BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

VŨ VĂN ĐÔNG

MỘT PHƯƠNG PHÁP BẢO TOÀN TÍNH RIÊNG TƯ TRONG KHAI THÁC LUẬT KẾT HỢP TRÊN CƠ SỞ DỮ LIỆU PHÂN TÁN NGANG

LUẬN VĂN THẠC SĨ

Chuyên ngành: Công Nghệ Thông Tin

Mã ngành: 60480201

TP. HỒ CHÍ MINH, tháng 02 năm 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

VŨ VĂN ĐÔNG

MỘT PHƯƠNG PHÁP BẢO TOÀN TÍNH RIÊNG TƯ TRONG KHAI THÁC LUẬT KẾT HỢP TRÊN CƠ SỞ DỮ LIỆU PHÂN TÁN NGANG

LUẬN VĂN THẠC SĨ

Chuyên ngành: Công Nghệ Thông Tin

Mã ngành: 60480201

CÁN BỘ HƯỚNG DẪN KHOA HỌC: TS. CAO TÙNG ANH

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: TS. Cao Tùng Anh

Cao Tùng Anh

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:

TT	Họ và Tên	Chức danh Hội đồng
1	GS.TSKH. Hoàng Văn Kiếm	Chủ tịch
2	PGS.TS. Võ Đình Bảy	Phản biện 1
3	TS. Nguyễn Thị Thúy Loan	Phản biện 2
4	TS. Lê Văn Quốc Anh	Ủy viên
5	TS. Lê Tuấn Anh	Ủy viên, Thư ký

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

Chủ tịch Hội đồng đánh giá LV

TRƯỜNG ĐH CÔNG NGHỆ TP. HCM **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

PHÒNG QLKH – ĐTSĐH **Độc lập – Tự do – Hạnh phúc**

TP. HCM, ngày 15 *tháng 02 năm 2016*

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ tên học viên: Vũ Văn Đông Giới tính: Nam

Ngày 12 tháng 10 năm sinh: 1978 Nơi sinh: Hà Nội

Chuyên ngành: Công nghệ thông tin MSHV: 1441860007

I- Tên đề tài:

MỘT PHƯƠNG PHÁP BẢO TOÀN TÍNH RIÊNG TƯ TRONG KHAI THÁC LUẬT KẾT HỢP TRÊN CƠ SỞ DỮ LIỆU PHÂN TÁN NGANG

II- Nhiệm vụ và nội dung:

- Tìm hiểu các thuật toán khai thác tập phổ biến, luật kết hợp.
- Tìm hiểu các thuật toán bảo toàn tính riêng tư trong khai thác dữ liệu trên cơ sở dữ liệu phân tán ngang.
- Xây dựng ví dụ cho thuật toán đã nghiên cứu.
- Xây dựng chương trình Demo.

III- Ngày giao nhiệm vụ **:** 15/07/2015

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

V- Cán bộ hướng dẫn **: TS. Cao Tùng Anh**

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

LỜI CAM ĐOAN

Tôi xin cam đoan đây 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 cũng như các trích dẫn hay tài liệu học thuật tham khảo đã được cảm ơn đến tác giả và các thông tin trích dẫn trong Luận văn đã được chỉ rõ nguồn gốc.

Học viên thực hiện Luận văn

Vũ Văn Đông

LỜI CÁM ƠN

Trước hết, cho tôi được gửi lời cảm ơn đến sự hướng dẫn và giúp đỡ tận tình của **Thầy Cao Tùng Anh**.

Xin cảm ơn các Thầy/Cô trong Khoa CNTT trường Đại Học Công Nghệ TP. HCM đã giúp đỡ và cung cấp cho tôi những kiến thức quí giá trong suốt thời gian học tập và nghiên cứu thực hiện luận văn.

Xin cám ơn các Thầy/Cô thuộc phòng QLKH&ĐTSĐH đã tạo rất nhiều điều kiện thuận lợi cho tôi trong suốt quá trình theo học tại Trường.

Tôi cũng xin gửi lời cảm ơn đến gia đình, bạn bè và những người thân đã luôn quan tâm và giúp đỡ tôi trong suốt thời gian học tập và nghiên cứu hoàn thành luận văn này.

Luận văn không thể tránh khỏi những sai sót, rất mong nhận được ý kiến đóng góp của mọi người cho luận văn được hoàn thiện hơn.

Tôi xin chân thành cảm ơn.

TP. Hồ Chí Minh, ngày 15 tháng 02 năm 2016

Học viên thực hiện Luận văn

Vũ Văn Đông

TÓM TẮT

Trong những năm gần đây, khai thác luật kết hợp trên cơ sở dữ liệu phân tán đã nhận được sự quan tâm của các nhà nghiên cứu. Việc kết hợp dữ liệu phân tán (ngang hay dọc) từ nhiều cơ sở dữ liệu khác nhau sẽ cho phép khai thác được các luật có lợi cho tất cả các bên tham gia quá trình khai thác.

Tuy nhiên, khi khai thác dữ liệu từ nhiều bên sẽ nảy sinh vấn đề về tính riêng tư về dữ liệu của các bên tham gia cần được bảo vệ. Phần lớn dữ liệu của các bên đều có dữ liệu nhạy cảm và các bên tuy rất muốn cung cấp dữ liệu để khai thác được các luật dùng chung nhưng vẫn muốn bảo vệ tính riêng tư có trong dữ liệu của mình.

Để giải quyết các vấn đề như đã nêu ở trên, nội dung nghiên cứu của luận văn sẽ tập trung vào nghiên cứu các thuật toán khai thác luật kết hợp, khai thác luật kết hợp trên cơ sở dữ liệu phân tán ngang có bảo toàn tính riêng tư của các bên tham gia, viết chương trình thực nghiệm một thuật toán đã nghiên cứu.

ABSTRACT

In recent years, mining association rules in distributed database has received the attention of the researchers, The combination of distributed data (horizontal or vertical) from many different databases will mining association rules beneficial for all parties involve.

However, when data mining from multiple parties will arise issues of data privacy of the parties involved should be protected. Most data of each parties have sensitive data and the parties but wanted to provide data for mining association rules but they still want to protect the privacy of their data.

To solve the problem as stated above, research contents of the thesis will focus on the study of algorithms mining association rules, mining association rules in horizontal distributed database with privacy preserving of the parties, programing an algorithm had studied.

DANH MỤC CÁC TỪ VIẾT TẮT

Ký hiệu, viết tắt	Ý nghĩa tiếng Anh	Ý nghĩa tiếng Việt
CSDL		Cơ sở dữ liệu
DB	DataBase	Cơ sở dữ liệu
Conf	Confidence	Độ đo tin cậy
Sup	Support	Độ đo hỗ trợ
MST	Minsup	Ngưỡng hỗ trợ tối thiểu
MCT	Minconf	Ngưỡng tin cậy tối thiểu
FI	Frequent itemset	Tập phổ biến
PPDM	Privacy Preserving Data Mining	Bảo toàn tính riêng tư trong khai thác dữ liệu
SM	Safety Margin	Khoảng an toàn

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH

MỤC LỤC

PHẦN MỞ ĐẦU

1. LÝ DO CHỌN ĐỀ TÀI

Trong thời đại ngày nay, với sự phát triển vượt bậc của công nghệ thông tin và sự phổ biến của Internet. Lượng dữ liệu tại các hệ thống thông tin này ngày càng trở nên phong phú, đa dạng và thực sự khổng lồ. Trong tình hình đó, việc chắt lọc những thông tin quý giá từ những dữ liệu khổng lồ ngày càng có ý nghĩa hơn bao giờ hết, nó đóng vai trò chìa khóa thành công cho sự phát triển của các tổ chức, cá nhân. Các thông tin tìm được có thể được vận dụng để cải thiện hiệu quả hoạt động của hệ thống thông tin ban đầu, cải thiện thời gian tìm kiếm, hay đưa ra những dự đoán giúp cải thiện những quyết định trong tương lai,… Các kỹ thuật khai thác dữ liệu (data mining) ngày càng được quan tâm và ứng dụng rộng rãi trong nhiều lĩnh vực của cuộc sống như kinh tế, giáo dục, y tế, trong siêu thị,…

Phân tích luật kết hợp là một trong những phương pháp của khai thác dữ liệu. Nhiệm vụ của phương pháp này là phân tích dữ liệu trong CSDL nhằm phát hiện và đưa ra những mối liên hệ giữa các giá trị dữ liệu. Luật kết hợp thu được thường có dạng một mệnh đề có 2 vế: $A \rightarrow B$, trong đó A gọi là tiền đề, B gọi là mệnh đề kết quả. Luật kết hợp tuy khá đơn giản nhưng những thông tin mà luật mang lại là rất đáng kể, hỗ trợ không nhỏ trong quá trình ra quyết định. Tìm kiếm được các luật "hữu ích" từ CSDL tác nghiệp.

Một ứng dụng quan trọng của luật kết hợp là phân tích thị trường. Đó là việc phân tích thói quen mua hàng của khách để tìm sự kết hợp giữa các mặt hàng khác nhau trong một lần mua hàng của họ.

Ví dụ: Tổng hợp trong một số lần mua hàng tại siêu thị, nếu khách hàng mua kem đánh răng, thì họ thường sẽ mua bản chải đánh răng và khăn mặt. Nhưng thông tin như thế giúp người bán hàng lựa chọn mặt hàng và vị trí của chúng trên giá hàng. Do đó người bán có thể những mặt hàng thường được mua cùng nhau trong phạm vi gần kề để gây tác động tích cực tới việc mua của khách cho những mặt hàng này. Việc nhận ra các mặt hàng thường được mua cùng nhau, giúp người bán hàng có thể bán được nhiều hàng hơn. Do đó, doanh thu sẽ tăng.

Khai thác luật kết hợp nhằm tìm ra những mối liên kết đáng quan tâm hoặc những quan hệ tương quan trong một tập lớn các đối tượng. Trong giao dịch thương

mại khám phá mối quan hệ trong số lượng lớn các bản ghi giao dịch có thể giúp nhiều nhà kinh doanh xử lý giải quyết các vấn đề một cách hiệu quả hơn.

Trong những năm gần đây, một số tác giả đề xuất hướng nghiên cứu khai thác dữ liệu trên CSDL phân tán [4, 5, 10]. Dữ liệu được lưu trữ trên nhiều vị trí và được kết nối với nhau bởi hệ thống mạng. Theo lý thuyết CSDL phân tán có thể được tái thiết lại giữa các vị trí thành một CSDL tập trung. Tuy nhiên nếu làm như vậy mất nhiều chi phí cho việc kết hoặc hội CSDL. Ngoài ra việc gửi dữ liệu của các bên tham gia để tạo ra CSDL tập trung có thể làm lộ thông tin nhạy cảm về dữ liệu của các bên tham gia. Luận văn sẽ tập trung nghiên cứu các thuật toán khai thác tập phổ biến và luật kết hợp trên CSDL phân tán ngang có quan tâm đến việc bảo toàn tính riêng tư của các bên tham gia cung cấp dữ liệu cho quá trình khai thác.

2. MỤC TIÊU VÀ PHẠM VI NGHIÊN CỨU

Một số thuật toán khai thác luật kết hợp trên CSDL phân tán bảo toàn tính riêng tư đã được nhiều tác giả đề xuất [1, 3, 6, 9, 11]. Tuy nhiên, một số vấn đề vẫn còn tồn tại với các thuật toán như: Chi phí thực hiện, thời gian thực hiện, …. trong CSDL phân tán, chi phí thực hiện chủ yếu được tính qua quá trình truyền và nhận dữ liệu giữa các bên tham gia khai thác, các thuật toán khai thác trên CSDL phân tán cũng tính toán giảm các chi phí này. Ngoài ra ở một số thuật toán, khả năng bị tấn công và bị lộ thông tin vẫn còn cao [4]. Điều này có nghĩa là người tham gia phải chấp nhận một tỷ lệ bị lộ tính riêng tư trong dữ liệu của mình cho chính quá trình sử dụng của họ.

Đề tài này tập trung vào việc nghiên cứu các thuật toán khai thác tập phổ biến, khai thác luật kết hợp và khai thác trên CSDL phân tán ngang bảo toàn tính riêng tư của các bên tham gia khai thác. Theo đánh giá của các tác giả [6] thì mô hình khai thác này đảm bảo tính riêng tư an toàn cho các bên tham gia khai thác và giảm được chi phí trong quá trình truyền và nhận dữ liệu giữa các bên. Từ mô hình [6] luận văn cũng mạnh dạn đề xuất một thay đổi nhỏ trong bước khai thác tập phổ biến để làm giảm thời gian khai thác tại mỗi bên, Ngoài ra, luận văn cũng trình bày phần cài đặt chương trình thực nghiệm cho mô hình để kiểm tra tính đúng đắn của mô hình đã nghiên cứu.

3. PHƯƠNG PHÁP NGHIÊN CỨU

Nghiên cứu tổng quan về khai thác dữ liệu, tiến hành thu thập và nghiên cứu các tài liệu có liên quan đến đề tài.

Tìm hiểu các thuật toán khai thác dữ liệu, khai thác dữ liệu trên CSDL phân tán ngang có quan tâm đến việc bảo vệ tính riêng tư của các bên tham gia.

Xây dựng chương trình thực nghiệm cho mô hình thuật toán đã nghiên cứu.

4. BỐ CỤC LUẬN VĂN

Luận văn được tổ chức có 3 chương, phần mở đầu và phần kết luận. Chương 1: Trình bày tổng quan về khai thác dữ liệu. Chương 2: Trình bày một số thuật toán khai thác CSDL phân tán ngang có bảo toàn tính riêng tư của các bên tham gia. Chương 3: Trình bày một mô hình mới đề xuất trong khai thác luật kết hợp trên CSDL phân tán ngang bảo toàn tính riêng tư của các bên tham gia và chương trình thực nghiệm.

CHƯƠNG 1
TỔNG QUAN VỀ KHAI THÁC DỮ LIỆU

1.1 GIỚI THIỆU ĐỀ TÀI

Sự phát triển mạnh mẽ của mạng Internet hiện nay dẫn đến sự bùng nổ của thông tin, tri thức và với khối lượng dữ liệu ngày càng lớn đã thúc đẩy một lĩnh vực nghiên cứu đầy tiềm năng là khai thác tri thức và khai thác dữ liệu. Chúng ta đang bị ngập trong khối dữ liệu khổng lồ nhưng những dữ liệu thật sự có giá trị cho chúng ta thì rất nhỏ. Do đó, việc khai thác dữ liệu (data mining) là quá trình giúp chúng ta có được những dữ liệu có giá trị từ khối dữ liệu khổng lồ đó.

Khai thác dữ liệu là quá trình tìm kiếm các mẫu mới, những thông tin tiềm ẩn trong các khối dữ liệu khổng lồ, khai thác có thể dự đoán những xu hướng trong tương lai, hay giúp cho các công ty kinh doanh ra các quyết định kịp thời, hay dựa trên những sự kiện trong quá khứ của các hệ hỗ trợ ra quyết định (decision support systems - DSSs). Với các ưu điểm trên, khai thác dữ liệu được ứng dụng rộng rãi trong các lĩnh vực như thương mại, tài chính, y học, giáo dục và các lĩnh vực khác.

Một ví dụ tiêu biểu cho việc khai thác tập phổ biến là phân tích giỏ hàng. Quá trình phân tích này tập trung phân tích thói quen mua sắm của khách hàng bằng cách tìm ra sự kết hợp giữa các danh mục khác nhau từ trong giỏ hàng của họ. Việc khám phá ra những sự kết hợp này giúp ích cho các nhà bán lẻ mở rộng phân phối sản phẩm bởi họ thấu hiểu được những lợi nhuận có được từ những danh mục được khách hàng mua thường xuyên. Cho một ví dụ thực tiễn hơn, nếu khách hàng mua sữa, khả năng họ mua bánh mì trên cùng một lần đi siêu thị là như thế nào? Những thông tin này sẽ giúp cho các nhà bán lẻ tăng doanh thu và giúp họ lựa chọn kế hoạch tiếp thị và trưng bày sản phẩm.

Kết quả phân tích giỏ hàng có thể giúp bạn lên kế hoạch tiếp thị, chiến lược quảng cáo, trưng bày sản phẩm hay lập danh mục bán hàng giảm giá …Ví dụ, kết quả phân tích cho thấy nếu khách hàng mua một máy vi tính thì có thể mua kèm phần mềm diệt vi rút. Từ đó, bạn sẽ có kế hoạch trưng bày sản phẩm hợp lý hơn (Thông tin về máy tính được hiển thị kèm theo phần mềm diệt vi rút được khuyến khích mua).

Từ phân tích giỏ hàng bạn cũng có thể tìm ra một số quy tắc hay luật kết hợp có ích. Ví dụ, thông tin khách hàng mua máy vi tính và cũng mua phần mềm diệt vi rút đã đưa ra luật kết hợp như sau:

Computer → antivirus_software [support = 2%, confidence = 60%]

Độ hỗ trợ (support) và độ tin cậy (confidence) của luật là hai độ đo được quan tâm nhất. Luật có support=2%, nghĩa là số lần giao dịch mà máy vi tính và phần mềm diệt vi rút được mua cùng nhau chiếm 2% trong tổng số các giao dịch; confidence=60%, nghĩa là có 60% khách hàng mua máy vi tính thì cũng sẽ mua phân mềm diệt vi rút.

Luật kết hợp được quan tâm nếu nó thỏa mãn cả hai ngưỡng độ hỗ trợ nhỏ nhất (minimum support threshold) và độ tin cậy nhỏ nhất (minimum confidence threshold).

Phần lớn các thuật toán khai thác dữ liệu hiện nay thường thực hiện trên CSDL phân tán ngang và có quan tâm đến việc bảo toàn tính riêng tư về dữ liệu của các bên tham gia. Với luận văn này, tác giả muốn trình bày một số thuật toán hiện nay có thể khai thác được các luật từ CSDL phân tán ngang cho các bên tham gia, từ đó có thể ứng dụng vào công việc mang lại lợi ích cho các bên và bảo toàn tính riêng tư về dữ liệu của các bên tham gia khai thác**.** Việc cài đặt chương trình thực nghiệm cũng là một đóng góp nhỏ của luận văn.

1.2 KHAI THÁC TẬP PHỔ BIẾN VÀ LUẬT KẾT HỢP.

1.2.1 Một số khái niệm.

Khi dữ liệu được tổ chức theo một cấu trúc, được xử lý và mang đến cho con người những ý nghĩa, hiểu biết nào đó thì khi đó nó trở thành thông tin. Một số người có thể quan niệm thông tin là quan hệ giữa các dữ liệu. Các dữ liệu được sắp xếp theo một thứ tự hoặc được tập hợp lại theo một ràng buộc nào đó sẽ chứa đựng thông tin. Nếu những ràng buộc dữ liệu này được chỉ ra một cách rõ ràng, có ý nghĩa thì đó là các tri thức**.**

1.2.1.1 Tri thức: Là các thông tin tích hợp, bao gồm các sự kiện và mối quan hệ giữa chúng, đã được nhận thức, khám phá, hoặc nghiên cứu. Tri thức có thể được xem như là dữ liệu trừu tượng và tổng quát ở mức độ cao.

1.2.1.2 Khám phá tri thức:

Là quá trình rút trích ra các tri thức chưa được nhận ra, tiềm ẩn trong các tập dữ liệu lớn một cách tự động. Khám phá tri thức hay phát hiện tri thức trong CSDL là một quá trình gồm một loạt các bước phân tích dữ liệu nhằm rút ra được các thông tin có ích, xác định được các giá trị, quy luật tiềm ẩn trong các khuôn mẫu hay mô hình dữ liệu.

1.2.1.3 Khai thác dữ liệu: Là một bước trong quá trình khám phá tri thức, gồm các thuật toán khai thác dữ liệu chuyên dùng với một số quy định về hiệu quả tính toán chấp nhận được để tìm ra các mẫu, các mô hình dữ liệu hoặc các thông tin có ích. Nói cách khác, mục tiêu của khai thác dữ liệu là rút trích ra những thông tin có giá trị tồn tại trong CSDL nhưng ẩn trong khối lượng lớn dữ liệu.

1.2.1.4 Dữ liệu giao dịch: Cho $I = \{i_1, i_2, \ldots, i_n\}$ là tập tất cả các mục dữ liệu (mặt hàng). $T = \{t_1, t_2, \ldots, t_m\}$ là tập tất cả các giao dịch trong CSDL giao dịch *D*. CSDL được cho là quan hệ hai ngôi $\delta \subseteq I \times T$. Nếu mục $i \in I$ xảy ra trong giao dịch $t \in T$ thì ta viết là $(i, t) \in \delta$, *ký hiệu* $i \, \delta \, t$.

Ví dụ về bảng dữ liệu của một cơ sở dữ liệu giao dịch:

Bảng 1.1 Cơ sở dữ liệu giao dịch

Mã giao dịch	Nội dung giao dịch
1	*A, C, T, W*
2	*C, D, W*
3	*A, C, T, W*
4	*A, C, D, W*
5	*A, C, D, T, W*
6	*C, D, T*

1.2.1.5 Độ hỗ trợ:

Cho CSDL giao dịch *D* và tập dữ liệu $X \subseteq I$. Độ hỗ trợ của *X* trong *D*, ký hiệu $\sigma(X)$, được định nghĩa là số giao dịch mà *X* xuất hiện trong *D*.

1.2.1.6 Tập phổ biến:

$X \subseteq I$ được gọi là phổ biến nếu $\sigma(X) \geq$ Minsup (với Minsup là giá trị do người dùng chỉ định). Tập phổ biến ký hiệu là FI (Frequent itemset)

1.2.1.7 Tính chất của tập phổ biến:

Mọi tập con của tập phổ biến cũng là tập phổ biến: Nghĩa là, Nếu X phổ biến thì mọi $Y \subset X$ cũng phổ biến.

Mọi tập cha của tập không phổ biến cũng không phổ biến: Nghĩa là, Nếu X không phổ biến thì mọi Y ($X \subseteq Y$) cũng không phổ biến.

1.2.1.8 Một luật kết hợp có dạng:

$A \rightarrow B$, với $A \subset I$, $B \subset I$ và $A \cap B = \emptyset$. Luật $A \rightarrow B$ ngầm chứa trong D với độ đo Supp s, trong đó s là tỷ lệ các giao dịch trong D chứa $A \cup B$, được diễn tả bằng xác suất $P(A \cup B)$. Luật $A \rightarrow B$ có độ đo Conf c trong tập D, thì c là tỷ lệ giữa các giao dịch trong D chứa A thì chứa luôn B, được diễn tả bằng xác suất P(B/A).

1.2.2 Khai thác tập phổ biến và luật kết hợp.

Cho tập $I = \{I_1, I_2, \ldots, I_m\}$ là một tập các mục dữ liệu. Cho D là bộ dữ liệu cần khai thác, và là một tập trong CSDL giao dịch. Mỗi giao dịch T là một tập các mục dữ liệu và $T \subseteq I$. Mỗi giao dịch có một định danh, được gọi là TID. Cho A là tập các mục dữ liệu. Một giao dịch T được gọi là chứa A khi và chỉ khi $A \subseteq T$.

Một luật kết hợp có dạng $A \rightarrow B$, với $A \subset I$, $B \subset I$ và $A \cap B = \emptyset$. Luật $A \rightarrow B$ ngầm chứa trong D với độ đo Supp s, trong đó s là tỷ lệ các giao dịch trong D chứa $A \cup B$, được diễn tả bằng xác suất $P(A \cup B)$. Luật $A \rightarrow B$ có độ đo Conf c trong tập D, thì c là tỷ lệ giữa các giao dịch trong D chứa A thì chứa luôn B, được diễn tả bằng xác suất P(B/A). Nghĩa là:

$$Supp\ (A \rightarrow B) = P(A \cup B)$$

$$Conf\ (A \rightarrow B) = P(B/A).$$

Những luật thỏa mãn cả hai ngưỡng Minsup và Minconf được gọi là mạnh.

Một tập các mục dữ liệu đơn (items) được gọi là itemset. Một itemset chứa k items được gọi là k-itemset. Chẳng hạn tập {computer, antivirus_software} là 2-itemset. Độ phổ biến của một itemset là số lượng các giao dịch có chứa itemset. Thường được biết với các tên là support count, hay count của itemset.

Nếu độ đo support count của một itemset I thỏa ngưỡng min_sup cho trước thì I là một tập phổ biến. Một tập phổ biến gồm k-items được ký hiệu là FI.

Độ đo Conf của luật A→B có thể thu được từ độ đo support của A và của A ∪ B. Do đó, một khi độ đo support của A, B và A ∪ B được tìm thấy, ta có thể kiểm tra 2 luật kết hợp A→B và B→A xem chúng có mạnh hay không. Như vậy, vấn đề khai thác các luật kết hợp có thể chuyển về bài toán khai thác các tập phổ biến.

Phát biểu bài toán:

Cho một tập các mục *I*, một cơ sở dữ liệu giao dịch *D*, ngưỡng hỗ trợ Minsup, ngưỡng tin cậy Minconf. Tìm tất cả các luật kết hợp X→Y trên CSDL *D* sao cho: sup(X →Y) ≥ Minsup và Conf(X→Y) ≥ Minconf. Bài toán khai thác luật kết hợp có thể được chia ra làm 2 bài toán con được phát biểu trong thuật toán sau:

Nội dung thuật toán

Vào: *I*, *D*, Minsup, Minconf

Ra: Các luật kết hợp thỏa mãn Minsup và Minconf

Các bước thực hiện:

(1) Tìm tất cả các tập mục phổ biến từ CSDL *D* tức là tìm tất cả các tập mục có độ hỗ trợ lớn hơn hoặc bằng Minsup.

(2) Sinh ra các luật từ các tập mục phổ biến (large itemsets) sao cho độ tin cậy của luật lớn hơn hoặc bằng Minconf.

Tùy theo ngữ cảnh các thuộc tính dữ liệu, cũng như phương pháp sử dụng trong các thuật toán; người ta có thể phân bài toán khai thác luật kết hợp ra nhiều nhóm khác nhau. Chẳng hạn, nếu giá trị của các thuộc tính có kiểu boolean thì ta gọi là khai thác luật kết hợp Boolean (Mining Boolean Association Rules)…

Apriori là thuật toán khai thác tập phổ biến và từ đó có thể khai thác luật kết hợp do RaKesh Agrawal, Tomasz Imielinski, Anin Sawami đưa ra vào năm 1993, là nền tảng cho việc phát triển những thuật toán sau này. Thuật toán sinh tập mục ứng cử từ những tập mục phổ biến ở bước trước, sử dụng kỹ thuật "tỉa" để bỏ đi tập mục ứng cử không thỏa mãn ngưỡng hỗ trợ cho trước.

1.2.2.1 Thuật toán Apriori khai thác tập phổ biến.

Input: *D*, cơ sở dữ liệu của các giao tác; Minsup, ngưỡng độ hỗ trợ tối thiểu.

Output: *L*, các tập item phổ biến trong *D*.

Method:

(1) L_1 = find_frequent_1-itemsets(D);

(2) **for** (k = 2; $L_{k-1} \neq 0$; k++) {

(3) C_k= **apriori_gen**(L_{k-1});

(4) **for each** giao tác $t \in D$ { // quét D để đếm

(5) C_t= subset(C_k, t); // lấy các tập con của t mà là các ứng viên

(6) **for each** ứng viên $c \in C_t$

(7) c.count++;

(8) }

(9) $L_k = \{c \in C_k | c.count \geq \text{Minsup}\}$

(10) }

(11) **return** $L = \cup_k L_k$;

procedure apriori_gen(L_{k-1}:tập (k-1) item phổ biến)

(1) **for each** tập item $l_1 \in L_{k-1}$

(2) **for each** tập item $l_2 \in L_{k-1}$

(3) **if** ($l_1[1] = l_2[1]$) $\wedge$ ($l_1[2] = l_2[2]$) $\wedge$ … $\wedge$ ($l_1[k\text{-}2] = l_2[k\text{-}2]$) $\wedge$ ($l_1[k\text{-}1] < l_2[k\text{-}1]$) **then** {

(4) $c = l_1$ kết l_2; // bước kết: phát sinh các ứng viên

(5) **ifhas_infrequent_subset**(c, L_{k-1}) then

(6) **delete**c; // bước xén tỉa: loại bỏ các ứng viên không đạt

(7) **elseadd** c **to** C_k;

(8) }

(9) **return** C_k;

procedurehas_infrequent_subset(c: ứng viên tập k item;

L_{k-1}: các tập (k-1) item phổ biến); // sử dụng kiến thức trước

(1) **for each** tập con (k-1) s **of** c

(2) **if** $s \notin L_{k-1}$ **then**

(3) **return** TRUE;

(4) **return** FALSE;

Trong thuật toán này, giai đoạn đầu đơn giản chỉ là việc tính độ hỗ trợ của các mục. Để xác định L_1, ta chỉ giữ lại các mục có độ hỗ trợ lớn hơn hoặc bằng Minsup.

Trong các giai đoạn thứ k sau đó (k >1), mỗi giai đoạn gồm có 2 pha:

Pha thứ 1: Các (k-1)-itemset phổ biến trong tập L_{k-1} tìm được trong giai đoạn thứ k-1 được dùng để sinh ra các tập mục ứng cử C_k bằng cách thực hiện hàm *apriori_gen().*

Pha thứ 2: CSDL *D* sẽ được quét để tính độ hỗ trợ cho mỗi tập mục ứng cử trong C_k. Các tập mục ứng cử trong C_k mà được chứa trong giao dịch t có thể được xác định một cách hiệu quả bằng việc sử dụng cây băm.

Hàm apriori_gen() thực hiện hai bước:

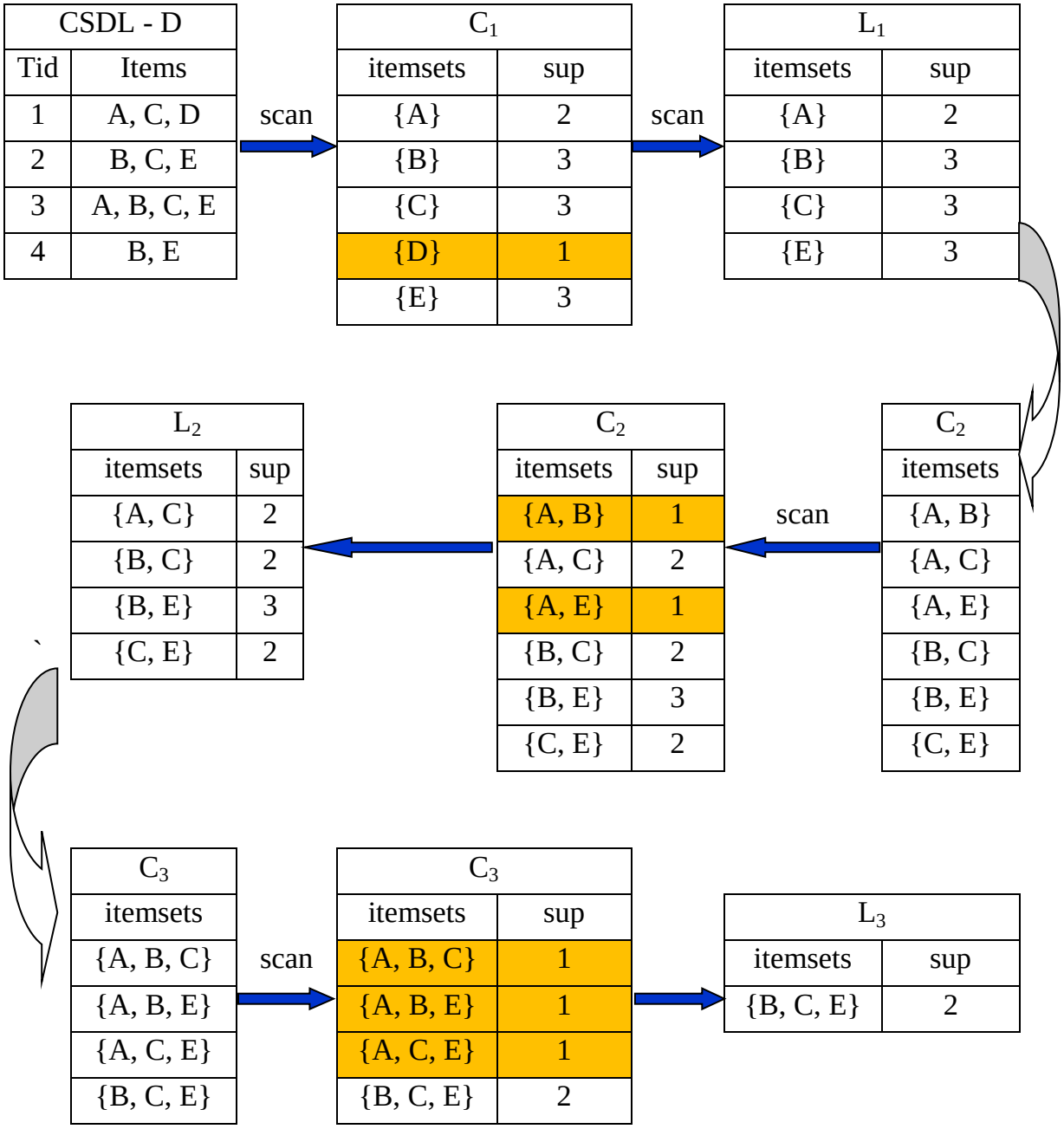
Bước kết nối (Join step): Để tìm L_k, một tập ứng viên các tập *k* item được sinh bởi việc kết L_{k-1} với nó. Tập các ứng viên này được đặt là C_k. Gọi l_1 và l_2 là các tập item trong L_{k-1}. Ký hiệu $l_i[j]$ chỉ tới item thứ *j* trong l_i (vd: $l_1[k–2]$ chỉ tới item cuối thứ 2 trong l_1). Với quy ước, Apriori giả sử các item trong một giao tác hay tập item đã được sắp xếp theo thứ tự từ điển. Đối với tập (*k*–1) item, l_i, nghĩa là các item được sắp xếp thành $l_i[1] < l_i[2] < \ldots < l_i[k\text{-}1]$. Phép kết, L_{k-1} kết L_{k-1}, được thực hiện, với các phần tử của L_{k-1} là khả kết nếu (*k*–2) items đầu tiên của chúng là chung. Do đó, các phần tử l_1 và l_2 của L_{k-1} được kết nếu $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \ldots \wedge (l_1[k–2] = l_2[k–2]) \wedge (l_1[k–1] < l_2[k–1])$. Điều kiện $l_1[k–1] < l_2[k–1]$ đơn giản là bảo đảm rằng không có các bản sao được phát sinh. Tập item tạo ra bởi việc kết l_1 và l_2 là $l_1[1], l_1[2], \ldots, l_1[k\text{-}2], l_2[k\text{-}1]$.

Bước cắt tỉa: C_k là tập cha của L_k, do đó, những phần tử của nó có thể hoặc không thể phổ biến, nhưng tất cả các tập *k* item phổ biến thuộc C_k. Việc quét cơ sở dữ liệu để xác định số lượng của mỗi ứng viên trong C_k sẽ cho kết quả trong việc xác định của L_k (Vd: tất cả ứng viên có số lượng không nhỏ hơn độ hỗ trợ tối thiểu là phổ biến theo định nghĩa và do đó thuộc về L_k). Tuy nhiên, C_k có thể khổng lồ và nó có thể đòi hỏi việc tính toán cực nhọc. Để giảm kích thước của C_k, tính chất Apriori được sử dụng như sau. Vài tập (*k*–1) items là không phổ biến thì không thể là tập con của một tập *k* items phổ biến. Sau đó, nếu vài tập con (*k*–1) items của ứng viên tập *k* items không thuộc L_{k-1}, thì ứng viên cũng không thể là phổ biến và có thể

loại bỏ khỏi C_k. Việc kiểm tra tập con này có thể hoàn thành một cách nhanh chóng bằng cách giữ một cây băm (hash tree) của tất cả các tập item phổ biến.

Thuật toán Apriori-TID dựa vào ý tưởng "không cần thiết phải sử dụng cùng một thuật toán cho tất cả các giai đoạn lên trên dữ liệu". Như đã đề cập ở trên, thuật toán Apriori thực thi hiệu quả ở các giai đoạn đầu, thuật toán Apriori-TID thực thi hiệu quả ở các giai đoạn sau. Phương pháp của thuật toán Apriori-Hybrid là sử dụng thuật toán Apriori ở các giai đoạn đầu và chuyển sang sử dụng thuật toán Apriori-TID ở các giai đoạn sau.

Ví dụ 1.1: Cho cơ sở dữ liệu giao dịch *D*, *I* = {A, B, C, D, E}. Áp dụng thuật toán Apriori để tìm các tập phổ biến thỏa *Minsup = 2*



Hình 1.1 Một ví dụ thuật toán Apriori

Như vậy, với cơ sở dữ liệu ví dụ sau 3 bước của thuật toán Apriori ta thu được tám tập phổ biến: {{A}, {B}, {C}, {E}, {AC}, {BC}, {BE}, {BCE}}.

Thuật toán Apriori cho thấy hiệu suất tốt với tập dữ liệu thưa, ví dụ như: dữ liệu kinh doanh, dữ liệu thị trường, nơi mà các tập phổ biến rất ít. Tuy nhiên, với tập dữ liệu phức tạp, dày như là dữ liệu viễn thông, tập dữ liệu về điều tra dân số trong đó có rất nhiều mẫu phổ biến dài thì hiệu quả của thuật toán Apriori bị giảm rất nhiều. Sự giảm hiệu suất này là do các lý do: Thứ nhất, thuật toán này thực hiện

nhiều lần duyệt qua CSDL để tìm ra tập phổ biến với một ngưỡng hỗ trợ Minsup nào đó, số lần duyệt bằng độ dài của tập phổ biến tìm được. Thứ hai, có thể thấy thuật toán Apriori là thuật toán đúng đắn để kiểm tra toàn bộ các mẫu phổ biến. Tuy nhiên, để khám phá được mẫu phổ biến có kích thước là n thì cần phải sinh và kiểm tra 2^n- 2 mẫu phổ biến tiềm năng (Số lượng tập con có thể có ngoại trừ tập rỗng). Khi mà n lớn thì các phương thức khai thác mẫu phổ biến phụ thuộc vào tốc độ xử lý của phần cứng. Nói một cách khác, thuật toán Apriori trên thực tế không khả thi để khai thác tập mẫu phổ biến lớn mà chỉ áp dụng cho tập mẫu phổ biến có kích thước n nhỏ. Mặt khác, trong nhiều vấn đề của thế giới thực (ví dụ như: Mẫu sinh học, dữ liệu điều tra dân số, vv…) tìm các tập phổ biến có kích thước dài khoảng 30 hoặc 40 thì không phải là không có.

1.2.2.2 Phương pháp IT-Tree

Cấu trúc IT-Tree (Itemset Tidset-tree) và các lớp tương đương [12]

Cho I là tập các danh mục và $X \subseteq I$. Ta định nghĩa một hàm p(X,k) = X[1:k] gồm k phần tử đầu của X và một quan hệ tương đương dựa vào tiền tố (prefix-based) θ_K trên itemset như sau:

$$\forall X, Y \subseteq I, X \equiv_{\theta_k} Y \Leftrightarrow p(X,k) = p(Y,k)$$

Nghĩa là, hai itemset có cùng một lớp tương đương khi và chỉ khi chúng chia sẻ chung k phần tử đầu phổ biến. Mỗi nút trong cây IT-Tree đại diện cho một cặp Itemset-Tidset X x $t(X)$, thực tế là một lớp tiền tố. Tất các các nút con của nút X thuộc về lớp tương đương của nó bởi vì chúng chia sẻ cùng tiền tố X.

Ký hiệu một lớp tương đương là $[P]= \{l_1,l_2,...,l_n\}$, trong đó P là nút cha và mỗi l_i là một mục dữ liệu đơn, đại diện cho nút Pl_i x $t(Pl_i)$. Chẳng hạn, nút gốc của cây tương ứng với lớp [ ] = {A,C,D,T,W}, nút trái cùng của gốc là lớp [A] chứa tất cả các itemset chứa A là tiền tố, nghĩa là tập {C,D,T,W}. Như vậy, mỗi lớp thành viên đại diện cho một con của nút cha. Một lớp đại diện cho các mục dữ liệu mà các mục dữ liệu đó là tiền tố để có thể mở rộng thành các lớp phổ biến mới. Rõ ràng, không có cây con nào của một tiền tố không phổ biến được xem xét. Sức mạnh của phương pháp lớp tương đương là nó chia không gian tìm kiếm ban đầu thành các vấn đề nhỏ độc lập. Đối với mỗi nút gốc con của nút X, có thể xem nó như một vấn đề mới hoàn toàn, mỗi nút có thể sinh ra các mẫu dưới nó.

Thuật toán phát sinh tập phổ biến [12]

Đầu vào: Lớp tương đương **[*P*]** ban đầu chứa tất cả các tập phổ biến 1-itemset và ngưỡng phổ biến Minsup.

Kết quả: tập **FI** gồm tất cả các tập phổ biến của CDSL.

Phương pháp thực hiện:

```
ECLAT()
    [∅] = {i ∈ I: σ(i) Minsup}
    ENUMERATE_FREQUENT([∅])
ENUMERATE_FREQUENT([P])
  for all l_i ∈ [P] do
    [P_i] = ∅
    for all l_j ∈ [P], with j > i do
      I = l_j
      T = t( l_i ) ∩ t( l_j )
      if |T| ≥ Minsup then
        [P_i] = [P_i] ∪ { I × T }
    ENUMERATE_FREQUENT([P_i])
    Delete [P_i]
```

Hình 1.2 Thuật toán sinh tập phổ biến thỏa Minsup

Minh hoạ thuật toán

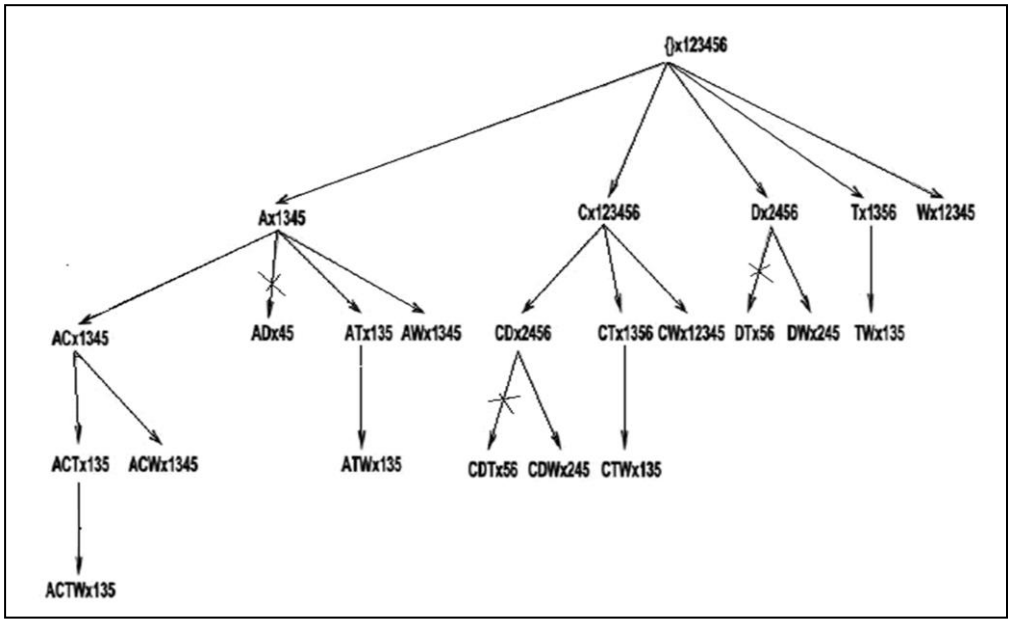
Xét CSDL ở bảng 1.1 với Minsup = 50% (chứa từ 3 TID trở lên). Ta có cây tìm kiếm minh họa cho quá trình tìm tập phổ biến như hình 1.3.

Nhận xét

Cây tìm kiếm IT-Tree luôn lệch trái do:

i) Ứng với mỗi lớp tương đương l_i, ta cần xét với mọi l_j ($j > i$), cho nên i càng nhỏ thì số lượng j cần xét càng lớn.

ii) Khi $| t(l_i) | > | t(l_j) |$ thì phần giao nhau giữa $t(l_i)$ với các lớp tương đương khác thường sẽ lớn hơn phần giao của $t(l_j)$ với các lớp tương đương còn lại.



Hình 1.3 Cây tìm kiếm tập FI thỏa ngưỡng Minsup = 50%

ENUMERATE_FREQUENT_SORT([*P*])

SORT([*P*])

for all $l_i \in [P]$ do

$[P_i] = \varnothing$

for all $l_j \in [P]$, with j > i do

$I = l_j$

$T = t(l_i) \cap t(l_j)$

if $|T| \geq$ *Minsup* then

$\{[P_i] = [P_i] \cup \{ I \times T \}$

ENUMERATE_FREQUENT_SORT($[P_i]$)

Delete $[P_i]$

Hình 1.4 Thuật toán tìm FI bằng thuật toán sắp xếp

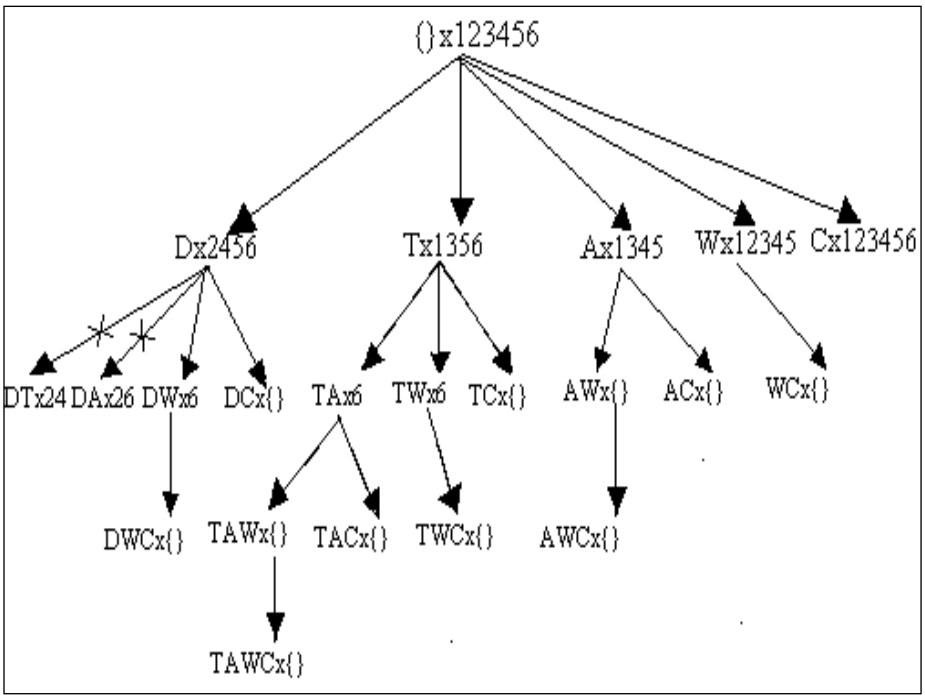
Với các nhận xét trên, ta thấy:

i) Không thể cải thiện được, còn để tránh.

ii) Ta chỉ cần sắp xếp các l_i trong lớp tương đương [*P*] theo chiều tăng

dần của độ hỗ trợ. Và với sự cải tiến này, cây IT-Tree sẽ ít lệch trái hơn.

Hình 1.5 minh họa cây tìm kiếm IT-Tree với phương pháp sắp xếp. Có thể thấy cây ít lệch trái hơn và số tập phát sinh không thỏa ngưỡng Minsup ít hơn (trong trường hợp có sắp xếp và trong trường hợp không sắp xếp). Điều này dẫn đến thời gian tính toán sẽ nhanh hơn và quá trình tìm kiếm ít tốn không gian bộ nhớ hơn (do cơ chế đệ qui cần phải lưu lại các nhánh con bên phải để xử lý sau trước khi gọi đệ qui). Tuy nhiên, có thể thấy các nút con được phát sinh ra trên cùng một mức của một nút cha nào đó thường đã thỏa điều kiện sắp tăng nên ta chỉ cần sắp xếp ở mức 1 của cây, các mức còn lại không cần sắp xếp bởi vì thường nó sẽ được thừa hưởng kết quả từ mức trước đó.



Hình 1.5 Cây tìm kiếm tập FI thỏa ngưỡng Minsup = 50% có sắp xếp

Diffset để tính nhanh độ hỗ trợ [11],[12]

Giả sử chúng ta đang thao tác trên IT-pair sử dụng định dạng dữ liệu dọc (vertical). Các thuật toán khai thác dữ liệu sử dụng định dạng dọc cho thấy rất hiệu quả và thực thi tốt hơn cách tiếp cận theo định dạng ngang (horizontal). Lợi ích chính của việc sử dụng định dạng dọc là:

i) Tính toán độ hỗ trợ đơn giản và nhanh hơn. Chỉ đòi hỏi tính phần giao trên các giao tác và được hỗ trợ tốt bởi các CSDL hiện hành. Nói cách

khác, tiếp cận theo định dạng ngang đòi hỏi một cấu trúc dữ liệu phức tạp hơn.

ii) Nó tự động tỉa các thông tin không liên quan, chỉ có các định danh giao tác (tid) có liên quan với tần số xác định được giữ lại sau mỗi lần giao. Đối với những CSDL có nhiều giao tác, phương pháp dọc làm giảm số thao tác I/O trên CSDL.

Mặc dù có rất nhiều thuận lợi trong phương pháp dọc, nhưng khi số phần tử của Tidset lớn (với nhiều mục dữ liệu phổ biến), phương pháp này bắt đầu chịu tổn thất bởi vì thời gian tính phần giao quá lớn. Hơn nữa, kích thước của các Tidset được sinh ra tức thời cũng rất lớn, đòi hỏi dữ liệu phải được giảm bớt và ghi tạm lên đĩa. Vì vậy, trên các CSDL đặc, với đặc điểm là có nhiều mục dữ liệu và tần số xuất hiện cao, phương pháp dọc làm giảm nhanh chóng các thuận lợi của chúng. Chính vì vậy, Zaki và các đồng sự đã đưa ra cách biểu diễn dữ liệu dọc có tên là Diffset (Difference of two Tidset) được đề nghị trong [11]. Diffset lưu vết các sự khác nhau trong các tid của các mẫu ứng viên từ mẫu phổ biến cha của nó. Các khác nhau này sẽ truyền đi theo mọi hướng từ một nút đến các con của nó bắt đầu từ gốc. Diffset làm giảm kích thước bộ nhớ yêu cầu để lưu kết quả tức thời. Vì thế, thậm chí ngay cả dữ liệu đặc, làm việc trên toàn bộ các mẫu của các thuật toán khai thác dọc có thể phù hợp hoàn toàn trong bộ nhớ chính. Vì Diffset là một phần nhỏ của kích thước Tidset nên thao tác giao nhau được thực thi khá hiệu quả.

Một cách hình thức hơn, xét một lớp với tiền tố P. Gọi d(X) là Diffset của X (theo khía cạnh là một Tidset tiền tố) là toàn bộ các tid hiện hành. Giả sử PX và PY là hai lớp thành viên bất kỳ của P. Theo định nghĩa của độ hỗ trợ thì $t(PX) \subseteq t(P)$ và $t(PX) \subseteq t(P)$. Hơn nữa, có thể tính được độ hỗ trợ của *PXY* bằng cách kiểm tra số phần tử của $t(PX) \cap t(PY) = t(PXY)$.

Bây giờ, giả sử chưa có $t(PX)$ nhưng có $d(PX)$ (được tính = $t(P) - t(X)$). Tương tự, giả sử cũng có $d(PY)$. Làm sao tính *PXY* ($d(PXY)$) nếu *PXY* là phổ biến? Có thể tính độ hỗ trợ của *PXY* theo công thức:

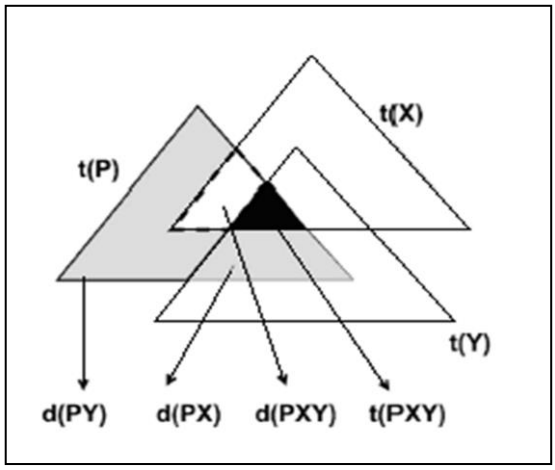
$$\sigma(PXY) = \sigma(PX) - |d(PXY)|$$

Mà theo định nghĩa, ta có $d(PXY) = t(PX) - t(PY)$. Nhưng chúng ta chỉ có

Diffset và không có Tidset như công thức yêu cầu, điều này rất dễ giải quyết vì ta có:

$$\begin{aligned} d(PXY) &= t(PX) - t(PY) = t(PX) - t(PY) + t(P) - t(P) \\ &= (t(P) - t(PY)) - (t(P) - t(PX)) \\ &= d(PY) - d(PX) \end{aligned}$$

Nói cách khác, thay vì tính d(PXY) theo sự khác nhau của các Tidset là t(PX) – t(PY), chúng ta tính nó dựa vào sự khác nhau của các Diffset là d(PY) – d(PX).



Hình 1.6 Các miền khác nhau của Tidset và Diffset [11]

Hình 1.6 minh họa các miền khác nhau của các Tidset và Diffset của một lớp tiền tố được cho và bất kỳ hai thành viên nào của nó.

Thuật toán sinh tập FI sử dụng Diffset

ENUMERATE_FREQUENT_DIFF([*P*])

SORT([*P*])

for all $l_i \in [P]$ do

$[P_i] = \varnothing$

for all $l_j \in [P]$, with j > i do

$I = l_j$

if $P = \varnothing$ then

$T = t(l_i) \setminus t(l_j)$

else

$T = d(l_j) \setminus d(l_i)$

if $\sigma(l_i)$ - $|T| \geq$ *Minsup* then

$[P_i] = [P_i] \cup \{ I \times T \}$

ENUMERATE_FREQUENT_DIFF($[P_i]$)

Delete $[P_i]$

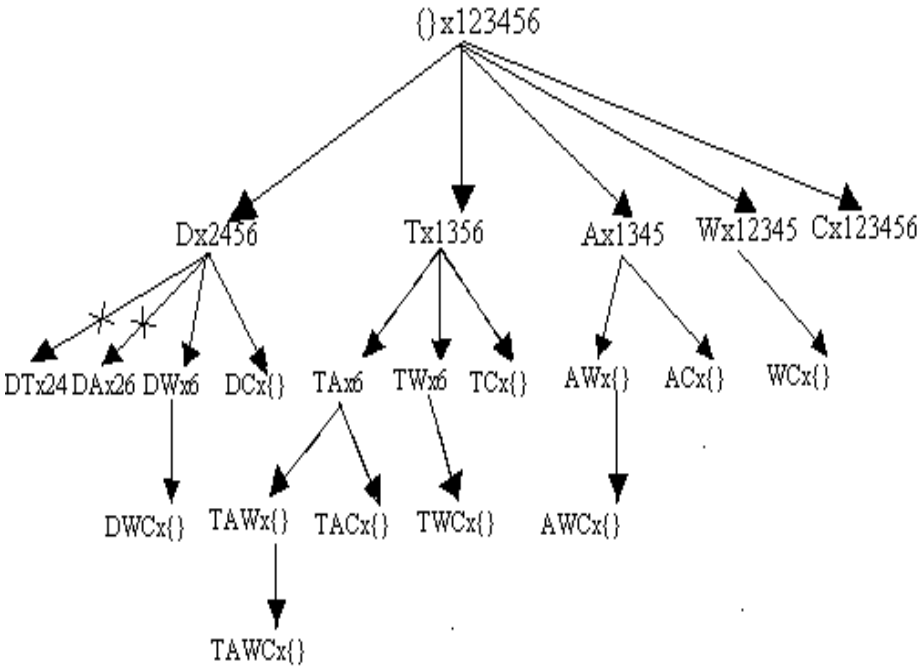
Hình 1.7 Thuật toán sinh tập FI sử dụng Diffset [11]

Cây tìm kiếm IT-Tree với Diffset

Hình 1.8 minh họa việc tìm kiếm trên IT-Tree của thuật toán sinh tập FI thỏa ngưỡng Minsup = 50% sử dụng *Diffset*. Có thể thấy $|T|$ ứng với mỗi nút $I \times T$ trên cây IT-Tree sử dụng *Diffset* nhỏ hơn $|T|$ trên cây sử dụng Tidset. Điều này dẫn đến kích thước vùng nhớ yêu cầu để lưu trữ *Diffset* sẽ nhỏ hơn rất nhiều so với sử dụng Tidset và thao tác tính *T* cũng sẽ nhanh hơn.

Xét $d(DT) = t(D) - t(T) = 2456 - 1356 = 24$

Xét $d(DWC) = d(DC) - d(DW) = \varnothing - 6 = \varnothing$



Hình 1.8 Cây tìm kiếm IT-Tree sử dụng Diffset [11]

1.2.3 Thuật toán khai thác luật kết hợp

Input : tập phổ biến: FI, ngưỡng tin cậy *Minconf*

Output : tập các luật kết hợp AR

Method:

(1) SORT (FI) // hàm sắp xếp tập FI tăng theo k-itemset

(2) AR = ∅

(3) **For each** $f_i \in$FI with $|f_i| > 1$ do

(4) **For each** $f_j \in$ FI with j < i do

(5) **if** $f_j \subset f_i$ **then**

(6) Conf = Sup(f_i)/ Sup(f_j)

(7) **if** Conf≥*Minconf* **then**

(8) AR = AR∪{$f_j \rightarrow f_i \setminus f_j$ (Sup(f_i), Conf)}

(9) **return** AR

Với FI = {{A}, {C}, {D}, {T}, {W}, {AC}, {AT}, {AW}, {CD}, {CT}, {CW}, {DW}, {TW}, {ACT}, {ACW}, {ATW}, {CDW}, {CTW}, {ACTW}},19 tập phổ biến và *Minconf*=80%. Sau khi chạy thuật toán khai thác luật kết hợp trên ta

có: tập các luật như sau: AR= {D→C; T→C; A→W; W→A; A→C; C→W; W→C; DW→C; AT→W; TW→A; AT→C; A→CW; W→AC; AC→W; AW→C; CW→A; AT→CW; TW→AC; ACT→W; ATW→C; CTW→A}.

Kết chương

Trong chương 1 của luận văn đã trình bày một số khái niệm cơ bản về tập phổ biến, luật kết hợp và 2 thuật toán: khai thác tập phổ biến sử dụng thuật toán Apriori và sử dụng phương pháp IT-Tree của các tác giả khác nhau [11, 12], Thuật toán khai thác luật kết hợp cũng được tác giả trình bày trong chương này. Đây là hai thuật toán thường sử dụng trong khai thác tập phổ biến, luật kết hợp trên CSDL phân tán bảo toàn tính riêng tư sẽ trình bày trong chương 2 và chương 3.

CHƯƠNG 2
MỘT SỐ THUẬT TOÁN BẢO TOÀN TÍNH RIÊNG TƯ TRONG KHAI THÁC TRÊN CSDL PHÂN TÁN NGANG

2.1 GIẢI THUẬT KHAI THÁC TẬP PHỔ BIẾN ĐẢM BẢO TÍNH RIÊNG TƯ VÀ CHỐNG THÔNG ĐỒNG TRÊN CSDL PHÂN TÁN NGANG

2.1.1 Giao thức đảm bảo tính riêng tư trong tính độ phổ biến toàn cục

Trong phần này tác giả luận văn trình bày phần tìm hiểu của mình về nghiên cứu của các tác giả trong [1, 11]. Để xây dựng giao thức tính độ phổ biến toàn cục, trước hết, các tác giả định rằng tất cả m bên đều biết một số nguyên A thỏa điều kiện A ≥ max {$|^1DB|$, $|^2DB|$, …, $|^mDB|$}, việc tiết lộ giá trị A như vậy không làm ảnh hưởng lớn đến tính riêng tư, tuy nhiên trong phần sau, chúng tôi sẽ đề xuất một giao thức chọn A an toàn.

Đặt: $^ix = \frac{|^iDB|}{A+\varepsilon}$ và $^iy = \frac{(|^iX|+\varepsilon)}{A+\varepsilon}$

Với ε là số thực rất bé được biết trước bởi tất cả các bên, (ε≤ 1, trong thực tế ta có thể chọn ε = 1). Khi đó ta có:

$$0 < {}^ix < 1 \text{ và } 0 < {}^iy < 1$$

Mỗi bên S_i phát sinh một số thực ngẫu nhiên $^ic \in (0,1)$. Áp dụng giao thức tính tích của hai tổng đảm bảo riêng tư SPoS trong [1], ta tính được:

$$p_1 = (^1x + {}^2x + ... + {}^mx)(^1c + {}^2c + ... + {}^mc) \quad (2.1)$$

$$p_2 = (^1y + {}^2y + ... + {}^my)(^1c + {}^2c + ... + {}^mc) \quad (2.2)$$

Chia (2.1) cho (2.2) ta được:

$$\frac{P_1}{P_2} = \frac{\sum_{i=1}^{m} {}^ix}{\sum_{i=1}^{m} {}^iy} = \frac{\sum_{i=1}^{m} |^iX| + m\varepsilon}{\sum_{i=1}^{m} |^iDB|} \quad (2.3)$$

Trong khai thác dữ liệu, m (số lượng các bên) nhỏ hơn rất nhiều so với số lượng giao tác trong CSDL, hơn nữa ε rất bé (ε≤ 1), nên thành phần mε có thể bỏ qua trong công thức (2.3). Do vậy ta có:

$$\frac{P_1}{P_2} = \frac{\sum_{i=1}^{m} |^i X| + m\varepsilon}{\sum_{i=1}^{m} |^i DB|} \approx \frac{\sum_{i=1}^{m} |^i X|}{\sum_{i=1}^{m} |^i DB|} = \sigma(X) \qquad (2.4)$$

Công thức (2.4) cho ta độ phổ biến toàn cục của itemset X.

2.1.2 Giải thuật khai thác tập phổ biến

Để tiết kiệm bộ nhớ và tăng tốc độ xử lý cục bộ tại mỗi S_i, các tác giả đã chọn thuật toán khai thác tập phổ biến dựa trên cấu trúc chuỗi bit động [2], dữ liệu liên quan đến mỗi tập mục dữ liệu được lưu trữ bởi một chuỗi bit động (DBS : Dynamic Bit String). Kết hợp với giao thức tính độ phổ biến toàn cục được xây dựng trong phần 2.1.1 để có thể áp dụng trên CSDL phân tán ngang, đảm bảo riêng tư.

Mỗi S_i sử dụng một FITree (Frequent- ITree) để lưu trữ tập phổ biến toàn cục, mỗi nút trong FITree gồm các thông tin: Itemset X, DBS(X), $\sigma(^iX)$ và $\sigma(X)$.

Một số ký hiệu sử dụng trong thuật toán:

^iLL_k: Tập itemset phổ biến cục bộ tại S_i trong lần duyệt thứ k.

C_k: Tập ứng viên toàn cục ở lần duyệt thứ k.

^iBT: CSDL của S_i được nén, mỗi phần tử của ^iBT gồm 3 phần: Itemsret X, DBS(X) và độ phổ biến toàn cục $\sigma(X)$.

Thuật toán khai thác CSDL phân tán ngang bảo đảm tính riêng tư:

CREATE_FITREE(^{i}DB, {S_j/j = 1, 2, …, m})

1. Áp dụng cấu trúc chuỗi bit động, nén CSDL ^{i}DB của S_i vào ^{i}BT.

2. A = **UPPER_BOUND**(|^{i}DB|}).

3. Phát sinh ngẫu nhiên số thực $^i c \in (0, 1)$.

4. $P_1 = SPoS(^i c, \frac{|^i DB|}{A+\varepsilon})$.

5. L= ∅ // L là tập các Item phổ biến.

6. k=1;

7. ^iLL_k =Tập phổ biến cục bộ ở S_i.

8. C_k=SECURE_UNION(^iLL_k).

9. **For Each** $X \in C_k$.

10. $\sigma(X) = \text{SECURE\_SUPPORT}(X)$

11. **If** $\sigma(X) \geq$ Minsup **then**

12. $L = L \cup \{X\}$

13 **End for**

14. Khởi tạo FITree = Empty

15. FITree.Children=L.

16. EXTEND _FITREE(FITree, Minsup,0).

17. **Return** FITree.

Hình 2.1 Thủ tục CREATE_FITREE

Mỗi bên S_i (i=1,2,.., m) sử dụng thủ tục CREATE_FITREE sau đây để có được FITree toàn cục

SECURE_SUPPORT(X).

1. $P_2 = SPoS(\frac{|X|+\varepsilon}{A+\varepsilon}, {}^{i}c)$

2. $\sigma(X) = \frac{P_1}{P_2}$

3. ***Return*** $\sigma(X)$.

Hình 2.2 Thủ thục SECCURE_SUPPORT(X)

Thủ tục SECURE_SUPPORT(X) là sự cài đặt của giao thức tính độ phổ biến toàn cục của itemset X được xây dựng trong phần 2.1.1. Giao thức SPoS($^{i}x_1$, $^{i}x_2$) được vận dụng vào thủ tục để tính giá trị trung gian $P = \sum_{k=1}^{m} {}^{k}x_1 \sum_{k=1}^{m} {}^{k}x_2$ trong tính toán bảo vệ tính riêng tư các giá trị $^{i}x_1$, $^{i}x_2$.

Sau khi các nút con của nút gốc trong FITree (gồm các L-itemset phổ biến toàn cục) được tạo (từ dòng 1 đến dòng 14). Thủ tục EXTEND_FITREE được gọi một cách đệ qui để mở rộng và hoàn thiện FITree chứa tập đầy đủ các itemset phổ biến toàn cục.

Từ tính chất "*Một itemset là phổ biến toàn cục thì phải phổ biến cục bộ ít nhất tại một bên nào đó*" [3], các tác giả đã sử dụng phép hợp an toàn (Secure Union) trong [4] để tìm tập itemset ứng viên trong mỗi bước xử lý.

EXTEND _FITREE (FITree, Minsup, k).

1. k=k+1.

2. **For** l=1 **To** FITree.Children.Count-1

3. X_i = To FITree.Children[i].

4. ${}^iC_k$=Tập phổ biến cục bộ ở S_i phát sinh từ FITree.

5. C_k= SECURE_UNION (${}^iC_k$).

6. **End for**

7. **For** j = l+1 **to** FITree.Children.Count

8. X_j= FITree.Children[j]

9. **If** $({}^l(X_i \cup X_j) \in C_k)$ **then**

10. $$P_2 = SPoS(\frac{|{}^i(X_i \cup X_j)| + \varepsilon}{A + \varepsilon}, {}^i c)$$

11. $\sigma(X_i \cup X_j) = SECURE\_SUPPORT(X_i \cup X_j)$

12. **If** $\sigma(X_i \cup X_j) \geq$ Minsup **then**

13. X_i.children.Add($X_i \cup X_j$)

14. FITree.DBS=Empty;

15. **End for**

16. **EXTEND _FITREE** (X_i, Minsup,k).

Hình 2.3 Thủ tục EXTEND_FITREE

UPPER_BOUND(^{i}DB)

1. Phát sinh số nguyên ngẫu nhiên r_i
2. **If** (i=1) **then //S_i** là master
3. Gởi $v_1 = r_1 + |^1DB|$ đến S_2.
4. Nhận v_m từ S_m.
5. Gởi v_m đến tất cả các S_j ($j \neq i$).
6. **Else //**S_i không phải là master
7. Nhận v_{i-1} từ S_{i-1}
8. Gởi $v_i=\max\{v_{i-1}, r_i + |^iDB|\}$ đến $S_{(i \bmod m)+1}$
9. Nhận v_m từ S_1
10. **Return** v_m

Hình 2.4 Thủ tục UPPER_BOUND

Ví dụ 2.1: (minh họa thuật toán) bảng 2.1 cho dữ liệu minh họa cho thuật toán với trường hợp cụ thể gồm hai bên S_1, S_2 với Minsup=40% như sau:

Bảng 2.1 Minh họa hệ thống gồm hai bên S_1, S_2

S_1	
Trans	Items
1	A, B
2	A, C
3	A, B, C
4	B, C
5	A, C, D

FITree={}

S_2	
Trans	Items
6	C, D
7	A, B, D
8	A, B, C
9	A, B

FITree={}

Kết quả của bước nén các CSDL cục bộ (dòng 1) để đưa vào bộ nhớ trong:

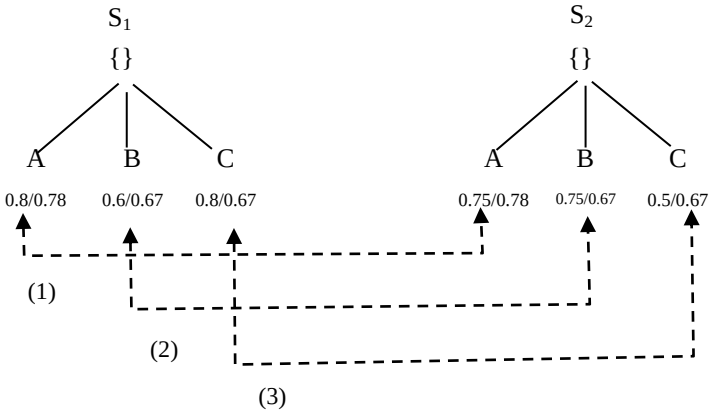
- Nén CSDL ^{1}DB: ^{1}BT={(A, 29, ?), (B, 22, ?), (C, 15, ?),(D, 1, ?)}
- Nén CSDL ^{2}DB: ^{2}BT={(A, 7, ?), (B, 7, ?), (C, 10, ?),(D, 12, ?)}

(Sử dụng ký hiệu ? để biểu diễn cho độ phổ biến toàn cục chưa biết của các itemsets).

Tạo các nút con của nút gốc của FITree (dòng 13 đến dòng 15 của thuật toán):

- Tập ứng viên toàn cục C_1 = {A, B, C}.

- Lần lượt tính độ phổ biến toàn cục các itemset A, B và C (dòng 10, 11, 12), tất cả đều có độ phổ biến toàn cục lớn hơn Minsup nên L={A, B, C} là con của nút gốc FITree ở mỗi S_i (kết quả như hình 2.5).



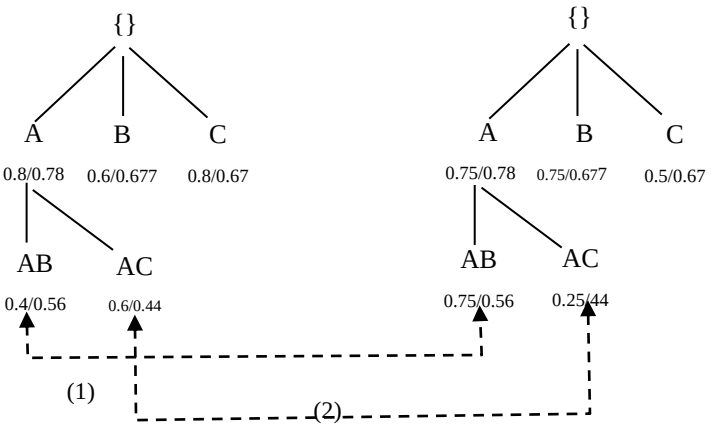
Hình 2.5 Kết quả FITree sau khi xử lý nút gốc [1]

Thủ tục EXTEND_FITREE để mở rộng và hoàn thiện FITree

Tạo các nút con cho nút A (từ dòng 6-12 thủ tục EXTEND_FITREE):

- Tập ứng viên toàn cục: C_2 = {AB, AC}

- Lần lượt tính độ phổ biến toàn cục cho AB, AC. Cả hai có độ phổ biến toàn cục lớn hơn Minsup nên đều là nút con của nút A trong từng FITree trong mỗi S_i (kết quả như hình 2.6).



Hình 2.6 Kết quả FITree sau khi xử lý nút A [1]

Để tiết kiệm bộ nhớ, thủ tục sẽ hủy DBS của nút A sau khi xử lý (dòng 13).

Tạo các nút con cho nút AB:

Độ phổ biến toàn cục của itemset ABC trên cả S_1 và S_2 lần lượt là 0.2 và 0.25, cả hai đều nhỏ hơn Minsup = 40% nên tập ứng viên toàn cục tương ứng $C_3 = \varnothing$, không có nút con của nút AB.

Tạo các nút con cho nút B:

- Tập ứng viên toàn cục ứng với nút B là C_4={BC}.
- Độ phổ biến của BC là 0.33 < Minsup nên không tạo nên nút con của B.

Kết thúc thuật toán, hình 2.6 cũng là tình trạng cuối cùng của FITree, tập phổ biến tìm được là tập các itemsets tương ứng với các nút trong FITree.

2.1.3 Đánh giá thuật toán

2.1.3.1 Đánh giá mức độ bảo vệ riêng tư

Mức độ đảm bảo riêng tư của thủ tục UPPER_BOUND: Trong thủ tục này, mỗi S_i đều cộng thêm vào $|^iDB|$ của mình một số nguyên ngẫu nhiên r_i trước khi trao đổi với bên khác, do vậy $|^iDB|$ của S_i được đảm bảo riêng tư và cũng chống lại khả năng thông đồng.

Mức độ đảm bảo riêng tư của giao thức tìm tập ứng viên toàn cục (SECURE_UNION): Ở đây sử dụng phép hợp đảm bảo riêng tư trong [1] để tìm tập ứng viên toàn cục. Trong [1], các tác giả đã chứng minh giao thức này là an toàn trong cả môi trường nguy hiểm (malicious) và có khả năng chống thông đồng.

Mức độ đảm bảo riêng tư của giao thức tính độ phổ biến toàn cục cho các itemset: Các tác giả trong [1] đã chứng minh giao thức SPoS là an toàn theo mức độ an toàn của hệ mã hóa sử dụng. Hơn nữa, giao thức SPoS có mức độ bảo vệ riêng tư và chống thông đồng hoàn toàn (full - private). Tuy nhiên chúng ta cũng cần phải xem xét cụ thể khi áp dụng giao thức này để tính độ phổ biến toàn cục.

Độ phổ biến toàn cục của itemset X được xác định thông qua công thức (2.4). Trong đó mức độ đảm bảo riêng tư trong tính toán giá trị P_1, P_2 được xác định theo hai giao thức SPoS và SUPPER_BOUND (full - private).

Theo định lý tổng hợp [1], để đánh giá mức độ duy trì tính riêng tư của toàn bộ thuật toán, ta xem các giao thức con đảm bảo riêng tư đã dùng như các hộp đen

và xem xét mức độ riêng tư của giao thức tính độ phổ biến toàn cục của itemset X. Độ phổ biến toàn cục của X được tính theo công thức.

$$\sigma(X) = \frac{\sum_{i=1}^{m} |^i X|}{\sum_{i=1}^{m} |^i DB|} \qquad (2.5)$$

Các tác giả đã giả thiết các trường hợp có thể xảy ra sau đây:

Trường hợp có ít hơn m - 1 bên thông đồng:

Phương trình (2.5) luôn có nhiều hơn 4 biến, do vậy độ phổ biến cục bộ của X trong các bên còn lại vẫn được giữ bí mật.

Trường hợp có m - 1 bên S_{i1}, S_{i_2},..., $S_{i_{m-1}}$ thông đồng với nhau:

Nếu tất cả S_{i1}, S_{i_2},..., $S_{i_{m-1}}$ tham gia trong giao thức tính độ phổ biến toàn cục của X với kích cỡ CSDL cũng như độ phổ biến cục bộ bằng 0, ta có:

$$\sigma(\mathrm{X}) = \frac{\sum_{\mathrm{i}=1}^{\mathrm{m}} |^{\mathrm{i}} \mathrm{X}|}{\sum_{\mathrm{i}=1}^{\mathrm{m}} |^{\mathrm{i}} \mathrm{DB}|} = \frac{|^{\mathrm{i_m}} \mathrm{X}|}{|^{\mathrm{i_m}} \mathrm{DB}|}$$

Khi đó, các S_{i_1}, S_{i_2},..., $S_{i_{m-1}}$ sẽ biết được độ phổ biến cục bộ của X tại S_{i_m}. Tuy nhiên, trong mô hình khai thác CSDL từ nhiều bên, các bên thực hiện theo đúng giao thức đã được định sẵn, độ phổ biến cục bộ của itemset X có thể bằng 0 nhưng số lượng giao tác trong mỗi CSDL phải lớn hơn (rất nhiều) so với 0. Do vậy, việc suy luận chính xác độ phổ biến cục bộ của X trong S_{i_m} là không thể.

Nếu độ phổ biến của X lớn hơn 0 tại ít nhất một bên trong m-1 bên S_{i_1}, S_{i_2},..., $S_{i_{m-1}}$, giao thức tính độ phổ biến toàn cục được xây dựng cũng đảm bảo tính riêng tư hoàn toàn như giao thức SPoS.

Tóm lại, giao thức tính độ phổ biến toàn cục của luận văn đề xuất đảm bảo tính riêng tư hoàn toàn với môi trường semi-honest.

2.1.3.2 Đánh giá chi phí truyền thông

Theo thuật toán được xây dựng, độ phổ biến toàn cục của mỗi itemset X được tính bởi công thức (2.5)

P_1 và giá trị A sử dụng trong thuật toán chỉ được tính một lần trong giai đoạn khởi tạo, do vậy ta chỉ cần đánh giá chi phí truyền thông phát sinh từ việc tính P_2.

Mỗi giá trị P_2 được tính tương ứng với một lần thực hiện giao thức SPoS, số lượng thông điệp truyền đi trong hệ thống là: 4m(m-1) với m là số lượng các bên. Tuy nhiên, quá trình trao đổi thông điệp xảy ra song song trên từng cặp hai thành viên độc lập và trong [1] đã chứng minh rằng thời gian chạy của giao thức SPoS chỉ là tuyến tính theo m (O(m)), như vậy thời gian để tính P_2 cũng là O(m), nếu thời gian thực hiện các xử lý cục bộ là như nhau giữa các bên và không xét đến phép hợp an toàn để tỉa tập ứng viên, tổng thời gian chạy của toàn bộ thuật toán tìm tập phổ biến được xây dựng cũng là O(m).

2.2 GIAO THỨC KHAI THÁC CSDL PHÂN TÁN NGANG BẢO ĐẢM TÍNH RIÊNG TƯ.

2.2.1 Đặt vấn đề

Khi tiếp cận việc khai thác trên CSDL phân tán, các CSDL chia sẻ mô hình khai thác cục bộ để tìm ra kết quả khai thác cuối cùng các tác giả trong [8] đã trình bày một giao thức khai thác trên CSDL phân tán ngang bảo toàn tính riêng tư mà tác giả luận văn sẽ trình bày sau đây. Do mô hình dữ liệu cục bộ có thể chứa thông tin riêng tư nhạy cảm, do đó cần có giao thức để bảo đảm tính riêng tư cho các dữ liệu này. Các tác giả đã đề xuất giao thức theo cách, trước hết tìm tập ứng viên rút gọn sau đó tìm itemset phổ biến toàn cục của tập ứng viên rút gọn này. Mục tiêu đặt ra là kết quả khai thác chính xác, không tiết lộ dữ liệu cục bộ, độ hỗ trợ của các itemset cục bộ và kích thước dữ liệu cục bộ, đồng thời giao thức có chi phí truyền thông thấp. Các tác giả dựa vào giao thức do Mahmoud Hussein và các đồng nghiệp đề xuất năm 2008 trong [5] với hai điểm cải tiến chính: tìm tập phổ biến tối đại (MFI) cục bộ và tính độ hỗ trợ toàn cục sử dụng mã hóa Paillier, từ đó nâng cao tính riêng tư, có chi phí truyền thông thấp hơn và kết quả khai thác hoàn toàn chính xác.

Bảo đảm tính riêng tư: trong giao thức đề xuất các tác giả đã tập trung vào bảo đảm các vấn đề : bảo đảm không tiết lộ tập phổ biến tối đại (MFI_i) cục bộ của các bên và bảo đảm độ hỗ trợ toàn cục không bị tiết lộ trong quá trình khai thác.

2.2.2 Cơ sở lý thuyết

Tập phổ biến tối đại (MFI: Maximal Frequent Itemsets): M là tập phổ biến tối đại nếu M là tập phổ biến và không tồn tại tập phổ biến S khác M mà $M \subset S$. Ta có: |MFI| << |FI|.

Mã hóa Paillier: là hệ mã có tính chất đồng hình theo phép cộng, ta có thể tính tổng của các bản rõ dựa vào tích của các bản mã, nghĩa là:

$$E_g(m_1) * E_g(m_2) = E_g(m_1 + m_2)$$

Sau khi giải mã sẽ là tổng của hai bản rõ, tức là:

$$D(E_g(m_1) * E_g(m_2)) = m_1 + m_2.$$

Như vậy ta có thể tính tổng $m_1 + m_2$ mà không cần biết m_1 và m_2.

Mã hoá Paillier dựa vào song ánh:

$$E_g \begin{cases} Z_N * Z_N^* \rightarrow Z_{N^2}^* \\ (x,y) \rightarrow g^x y^N (\mathrm{mod} N^2) \end{cases}$$

Với N = p*q với p và q là số nguyên tố lớn, $g \in Z_{N^2}^*$ có bậc là bội khác không của N (có thể chọn g = n + 1). Các bước thực hiện:

Bước 1: *Phát sinh khoá*: public key (N, g) và private key (p, q).

Bước 2: *Mã hoá*: để mã hoá thông điệp m với m < N, chọn ngẫu nhiên $r \in Z_N^*$, sử dụng public key(N, g) tính:c = $E_g(m) = g^m r^N (\mathrm{mod}\ N^2)$, với c là bản mã của m.

Bước 3: *Giải mã*: để giải mã bản mã c sử dụng private key p và q, tính m = $[c]_g$ như sau:

$$[c]_g = \frac{(c^\lambda (\mathrm{mod}\ N^2) - 1)/N}{(g^\lambda (\mathrm{mod}\ N^2) - 1)/N} (\mathrm{mod}\ N)$$

Với λ = lcm (p - 1, q - 1), với lcm là bội số chung nhỏ nhất. Ta có $[c]_g$ không thể tính được nếu không có p và q.

Do bản rõ $m \in Z_N$, bản mã $c \in Z_{N^2}^*$ nên với khoá t (bit) kích thước bản mã là 2*t (bit) khi mã hoá một phần tử. Vì r được chọn ngẫu nhiên nên bản mã c của thông điệp m là ngẫu nhiên, do đó hệ thống mã hoá Paillier là một phép mã hoá theo xác suất.

2.2.3 Giao thức khai thác

Giao thức các tác giả trong [8] đề xuất dựa trên ý tưởng của giao thức do M.Hussen đề xuất. Tuy nhiên, Giao thức đề xuất có sự 2 khác biệt là: đề xuất sử dụng $\bigcup_{i=1}^{n} MFI_i$ để phát sinh tập ứng viên thay cho $\bigcup_{i=1}^{n} FI_i$ và sử dụng mã hoá Paillier để tính

độ hỗ trợ toàn cục thay cho phép hợp nhằm tăng sự bảo toàn tính riêng tư cho giao thức. Để thấy rõ có thể sử dụng $\bigcup_{i=1}^{n} MFI_i$ là tập ứng viên các tác giả giới thiệu 2 bổ đề:

Bổ đề 2.1

Một itemset phổ biến toàn cục thì phải phổ biến cục bộ tại ít nhất một site.

Chứng minh

Giả sử itemset X phổ biến toàn cục và không phổ biến cục bộ tại bất kỳ site nào. Với S_i là độ hỗ trợ của X tại site i, S là độ hỗ trợ toàn cục, ta có:

$$\forall s_i, s_i < s \Rightarrow \sum_{i=1}^{n} s_i * D_i < s*D \Rightarrow \text{X không phổ biến toàn cục}$$

Điều này mâu thuẫn với giả thiết nên suy ra điều phải chứng minh.

Bổ đề 2.2

$\bigcup_{i=1}^{n} MFI_i$ xác định tất cả các itemset phổ biến toàn cục (với MFI_i là MFI tại site S_i).

Chứng minh

Từ bổ đề 2.1, nếu X là một itemset phổ biến toàn cục thì X phải phổ biến cục bộ tại ít nhất một site. Giả sử X phổ biến cục bộ tại các site k thì X phải được xác định trong MFI_k do đó cũng được xác định trong $\bigcup_{i=1}^{n} MFI_i$.

Từ bổ đề 2.2, có thể chọn $\bigcup_{i=1}^{n} MFI_i$ để phát sinh tập itemset ứng viên toàn cục.

Với mã hoá Paillier ta có thể tính độ hỗ trợ toàn cục của các ứng viên và có thể tính tổng các độ hỗ trợ ở dạng mã hoá, tức là:

$E (sup_1 + \ldots + sup_{n-1}) = E (sup_1) * \ldots * E (sup_{n-1})$

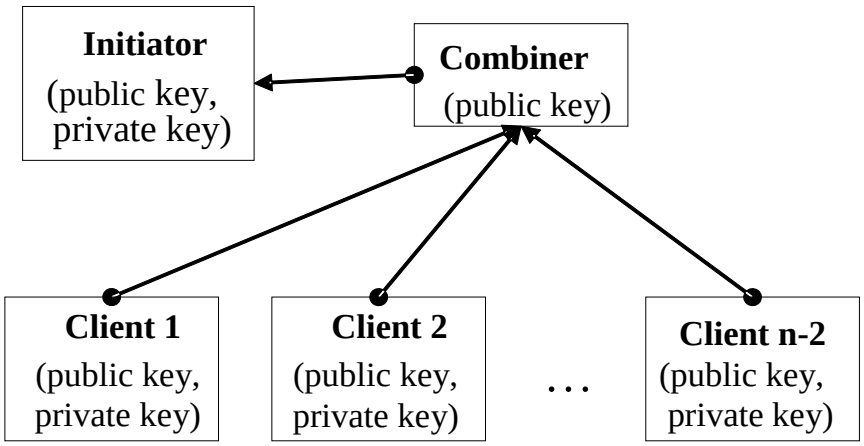
Sau khi giải mã sẽ là độ hỗ trợ toàn cục của ứng viên.

Giao Thức [8]

Giao thức gồm 3 giai đoạn (giai đoạn đầu, giai đoạn hai và giai đoạn kết thúc). Hai giai đoạn đầu mỗi giai đoạn có 3 bước, tại giai đoạn kết thúc Initiator tìm các luật toàn cục mạnh và gửi về cho các bên. Sơ đồ hoạt động của các bên trong giao thức như hình 3.1.

Khởi đầu: Bên Initiator gửi khóa công khai (public key) và khóa bí mật (private key) theo mã hóa Paillier đến tất cả các bên (trừ Combiner chỉ có public

key). Như vậy, trừ Combiner, các bên còn lại đều có 2 bộ khóa public key và private key



Hình 2.7 Giao thức đảm bảo tính riêng tư [8]

Giai đoạn đầu:

Bước 1: Mỗi bên thực hiện tìm tập tối đại (MFI) của mình một cách độc lập (trừ Initiator) và mã hóa tập tối đại bằng mã khóa private key (bên Combiner không mã hóa). Sau đó các bên gửi dữ liệu đã mã hóa của mình cho Combiner.

Bước 2: Combiner nhận dữ liệu của các bên và vì không có khóa private key nên không thể biết MFI của các bên. Sau đó Combiner trộn dữ liệu nhận được từ các bên với dữ liệu MFI của mình và gửi đến Initiator.

Bước 3: Initiator nhận được dữ liệu đã trộn nên không biết dữ liệu của bên nào, Initiator giải mã dữ liệu nhận được từ Combiner và kết hợp với tập phổ biến tối đại của nó để tìm MFI toàn cục, trong đó, mỗi tập phổ biến tối đại không là tập con của tập phổ biến tối đại khác. Sau đó Initiator gửi MFI toàn cục cho tất cả các bên. Mỗi bên tự phát sinh các tập phổ biến của mình theo thứ tự xác định từ MFI toàn cục.

Giai đoạn hai:

Bước 1: Mỗi bên (trừ Initiator) tính độ hỗ trợ các tập phổ biến của mình và mã hóa bằng cách sử dụng mã khóa Paillier. Sau đó các bên gửi dữ liệu đã mã hóa cho Combiner. Mã hóa độ hỗ trợ của tập phổ biến X tại bên S_i được ký hiệu là $E(X.sup_i)$

Bước 2: Với mỗi X Combiner tính toán :

$$E\left(X.\sup_{Combiner}\right) = E\left(X.sup_{Combiner}\right)^{*} \prod_{k=1}^{n-2} E\left(X.sup_{k}\right)$$

Sau đó mã hóa và gửi cho Initiator.

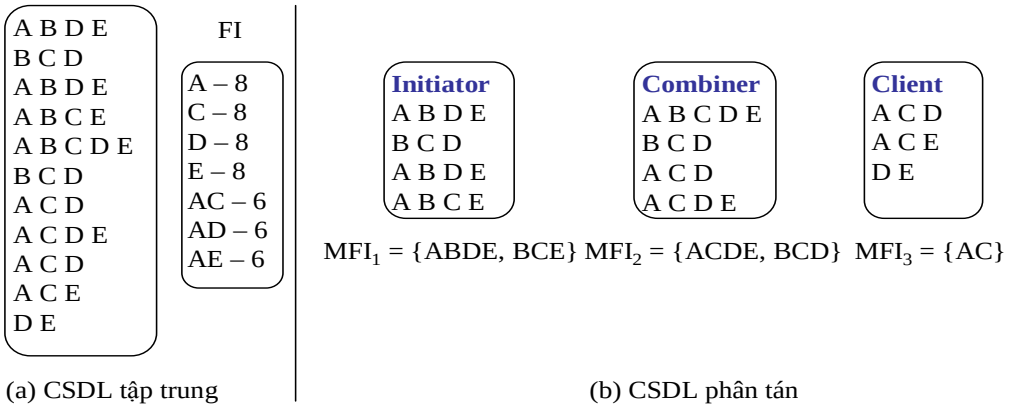
Bước 3: Initiator giải mã dữ liệu nhận được từ Combiner và tính toán độ hỗ trợ toàn cục cho tất cả các bên theo công thức:

$$X.sup = D\left(E(X.sup_{Combiner})\right) + X.sup_{Initiator}$$

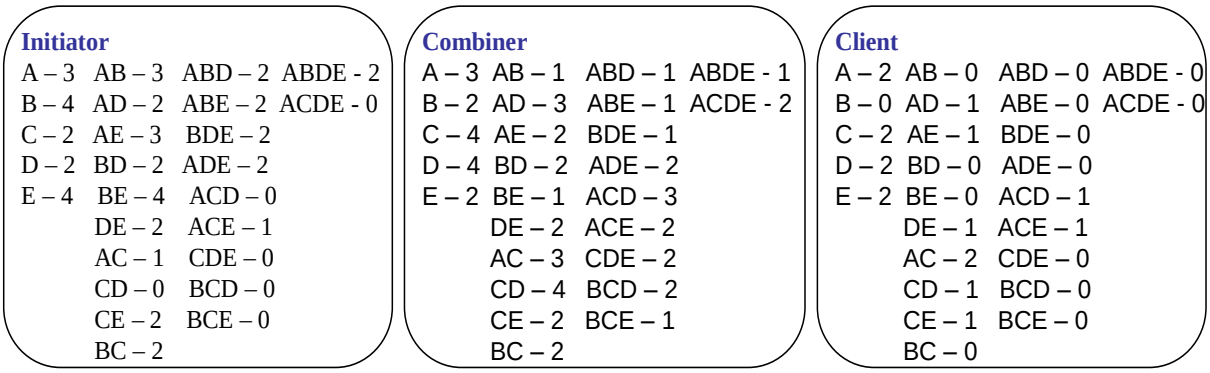
Giai đoạn hoàn tất:

Các bên cùng tính $|DB_i| = \sum_{i=1}^{n} |DB_i|$ theo cách thực hiện như ở giai đoạn 2. Sau đó Initiator tìm các luật toàn cục mạnh và gửi về cho các bên.

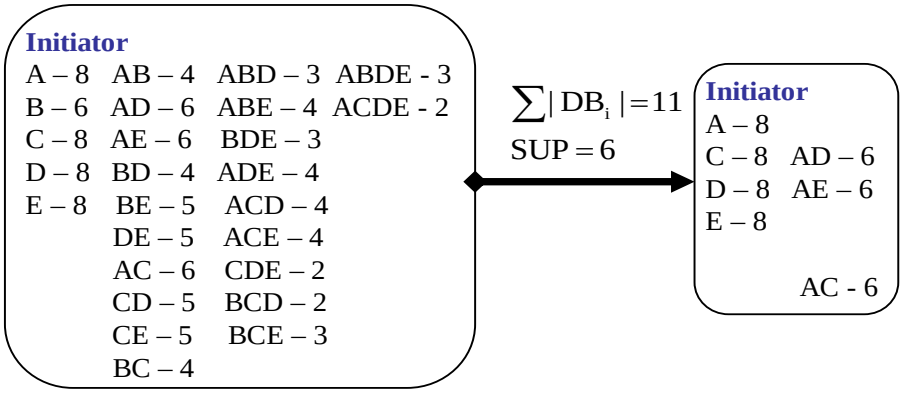
Ví dụ 2.2 (minh họa thuật toán): Giả sử chúng ta có CSDL ban đầu gồm CSDL tập trung (a) và được phân tán ra 3 bên gồm: 1- Initiator, 2- Combiner và 3- Client (b). Với ngưỡng hỗ trợ là 50% ta có MFI_i là tập phổ biến tối đại tương ứng với thứ tự các bên (như hình 2.8). Từ đó Initiator có thể tính được MFI toàn cục = {ABDE, ACDE, BCE, BCD} nhưng không biết MFI_i.



Hình 2.8 CSDL tập trung và CSDL phân tán [8]



Hình 2.9 Các bên tính độ hỗ trợ cục bộ [8]



Hình 2.10 Tính độ hỗ trợ toàn cục và tập phổ biến toàn cục [8]

Qua ví dụ ta nhận thấy kết quả khai thác tập phổ biến trên CSDL tập trung (hình 2.8) và kết quả khai thác trên CSDL phân tán ngang có bảo toàn tính riêng tư (hình 2.10) với cùng ngưỡng hỗ trợ là hoàn toàn giống nhau.

2.2.4 Đánh giá giao thức

Về tính riêng tư [8]

Bước tìm tập ứng viên, Combiner nhận dữ liệu đã được mã hoá từ các bên và không có private key nên không thể giải mã, Combiner trộn các MFI cục bộ nên sau khi giải mã Initiator không thể biết được MFI nào của site nào.

Bước tính độ hỗ trợ toàn cục, Combiner tính tích các độ hỗ trợ ở dạng mã hoá nên Initiator không thể biết chính xác độ hỗ trợ của từng itemset của các site khác. Với mã hoá Paillier có bản mã là ngẫu nhiên nên có tính riêng tư cao hơn so với giao thức MHS [5].

Từ đó ta có thể khẳng định giao thức không tiết lộ dữ liệu cục bộ, các itemset cùng độ hỗ trợ $|DB_i|$ và có tính riêng tư cao hơn so với giao thức MHS.

Về độ chính xác [8]

Từ bổ đề 2.2 tập itemset phổ biến toàn cục là tập con của tập ứng viên, sau đó ta tính độ hỗ trợ toàn cục của các itemset ứng viên và sẽ tìm ra itemset phổ biến toàn cục. Cụ thể trong mỗi bước:

Bước tìm tập ứng viên, Combiner chỉ thực hiện phép trộn và không làm thay đổi dữ liệu nhận được nên sau khi Initiator giải mã sẽ nhận được chính xác MFI cục bộ của các site.

Bước tính độ hỗ trợ toàn cục, do sử dụng mã khóa Paillier nên Combiner có thể tính tổng các độ hỗ trợ ở dạng mã hoá nên Initiator sau khi giải mã sẽ nhận được chính xác tổng độ hỗ trợ của (n – 1) bên.

Kết chương

Trong chương 2 của luận văn đã trình bày một giải thuật và một giao thức bảo toàn tính riêng tư trong khai thác dữ liệu phân tán ngang, Qua từng giải thuật và giao thức các tác giả cũng đã cho ví dụ để minh họa các bước sau đó đánh giá về khả năng bảo toàn tính riêng tư, khả năng thông đồng của một số bên tham gia khai thác để làm lộ dữ liệu của một hay nhiều bên còn lại. Các giải thuật đều được đánh giá là an toàn trong môi trường "Bán thân thiện" để khai thác tập phổ biến và tập phổ biến tối đại. Một thuật toán mới về bảo toàn tính riêng tư trong khai thác luật kết hợp trên cơ sở dữ liệu phân tán ngang sẽ được tác giả luận văn trình bày trong chương 3 của luận văn.

CHƯƠNG 3
THUẬT TOÁN BẢO TOÀN TÍNH RIÊNG TƯ TRONG KHAI THÁC LUẬT KẾT HỢP TRÊN CSDL PHÂN TÁN NGANG

3.1 CƠ SỞ NGHIÊN CỨU

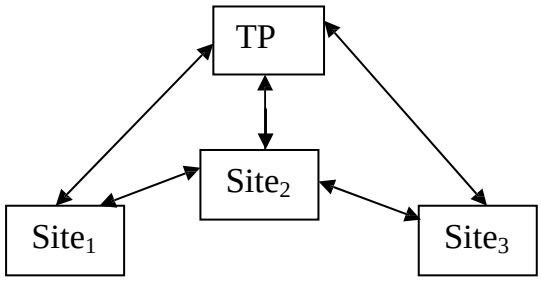
Dựa trên các nghiên cứu của các tác giả mà luận văn đã trình bày ở chương 2 nghiên cứu về khai thác tập phổ biến và phổ biến đóng trên CSDL phân tán ngang bảo toàn tính riêng tư của các bên tham gia khai thác. Luận văn sẽ trình bày một nghiên cứu về thuật toán đề xuất mới của các tác giả trong [6]. Trong mô hình đề xuất này các tác giả sử dụng kỹ thuật bảo toàn tính riêng tư trong khai thác luật kết hợp trên CSDL phân tán ngang với đề xuất là sử dụng một bên được coi là đáng tin cậy (Trusted Party - TP). Phương pháp đề nghị giải quyết vấn đề bảo toàn tính riêng tư bằng cách sử dụng một kỹ thuật mã hóa dựa trên tổng an toàn, một kỹ thuật mã hóa để bảo vệ dữ liệu hoặc thông tin từ những người khác truy cập trong một môi trường cơ sở dữ liệu phân tán và dành cho các mô hình được coi là bán trung thực. Trong nghiên cứu này để bảo vệ các tập phổ biến cục bộ của các bên tham gia từ một bên khác tấn công, thuật toán khóa công khai được cũng sử dụng. Ngoài việc đề xuất dựa vào tổng an toàn, việc bảo toàn tính riêng tư cũng được tăng cường bằng cách truyền các kết quả không rõ ràng giữa các bên trong quá trình khai thác tập phổ biến và khai thác luật. Trong phương pháp này, một bên đặc biệt được chỉ định và được gọi là "đáng tin cậy - TP". Bên TP sẽ khởi tạo quá trình tìm kiếm luật kết hợp mà không biết dữ liệu của bất cứ một bên nào, nhưng bằng cách lấy kết quả đã qua xáo trộn từ tất cả các bên để khai thác tập phổ biến và tập luật, mô hình đề xuất cũng được đánh giá là an toàn.

3.2 MÔ HÌNH KHAI THÁC TRÊN CSDL PHÂN TÁN NGANG

3.2.1 Mô hình đề xuất

Trong mô hình cơ sở dữ liệu phân tán ngang [6], có n bên tham gia khai thác, tất cả các bên tự quản lý cơ sở dữ liệu của mình. Ngoài ra có một bên đặc biệt được coi là đáng tin cậy (Trusted Party - TP) bên này có một số quyền đặc biệt để thực hiện các nhiệm vụ nhất định. Phương pháp đề xuất bao gồm nhiều công việc, thực hiện bởi cả TP cũng như các bên tham gia để tìm luật kết hợp toàn cục trong khi vẫn

đảm bảo tính riêng tư của dữ liệu của các bên tham gia. Sơ đồ dưới đây cho thấy phương pháp liên lạc và trao đổi thông tin giữa TP và các bên trong mô hình đề xuất.



Hình 3.1 Truyền nhận thông tin giữa các bên và TP [6]

Trong mô hình đề xuất này, cơ sở dữ liệu phân tán gồm n bên phân tán theo chiều ngang của dữ liệu và được gọi là $Site_1$, $Site_2$, ..., $Site_n$. Bên thứ i ($Site_i$) duy trì một cơ sở dữ liệu DB_i có chiều dài là $|DB_i|$ trong đó $1 \leq i \leq n$. Tổng số lượng giao dịch ở tất cả các bên là (|DB|) được tính theo công thức:

$$|DB| = |DB_1| + |DB_2| + \ldots + |DB_n|$$

Mỗi bên cần có các tập phổ biến toàn cục và độ hỗ trợ của chúng để tạo ra các luật kết hợp toàn cục. Vì vậy, mục đích là phải xác định các tập phổ biến và độ hỗ trợ của chúng dựa trên cơ sở dữ liệu ở tất cả các bên. Bất kỳ tập mục được cho là thường xuyên trên toàn cục khi và chỉ khi tổng độ hỗ trợ của nó ở tất cả các bên lớn hơn hoặc bằng với số lượng tối thiểu của các giao dịch cần thiết để hỗ trợ mục này trên toàn cục (độ hỗ trợ tối thiểu). Một mục dữ liệu có thể là phổ biến trên toàn cục chỉ khi nó là phổ biến trong ít nhất cơ sở dữ liệu của một hoặc nhiều bên. Tương tự, một mục dữ liệu có thể không phổ biến trên toàn cục chỉ khi mục này là không phổ biến trong ít nhất một bên.

Một điều rõ ràng là, không có ai muốn để lộ tập phổ biến cục bộ, độ hỗ trợ và kích thước cơ sở dữ liệu của mình cho bất kỳ bên nào cũng như bên TP. Để giải quyết vấn đề trên, phương pháp đề xuất cung cấp một quyền đặc biệt để TP có thể có tập phổ biến cục bộ tại các bên mà không cần lấy giá trị của độ hỗ trợ từ tất cả các bên để xác định tập phổ biến toàn cục. Mỗi chủ sở hữu dữ liệu là các bên chấp nhận cung cấp tập phổ biến cục bộ của mình ở dạng mã hóa để bên TP mà họ tin tưởng tạo ra tập phổ biến toàn cục. Để tìm luật kết hợp toàn cục trong cơ sở dữ liệu phân tán theo chiều ngang có kích thước n (> 2), một số nhiệm vụ cần được thực

hiện bởi cả TP cũng như các bên tham gia. Trong mô hình này một số thuật ngữ được sử dụng và được thể hiện trong bảng 3.1 sau:

Bảng 3.1 Một số thuật ngữ sử dụng trong thuật toán [6]

Ký hiệu	Ý nghĩa tiếng Anh
AS_j	Actual Support
GES_j	Global Excess Support for item set X_j
PS_{ij}	Partial Support of item set X_j at $Site_i$
RN_i	Random Number for $Site_i$
$Sign_i$	Sign used with random number for $Site_i$
SignSumRN	Sum of random numbers along with respective signs
$TotalPS_{ij}$	Sum of PS_{ij} of item set X_j, where i indicates site number varies from 0 to n
TP	Trusted Party
Minsup	Minimum Support threshold
Minconf	Minimum Confidence threshold

Mô hình đề xuất được thông qua phương pháp dựa trên tổng an toàn, phương pháp mã hóa để tìm luật kết hợp toàn cục và bảo toàn sự riêng tư của dữ liệu các bên tham gia. Các bước trong mô hình được đề xuất như sau:

Bước 1: Nhiệm vụ đầu tiên bắt đầu thực hiện bởi TP, TP gửi yêu cầu tìm tập phổ biến cho tất cả các bên bằng cách gửi khóa công khai (Public key), hỗ trợ tối thiểu (Minsup).

Bước 2: Khi nhận được khóa công khai và độ hỗ trợ tối thiểu, mỗi bên sẽ tìm tập phổ biến có trong cơ sở dữ liệu của họ bằng cách sử dụng thuật toán apriori để tìm tập phổ biến. Đối với tập phổ biến tìm được, các bên áp dụng thuật toán mã hóa để chuyển đổi các tập phổ biến sang dạng mã hóa bằng cách sử dụng khóa công khai sau đó gửi nó đến TP.

Bước 3: TP sau khi nhận được các tập phổ biến cục bộ đã mã hóa sẽ tiến hành giải mã các dữ liệu bằng cách sử dụng chìa khóa và chuẩn bị một danh sách trộn trong đó bao gồm các tập phổ biến cục bộ của tất cả các bên sau khi loại bỏ các tập dư thừa. Đối với mỗi bên, TP sẽ tạo ra một số ngẫu nhiên (RN_i) và một dấu hiệu (+ hoặc -). Danh sách trộn cùng với số ngẫu nhiên (RN_i) và một dấu hiệu ($Sign_i$)

được gửi đến mỗi bên tương ứng. Số này chỉ ra rằng các số ngẫu nhiên là để được thêm vào hoặc trừ ra khỏi độ hỗ trợ của nó (PS_{ij}).

Bước 4: Mỗi bên sẽ tính độ hỗ trợ một phần cho từng mục phổ biến có trong danh sách trộn được nhận từ TP bằng cách sử dụng công thức:

$$PS_{ij} = X_j.sup - \quad Minsup \quad \times | DB_i | + (Sign_i)\, RN_i$$

Tại bên thứ i, với i nằm trong khoảng từ 1 đến n và j là mục thứ j nằm trong danh sách trộn (có k phần tử) các tập phổ biến (có từ TP), j nằm trong khoảng từ 1 đến k. Mỗi bên sau đó gửi các giá trị PS_{ij} đã tính toán của nó cho tất cả các mục thường xuyên trong danh sách trộn vào tất cả các bên khác.

Bước 5: Mỗi bên sẽ tính toán Tổng PS_{ij} cho từng tập mục X_j bằng cách sử dụng công thức:

$$TotalPS_{ij} = \sum_{i=1}^{n} PS_{ij} \quad \text{với mỗi j = 1 đến k sau đó gửi đến TP.}$$

Bước 6: TP nhận được $TotalPS_{ij}$ là tổng độ hỗ trợ từ tất cả các bên cho mỗi tập phổ biến X_j.

Bước 7: TP xác minh giá trị của $TotalPS_{ij}$ nhận từ các bên cho tập phổ biến X_j, nơi mà i là số các bên ($Site_i$) thay đổi từ 1 đến n. Nếu có sự khác biệt xảy ra thì kết quả nhận được là sai, khi đó TP sẽ yêu cầu tất cả các bên thực hiện lại bước 5 một lần nữa để có được kết quả chính xác.

Bước 8: TP tính độ hỗ trợ toàn cục (GES_j) cho mỗi tập phổ biến X_j bằng cách sử dụng công thức:

$$GES_j = TotalPS_{ij} - SignSumRN$$

Với SignSumRN được tính bằng cách cộng tất cả các số ngẫu nhiên với các dấu hiệu của họ được TP đã tạo ra. Nếu giá trị tính toán của $GES_j \geq 0$ thì các mục phổ biến X_j là phổ biến trên toàn cục nếu không nó là không phổ biến trên toàn cục.

Bước 9: Đối với mỗi mục phổ biến toàn cục được thiết lập X_j, TP tìm độ hỗ trợ thực tế (AS_j) theo công thức:

$$AS_j = GES_j + Minsup * |DB|$$

$$\text{Với } |DB| = \sum_{i=1}^{n} |DB_i|$$

Bước 10: TP gửi danh sách bao gồm tất cả các tập phổ biến toàn cục và giá trị của nó cho tất cả các bên tham gia.

Bước 11: Mỗi bên có thể tạo ra các luật kết hợp với các độ tin cậy khác nhau bằng cách sử dụng các tập phổ biến toàn cục và độ hỗ trợ nhận được từ TP.

Ví dụ 3.1: Minh họa mô hình đề xuất

Mô hình đề xuất được minh họa bằng cách sử dụng ba cơ sở dữ liệu phân tán ngang phần khai thác luật kết hợp bảo toàn tính riêng tư của các bên tham gia. Trong mô hình mẫu này, các cơ sở dữ liệu theo được phân chia thành các mảnh là: DB_1, DB_2 và DB_3 được đặt tại $Site_1$, $Site_2$ và $Site_3$ tương ứng. Ngoài ba bên, sẽ tồn tại một bên đặc biệt được gọi là Trusted party (TP). Cơ sở dữ liệu mẫu tại $Bên_1$, $Bên_2$ và $Bên_3$ được đưa ra dưới đây:

Bảng 3.2 Cơ sở dữ liệu cục bộ tại $Site_1$

T-id/Item	A_1	A_2	A_3	A_4	A_5
T_1	1	0	0	1	0
T_2	1	1	0	1	1
T_3	0	1	1	0	1
T_4	0	0	1	1	1
T_5	1	1	0	1	1

Bảng 3.3 Cơ sở dữ liệu cục bộ tại $Site_2$

T-id/Item	A_1	A_2	A_3	A_4	A_5
T_1	0	1	1	1	1
T_2	0	0	1	1	1
T_3	1	1	1	1	0
T_4	1	1	0	1	1
T_5	1	1	0	0	1

Bảng 3.4 Cơ sở dữ liệu cục bộ tại $Site_3$

T-id/ Item	A1	A2	A3	A4	A5
T_1	1	0	0	1	1
T_2	1	1	1	0	1
T_3	1	0	1	1	1
T_4	1	0	1	1	0
T_5	1	0	1	1	1

Bước 1: TP yêu cầu ba bên gửi các tập phổ biến dưới dạng mã hóa cục bộ của bên mình bằng cách gửi hai giá trị là: ngưỡng hỗ trợ tối thiểu và khóa công khai.

Bước 2: Mỗi bên tính tập phổ biến cục bộ cho cơ sở dữ liệu của mình bằng cách sử dụng ngưỡng hỗ trợ tối thiểu 40% được gửi bởi TP và mã hóa theo khóa công khai cũng do TP gửi. Các tập phổ biến cục bộ (LF – Local Frequent) của các bên $Site_1$, $Site_2$ và $Site_3$, được đưa ra dưới đây:

LF_1= {A_1, A_2, A_3, A_4, A_5, (A_1, A_2), (A_1, A_4), (A_1, A_5), (A_2, A_4), (A_2, A_5), (A_3, A_5), (A_4, A_5), (A_1, A_2, A_4), (A_1, A_2, A_5), (A_1, A_4, A_5), (A_2, A_4, A_5), (A_1, A_2, A_4, A_5)}

LF_2= {A_1, A_2, A_3, A_4, A_5, (A_1, A_2), (A_1, A_4), (A_1, A_5), (A_2, A_3), (A_2, A_4), (A_2, A_5), (A_3, A_4), (A_3, A_5), (A_4, A_5), (A_1, A_2, A_4), (A_1, A_2, A_5), (A_2, A_3, A_4) (A_2, A_4, A_5), (A_3, A_4, A_5)}

LF_3= {A_1, A_3, A_4, A_5, (A_1, A_3),(A_1, A_4), (A_1, A_5), (A_3, A_4), (A_3, A_5), (A_4, A_5), (A_1, A_3, A_4, A_5)}

Bước 3: Sau khi nhận các tập phổ biến dưới dạng mã hóa từ các bên gửi về, TP trộn thành một danh sách các tập phổ biến sau khi loại bỏ các mục dữ liệu thừa (lặp lại 2 lần). Danh sách tập phổ biến từ các bên như sau:

{A_1, A_2, A_3, A_4, A_5, (A_1, A_2), (A_1, A_3), (A_1, A_4), (A_1, A_5), (A_2, A_3), (A_2, A_4), (A_2, A_5), (A_3, A_4), (A_3, A_5), (A_4, A_5), (A_1, A_3, A_4), (A_1, A_3, A_5), (A_1, A_4, A_5), (A_1, A_2, A_4), (A_1, A_2, A_5), (A_2, A_3, A_4), (A_2, A_4, A_5), (A_3, A_4, A_5), (A_1, A_2, A_4, A_5), (A_1, A_3, A_4, A_5)}

Sau đây là những con số ngẫu nhiên và dấu hiệu do TP tạo ra và gửi cùng với tập phổ biến đã trộn lẫn của các bên cho tất cả ba bên.

$Site_1$ received $RN_1 = 20$, $Sign_1 = (‘+’)$.

$Site_2$ received $RN_2 = 39$, $Sign_2 = (‘-‘)$.

$Site_3$ received $RN_3 = 41$, $Sign_3 = (‘-‘)$.

Bước 4: Mỗi bên tính toán độ hỗ trợ phần của mình và gửi đến tất cả các bên khác để tìm ra tổng của độ hỗ trợ. Tất cả ba địa điểm gửi tổng độ hỗ trợ cho tất cả các tập phổ biến trong danh sách trộn nhận được từ TP. TP cuối cùng tính được tập phổ biến toàn cục bằng cách so sánh độ hỗ trợ toàn cục do các bên gửi đến (GES)

của một mục dữ liệu với nơi mà nó được tính GES_i, bằng cách lấy $TotalPS_i$ trừ cho SignSumRN.

Các bước sau đây minh họa cho quá trình tìm kiếm cho biết hai tập trong danh sách trộn là có phổ biến trên toàn cục hay không?

Xét hai tập phổ biến {(A_3, A_5), (A_3, A_4, A_5)} trong danh sách trộn.

Cho X_1= (A_3, A_5) và X_2 = (A_3, A_4, A_5)

Từ các bảng 1, 2 và 3, kích thước của dữ liệu được mô tả dưới đây:

| DB_1 | = 5, | DB_2 | = 5, | DB_3 | = 5 kích thước cơ sở dữ liệu cục bộ tại các bên

là | DB | = $\sum_1^3 |DB_i|$ = 15

TP tính giá trị SignSumRN bằng cách thêm ba số ngẫu nhiên cùng với những dấu hiệu đã gửi cho các bên ở bước 3 bằng cách sau:

SignSumRN = (+) 20 + (-) 39 + (-) 41 = - 60

Độ hỗ trợ của X_1 tại các địa điểm khác nhau được tính như sau:

<u>Tại Bên 1</u>:

PS_{11}= X_1.sup – 40% của DB_1 + ($Sign_1$) RN_1

PS_{11} = 2 – 2 + 20 = 20

<u>Tại Bên 2</u>:

PS_{21} = X_1.sup – 40% của | DB_2 | + ($Sign_2$) RN_2

PS_{21} = 2 - 2 - 39 = - 39

<u>Tại Bên 3</u>:

PS_{31} = X_1.sup – 40% của | DB_3 | + ($Sign_3$) RN_3

PS_{31} = 3 - 2 - 41 = - 40

$Site_1$ gửi PS_{11}=20 của mình đến $Site_2$ và $Site_3$. Tương tự, $Site_2$ gửi PS_{21}=-39 đến $Site_1$ và $Site_3$. $Site_3$ gửi PS_{31} = -40 đến $Site_1$ và $Site_2$. Giá trị $TotalPS_{ij}$ được tính toán ở tất cả các Site:

$TotalPS_{11}$ = PS_{11} + (PS_{21} + PS_{31}) = 20 +(- 39 - 40) = -59

$TotalPS_{21}$ = PS_{21} + (PS_{11} + PS_{31}) = - 39 + (20 - 40) = -59

$TotalPS_{31}$ = PS_{31} + (PS_{11} + PS_{21}) = - 40 + (20 - 39) = -59

TP nhận được giá trị -59 như tổng độ hỗ trợ của tập X_1 được tính toán từ ba địa điểm và đảm bảo các tính toán được thực hiện bởi tất cả các bên này là chính

xác. TP sau đó tính độ hỗ trợ toàn cục (GES_1) bằng cách trừ SignSumRN từ $TotalPS_{11}$.

$$GES_1 = TotalPS_{11} - SignSumRN = -59 - (-60) = 1$$

Giá trị của $GES_1 = 1$ là lớn hơn hoặc bằng 0, do đó (A_3, A_5) được coi là tập phổ biến toàn cục bởi độ hỗ trợ thực tế (AS_1) của X_1 được tính bằng công thức sau:

$$AS_1 = GES_1 + Minsup * |DB| = 1 + 6 = 7 \quad với \quad |DB| = 15$$

Như vậy, tập mục (A_3, A_5) là tập phổ biến toàn cục có độ hỗ trợ là 7. Tiếp theo chúng ta tìm xem tập $X_2 = (A_3, A_4, A_5)$ có là tập phổ biến toàn cục hay không?

Độ hỗ trợ từng phần cho X_2 tại ba địa điểm được tính như sau:

Tại Bên 1:

$PS_{12} = X_2.sup - 40\%$ của $DB_1 + (Sign_1) RN_1 = 1 - 2 + 20 = 19$

Tại Bên 2:

$PS_{22} = X_2.sup - 40\%$ của $DB_2 + (Sign_2) RN_2 = 2 - 2 - 39 = -39$

Tại Bên 3:

$PS_{32} = X_2.sup - 40\%$ của $DB_3 + (Sign_3) RN_3 = 2 - 2 - 41 = -41$

Bước tiếp theo $Site_1$ gửi PS_{12}=19 của mình đến $Site_2$ và $Site_3$. Tương tự, $Site_2$ gửi PS_{21}=-39 đến $Site_1$ và $Site_3$. $Site_3$ gửi PS_{31} = -41 đến $Site_1$ và $Site_2$. Giá trị $TotalPS_{ij}$ được tính toán ở tất cả các site giống nhau và bằng.

$$TotalPS_{12} = PS_{12} + PS_{22} + PS_{32} = 19 + (-39 - 41) = -61$$

Mỗi bên gửi dữ liệu tính toán $TotalPS_{i2}$ đến TP. TP sau đó tính GES_2

$$GES_2 = TotalPS_{12} - SignSumRN = 59 - (-60) = -1$$

Giá trị của GES_2 là -1, thấp hơn số không, như vậy (A_3, A_4, A_5) được coi là không phổ biến trên toàn cục, mặc dù nó phổ biến tại $Site_2$ và $Site_3$.

Các bước trên được lặp lại cho tất cả các tập mục trong danh sách tập phổ biến cục bộ trộn từ các bên để tìm xem chúng có là phổ biến trên toàn cục hay không. Cuối cùng TP chuẩn bị một tập gồm toàn bộ các tập phổ biến toàn cục và độ hỗ trợ của chúng, sau đó TP gửi danh sách này đến tất cả các Site. Kết quả từ ví dụ trên được hiển thị trong bảng 3.5 dưới đây.

Mặc dù danh sách trộn từ tập phổ biến cục bộ của các bên gửi đến TP bao gồm 25 tập nhưng chỉ có 13 tập là phổ biến trên toàn cục.

Bảng 3.5 Tập phổ biến toàn cục và độ hỗ trợ của chúng

Item Set	Sup	Item Set	Sup	Item Set	Sup
A_1	11	(A_1, A_2)	6	(A_4, A_5)	9
A_2	8	(A_1, A_4)	9	(A_3, A_4)	7
A_3	9	(A_3, A_5)	7	(A_1, A_4, A5)	6
A_4	12	(A_1, A_5)	8		
A_5	12	(A_2, A_5)	7		

Mỗi bên có thể tạo ra các luật kết hợp toàn cục cho mỗi tập phổ biến toàn cục dựa trên ngưỡng tin cậy tối thiểu quy định. Các tính toán sau đây minh họa cách mà một luật kết hợp có thể được coi là luật mạnh hay yếu dựa vào ngưỡng tin cậy tối thiểu do người dùng định nghĩa.

Giả sử với ngưỡng Minconf = 65% và đối với tập thường xuyên (A_1, A_4, A_5), các luật kết hợp khác nhau có thể được tạo ra là: {$A_1 \rightarrow (A_4, A_5)$, $A_4 \rightarrow (A_1, A_5)$, $A_5 \rightarrow (A_1, A_4)$, $(A_1, A_4) \rightarrow A_5$, $(A_1, A_5) \rightarrow A_4$, $(A_4, A_5) \rightarrow A_1$}. Tất cả các luật kết hợp này không chắc là luật mạnh. Luật chỉ mạnh khi độ tin cậy của các luật là lớn hơn hay bằng độ tin cậy tối thiểu (Minconf).

Với luật $A_1 \rightarrow (A_4, A_5)$ ngưỡng tin cậy của luật sẽ là: Sup(A_1, A_4, A_5) / Sup(A_1) = 6/11 = 54% không thỏa điều kiện lớn hơn hay bằng Minconf=65% nên luật này không được coi là luật mạnh.

Với luật $(A_1, A_4) \rightarrow A_5$ độ tin cậy của luật là: Sup (A_1, A_4, A_5) / Sup(A_1, A_4) = 6/9 = 66% Thỏa điều kiện lớn hơn hay bằng Minconf nên luật $(A_1, A_4) \rightarrow A_5$ được coi là luật mạnh.

Kiểm tra tương tự với các luật trong số các luật đã liệt kê ta cũng có thêm các luật mạnh gồm: $(A_4, A_5) \rightarrow A_1$ với độ tin cậy của luật =66%, $(A_1, A_5) \rightarrow A_4$ với độ tin cậy là 75%.

3.2.2 Về việc bảo toàn tính riêng tư trong mô hình đề xuất

Một mô hình mới đã được các tác giả đề xuất [6], để đảm bảo tính riêng tư trong khai thác luật kết hợp trên cơ sở dữ liệu phân tán ngang. Mô hình đề xuất có thể được áp dụng cho số lượng các bên (n) là rất lớn và cho bất kỳ số lượng các giao dịch trong cơ sở dữ liệu của các bên. Nhiều công việc như phát hiện của tập phổ biến cục bộ tại các bên, tính độ hỗ trợ một phần và tổng độ hỗ trợ cho từng mục

trong danh sách trộn được thực hiện độc lập tại các bên. Do đó thời gian tính toán của mô hình đề xuất là ít. Hiệu quả của phương pháp đề xuất về tính riêng tư và liên lạc thông tin giữa các bên được thảo luận như sau:

• Vấn đề bảo mật được đảm bảo bằng cách sử dụng mã hóa và giải mã kỹ thuật tại thời điểm chuyển giao tập phổ biến từ các bên khác nhau về TP. Từ điều này, bên TP chỉ có thể biết tập phổ biến cục bộ của mỗi bên nhưng TP không thể biết độ hỗ trợ của bất kỳ tập phổ biến nào và không thể dự đoán bất cứ gì liên quan đến cơ sở dữ liệu của các bên.

• Tại thời điểm tính độ hỗ trợ một phần của một tập mục (X_i) tại mỗi bên, theo công thức $PS_{ij}= X_j.Sup – 40\%\ of\ DB_i + (Sign_i)\ RN_i$ với RN_i là một số ngẫu nhiên. Vì vậy, độ hỗ trợ một phần là hình thức ẩn hình sau đó được gửi đến các bên một cách an toàn. Mỗi bên không được có bất kỳ ý tưởng về những ký hiệu, số ngẫu nhiên được tạo ra từ TP và gửi đến các bên khác nhau và kích thước cơ sở dữ liệu của các bên cũng không được biết. Vì vậy, từ độ hỗ trợ một phần, không có bên nào có thể dự đoán các thông tin dữ liệu của bên khác. Bằng cách này, độ hỗ trợ một phần của một tập có thể được gửi đến tất cả các bên mà vẫn đảm bảo sự riêng tư của dữ liệu các bên. Do khái niệm tổng an toàn được sử dụng trong việc tính toán độ hỗ trợ một phần cũng tăng cường sự riêng tư của dữ liệu của các bên tham gia.

• Bên TP nhận được tổng độ hỗ trợ của mỗi tập phổ biến từ tất cả các bên để tìm những tập phổ biến toàn cục. Bởi vì là tổng độ hỗ trợ nên bên TP không thể tìm thấy thông tin dữ liệu, kích thước cơ sở dữ liệu của bất kỳ bên nào và độ hỗ trợ cục bộ của bất kỳ mục nào. Mặc dù bên TP là giao con số ngẫu nhiên, dấu hiệu cho tất cả các bên và tổng kích thước cơ sở dữ liệu được biết, nhưng TP cũng không thể dự đoán dữ liệu cá nhân của bất kỳ bên nào.

• Cuối cùng kết quả là các tập phổ biến toàn cục và độ hỗ trợ của chúng được gửi bởi bên TP cho tất cả các bên. Với những kết quả này, không có bên nào có thể dự đoán được độ hỗ trợ cục bộ của bất kỳ tập mục phổ biến toàn cục, vì tập phổ biến trên toàn cục có thể không phổ biến trong tất cả các

bên và bất kỳ chủ sở hữu bên nào cũng không thể dự đoán sự đóng góp của cơ sở dữ liệu các bên khác mà tạo thành tập phổ biến trên toàn cục.

Trong môi trường phân tán, chi phí truyền thông được đo bằng số lượng các thông tin liên lạc để truyền dữ liệu giữa tất cả các bên liên quan trong quá trình tìm kiếm luật kết hợp toàn cục.

• Hiệu quả của một thuật toán được đánh giá về chi phí truyền thông phát sinh trong quá trình trao đổi thông tin. Mô hình đề xuất đã giảm thiểu số lượng các quá trình chuyển dữ liệu bằng cách cho phép việc chuyển giao số lượng lớn các dữ liệu tại một thời điểm từ một bên đến một bên khác và bên TP đến các bên. Ví dụ mỗi bên sẽ gửi tập phổ biến cục bộ của mình một lần duy nhất đến bên TP và thậm chí cả các bên sẽ gửi độ hỗ trợ một phần của từng hạng mục đến các bên khác thay vì gửi một tập gồm nhiều độ hỗ trợ một phần của tất cả mục trong danh sách trộn đến các bên khác. Do đó các mô hình đề xuất có nhu cầu về thông tin liên lạc ít.

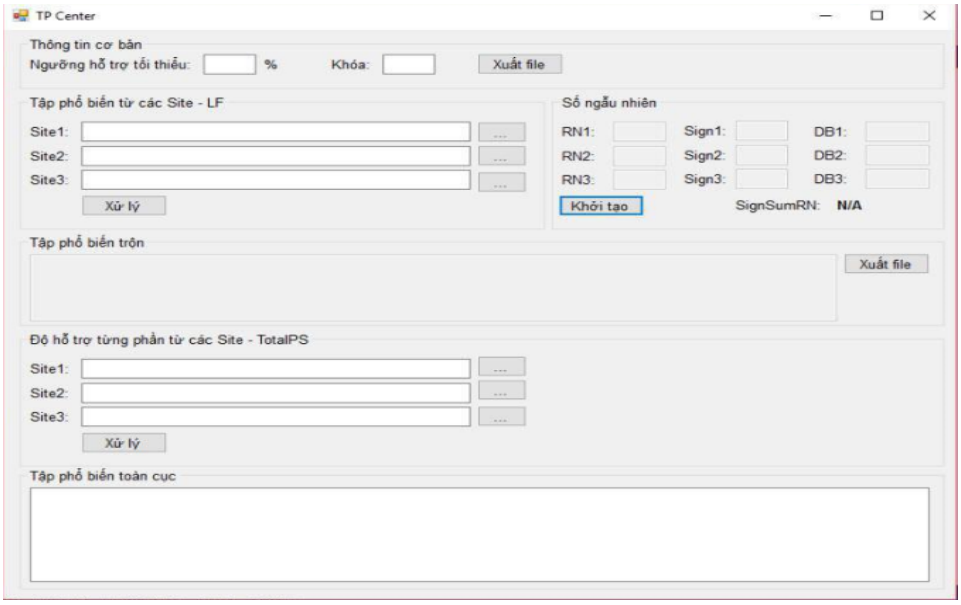
• Bên TP cũng gửi tất cả các tập phổ biến toàn cục cho tất cả các bên trong một lần. Do đó mô hình được đề xuất là kinh tế nhiều hơn về chi phí truyền thông vì nó sử dụng số lượng lớn dữ liệu truyền.

Các thảo luận ở trên khẳng định rõ rằng mô hình mà các tác giả đề xuất là hiệu quả cho việc tìm kiếm các luật kết hợp toàn cục trên cơ sở dữ liệu phân tán ngang bảo toàn tính riêng tư của dữ liệu cho các bên tham gia.

3.3 THỰC NGHIỆM MÔ HÌNH

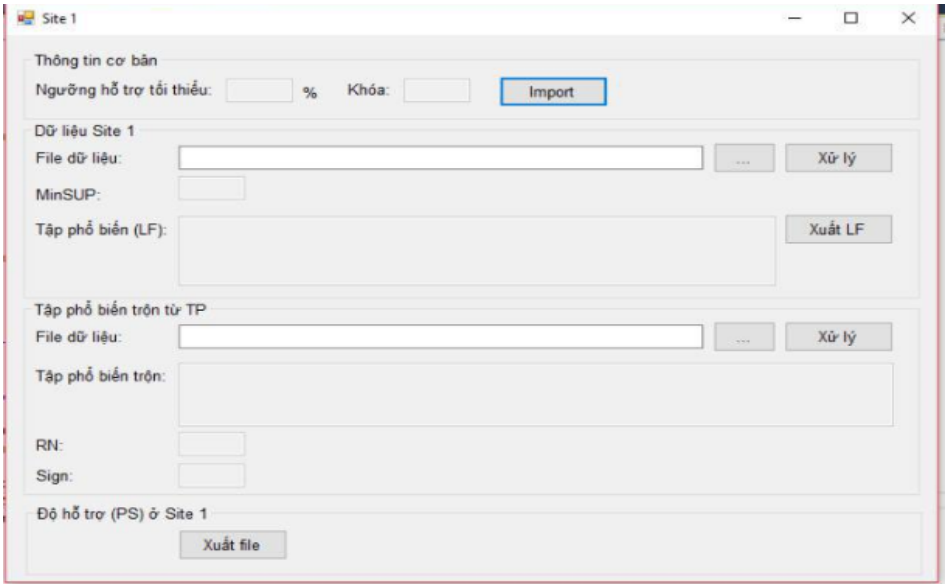
Để kiểm tra tính hiệu quả của mô hình khai thác luật kết hợp trên CSDL phân tán ngang đã trình bày trong mục 3.2. Chương trình áp dụng mô hình khai thác trên CSDL phân tán ngang bảo toàn tính riêng tư của các bên tham gia khai thác được viết bằng ngôn ngữ C# với giả sử gồm 4 bên tham gia khai thác gồm: Bên TP, Bên 1, Bên 2, Bên 3. Sẽ khai thác luật kết hợp toàn cục từ dữ liệu cục bộ của 3 Bên và một Bên TP là bên điều khiển quá trình khai thác. Giả sử chương trình được thực hiện trên 1 máy và thể hiện các chức năng của từng bên qua các màn hình khác nhau. Cấu hình máy tính thực hiện thực nghiệm là: Lenovo X1, bộ xử lý Core i5 và 2 GB bộ nhớ chính, chạy trên hệ điều hành Windows 10 - 64bit, DotNet FrameWork 4.5, Microsoft Visual Studio 2015.

Chương trình khởi động từ màn hình của bên điều khiển TP có cấu trúc như hình 3.2. Người sử dụng nhập ngưỡng hỗ trợ tối thiểu và mã khóa công khai rồi nhấn nút xuất file để tạo file gửi cho các bên tiến hành khai thác tập phổ biến cục bộ tại mỗi bên. Sau khi nhận được tập phổ biến cục bộ tại 3 bên như ví dụ. Bên TP sẽ tiến hành trộn các tập phổ biến này lại và xuất ra file và gửi cho các bên cùng với số phát sinh ngẫu nhiên và ký hiệu (+ hoặc -) để các bên tính độ hỗ trợ của các phần tử trong tập phổ biến toàn cục đã trộn và loại bỏ đi các tập trùng nhau.



Hình 3.2 Màn hình bên TP

Sau khi nhận lại tập phổ biến và giá trị đã tính toán của các bên. Bên TP sẽ tính bước cuối cùng để cho ra tập phổ biến toàn cục và độ hỗ trợ của chúng.



Hình 3.3 Màn hình của các Bên

Màn hình của các bên hoạt động giống nhau và có giao diện như hình 3.3. Các bên nhận ngưỡng hỗ trợ tối thiểu và mã khóa từ file do TP gửi đến bằng cách Import dữ liệu từ file. Sau đó nhận file dữ liệu của bên mình và tính ra tập phổ biến thỏa ngưỡng hỗ trợ tối thiểu do bên TP gửi đến. Để gửi tập phổ biến LF về TP sau khi đã mã hóa các bên xuất ra file để bên TP nhận. Bước tiếp theo, các bên nhận từ TP file chứa tập phổ biến đã trộn từ tập phổ biến cục bộ của các bên cùng với số ngẫu nhiên, ký hiệu. Bước cuối cùng là thực hiện tính toán cho tập phổ biến toàn cục theo số ngẫu nhiên và ký hiệu do TP gửi xuống ứng với dữ liệu của bên mình. Kết quả sẽ nhận được file từ TP gửi xuống là tập phổ biến toàn cục và độ hỗ trợ của chúng.

Chương trình mới chỉ thực hiện được trên một số CSDL nhỏ ở dạng ví dụ để kiểm tra tính đúng đắn của mô hình để khai thác. Phần mở rộng để chương trình có thể thực hiện trên CSDL thực sẽ được tác giả tiếp tục phát triển trong tương lai.

PHẦN KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Luận văn đã trình bày được phần lý thuyết về một số thuật toán và phương pháp khai thác luật kết hợp (Apriori, IT-Tree), lý thuyết về bảo toàn tính riêng tư trong khai thác dữ liệu, các thuật toán khai thác tập phổ biến và luật kết hợp trên CSDL phân tán ngang, bảo toàn tính riêng tư. Từ đó luận văn đề xuất thay đổi bước 2 khi các bên khai thác tại CSDL cục bộ của mình để giảm thời gian khai thác, luận văn cũng đề xuất thay thuật toán từ Apriori thành phương pháp IT-Tree.

Kết quả thực nghiệm trên mô hình khai thác trong CSDL phân tán giữa các bên tham gia cũng cho thấy tính đúng đắn của mô hình khai thác trên CSDL ngang bảo toàn tính riêng tư của các bên tham gia khai thác. Khi áp dụng trong mô hình thực tế với nhiều bên tham gia và khối lượng dữ liệu lớn thì việc thay đổi thuật toán trong các bước thực hiện khai thác tập phổ biến ở các bên sẽ giảm thời gian khai thác.

2. Hướng phát triển

Những đóng góp chính của luận văn hiện nay mới chỉ dừng lại ở việc nghiên cứu các thuật toán về khai thác luật kết hợp và góp ý đề xuất. Phần thực nghiệm của luận văn cũng chưa chạy được trên nhiều máy, online và trên CSDL thực tế, để có thể so sánh, đánh giá với một số mô hình khai thác các tập phổ biến có bảo toàn tính riêng tư. Phần này tác giả sẽ tiếp tục nghiên cứu thêm và hoàn thiện.

Ngoài ra, phần thực nghiệm của luận văn cần được mở rộng để so sánh và đánh giá với một số thuật toán khác có cùng mục đích khai thác các tập phổ biến trên CSDL phân tán ngang có bảo toàn tính riêng tư của các bên tham gia khai thác để kết quả khách quan hơn và phần thực nghiệm trên cần thủ nghiệm trên một số CSDL thực tế khác nhau.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Trần Quốc Việt, Cao Tùng Anh, Lê Hoài Bắc (2012), "*Đảm bảo tính riêng tư và chống thông đồng trong khai thác luật kết hợp trên dữ liệu phân tán ngang*", Chuyên san các công trình nghiên cứu, phát triển và ứng dụng công nghệ thông tin và truyền thông, Tạp chí công nghệ thông tin và truyền thông, số 7, Hà Nội 05/2012, tr 60-70.

[2] Võ Đình Bảy, Lê Hoài Bắc (2010), "*Chuỗi Bit Động: Cách tiếp cận mới để khai thác tập phổ biến*", ICTFIT' 2010, Hồ Chí Minh, Nhà xuất bản Khoa học & Kỹ thuật, tr 47-52.

Tiếng Anh

[3] Cheung David Wai-Lok, Han Jiawei, Ng. Vincent, Fu Ada Wai-Chee, and Fu Yongjian (1996), "*A fast distributed algorithm for mining association rules"*,IEEE In Proceedings of the 1996 International Conference on Parallel and Distributed Information Systems (PDIS'96), Miami Beach, Florida, USA, pp. 31-42.

[4] Estivill-Castro Vladimir, HajYasien Ahmed, (2007), "*Fast Private Association Rule Mining by a Protocol Securely Sharing Distributed Data"*, In Proceedings of the 2007 IEEE Intelligence and Security Informatics, New Brunswick, New Jersey, USA, May 23-24, pp. 324–330.

[5] Hussein Mahmoud, El-Sisi Ashraf, Ismail Nabil (2008), "*Fast Cryptographic Privacy Preserving Association Rules Mining on Distributed Homogenous Data Base",* Lecture Notes in Computer Science, Vol. 5178/2008, pp. 513-519.

[6] Lakshmi N. V. Muthu, Rani Dr. K Sandhya (2012), "*Privacy Preserving Association Rule Mining in Horizontally Partitioned Databases Using Cryptography Techniques*", International Journal of Computer Science and Information Technologies, Vol. 3 (1), 2012, pp. 3176-3182.

[7] Lindell Yehuda and Pinkas Benny (2008)*, "Secure Multiparty Computation for Privacy-Preserving Data Mining",* IACR, The Journal of Privacy and Confidentiality, Number 1, pp. 59-98.

[8] Nguyen Xuan Canh, Le Hoai Bac,Cao Tung Anh, (2012) "*An Enhanced Sheme for Priserving Associstion Rules Mining on Horizontally Distributed Databases*", IEEE RIVF International Conference on Computing & Communication Technologies, research, Innovation and Vision for the Future 27 Feb-01 Mar 2012, pp. 29-32.

[9] Verykios Vassilios, Bertino Elisa, Fovino Igor Nai, Parasiliti Loredana, Saygin Yücel, and Theodoridis Yannis, (2004)*, "State-of-the-art in privacy preserving data mining",* SIGMOD Record, 33(1), pp. 50-57.

[10] Yang Bin, Nakagawa Hiroshi, Sato Issei and Sakuma Jun (2010). "*Collusion-resistant privacy-preserving data mining"*, Proceedings of the 16th *ACM SIGKDD* international conference on Knowledge discovery and data mining, NY, USA pp. 483-492.

[11] Zaki Mohammed Javeed, Gouda Karam (2003), "*Fast Vertical Mining Using Diffsets*", Proceeding of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 326-335.

[12] Zaki Mohammed Javeed, Hsiao Ching-Jui, (2005) "*Efficient Algorithms for Mining Closed Itemsets and Their Lattice Structure*", IEEE Transactions on Knowledge and Data Engineering. 17(4): pp. 462-478.