei Han , Runying Mao, (2000) . "CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets", Proceedings of the ACM

SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery (DMKD'00), 21-30.

[41] Antonio Gomariz , Manuel Campos , Bart Goethals , (2013) . "ClaSP: an efficient algorithm for mining frequent closed sequences", Advances in Knowledge Discovery and Data Mining, LNA, 50-61.

[42] Ding-Ying, Hung W, L.P. Chen, (2004) . "An efficient algorithm for mining frequent sequences by a new strategy without support counting", Data Engineering, 2004. Proceedings. 20th International Conference on , 375 - 386.

[43] Thi-Thiet Pham, Jiawei Luo, Tzung-Pei Hong, Bay Vo, (2012) . "MSGPs: A Novel Algorithm for Mining Sequential Generator Patterns", Computational Collective Intelligence, Tech-nologies and Applications, Lecture Notes in Computer Science, 7654, 393–401.

[44] Jie Dong, Min Han, (2007) . "BitTableFI: An efficient mining frequent itemsets algorithm", Knowledge-Based Systems, 20(4), 329–335.

[45] Wei Songa, Bingru Yangb, Zhangyan Xuc, (2008) . "Index-BitTableFI: An improved algorithm for mining frequent itemsets", Knowledge-Based Systems, 21(6) , 507–513.

[46] 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, 39(8) , 7196–7206.

[47] Thi-Thiet Pham, Jiawei Luo, Bay Vo, (2013) . "An effective algorithm for mining closed sequential patterns and their minimal generators based on prefix trees", Journal International Journal of Intelligent Information and Database Systems, 7(4) , 324-339.

[48] Thien -Trang Van, Bay Vo, Bac Le, (2014) . "IMSR_PreTree: an improved algorithm for mining sequential rules based on the prefix-tree", Vietnam J Comput Sci , 1(2), 97–105.

[49] Jianyong Wang, Jiawei Han, Chun Li, (2007) . "Frequent closed sequence mining without candidate maintenance", IEEE transactions on knowledge and data enginneering, 19(8), 1042–1056.