BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

NGUYỄN QUANG NHÂN

MỘT THUẬT TOÁN CẢI TIẾN TRONG KHAI THÁC LUẬT KẾT HỢP BẢO TOÀN TÍNH RIÊNG TƯ

LUẬN VĂN THẠC SỸ

Chuyên ngành : Công nghệ thông tin

Mã số ngành: 60480201

TP. HỒ CHÍ MINH, tháng 02 năm 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

NGUYỄN QUANG NHÂN

MỘT THUẬT TOÁN CẢI TIẾN TRONG KHAI THÁC LUẬT KẾT HỢP BẢO TOÀN TÍNH RIÊNG TƯ

LUẬN VĂN THẠC SỸ

Chuyên ngành : Công nghệ thông tin

Mã số 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**

(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 … tháng … năm …

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	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 sau khi Luận văn đã được 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….. tháng….. năm 20…..

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ tên học viên: Nguyễn Quang Nhân .....Giới tính: Nam

Ngày, tháng, năm sinh: 15/04/1990.....Nơi sinh: An Giang

Chuyên ngành: Công nghệ thông tin.....MSHV: 1441860019

I- Tên đề tài:

MỘT THUẬT TOÁN CẢI TIẾN TRONG KHAI THÁC LUẬT KẾT HỢP BẢO TOÀN TÍNH RIÊNG TƯ

II- Nhiệm vụ và nội dung:

- Nghiên cứu tổng quan khai thác dữ liệu.
- Nghiên cứu khai thác dữ liệu bảo toàn tính riêng tư, các phương pháp.
- Nghiên cứu về Luật kết hợp, khai thác luật kết hợp.
- Nghiên cứu thuật toán Apriori.
- Nghiên cứu về khai thác luật kết hợp bảo toàn tính riêng tư.
- Nghiên cứu, giới thiệu Thuật toán khai thác luật kết hợp bảo toàn tính riêng tư, nâng cao tính thực thi.
- Xây dựng chương trình demo minh hoạ cho thuật toán Thuật toán khai thác luật kết hợp bảo toàn tính riêng tư, nâng cao tính thực thi.

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 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

Nguyễn Quang Nhân

LỜI CÁM ƠN

Tôi xin cảm ơn các thầy, cô giáo ở khoa Công nghệ thông tin trường Đại học Công Nghệ Thành Phố Hồ Chí Minh đã giảng dạy tôi trong suốt thời gian học tập tại trường và tạo điều kiện giúp đỡ tôi hoàn thành luận văn này.

Đặc biệt tôi xin bày tỏ lòng cảm ơn chân thành và biết ơn sâu sắc tới **TS. Cao Tùng Anh**. Người Thầy đã tận tận tình hướng dẫn tôi trong suốt thời gian nghiên cứu và làm luận văn tốt nghiệp này.

Tôi cũng xin gởi lời cám ơn đến các bạn học viên trong lớp cao học khoá 2014-2015 đã tạo điều kiện, chia sẻ những kiến thức để em có thể hoàn thành khoá học cũng như luận văn này.

Cảm ơn các bạn bè, đồng nghiệp đã cổ vũ động viên tôi trong suốt quá trình học tập tại trường.

Tuy đã có những cố gắng nhất định nhưng do thời gian và trình độ có hạn nên chắc chắn luận văn này còn nhiều thiếu sót và hạn chế nhất định. Kính mong nhận được sự góp ý của thầy cô và các bạn.

TÓM TẮT

Với nguồn thông tin khổng lồ hiện nay, việc thu thập và rút trích các thông tin có ích từ vô vàng dữ liệu là một công việc vô cùng quan trọng. Khai thác dữ liệu là một công cụ phân tích dữ liệu vô cùng mạnh mẽ, bởi trong đó quy định mô hình và các kiến thức được trích xuất từ các tập dữ liệu lớn. Trong quá trình khai thác dữ liệu, một vấn đề được đặt ra là phải đảm báo tính riêng tư của người dùng và thông tin của họ.

Trong khai thác dữ liệu, khai thác luật kết hợp là một trong các phương pháp quan trọng. Vì vậy khai thác luật kết hợp bảo toàn tình riêng tư là một việc hết sức cần thiết. Hiện nay có rất nhiều thuật toán khai thác luật kết hợp bảo toàn tính riêng tư, như thuật toán MASK[8]. Cả thuật toán MASK và một số thuật toán tối ưu hoá khác đều chỉ sử dụng phương pháp gây nhiễu dữ liệu. Tuy nhiên dữ liệu bị nhiễu loạn vẫn tồn tại sự liên quan đến dữ liệu thô ban đầu. Che dấu dữ liệu, phân vùng dữ liệu, giấu các luật nhạy cảm và lấy mẫu dữ liệu được áp dụng trong phương pháp hạn chế truy vấn để tránh lộ các dữ liệu thô ban đầu cần được bảo vệ.

Với các vấn đề nêu trên, học viên đã chọn đề tài "**MỘT THUẬT TOÁN CẢI TIẾN TRONG KHAI THÁC LUẬT KẾT HỢP BẢO TOÀN TÍNH RIÊNG TU**". Luận văn sẽ tập trung vào nội dung xử lý cải thiện hiệu quả thực thi trong khai thác luật kết hợp bảo toàn tính riêng tư, và chương trinh demo cho việc cải thiện hiệu quả thực thi. Nội dung chính gồm :

Mở đầu

Chương 1: Tổng quan lý thuyết

Chương 2: Khai thác luật kết hợp bảo toàn tính riêng tư

Chương 3: Thuật toán bảo toàn tính riêng tư trong khai thác luật kết hợp.

Kết Luận

ABSTRACT

With the innumerable information sources available today, the process of collecting and capturing useful data is a very important job. Data mining is an extremely powerful analysis tool as it requires proper models and methods to extract knowledges from large data sets. During the data mining process, one of the biggest security concerns is to ensure the privacy of users and their information.

In data mining, mining association rule is one of the important methods. The use of association rule to preserve the privacy is a very necessary job. Currently there are many mining association rule algorithm to preserve privacy, such as MASK algorithm [8] . Both MASK algorithm and other optimization algorithms are only using data jamming methods. However distorted data still remains connected to the raw data. Data masking, data partitioning,hiding sensitive rules, and data sampling methods are applied by using the limiting queries method to avoid revealing raw data that needs to be protected.

With all of the issues mentioned prior, I chose the topic "**AN IMPROVED ALGORITHM IN MINING ASSOCIATION RULE FOR PRIVACY PRESERVATION"**. Dissertation will focus on improving effective enforcement of mining association rule to preserve privacy, and demo program for improving implementation efficiency. The main contents include:

Introduction

Chapter 1 : Overview Theory

Chapter 2 : Mining Association Rule for Privacy Preservation

Chapter 3 : Algorithm for Mining Association Rule for Privacy Preservation

Conclusion

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa tiếng Anh	Ý nghĩa tiếng Việt
Conf	Confindence	Độ tin cậy
CSDL	Cơ Sở Dữ Liệu	
EMASK	Efficient MASK	Thuật toán EMASK
FCI	Frequent Closed Itemset	Tập phổ biến đóng
MASK	Mining Associations with Secrecy Konstraints	Thuật toán MASK, khai thác luật kết hợp bảo toàn tính riêng tư
Maxconf	Maximally confidence	Độ tin cây tối đại
Maxsup	Maximally suppport	Độ hỗ trợ tối đại
MFI	Maximally Frequent Itemset	Tập phổ biến tối đại
Minconf	Minimum Confidence	Độ tin cậy tối thiểu
Minsup	Minimum support	Độ hỗ trợ tối thiểu
MMASK	Modified MASK	Một thuật toán cải tiến từ thuật toán MASK
PPDM	Privacy Preserving Data Mining	Khai thác dữ liệu bảo toàn tính riêng tư
SM	Safety Margin	Tham số khoảng an toàn
SMC	Secure Multiparty Computation	Tính toán bảo mật đa thành phần.
SQL	Structured Query Language	Ngôn ngữ truy vấn mang tính cấu trúc
Supp	Support	Độ hỗ trợ

DANH MỤC CÁC BẢNG

DANH MỤC HÌNH ẢNH

MỞ ĐẦU

Đến thời điểm hiện tại, số lượng dữ liệu trong thế giới của chúng ta đang ngày càng bùng nổ. Thu thập thông tin có giá trị từ vô vàng dữ liệu khổng lồ là một công việc vô cùng quan trọng. Khai thác dữ liệu là một công cụ phân tích dữ liệu vô cùng mạnh mẽ, bởi trong đó quy định mô hình và các kiến thức được trích xuất từ các tập dữ liệu lớn. Tuy nhiên, có một vấn đề phát sinh đó là sự riêng tư của người dùng và thông tin của họ không thể được bảo vệ một cách hiệu quả trong quá trình khai thác dữ liệu [8]. Ví dụ, thông qua việc khai thác dữ liệu bệnh lý của bệnh nhân, các quy tắc liên quan đến các bệnh khác nhau, có thể được kết nối từ nhiều hồ sơ của nhiều bệnh nhân (một bệnh nhân mắc bệnh tiểu đường, tại cùng một thời điểm đó có thể mắc các bệnh liên quan đến tim mạch như huyết áp cao, tai biến mạch máu não,…). Trong quá trình khai thác dữ liệu bệnh lý bệnh nhân đó chắc chắn sẽ gây ra sự tiếp xúc của các dữ liệu, trường hợp này dẫn đến sự rò rỉ thông tin riêng tư của các bệnh nhân.

Vì vậy, vấn đề đặt ra là làm thể nào để đảm bảo sự riêng tư và an ninh thông tin trong quá trình khai thác, phân tích dữ liệu khổng lồ.

Trong lĩnh vực khai thác dữ liệu, khai thác luật kết hợp đóng vai trò hết sức quan trọng. Lần đầu tiên, việc khai thác luật kết hợp trong khai thác dữ liệu được đề xuất bởi Agrawal và các cộng sự vào năm 1993 [6]. Kể từ khi mối tương quan hoặc sự liên quan giữa các kết quả trong quá trình khai thác thông tin có thể tìm thấy bởi luật kết hợp, thì nó đã được áp dụng rộng rãi trong các quyết định của Chính phủ, doanh nghiệp và các cá nhân [11].

Khi nhắc đến các thuật toán khai thác luật kết hợp trong khai thác dữ liệu bảo toàn tính riêng tư, việc áp dụng đơn lẻ các phương pháp dẫn đến hiệu quả thực thi không cao. Để khắc phục vấn đề trên, trong báo cáo luận văn này sẽ đề cập đến một thuật toán mới giúp cải thiện hiệu quả thực thi và có kết quả tốt hơn trong việc bảo vệ tính riêng tư của thông tin.

Luận văn bao gồm tổng cộng 5 phần: Bao gồm phân Mở Đầu, Chương 1, Chương 2, Chương 3**,** và phần Kết luận–đánh giá.

MỞ ĐẦU: Sẽ giới thiệu tổng quan lĩnh vực nghiên cứu, tinh hình nghiên cứu.

Chương 1: Tổng quan lý thuyết.

Trong Chương 1 sẽ sẽ được chia thành bốn phần

- Phần một là trình bày các khái niệm về Khai thác dữ liệu, Tính riêng tư, và khai thác dữ liệu có bào toàn tính riêng tư.
- Phần hai giới thiệu về các phương pháp Khai thác dữ liệu bảo toàn dữ liệu bảo toàn tính riêng tư. Bao gồm 3 phương pháp.
- Phần ba giới thiệu các phương che dấu dữ liệu nhạy cảm.
- Phần bốn sẽ là các khái niệm vế luật kết hợp, khai thác luật kết hợp để làm gì, và thuật toán Apriori để tìm luật kết hợp.

Chương 2: Khai thác luật kết hợp bảo toàn tính riêng tư

Chương 2 bao gồm bốn phần chính.

- Phần một là Bài toán cần Khai thác luật kết hợp bảo toàn tính riêng tư
- Phần hai trình bày các kỹ thuật dùng để áp dụng khai thác luật kết hợp bảo toàn tính riêng tư.
- Phần ba là thuật toán MASK(Mining Associations with Secrecy Konstraints), một thuật toán khai thác luật kết hợp bảo toàn tính riêng tư. Trong phần này bao gồm giới thiệu tình hình nghiên cứu liên quan đến thuật toán; bài toán đặt ra và thuật toán xử lý. Một số biến thể liên quan.
- Phần bốn giới thiệu về lý thuyết giàn giao, việc áp dụng giàn giao trong việc ẩn tập mục nhạy cảm. Mặt hạn chế trong việc áp dụng này, từ đó đưa ra hướng giải quyết trong Chương 3.

Chương 3: Thuật toán bảo toàn tính riêng tư trong khai thác luật kết hợp.

Trong Chương 3 sẽ gồm 3 phần

- Phần một sẽ giới tổng quan về thuật toán.
    - Phần hai giới thiệu bài toán, cách xử lý và thuật toán, ví dụ minh họa.
    - Phần ba là chương trình minh họa cho thuật toán.

Kết luận: Trong phần này sẽ nêu lên các kết quả đạt được sau khi thực hiện luận văn, các vấn đề còn tồn đọng các mặt hạn chế của luận văn, cuối cùng là kiến nghị hướng phát triển tiếp theo của luận văn.

CHƯƠNG 1 LÝ THUYẾT TỔNG QUAN

1.1 Các khái niệm

1.1.1 Khai thác dữ liệu

Khai thác dữ liệu (data mining) là quá trình khám phá các tri thức mới và các tri thức có ích ở dạng tiềm năng trong nguồn dữ liệu đã có. Cụ thể hơn 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.

Khai thác dữ liệu là một bước trong các bước của quá trình Khai thác tri thức- KDD (Knowledge Discovery in Database) và KDD được xem là các quá trình khác nhau theo thứ tự sau đây:

- Trích chọn dữ liệu (data selection): Là bước trích chọn những tập dữ liệu cần được khai phá từ các tập dữ liệu lớn (databases, data warehouses, data repositories) ban đầu theo một số tiêu chí nhất định.
- Tiền xử lý dữ liệu (data preprocessing): Đây là giai đoạn làm sạch dữ liệu và cấu hình lại, ở đây một số kỹ thuật được áp dụng để đối phó với tính không đầy đủ, nhiễu, và không phù hợp của dữ liệu. Bước này cũng cố gắng để giảm dữ liệu bằng cách sử dụng chức năng tổng hợp và nhóm, các phương pháp nén dữ liệu, histograms, lấy mẫu, ... Ngoài ra, các kỹ thuật rời rạc hoá dữ liệu (Bining, rời rạc hóa dựa vào histograms, dựa vào entropy, dựa vào phân khoảng, …) có thể được sử dụng để làm giảm số lượng các giá trị cho một thuộc tính liên tục. Sau bước này, dữ liệu được làm sạch , hoàn chỉnh, thống nhất.
- Biến đổi dữ liệu (data transformation): Trong bước này, dữ liệu được chuyển dạng hoặc hợp nhất thành dạng thích hợp cho khai thác dữ liệu. Biến đổi dữ liệu có thể liên quan đến việc làm mịn và chuẩn hóa dữ liệu. Sau bước này, dữ liệu đã sẵn sàng cho bước khai thác dữ liệu.
- Khai thác dữ liệu (data mining): Đây được xem là bước quan trọng nhất trong quá trình KDD. Nó áp dụng một số kỹ thuật khai thác dữ liệu (chủ yếu

là từ học máy và các lĩnh vực khác) để khai thác, trích chọn được những mẫu (patterns) thông tin, những mối liên hệ (relationships) đặc biệt trong dữ liệu.

- Biểu diễn và đánh giá tri thức (knowledge representation and evaluation): Những mẫu thông tin và mối liên hệ trong dữ liệu đã được khai thác ở bước trên được chuyển dạng và biểu diễn ở một dạng gần gũi với người sử dụng như đồ thị, cây, bảng biểu, luật, ... Đồng thời bước này cũng đánh giá những tri thức khám phá được theo những tiêu chí nhất định.

1.1.2 Tính riêng tư

Tính riêng tư là tính chất của các dữ liệu nhạy cảm: như định danh, tên, địa chỉ, điện thoại, thu nhập,… của các cá nhân, một số dữ liệu thống kê của các tổ chức hay doanh nghiệp. Các thông tin này thuộc về thông tin cá nhân, bí mật kinh doanh, nếu để lộ ra ngoài sẽ gây bất lợi cho cá nhân hay tổ chức,… hay các thông tin theo quy định của pháp luật không nên tiết lộ như thông tin bảo hiểm y tế, thông tin tài khoản ngân hàng,…. Những loại thông tin như trên được gọi là những thông tin có tính riêng tư hay tri thức nhạy cảm.

1.1.3 Khai thác dữ liệu bảo toàn tính riêng tư

Khai thác dữ liệu bảo toàn tính riêng tư PPDM (Privacy Preserving Data Mining) là hướng nghiên cứu bảo vệ tính riêng tư của dữ liệu lẫn tri thức trước và sau khi thực hiện khai thác trên dữ liệu.

Một số ví dụ minh hoạ:

Ví dụ 1: Dữ liệu về định danh, tên, địa chỉ, điện thoại, thu nhập, … của một cá nhân cần phải được sửa đổi hoặc loại bỏ bớt theo cách nào đó để cho người sử dụng dữ liệu không thể vi phạm tính riêng tư của họ.

Ví dụ 2: Siêu thị A có một cơ sở dữ liệu về các giao dịch bán hàng. A biết rằng cở sở dữ liệu này chứa dựng một số tri thức rất có lợi cho hoạt động kinh doanh. Siêu thị B mong muốn cùng được chia sẻ dữ liệu với A. Vì mối quan hệ, A đồng ý nhưng vì liên quan đến chiến lược kinh doanh, trước khi gửi cơ sở dữ liệu cho B, A đã thay đổi dữ liệu theo chiều hướng giấu đi những tri thức nhạy cảm mà A cho là quan trọng và không muốn tiết lộ.

Ví dụ 3: Cơ quan tình báo của một nước A quan sát hoạt động X = (x1, x2, …, xn) trong một thời gian dài. Cơ quan tình báo của B cũng quan sát một hoạt

động Y (y1, y2, …, ym) trong một thời gian dài. Họ muốn tìm ra những hoạt động của Y có tương quan với bất kỳ hoạt động nào của X hay không. Kết quả của sự cộng tác có thể giúp cả 2 nước hiểu ra khuynh hướng hoạt động của các đối tượng, như các hành vi của các tổ chức bị nghi ngờ là khủng bố, những hoạt động quân sự. Tuy nhiên cả A lẫn B đều không muốn tiết lộ những thông tin của nó cho những nước khác vì họ không hoàn toàn tin tưởng lẫn nhau. Rất có thể rằng B có thể sử dụng các thông tin tình báo của A, chẳng hạn như đem bán, để làm hại lại A.

Trường hợp đầu tiên liên quan đến vấn đề giấu dữ liệu nhạy cảm. Trường hợp thứ hai là bài toán thay đổi dữ liệu để việc chia sẻ dữ liệu không làm mất đi một số tri thức nhạy cảm. Ở ví dụ sau cùng, hai hay nhiều tổ chức đều có dữ liệu riêng và cùng muốn khai thác trên dữ liệu của chung, nhưng không ai muốn tiết lộ dữ liệu của mình.

1.2 Phân loại các phương pháp PPDM

Có nhiều cách tiếp cận dùng cho PPDM. Có nhiều cách phân loại khác nhau. Mỗi cách phân loại giúp ta hiểu vấn đề ở một khía cạnh khác nhau.

1.2.1 Phương pháp 1:

Có thể phân loại chúng dựa trên các tiêu chí như sau:

1) Sự phân bố dữ liệu (Data distribution): Dữ liệu tập trung hoặc dữ liệu phân tán. Trong trường hợp dữ liệu là phân tán thì phân tán ngang hoặc phân tán dọc.

2) Phương pháp sửa đổi dữ liệu (Data modification): Sửa đổi các giá trị nguyên thủy của CSDL trước khi gửi cho nhiều người nhận nhằm bảo vệ tính riêng tư. Kỹ thuật sửa đổi này phải phù hợp với chính sách riêng tư đang được sử dụng. Có thể liệt kê các phương pháp như sau:

a) Thay giá trị thực sự thành giá trị mới (ví dụ đổi 1 thành 0 hoặc làm nhiễu dữ liệu).

b) Làm cản trở quá trình phân tích dữ liệu bằng cách thay thế giá trị đã có thành "?".

c) Gom lại hoặc trộn lại, là sự kết hợp nhiều giá trị thành một phân loại thô hơn.

d) Đổi chỗ giữa các giá trị trong từng record.

e) Tạo mẫu: chỉ cho chia sẻ những dữ liệu mang tính chất chung.

3) Thuật toán khai khoáng (Data mining Algorithm): Các thuật toán khai khoáng gồm, phân lớp, cây quyết định, tìm tập phổ biến và luật kết hợp, gom nhóm, tập thô và mạng Bayesian.

4) Giấu dữ liệu hoặc giấu luật (Data or rule hiding): Gồm việc giấu dữ liệu thô hoặc dữ liệu kết hợp dạng luật. Có nhiều phương pháp dùng cho việc giấu dữ liệu kết hợp dưới dạng luật vì độ phức tạp cao hơn. Giảm bớt dữ liệu khi chia sẻ sẽ làm cho việc suy diễn yếu hơn hoặc cho ra giá trị suy diễn có độ tin cậy thấp. Quá trình này gọi là rule confusion.

5) Bảo vệ riêng tư (Privacy preservation): Là quan trọng nhất, liên quan đến các kỹ thuật bảo vệ tính riêng tư dùng để sửa đổi dữ liệu có chọn lọc. Sửa đổi dữ liệu có chọn lọc nhằm cho dữ liệu vẫn có tính thiết thực cao nhưng không ảnh hưởng đến tính riêng tư. Các kỹ thuật này gồm có:

a) Kỹ thuật dựa trên Heuristic (Heuristic-based techniques) như là chỉnh sửa thích nghi, tức là chỉ chỉnh sửa một cách có chọn lọc để giảm thiểu việc mất đi tính thiết thực của dữ liệu sau khi đã chỉnh sửa.

b) Kỹ thuật dựa trên phương pháp mã hóa (Cryptographic-based techniques) chẳng hạn như kỹ thuật bảo mật tính toán đa thành phần SMC (Secure multiparty computation), trong đó có nhiều người tham gia vào một hệ thống phân tán, mỗi người có một dữ liệu đầu vào (input) và tham gia quá trình tính toán dựa trên một/ một số dữ liệu đầu vào khác để cho ra kết quả cuối cùng (output). Từng người tham gia chỉ biết giá trị input của người đó và kết quả trả về, ngoài ra không biết gì hơn.

c) Kỹ thuật dựa trên sự tái tạo (Reconstruction-based techniques): Sự phân bố của dữ liệu nguyên thủy được tái tạo lại từ dữ liệu ngẫu nhiên.

1.2.2 Phương pháp 2:

Có thể chia các kỹ thuật PPDM ra làm 2 nhóm:

- Nhóm 1: Chia sẻ dữ liệu (Data-sharing techniques): Gồm các thuật toán làm thay đổi dữ liệu ban đầu để giấu đi dữ liệu nhạy cảm. Có thể chia ra làm 3 loại:

a) Xóa bớt phần từ (item restriction – based): Là làm giảm độ hỗ trợ hoặc độ tin cậy (trong bài toán tìm luật kết hợp) của luật bằng cách xóa giao tác hoặc một/ một số item của một giao tác để giấu luật nhạy cảm.

b) Thêm phần từ (item addition-based): Thêm item ảo vào các giao tác nhằm giấu đi một số luật nhạy cảm (và phát sinh tri thức không có thật).

c) Thay bằng giá trị không rõ (unknown - ?): Để giấu tri thức nhạy cảm.

- Nhóm 2: Chia sẻ tri thức khám phá được từ dữ liệu (Pattern-sharing techniques): Gồm các thuật toán giấu luật khai phá được chứ không phải giấu dữ liệu. Các giải pháp thuộc loại này tìm cách loại bỏ các luật nhạy cảm trước khi chia sẻ luật hoặc chia sẻ theo kiểu bảo mật tính toán đa thành phần SMC.

1.2.3 Phương pháp 3:

Khai thác dữ liệu bảo toàn tính riêng tư được thực hiện ở các cấp độ sau:

- Cấp độ 1: Gồm các kỹ thuật áp dụng trên dữ liệu thô ban đầu với mục đích tránh mất dữ liệu hoặc tri thức nhạy cảm. Hoặc kỹ thuật bảo vệ tính riêng tư của 2 hay nhiều người tham gia muốn khai thác trên dữ liệu chung nhưng không muốn mất thông tin riêng tư trên dữ liệu của từng người.
- Cấp độ 2: Gồm các kỹ thuật đảm bảo tính riêng tư được nhúng trong thuật toán khai thác dữ liệu. Thông thường, những chuyên gia về dữ liệu dùng các ràng buộc trước khi hoặc trong khi thực hiện khai thác.
- Cấp độ 3: Gồm các kỹ thuật áp dụng trên kết quả của quá trình khai thác nhằm đạt được cùng mục đích như ở cấp độ 1.

1.3 Các phương pháp giấu dữ liệu nhạy cảm:

1.3.1 Làm xáo trộn (Perturbation)

Với kỹ thuật này, giá trị nguyên thủy của dữ liệu bị thay đổi thành một giá trị khác hoặc thêm nhiễu. Trong các cơ sở dữ liệu nhị phân (còn gọi là cơ sở dữ liệu giỏ hàng siêu thị), người ta làm xáo trộn dữ liệu bằng cách thay đổi giá trị 1 bằng giá trị 0 và/hoặc 0 thành 1. Ngoài ra, có thể thêm nhiễu trên dữ liệu bằng cách thay giá trị x bởi giá trị (x + r), với r là một giá trị ngẫu nhiên lấy từ một phân bố xác suất nào đó. Phương pháp làm việc trên dữ liệu sau khi làm xáo trộn tùy thuộc vào thuật toán khai khoáng.

1.3.2 Ngăn chặn (Blocking)

Là việc thay đổi giá trị nguyên thủy bởi một ký hiệu mang ý nghĩa là "không biết". Thường người ta dùng ký hiệu dấu "?" để biểu thị cho giá trị không biết này.

Ưu điểm: Kỹ thuật blocking có thể ngăn cản quá trình khai khoáng cho ra những tri thức nhạy cảm, mà lại không sinh ra những tri thức sai gây ảnh hưởng đến người dùng dữ liệu.

1.3.3 Gom hoặc trộn (Aggregation / Merging)

Là việc giấu dữ liệu chi tiết bằng cách kết hợp các thuộc tính hoặc các đối tượng (object) lại tương ứng thành 1 thuộc tính hoặc 1 đối tượng. Phương pháp này thường được dùng trong quá trình tiền xử lý dữ liệu phục vụ cho mục đích khai thác, nhằm bỏ bớt dữ liệu, hoặc giảm sự biến động trên giá trị của dữ liệu.

1.3.4 Đổi chổ (Swapping)

Là việc đổi chỗ các giá trị giữa các mẫu tin với nhau trong cơ sở dữ liệu. Phương pháp này được giới thiệu đầu tiên vào năm 1980

Ta có ví dụ:

#	Tuổi	Thu nhập
1	21	20000
2	24	30000
3	35	30000
4	36	25000
5	45	55000
6	50	15000

(a)

#	Tuổi	Thu nhập
1	21	15000
2	24	30000
3	35	30000
4	36	55000
5	45	25000
6	50	20000

(b)

#	Tuổi	Thu nhập
1	24	15000
2	21	30000
3	36	30000
4	35	55000
5	50	25000
6	45	20000

(c)

Hình 1.1 Ví dụ cho phương pháp đỗi chỗ.

(a) Dữ liệu nguyên thủy.

(b) Dữ liệu sau khi đổi chỗ ngẫu nhiên trên trường Thu nhập, xảy ra trên các cặp mẫu tin 1 và 6, 2 và 3, 4 và 5.

(c) Dữ liệu sau khi đổi chỗ ngẫu nhiên trên trường Tuổi, xảy ra trên các cặp mẫu tin 1 và 2, 3 và 4, 5 và 6.

Một số nhận xét:

- Xác suất một lần đổi chỗ giấu được thông tin của 1 mẫu tin tỉ lệ nghịch với tần suất giá trị đó xuất hiện trên mẫu tin. Điều này có thể chấp nhận được trên dữ

liệu có kích thước lớn. Một giá trị thu nhập xuất hiện thường xuyên trên tập tin sẽ khó xác định là ứng với giá trị của mẫu tin nào so với một giá trị thu nhập xuất hiện trên tập tin với tần suất thấp.

- Việc đổi chỗ ngẫu nhiên trên các trường khác nhau có thể xảy ra trên các mẫu tin khác nhau.
- Các lần đổi chỗ diễn ra một cách độc lập, trên các trường khác nhau và có thể giấu được thông tin chính xác của từng mẫu tin.
- Theo khuyến cáo, nên thực hiện đổi chỗ trên các trường nhạy cảm, ví dụ như Thu nhập, Tuổi,…

1.3.5 Lấy mẫu:

Khi khảo sát một quần thể, nếu căn cứ vào tất cả các cá thể của quần thể thì không khả thi và tốn rất nhiều chi phí. Phương pháp lấy mẫu được dùng để chọn ra các mẫu trong quần thể. Sự ước lượng về quần thể dựa trên thông tin của những mẫu được chọn này. Vì vậy, tập mẫu phải đủ lớn để đại diện tốt cho quần thể, nhưng phải đủ nhỏ để có thể quản lý được. Phương pháp lấy mẫu thường được dùng trong khai thác dữ liệu để giảm chi phí tính toán và thời gian xử lý dữ liệu thay vì phải xử lý toàn bộ dữ liệu. Có 2 phương pháp lấy mẫu chính: lấy mẫu ngẫu nhiên và lấy mẫu không ngẫu nhiên. Có 5 cách lấy mẫu ngẫu nhiên:

1) Lấy ngẫu nhiên đơn giản (Simple random sampling): Lấy ngẫu nhiên một cá thể từ quần thể và lấy các cá thể được chọn làm mẫu đại diện cho cả quần thể. Mỗi cá thể có thể trở thành mẫu khảo sát với xác suất bằng nhau. Có 2 cách lấy:

- Lấy mẫu không giữ lại giá trị mẫu (Sampling without replacement): Cứ mỗi cá thể được chọn ra làm mẫu thì cá thể đó sẽ bị loại bỏ khỏi quần thể.
- Lấy mẫu vẫn giữ lại giá trị mẫu (Sampling with replacement): Cá thể làm mẫu vẫn được giữ lại trong quần thể. Cùng một cá thể có thể được chọn ra làm giá trị mẫu nhiều lần.

Ví dụ: Cần chọn ra 2000 người từ cuốn danh bạ điện thoại được đánh số tuần tự theo tên người đăng ký. Phương pháp lấy mẫu ngẫu nhiên đơn giản sẽ sinh ra 2000 số ngẫu nhiên (trùng nhau hoặc không trùng nhau) và sẽ lấy thông tin ứng với các số ngẫu nhiên vừa tạo để cho ra tập mẫu.

Ưu điểm: Đơn giản và dễ ứng dụng khi thực hiện lấy mẫu trên quần thể nhỏ.

Khuyết điểm: Do từng cá thể trong quần thể phải có mặt trước khi thực hiện phương pháp này nên phương pháp này khó áp dụng đối với quần thể lớn.

2) Lấy mẫu có hệ thống (Systematic sampling): Còn được gọi là lấy mẫu theo đoạn vì lần chọn mẫu này cách lần chọn mẫu kia một khoảng bằng nhau về số lượng cá thể (bị bỏ qua không chọn làm mẫu). Lần chọn đầu tiên là ngẫu nhiên. Cứ sau k lần bỏ qua không chọn thì bắt đầu chọn tiếp, k không đổi trong suốt quá trình chọn mẫu. Phương pháp này thường được áp dụng trong công nghiệp khi muốn chọn mẫu để kiểm tra dây chuyền sản xuất.

Ví dụ: Nhà sản xuất quyết định chọn mỗi cá thể xuất hiện lần thứ hai mươi sau cá thể được chọn trước kia trên dây chuyền để kiểm tra chất lượng.

Ưu điểm: dễ thực hiện, chỉ chọn ngẫu nhiên cho lần chọn mẫu đầu tiên nhưng khả năng được chọn của từng cá thể trong quần thể giống nhau.

Khuyết điểm: Cần biết trước quần thể lấy mẫu nếu biết kích thước tập mẫu và khoảng cách giữa 2 lần lấy mẫu.

3) Lấy mẫu theo phân đoạn (Stratified sampling): Chia dữ liệu thành nhiều phân đoạn, và sau đó vận dụng phương pháp lấy mẫu ngẫu nhiên đơn giản hoặc lấy mẫu có hệ thống đối với từng phân đoạn.

Ví dụ: Ban giám hiệu của 1 trường học có 1000 sinh viên muốn khảo sát về một vấn đề A trên các sinh viên ở các năm khác nhau. Để đảm bảo tập mẫu có tính đại diện cho sinh viên ở từng năm, ban giám hiệu dùng phương pháp lấy mẫu theo phân đoạn, sinh viên thuộc năm thứ k sẽ thuộc phân đoạn thứ k.

Ưu điểm: Phương pháp này phù hợp với những khảo sát dựa trên những thuộc tính có thể phân đoạn đơn giản, dễ quan sát và có liên quan mật thiết với chủ trương của cuộc khảo sát. Phương pháp này cho phép chọn mẫu trong phân đoạn này nhiều hơn trong phân đoạn khác, có thể vì dữ liệu ứng với các cá thể trong phân đoạn này có nhiều biến động hơn so dữ liệu ứng với các cá thể thuộc phân đoạn khác, và vì thế cần phải khảo sát chúng.

4) Lấy mẫu theo nhóm (Cluster sampling): Đôi khi việc lấy mẫu khó thực hiện trên toàn bộ quần thể. Phương pháp lấy mẫu theo nhóm chia quần thể ra nhiều nhóm, chọn ngẫu nhiên một số nhóm làm đại diện cho quần thể.

Ví dụ: Có một cuộc khảo sát toàn quốc tìm môn thể thao được yêu thích nhất của học sinh lớp 12. Nếu khảo sát toàn bộ học sinh lớp 12 trên toàn quốc thì tốn nhiều thời gian và chi phí. Thay vì vậy, vận dụng phương pháp lấy mẫu theo nhóm, 100 trường học cấp 3 được chọn ngẫu nhiên, mỗi học sinh học lớp 12 trong 100 trường này được khảo sát về môn thể thao yêu thích.

Ưu điểm: giảm chi phí, làm đơn giản hóa việc khảo sát và quản lý thuận tiện hơn.

Khuyết điểm: với cùng kích thước tập mẫu thì phương pháp này cho kết quả có độ chính xác thấp hơn so với phương pháp lấy mẫu ngẫu nhiên đơn giản vì có khả năng mẫu lấy sai.

5) Lấy mẫu nhiều giai đoạn (Multi-stage sampling): Việc lấy mẫu trải qua tối thiểu là 2 giai đoạn, mỗi giai đoạn giống phương pháp lấy mẫu theo nhóm, nhưng không phải chọn tất cả cá thể trong nhóm đã chọn mà tiếp tục chọn mẫu trong từng nhóm.

Ví dụ: Để khảo sát tình hình bầu cử trong cả nước, đầu tiên là phân nhóm theo tỉnh hoặc thành phố, chọn một số tỉnh hoặc thành phố nào đó để khảo sát. Kế tiếp là chọn một số phường/ xã trong từng tỉnh/ thành phố được chọn, sau đó là chọn một số ấp/ khu phố và cuối cùng là chọn một số nhà để khảo sát.

Ưu điểm: tiện lợi, kinh tế và hiệu quả.

Khuyết điểm: vì cơ bản dựa trên phương pháp lấy mẫu theo nhóm nên độ chính xác thấp.

6) Phương pháp lấy mẫu thứ hai là lấy mẫu không ngẫu nhiên. Có 3 cách:

- Lấy mẫu theo phần (Quota sampling): giống phương pháp lấy mẫu theo phân đoạn nhưng khi chọn mẫu trong từng phân đoạn thì chọn không ngẫu nhiên.

Ví dụ: Khảo sát 1 vấn đề A trong 1000 sinh viên ở nhiều năm khác nhau. Đầu tiên chia 1000 sinh viên ra làm các phân đoạn theo năm học. Sau đó, người lấy mẫu định ra trong 100 sinh viên cần chọn ra để khảo sát thì mỗi phân đoạn sẽ lấy bao nhiêu sinh viên. Đến đây, theo cách lấy mẫu không ngẫu nhiên, phương pháp ngẫu nhiên đơn giản hoặc lấy mẫu có hệ thống không được dùng. Giả sử đối với sinh viên năm 4 thì phải chọn ra 15 sinh viên (chiếm 15%). 15 sinh viên này được

chọn theo cách, có thể là 15 sinh viên năm 4 đầu tiên đi vào trường hoặc 15 sinh viên năm 4 ngồi trên các dãy bàn đầu trong một lớp học nào đó.

Phương pháp này thường dùng để khảo sát thị trường hoặc khảo sát ý kiến các nhà nghiên cứu.

Ưu điểm: chi phí thấp, dễ thực hiện.

Khuyết điểm: đảm bảo tập mẫu đại diện được quần thể theo một tiêu chuẩn nào đó (ví dụ sinh viên năm thứ mấy) nhưng có thể không mang tính đại diện xét trên các tiêu chuẩn khác. Vì không dựa trên phương pháp chọn ngẫu nhiên nên có những cá thể không có cơ hội được chọn. Người ta nói phương pháp lấy mẫu này có độ lệch (biased).

- Lấy mẫu tiện lợi (Convenience Sampling): không tạo ra một tập mẫu đại diện cho quần thể vì dựa trên nguyên tắc là cá thể sẽ được chọn làm mẫu nếu chúng được biết đến một cách dễ dàng và thuận tiện.

Ví dụ: Mẫu được lấy là 10 xe hơi đầu tiên vào bãi đậu xe, hoặc 10 người nữ ở hàng ghế đầu tiên trong một buổi hòa nhạc.

Ưu điểm: Dễ thực hiện.

Khuyết điểm: Là phương pháp lấy mẫu có độ lệch.

- Lấy mẫu tự nguyện (Volunteer sampling): thường được dùng bởi các đài truyền thanh hoặc truyền hình để khảo sát ý kiến công chúng về vấn đề gì đó. Nhiều người sẽ gọi điện hoặc nhắn tin để biểu quyết cho vấn đề đặt ra trong một khoảng thời gian định trước, không giới hạn về số lượt người tham gia.

Ưu điểm: chi phí ít và dễ quản lý.

Khuyết điểm: không giới hạn được số lần biểu quyết của cùng một người nên không chắc chắn rằng tập mẫu có được mang tính đại diện. Phương pháp này có độ lệch, thiên về những người xem truyền hình hoặc nghe đài trong thời gian biểu quyết và có thể dùng điện thoại để biểu quyết.

1.4 Luật kết hợp

1.4.1 Định nghĩa:

Cho I={I_1, $I_{2,}$, …., $I_{m,}$} là tập hợp của m tính chất riêng biệt. Giả sử D là CSDL, với các bản ghi chứa một tập con T các tính chất (có thể coi như T ⊆ I), các bản ghi đều có chỉ số riêng. Một luật kết hợp là một mệnh đề kéo theo có dạng

X→Y, trong đó X, Y ⊆ I, thỏa mãn điều kiện X∩Y=∅. Các tập hợp X và Y được gọi là các tập hợp tính chất (itemset). Tập X gọi là nguyên nhân, tập Y gọi là hệ quả.

Có hai độ đo quan trọng đối với luật kết hợp: Độ hỗ trợ (support), và độ tin cậy (confidence), được định nghĩa sau đây.

1.4.2 Định nghĩa "Độ hỗ trợ":

Định nghĩa 1.1: Độ hỗ trợ của một tập hợp X trong CSDL D là tỷ số giữa các bản ghi T ∈ D có chứa tập X và tổng số bản ghi trong D (hay là phần trăm của các bản ghi trong D có chứa tập hợp X), ký hiệu là support(X) hay Supp(X) (Support sẽ tự sinh ra khi cài đặt thuật toán)

$$\mathrm{Supp}(X) = \frac{|\{T \in D : X \subset T\}|}{|D|}$$

Ta có: 0 ≤ Supp(X) ≤ 1 với mọi tập hợp X.

Định nghĩa 1.2: Độ hỗ trợ của một luật kết hợp X→Y là tỷ lệ giữa số lượng các bản ghi chứa tập hợp X ∪ Y, so với tổng số các bản ghi trong D. Ký hiệu Supp(X→Y)

$$\mathrm{Supp}(X \to Y) = \frac{|\{T \in D : X \cup Y \subseteq T\}|}{|D|}$$

- Khi chúng ta nói rằng độ hỗ trợ của một luật là 50%, có nghĩa là có 50% tổng số bản ghi chứa X∪Y. Như vậy, độ hỗ trợ mang ý nghĩa thống kê của luật.
- Trong một số trường hợp, chúng ta chỉ quan tâm đến những luật có độ hỗ trợ cao (Ví dụ như luật kết hợp xét trong cửa hàng tạp phẩm). Nhưng cũng có trường hợp, mặc dù độ hỗ trợ của luật thấp, ta vẫn cần quan tâm (ví dụ luật kết hợp liên quan đến nguyên nhân gây ra sự đứt liên lạc ở các tổng đà điện thoại).

1.4.3 Định nghĩa "Độ tin cậy":

Định nghĩa 1.3: Độ tin cậy của một luật kết hợp X→Y là tỷ lệ giữa số lượng các bản ghi trong D chứa X∪Y với số bản ghi trong D có chứa tập hợp X. Ký hiệu độ tin cậy của một luật là Conf(r). Ta có 0 ≤ Conf(r) ≤ 1

Nhận xét: Độ hỗ trợ và độ tin cậy có xác suất sau

Supp(X→Y)=P(X∪Y)

Conf (X→Y) = P(Y/X)=Supp(X∪Y)/Supp(X)

Có thể định nghĩa độ tin cậy như sau:

Định nghĩa 1.4: Độ tin cậy của một luật kết hợp X→Y là tỷ lệ giữa số lượng các bản ghi của tập hợp chứa XUY, so với tổng số các bản ghi chứa X.

- Nói rằng độ tin cậy của một luật là 90%, có nghĩa là có tới 90% số bản ghi chứa X chứa luôn cả Y. Hay nói theo ngôn ngữ xác suất là: " Xác suất có điều kiện để sảy ra sự kiện Y đạt 85%". Điều kiện ở đây chính là: "Xảy ra sự kiện X".
- Như vậy, độ tin cậy của luật thể hiện sự tương quan (correlation) gữa X và Y. Độ tin cậy đo sức nặng của luật, và người ta hầu như chỉ quan tâm đến những luật có độ tin cậy cao. Một luật kết hợp đi tìm các nguyên nhân dẫn tới hỏng hóc của hệ thống tổng đài, hay đề cập đến những mặt hàng thường hay được khách hàng mua kèm với mặt hàng chính mà độ tin cậy thấp sẽ không có ích cho công tác quản lý.
- Việc khai thác các luật kết hợp từ CSDL chính là việc tìm tất cả các luật có độ hỗ trợ và độ tin cậy do người sử dụng xác định trước. Các ngưỡng của độ hỗ trợ và độ tin cậy được ký hiệu là *Minsup* và *Minconf*.

Ví dụ: Khi phân tích giỏ hàng của người mua hàng trong một siêu thị ta được luật kiểu như: 85% khách hàng mua sữa thì cũng mua bánh mì, 30% thì mua cả hai thứ. Trong đó: "mua sữa" là tiền đề còn "mua bánh mì " là kết luận của luật. Con số 30% là độ hỗ trợ của luật còn 80% là độ tin cậy của luật.

- Chúng ta nhận thấy rằng tri thức đem lại bởi luật kết hợp dạng trên có sự khác biệt rất nhiều so với những thông tin thu được từ các câu lệnh truy vấn dữ liệu thông thường như SQL. Đó là những tri thức, những mối liên hệ chưa biết trước và mang tính dự báo đang tiềm ẩn trong dữ liệu. Những tri thức này không đơn giản là kết quả của phép nhóm, tính tổng hay sắp xếp mà là của một quá trình tính toán khá phức tạp.

1.4.4 Định nghĩa "Tập hợp":

Định nghĩa 1.5: Tập hợp X được gọi là tập hợp thường xuyên (Frequent itemset) nếu có Supp(X) ≥ *Minsup*, với *Minsup* là ngưỡng độ hỗ trợ cho trước. Kí hiệu các tập này là FI

Tính chất 1.1: Giả sử A,B ⊆ I là hai tập hợp với A ⊆ B thì Supp(A) ≥ Supp(B). Như vậy, những bản ghi nào chứa tập hợp B thì cũng chứa tập hợp A.

Tính chất 1.2: Giả sử A, B là hai tập hợp, A,B ⊆ I, nếu B là tập hợp thường xuyên và A ⊆ B thì A cũng là tập hợp thường xuyên.

Thật vậy, nếu B là tập hợp thường xuyên thì Supp(B) ≥ *Minsup*, mọi tập hợp A là con của tập hợp B đều là tập hợp thường xuyên trong cơ sở dữ liệu D vì Supp(A) ≥ Supp(B) (Tính chất 1.1).

Tính chất 1.3: Giả sử A, B là hai tập hợp, A ⊆ B và A là tập hợp không thường xuyên thì B cũng là tập hợp không thường xuyên.

Định nghĩa 1.6: Một tập mục X được gọi là đóng (closed) nếu không có tập cha nào của X có cùng độ hỗ trợ với nó, tức là không tồn tại một tập mục X' nào mà X' ⊃ X và t(X) = t(X') (với t(X) và t(X') tương ứng là tập các giao chứa tập mục X và X'). Ký hiệu tập phổ biến đóng là FCI.

Định nghĩa 1.7: Nếu X là phổ biến và không tập cha nào của X là phổ biến, ta nói rằng X là một tập phổ biến tối đại (maximally frequent itemset). Ký hiệu tập tất cả các tập phổ biến lớm nhất là MFI. Dễ thấy MFI ⊆ FCI ⊆FI.

- Khai thác luật kết hợp là công việc phát hiện ra (tìm ra, khám phá, phát hiện) các luật kết hợp thỏa mãn các ngưỡng độ hỗ trợ(σ) và ngưỡng độ tin cậy(α) cho trước. Bài toán khai thác luật kết hợp được chia thành hai bài toán nhỏ, hay như người ta thường nói, việc giải bài toán trải qua hai pha:

+ Pha 1: Tìm tất cả các tập phổ biến (tìm FI) trong CSDL T.

+ Pha 2: Sử dụng tập FI tìm được ở pha 1 để sinh ra các luật tin cậy (interesting rules). Ý tưởng chung là nếu gọi ABCD và AB là các tập mục phổ biến, thì chúng ta có thể xác định luật AB→CD với tỷ lệ độ tin cậy:

$$\text{Conf} = \frac{\text{Supp(ABCD)}}{\text{Supp(AB)}}$$

- Nếu Conf ≥ *Minconf* thì luật được giữ lại (và thỏa mãn độ hỗ trợ tối thiểu vì ABCD là phổ biến).

- Trong thực tế, hầu hết thời gian của quá trình khai thác luật kết hợp là thực hiện ở pha 1. Nhưng khi có những mẫu rất dài (mẫu chứa nhiều mục)xuất hiện trong dữ liệu, việc sinh ra toàn bộ các tập phổ biến (FI) hay các tập đóng (FCI) là không thực tế. Hơn nữa, có nhiều ứng dụng mà chỉ cần sinh tập phổ biến tối đại (MFI) là đủ, như khám phá mẫu tổ hợp trong các ứng dụng sinh học.

- Có rất nhiều nghiên cứu về các phương pháp sinh tất cả các tập phổ biến và tập phổ biến lớn nhất một cách có hiệu quả. Khi các mẫu phổ biến (frequent patterm) dài có từ 15 đến 20 items thì tập FI, thậm chí cả tập FCI trở nên rất lớn và hầu hết các phương pháp truyền thống phải đếm quá nhiều tập mục mới có thể thực hiện được. Các thuật toán dựa trên thuật toán Apriori, đếm tất cả 2k tập con của mỗi k-itemsets mà chúng quét qua, và do đó không thích hợp với các itemsets dài được. Các phương pháp khác sử dụng "look aheads" để giảm số lượng tập mục được đếm. Tuy nhiên, hầu hết các thuật toán này đều sử dụng tìm kiếm theo chiều rộng, ví dụ: tìm tất cả các k – itemsets trước khi tính đến các (k+1) – itemsets.
- Cách làm này hạn chế hiệu quả của lookaheads, vì các mẫu phổ biến dài hơn mà hữu ích vẫn chưa được tìm ra.

1.5 Thuật toán Apriori

1.5.1 Nguyên lý Apriori

Với mọi tập không phổ biến thì mọi tập chứ nó không là tập phổ biến. Dựa vào nguyên lý này, người ta thiết kế thuật toán Apriori như sau:

- Sinh các tập ứng viên dài (k + 1) từ các tập mục phổ biến có độ dài k (Độ dài tập mục là số phần từ của nó).
- Kiểm tra các tập ứng viên theo CSDL để loại bỏ các tập không phổ biến (có độ hỗ trợ < min_support).

1.5.2 Thuật toán Apriori

Bước đầu tiên, thực hiện duyệt CSDL để tìm các mục riêng biệt trong CSDL và độ hỗ trợ tương ứng của nó. Tập thu được là C_1. Duyệt tập C_1, loại bỏ các tập có độ hỗ trợ < min_support, các tập mục còn lại của C_1 là các tập 1-Itemset(L_1) phổ biến. Sau đó kết nối L_1 với L_1 để được các tập các tập 2-Itemset C_2. Duyệt CSDL để xác định độ hỗ trợ của các tập mục trong C_2. Duyệt C_2, loại bỏ các tập mục có độ hỗ trợ < min_support, các tập mục còn lại của C_2 là các tập 2-Itemset(L_2) phổ biến. L_2 được sử dụng để sinh ra L_3, và cứ tiếp tục các bước nhu vậy cho đến khi tìm được tập mục k-Itemset(L_k) mà $L_k = \emptyset$ (nghĩa là không còn tập phổ biến nào được tìm thấy nữa) thì dừng lại.

Tập các tập mục phổ biến của CSDL là: $\cup_{i-1}^{k} = L_1$.

Để tăng hiệu quả của thuật toán, trong quá trình sinh các tập ứng viên, ta sử dụng tính chất của tập mục phổ biến để làm giả số lượng tập các ứng viên, không phải là tập phổ biến được sinh ra. Ta có tính chất: Tập các tập con khác rỗng của tập mục phổ biến đều là tập mục phổ biến.

Mô tả thuật toán:

(1) Duyệt (scan) toàn bộ CSDL giao dịch để có được support S của 1-Itemset, so sánh S với min_support, để có được 1-Itemset(L_1).

(2) Sử dụng L_{k-1} nối (join) L_{k-1} để sinh ra tập ứng viên k-Itemset. Loại bỏ các Itemsets không phải là tập phổ biến thu được k-Itemsets.

(3) Duyệt (scan) toàn bộ CSDL giao dịch để có được support S của mỗi tập ứng viên k-Itemset, so sánh S với min_support, để thu được tập phổ biến k-Itemset(L_k).

(4) Lập lại từ bước thứ (2) cho đến khi các tập ứng viên C trống (không tìm thấy tập phổ biến).

(5) Với mỗi tập phổ biến I, sinh tất cả các tập con s không rỗng của I.

(6) Với mỗi tập con s không rỗng của I, sinh ra các luật $s => (I - s)$ nếu độ tin cậy (Confidence) của nó ≥ min_conf.

Mã giả thuật toán Apriori:

```
INPUT:
      - Cơ sở dữ liệu giao dịch D = {t | t giao dịch}
      - Độ hỗ trợ minsup > 0
OUTPUT:
      - Tập hợp tất cả các tập phổ biến.
mincount = minsup * |D|;
F1 = {Các tập phổ biến có độ dài bằng 1}
for (k = 1; Fk ≠ 0; k++ ) do begin{
      Ck+1 = apriori_gen(Fk)  //sinh mọi ứng viên có độ dài là k+1
      for  t ∈ D do begin{
            Ct = {c ∈ Ck+1 | c ⊆ t } //mọi ứng viên chứa trong t
            for c ∈ Ct do
                  c.count++
}
```

Hình 1.2 Mã giả thuật toán Apriori [3]

Nhật xét:

- Trong mỗi bước k, thuật toán Apriori đều phải duyệt cơ sở dữ liệu D.
- Khởi động, duyệt D để có F_1.
- Các bước k sau đó, duyệt D để tính số lượng giao dịch t thoả ứng viên c của C_{k+1}: Mỗi giao dịch t chỉ xem xét một lần cho mọi ứng viên c thuộc C_{k+1}. Mỗi bước k thực hiện hai thủ tục sau:

Bước nối: Sinh các tập mục R_{k-1} là ứng viên tập phổ biến có độ dài k+1 bằng cách kết hợp hai tập phổ biến R_k và Q_k có độ dài k và trùng nhau ở k-1 mục đầu tiên:

$R_{k-1} = P_k \cup Q_k = \{i_1, i_2, \dots, i_{k-1}, i_k\}$ với

$P_k = \{i_1, i_2, \dots, i_{k-1}, i_k\}$ và $Q = \{i_1, i_2, \dots, i_{k-1}, i_{k'}\}$

Trong đó $i_1 \le i_2 \le \dots \le i_{k-1} \le i_k \le i_{k'}$

Bước tỉa: Trong bước này ta giữ lại các R_{k+1}, thoả tính chất Apriori ($\forall X \subseteq R_{k+1}$ và $|X| = k \Rightarrow X \in F_k$), nghĩa là đã loại (tỉa) bớt đi các ứng viên R_{k+1} không đáp ứng tính chất này.

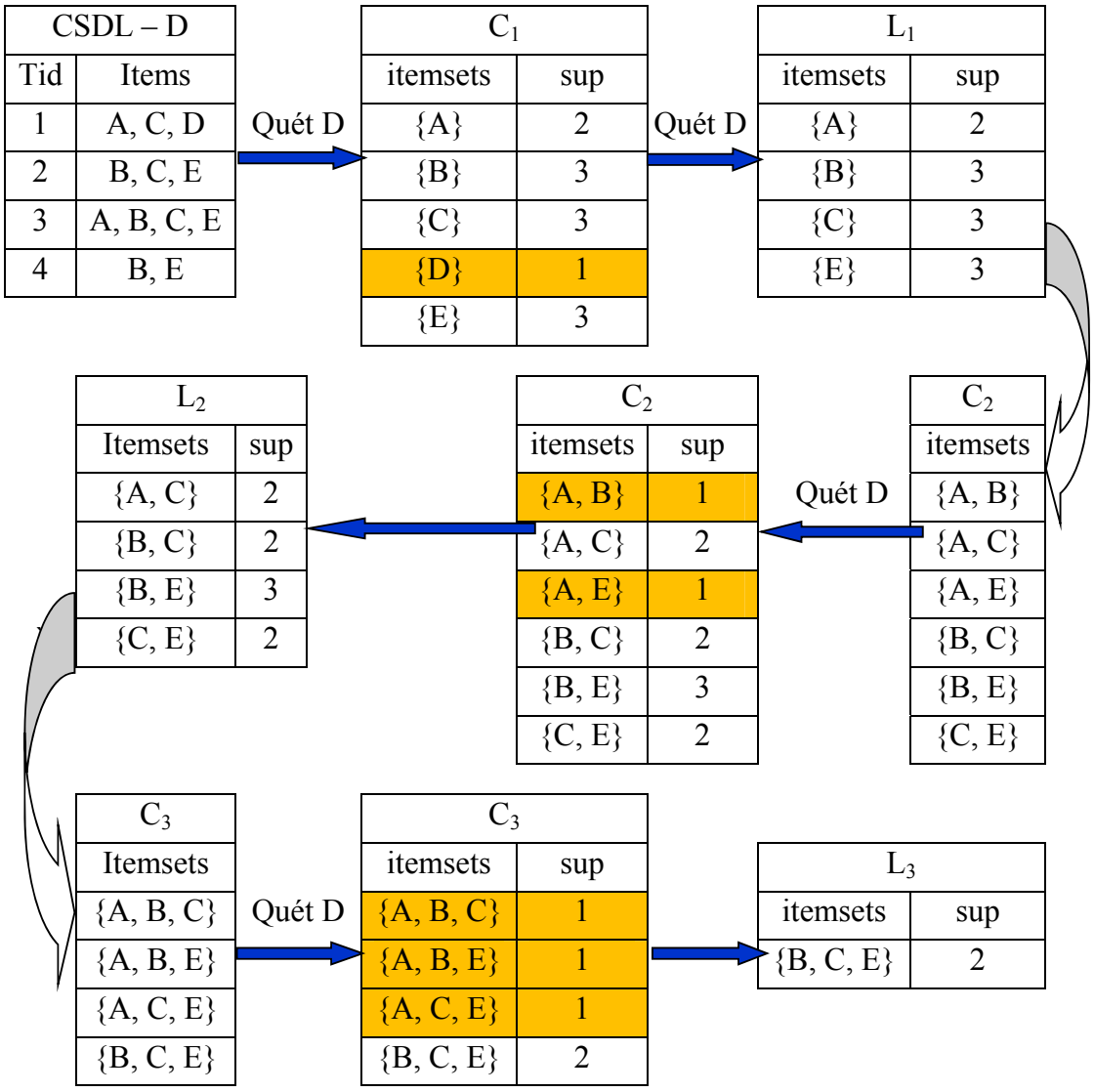
Ta có mã giả:

```
for mọi tập phổ biến I1 ∈ Lk
for mọi tập phổ biến I2 ∈ Lk
if(I1[1] = I2[1]) ∧ (I1[2] = I2[2]) ∧ … ∧ (I1[k − 1] = I2[k − 1]) ∧(I1[k] = I2[k])
      then{
      c = I1 ⇔ I2; //join step:generate candidates
}
Procedura has_infrequent_subset(c:tập ứng viên độ dài k+1; Lk: tập các
mục phổ biến độ dài k); // tri thức đã có
      for mỗi tập con s độ dài k của c
            if  s ∈ Lk
            then  Return TRUE
      Return FALSE
```

Hình 1.3 Mã giả cho thuật toán Apriori_gen [3]

1.5.3 Ví dụ minh họa thuật toán Apriori:

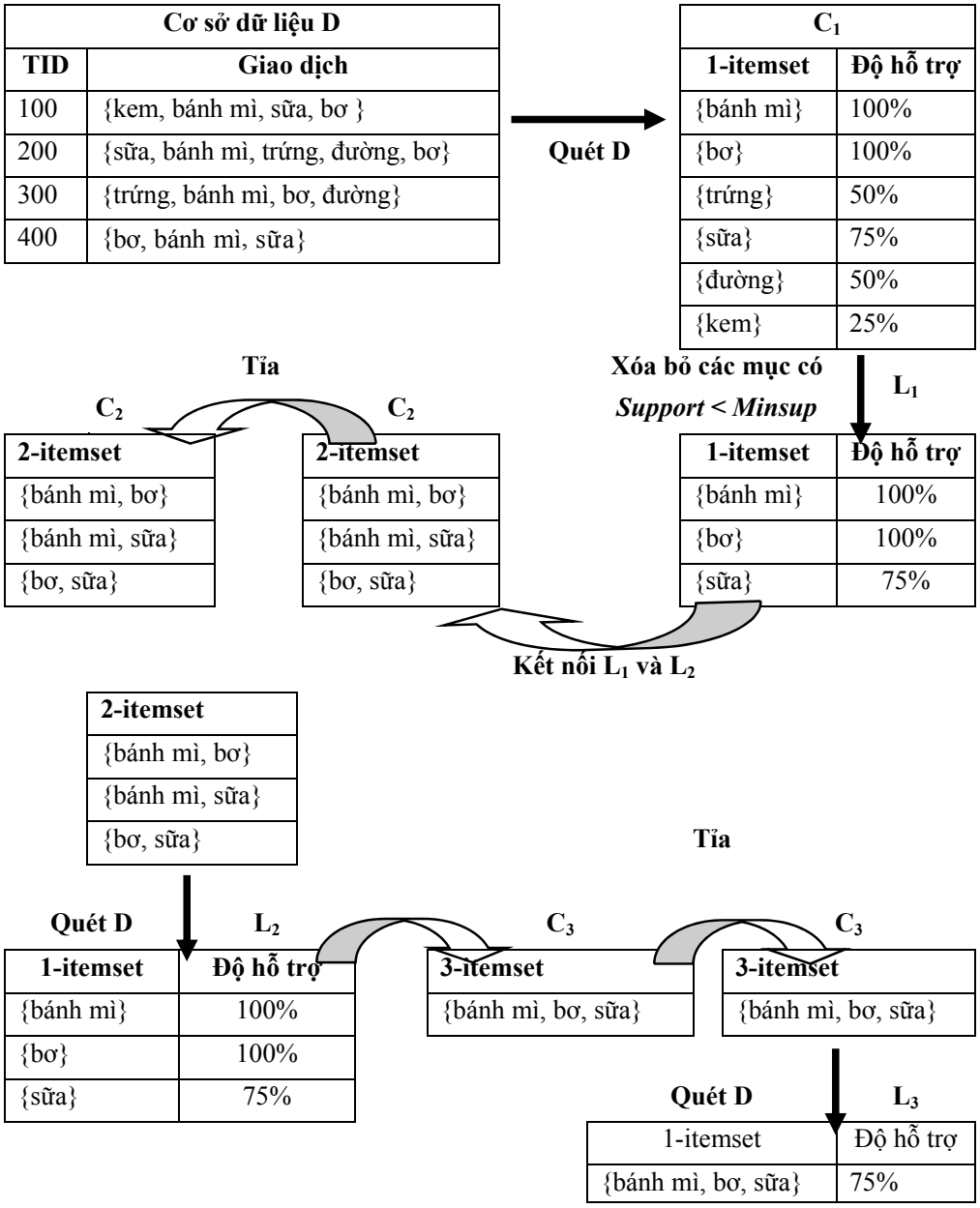
Ví dụ 1: Cho cơ sở dữ liệu giao dịch D, I = {A, B, C, D, E}. Áp dụng thuật toán Apriori để tìm các tập phổ biến thỏa *Minsup* = 2.



Hình 1.4 Ví dụ 1 cho thuật toán Apriori

Sau khi áp dụng thuật toán Apriori các tập phổ biến thu được trong Hình 1-4 là $L = L_1 \cup L_2 \cup L_3 = \{\{A\}; \{B\}; \{C\}; \{E\}; \{A, C\}; \{B, C\}; \{B, E\}, \{C, E\}, \{B, C, E\}\}$

Ví dụ 2: Cho cơ sở dữ liệu giao dịch D, I = {bánh mì, bơ, trứng, sữa, đông sương, kem}. Áp dụng thuật toán Apriori để tìm các tập phổ biến thỏa *Minsup* = 60%.



Hình 1.5 Ví dụ 2 cho thuật toán Apriori

Sau khi áp dụng thuật toán Apriori các tập phổ biến thu được trong Hình 1-3 là

$L = L_1 \cup L_2 \cup L_3 =$

$\{\{bánh\ mì\}; \{bơ\}; \{sữa\}; \{bánh\ mì, bơ\}; \{bánh\ mì, sữa\}; \{bơ, sữa\}; \{bánh\ mì, bơ, sữa\}\}$

CHƯƠNG 2 KHAI THÁC LUẬT KẾT HỢP BẢO TOÀN TÍNH RIÊNG TƯ

2.1 Bài toán

Gọi $I=\{I_1, I_2, \ldots, I_m\}$ là tập m thuộc tính riêng biệt, mỗi thuộc tính gọi là item. Gọi D là một tập hợp các giao tác. Mỗi giao tác T là một itemset, $T \subseteq I$. Một luật kết hợp là một quan hệ có dạng $X \Rightarrow Y$, trong đó $X \subset I$, $Y \subset I$ là các itemset, và $X \cap Y = \varnothing$. Có luật kết hợp $X \Rightarrow Y$ trên D với độ hỗ trợ (support) s và độ tin cậy c nếu:

$$s = |X \cup Y| / |D| \geq Minsup \text{ và } c = |X \cup Y| / |X| \geq Minconf$$

Trong đó *Minsup* và *Minconf* là hai giá trị ngưỡng hỗ trợ và ngưỡng tin cậy.

2.2 Các kỹ thuật khai thác luật kết hợp bảo toàn tính riêng tư

2.2.1 Kỹ thuật chỉnh sửa dữ liệu nhị phân

Đây là bài toán dùng kỹ thuật xáo trộn để thay đổi giá trị nguyên thủy của một cơ sở dữ liệu nhị phân nhằm giấu một số luật kết hợp cho trước [9]. Tác giả dựa trên một số heuristic để thay đổi giá trị của dữ liệu. Bài toán được phát biểu như sau:

Cho một cơ sở dữ liệu D và hai giá trị ngưỡng hỗ trợ và ngưỡng tin cậy *Minsup* và *Minconf*, cùng với tập luật nhạy cảm Rh. Hãy thay đổi D thành D' sao cho trên D' không thể khai thác được tập luật Rh nhưng vẫn có thể khai thác được các luật trong tập R-Rh, với R là tập hợp các luật kết hợp khai thác được từ D.

Việc giải quyết bài toán trên bằng cách thay đổi các giao tác để giảm độ hỗ trợ của những large itemset là một bài toán NP- khó. Vì vậy, giải pháp là dựa trên một số heuristic để chuyển đổi D thành D' và giữ ở mức tối đa có thể được số lượng luật khai thác được trong tập luật R – Rh.

Có hai giải pháp được đề ra. Thứ nhất là giấu những frequent itemset sinh ra các luật trong tập luật Rh (tức là giảm độ hỗ trợ của chúng). Thứ hai là giảm độ tin cậy của những luật nhạy cảm.

Có 4 chiến lược được đề ra để hiện thực hai giải pháp trên. Gọi $X \rightarrow Y$ là một luật muốn giấu, $X \rightarrow Y \in Rh$:

– Tăng độ hỗ trợ trên X cho đến khi độ tin cậy của luật bé hơn *Minsup*.

– Giảm độ hỗ trợ trên Y cho đến khi độ tin cậy hoặc độ hỗ trợ của luật bé hơn giá trị ngưỡng tương ứng.

– Giảm độ hỗ trợ trên X ∪ Y cho đến khi độ tin cậy hoặc độ hỗ trợ của luật bé hơn giá trị ngưỡng tương ứng.

– Giảm độ hỗ trợ của những itemset sinh ra luật cho đến khi chúng bé hơn *Minsup*.

Thực tế, hầu hết các thuật toán khi thực hiện giấu luật đều sinh ra hiệu ứng lề, đó là hiện tượng mất luật (lost rule) hoặc sinh ra luật mới (ghost rule). Các luật bị mất đi là do ảnh hưởng của việc thay đổi cơ sở dữ liệu D ban đầu. Các luật này lẽ ra phải vẫn còn tồn tại trong cơ sở dữ liệu D'. Các luật mới sinh ra thật sự không hiện diện trong cơ sở dữ liệu D, nhưng do thuật toán thay đổi cơ sở dữ liệu làm xuất hiện những luật không có thật này. Các heuristic được dùng để giấu Rh sẽ cân nhắc trong quá trình thay đổi dữ liệu để giảm thiểu hiệu ứng lề này.

2.2.2 Kỹ thuật thay giá trị dữ liệu bằng giá trị unknown

Việc che dấu dữ liệu, thông tin nhạy cảm rất quan trọng trong khai thác dữ liệu, có rất nhiều kỹ thuật, như thay các giá trị thật của dữ liệu bằng các dữ liệu sai (false value), hoặc kỹ thuật tạo nhiễu dữ liệu trong một số trường hợp sẽ cho ra kết quả xấu. Giả sử một tổ chức ngành y cho công khai một số dữ liệu mà trước đó đã thực hiện qua kỹ thuật thay thế giá trị thật bởi giá trị sai nhằm giấu một số thông tin riêng tư. Các nhà nghiên cứu có thể dùng dữ liệu này cho mục đích khai thác dữ liệu tìm tri thức. Tri thức không đúng tìm từ dữ liệu không chính xác gây ra hậu quả nghiêm trọng, dẫn đến việc chẩn đoán bệnh và gây nguy hiểm đến tính mạng bệnh nhân.

Như vậy, trong nhiều trường hợp, giấu dữ liệu bằng cách thay thế giá trị thật bởi giá trị "không biết" sẽ an toàn hơn là thay bởi giá trị sai. Bài toán được xác định như sau:

Cho một cơ sở dữ liệu D, hãy làm cho tập hợp các luật nhạy cảm cho trước "mờ" đi (bằng cách làm cho độ tin cậy hoặc độ hỗ trợ giảm) dùng kỹ thuật thay thế

giá trị đã biết của dữ liệu bởi giá trị "không biết" với ràng buộc là giảm thiểu hiệu ứng lề ảnh hưởng đến tập luật không nhạy cảm. Kỹ thuật này gọi là blocking.

Ta hãy xét ví dụ sau, cho cơ sở dữ liệu D và D' ứng với thời điểm trước và sau khi áp dụng kỹ thuật giấu dữ liệu như đã mô tả ở trên như sau:

Bảng 2.1 Bảng dữ liệu minh hoạ kỹ thuật Blocking

A	B	C	D
1	1	1	0
1	0	1	1
0	0	0	1
1	1	1	0
1	0	1	1

Thuật toán blocking

A	B	C	D
1	1	1	0
1	0	?	1
?	0	0	1
1	1	1	0
1	0	1	1

Trên D, sup( ) = 80%, conf( ) = 80%

Tuy nhiên trên D', 60% 80%, 60%

100%

Như vậy, khi trên cơ sở dữ liệu xuất hiện loại giá trị mới là "không biết " (dùng ký hiệu "?") thì độ hỗ trợ và độ tin cậy sẽ trở thành một khoảng (interval) chứ không còn là một giá trị cụ thể [14]:

Độ hỗ trợ của luật là khoảng: [min_sup ( ), max_sup( )].

Độ tin cậy của luật khoảng: [min_conf ( ), max_conf( )].

Trong đó:

$$\max\_\text{conf}(\text{A} \rightarrow \text{B}) = \frac{\max\_\text{sup}(A \cup B) * 100}{\min\_\text{sup}(A)}$$

$$= \frac{|(A = 1) \wedge (\text{B} = 1)| + |(A = 1) \wedge (\text{B} = ?)| + |(A = ?) \wedge (\text{B} = 1)| + |(A = ?) \wedge (\text{B} = ?)|}{|\text{D}|}$$

Khi đó, độ hỗ trợ thực sự của luật A → B sẽ là một giá trị nào đó thuộc khoảng [min_sup (A → B), max_sup(A → B)]. Tương tự đối với độ tin cậy của luật.

Khi không có giá trị "không biết" trong cơ sở dữ liệu thì giá trị cực tiểu và cực đại của độ hỗ trợ chỉ là 1 giá trị là *Minsup*. Trong quá trình thay thế giá trị thật của dữ liệu thành "?" thì giá trị cực tiểu và cực đại bắt đầu tách ra, và bằng cách này, độ không tin chắc về một luật sẽ tăng, vì thế mà luật được giấu đi. Đây là nguyên lý của thuật toán giấu luật theo kỹ thuật này.

Nhận xét rằng một itemset A vẫn còn nhạy cảm khi $\min\_\text{sup}(\text{A}) \geq Minsup$. Itemset A sẽ không còn nhạy cảm khi max_sup(A) < *Minsup*. A vẫn còn khả năng là nhạy cảm, nhưng không chắc mấy, khi $\min\_\text{sup}(\text{A}) \leq Minsup \leq \max\_\text{sup}(A)$. Ta quan tâm đến khả năng thứ 3 này. Một cách tương tự khi xét đến độ tin cậy của luật. Như vậy, ta giấu luật bằng cách thay đổi dữ liệu (bởi giá trị "?") để giảm độ hỗ trợ tối thiểu hoặc giảm độ tin cậy tối thiểu cho đến khi bé hơn giá trị ngưỡng tương ứng. Tuy nhiên, vấn đề đặt ra là giảm bao nhiêu thì an toàn? Tác giả của [14] đề nghị khoảng an toàn SM (safety margin) và dùng SM như là một tham số điều khiển quá trình thay đổi cơ sở dữ liệu để giấu luật.

Từ các điều trên, ta có giải pháp như sau:

1. Giấu luật A → B bằng cách giảm min_sup (A ∪ B) cho đến khi min_sup (A ∪ B) bé hơn *Minsup* một khoảng an toàn SM (tức là min_sup (A ∪ B) < Minsup - SM). Để đạt được điều này, ta thay 1 bởi "?" ở các item thuộc (A ∪ B), cho đến khi min_sup (A ∪ B) < Minsup – SM. Khi đó maxsup (A ∪ B) sẽ không đổi. Đây là ý tưởng của thuật toán GIH.
2. Giấu luật A → B bằng cách giảm min_conf (A → B) cho đến khi min_conf (A ∪ B) < Minconf một khoảng an toàn SM (tức là min_conf (A ∪ B) < Minconf - SM), nghĩa là ta sẽ thay 1 và 0 bởi "?" để giảm min_conf (A → B) cho đến khi

min_conf (A ∪ B) < Minconf – SM. Để đạt được điều này, ta giảm min_sup (A ∪ B) và/hoặc tăng max_sup (A).

- Để giảm min_sup (A ∪ B), ta thay 1 bởi "?" ở các item thuộc A hoặc B. Tuy nhiên, nếu thay tại A thì min_sup (A) cũng giảm nên kéo theo sẽ làm tăng max_conf (A → B) (vì max_conf (A → B) = max_sup (A ∪ B)*100/min(A)). Nhưng khi giấu luật, max_conf của luật càng nhỏ càng tốt nên tốt hơn hết ta sẽ thực hiện thay 1 bởi "?" trên các thuộc tính tại B. Đây là ý tưởng của thuật toán CR.
- Để tăng max_sup (A), ta thay 0 bởi "?" tại các item thuộc A. Đây là ý tưởng của thuật toán CR2.

Thuật toán GIH:

INPUT: Cơ sở dữ liệu D, Minsup, SM, tập hợp L các large itemset, Lh là tập hợp các large itemset cần giấu.
OUTPUT: D' thoả điều kiện không thể tìm thấy Lh.

Begin

1.Sắp xếp Lh giảm dần theo kích thước và giảm dần theo độ hỗ trợ tối thiểu.
Foreach Z **in** Lh
{
2.Sắp xếp các giao tác hỗ trợ Z tăng dần theo kích thước giao tác.
3.Số_lần_lặp = |TZ| - (Minsup-SM)* |D|
For k =1 **to** số_lần_lặp **do**
{
4.Thay 1 bởi dấu "?" tại các item hỗ trợ Z ít nhất trong giao tác kế tiếp.
5.Cập nhật độ hỗ trợ của những itemset liên quan.
6.Cập nhật D.
}
}

End

Hình 2.1 Mã giả thuật toán GIH [1]

Thuật toán CR:

```
INPUT: CSDL nguồn D, Minconf, Minsup, SM, tập luật Rh cần giấu.
OUTPUT: CSDL D' do thay đổi D thỏa điều kiện không tìm thấy tập luật Rh.
Begin
        Foreach rule r in Rh do
        {
        1.Tr = {giao tác t thuộc D, t hỗ trợ toàn phần r}
        2.For each t in Tr
          Đếm số item trong t
        3.Sắp xếp Tr theo thứ tự tăng dần về số item.
          Repeat until (min_conf(r) < Minconf - SM)
        {
        4.Chọn giao tác đầu tiên t ∈ Tr.
        5.Chọn item j thuộc vế phải của r là item hỗ trợ vế phải của r nhiều nhất.
        6.Thay 1 bởi dấu "?" tại j của t.
        7.Tính lại minsup (r).
        8.Tính lại minconf (r).
        9.Tính lại minconf của những luật khác bị ảnh hưởng.
        10.Xóa t khỏi Tr.
        }
        11.Xóa r khỏi Rh .
        }
End
```

Hình 2.2 Mã giả cho thuật toán CR [1]

Thuật toán CR2:

INPUT: CSDL nguồn D, Minconf, Minsup, SM, tập luật Rh cần giấu.
OUTPUT: CSDL D' do thay đổi D thỏa điều kiện không tìm thấy tập luật Rh.
Begin
Foreach rule r in Rh do
{
1.T'lr = {giao tác t thuộc D, t hỗ trợ một phần vế trái r và t không hỗ trợ toàn phần vế phải r}
2.For each t in T'lr
Đếm số item thuộc vế trái của r trong t
3.Sắp xếp T'lr theo thứ tự giảm dần về số item đếm được ở bước 2.
Repeat until (min_conf(r) <Minconf – SM or min_sup (r) < Minsup – SM)
{
4.Chọn giao tác đầu tiên t ∈ T′lr.
5.Trong t, thay 0 bởi dấu "?" tại các item thuộc vế trái của r.
6.Tính lại max_sup cho vế trái của luật r.
7.Tính lại min_conf (r).
8.Tính lại min_conf của những luật khác bị ảnh hưởng.
9.Xóa t khỏi T'lr.
}
10.Xóa r khỏi Rh .
}
End

Hình 2.3 Mã giả cho thuật toán CR2 [1]

Kỹ thuật blocking có thể ngăn cản quá trình khai thác cho ra những tri thức nhạy cảm, mà lại không sinh ra những tri thức sai gây ảnh hưởng đến người dùng dữ liệu.

Về độ an toàn: Nếu vận dụng riêng lẻ từng thuật toán, thì đối phương có thể tái tạo lại cơ sở dữ liệu D từ D' một cách dễ dàng bằng cách thay "?" bởi "1" ở 2 thuật toán đầu hoặc thay "?" bởi "0" bởi thuật toán sau cùng. Vì vậy, tác giả đã đề nghị vận dụng phối hợp 3 thuật toán trên theo cách chọn thuật toán khác nhau khi giấu các luật khác nhau. Bằng cách này, tác giả đã phân tích khả năng đối phương phục hồi lại D từ D' là không thể [14].

2.2.3 Phương pháp tái tạo

Từng bộ dữ liệu đươc xem như là một vector ngẫu nhiên:

$$X = \{X_i\} \text{ với } X_i = 0 \text{ hoặc } X_i = 1$$

Theo phương pháp mà Shariq J. Rizvi [10] đề nghị, X bị làm nhiễu theo cách sau:

$$\mathrm{Y} = \mathrm{distort(X)}$$

Trong đó:

$$Y_i = X_i \; XOR \; \overline{r_i}$$

$\overline{r_i}$ là phần bù của r_i là một biến ngẫu nhiên có hàm mật độ $f(r) = \mathrm{bernoulli(p)}(0 \leq \mathrm{p} \leq 1)$. Như vậy r_i có giá trị là một với xác suất p và có giá trị là 0 với xác suất (1-p), và item thứ i của X giữ nguyên giá trị với xác suất p và bị thay đổi giá trị với xác suất (1-p).

Xác suất tái tạo lại giá trị 1 của cơ sở dữ liệu D ban đầu:

Gọi $\mathrm{s_i}$ là độ hỗ trợ thật sự của item I, tức là xác suất một khách hàng ngẫu nhiên mua một item thứ i là $\mathrm{s_i}$. Tác giả tính xác suất tái tạo lại giá trị 1 đối với item thứ i là:

$$R_1(p, s_i) = \frac{s_i \times p^2}{s_i \times p + (1 - s_i) \times (1 - p)} + \frac{s_i \times (1 - p)^2}{s_i \times (1 - p) + (1 - s_i) \times p}$$

Biểu thức trên tính xác suất tái tạo lại giá trị "1" đối với một item ngẫu nhiên i. Một cách tổng quát, xác suất tái tạo lại giá trị 1 là:

$$\mathrm{R_1(p)} = \frac{\sum_i \mathrm{s_i R_1(p, s_i)}}{\sum_i \mathrm{s_i}}$$

Biểu thức trên đạt giá trị cực tiểu khi tất cả các item trong cơ sở dữ liệu có cùng độ hỗ trợ, và tăng khi độ hỗ trợ của các item khác biệt nhiều. Thay $\mathrm{s_i}$ bởi $\mathrm{s_0}$ với $\mathrm{s_0}$ là độ hỗ trợ trung bình của một item trong cơ sở dữ liệu. Khi đó, xác suất tái tạo lại giá trị 1 là:

$$\mathrm{R_1(p)} = \frac{\mathrm{s_0} \times \mathrm{p^2}}{\mathrm{s_0} \times \mathrm{p} + (1 - \mathrm{s_0}) \times (1 - \mathrm{p})} + \frac{\mathrm{s_0} \times (1 - \mathrm{p})^2}{\mathrm{s_0} \times (1 - \mathrm{p}) + (1 - \mathrm{s_0}) \times \mathrm{p}}$$

Tương tự xác suất tái tạo giá trị 0 là:

$$\mathrm{R_0(p)} = \frac{(1 - \mathrm{s_0}) \times \mathrm{p^2}}{(1 - \mathrm{s_0}) \times \mathrm{p} + \mathrm{s_0} \times (1 - \mathrm{p})} + \frac{(1 - \mathrm{s_0}) \times (1 - \mathrm{p^2})}{\mathrm{s_0} \times \mathrm{p} + (1 - \mathrm{s_0}) \times (1 - \mathrm{p})}$$

Cho a là trọng số cho biết giá trị 1 quan trọng hơn giá trị 0 (một khách hàng muốn bảo vệ cả giá trị 1 lẫn 0 trong giao tác của họ, nhưng họ thường cho rằng

những giá trị 1 thì nhạy cảm và cần được bảo đảm riêng tư hơn là giá trị 0), một cách tổng quát, xác suất tái tạo được tính như sau:

$$R_p = aR_1(p) + (1-a)R_0(p)$$

Quá trình khai thác trên cơ sở dữ liệu được làm nhiễu được thực hiện như sau:

- Tính độ hỗ trợ cho 1-itemset:

  + Gọi T là ma trận dữ liệu thật ban đầu.

  + D là ma trận sau khi được làm nhiễu với xác suất p.

  + Xét 1 item ngẫu nhiên i.

  +Gọi c_1^T là giá trị 1 tại cột i của T, c_0^T có gí trị 0 tại cột i trong T. Tương tự, gọi c_1^D là giá trị 1 tại cột i của D, c_0^D có gí trị 0 tại cột i trong D. Công thức sau được dùng để ước lượng độ hỗ trợ của item i trong T:

$$C^T = M^{-1}C^D$$

Trong đó:

$$M = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}$$

$$C^D = \begin{bmatrix} c_1^D \\ c_0^D \end{bmatrix} \qquad C^T = \begin{bmatrix} c_1^T \\ c_0^T \end{bmatrix}$$

Ma trận M có được là do phương pháp làm nhiễu theo nguyên tắc, nếu một cột có n giá trị 1 trong T thì tương ứng sẽ được đoán chừng là pn giá trị 1 và (1-p)n giá trị 0 cho cùng cột đó trong D. Một cách tương tự đối với cột 0 trong T. Như vậy cho trước c_1^Dvà c_0^D, hoàn toàn có thể tính được giá trị của c_1^T, là độ hỗ trợ của item i.

- Ước lượng độ hỗ trợ của n-itemset:

Tổng quát hóa cho trường hợp 1-itemset, các ma trận tương ứng được định nghĩa như sau:

$$C^D = \begin{bmatrix} c_{2^n-1}^D \\ . \\ . \\ . \\ c_1^D \\ c_0^D \end{bmatrix} \qquad C^T = \begin{bmatrix} c_{2^n-1}^T \\ . \\ . \\ . \\ c_1^T \\ c_0^T \end{bmatrix}$$

Trong đó c_k^T là số bộ trong T có dạng nhị phân của k (tính trên n ký số) của 1 itemset, ví dụ c_3^T cho biết số bộ "11" trong T. Tương tự đối với c_D^T.

Ma trận M được định nghĩa như sau:

m_{ij} =xác suất 1 bộ có dạng c_j^T trong T, tương ứng thành một bộ có dạng c_i^D trong D. Ví dụ, ta có $m_{1,2}$ = đối với 2-itemset là xác suất bộ có dạng "10" được làm nhiễu thành bộ "01". $m_{1,2} = (1-p)(1-p)$. Điều này có được là do phương pháp làm nhiễu đã mô tả đã gây nhiễu từng cột trong n-itemset một cách độc lập.

2.3 Thuật toán MASK.

2.3.1 Tình hình nghiên cứu

Việc khai thác luật kết hợp trong khai thác dữ liệu bảo toàn tính riêng tư đã đạt được nhiều tiến bộ đáng kể.

Việc xáo trộn dữ liệu và hạn chế truy vấn là hai phương pháp cơ bản trong quá trình khai thác các dữ liệu cần bảo mật. Với sự hỗ trợ của phương pháp gây nhiễu dữ liệu, dữ liệu thô trước hết bị xáo trộn bởi việc chuyển đổi dữ liệu, rời rạc hoá và tăng thêm điều kiện nhiễu trong dữ liệu, sau đó đạt được mô hình dữ liệu mong muốn, kiến thức và các quy tắc tìm được [12].

Thuật toán MASK (liên kết khai thác bảo mật Konstraints) được đưa ra lần đầu bởi Rizvi và cộng sự năm 2002 [10]. Sử dụng các phương pháp xáo trộn ngẫu nhiên và tái phân phối đảm bảo tính riêng tư trong khai thác luật kết hợp.

2.3.2 Thuật toán MASK

Bằng cách sử dụng phương pháp ngẫu nhiên, dữ liệu gốc bị bóp méo trong thuật toán MASK. Với sự giúp đỡ của các phương pháp hỗ trợ tái tạo, các tập hạng mục phổ biến có thể thu được từ các dữ liệu bị bóp méo.

Giả sử rằng bộ cơ sở dữ liệu được tạo ra bởi các giá trị 0 và giá trị 1(giá trị 1 chỉ sự tồn tại của thuộc tính, thuộc tính không tồn tại khi giá trị là 0), xác suất mà mỗi mục dữ liệu lưu trữ với giá trị ban đầu là p, và ngược lại thì xác suất là 1-p. Tất cả các cơ sở dữ liệu sẽ được biến đổi theo cách tương tự để tạo thành một cơ sở dữ liệu mới, và việc khai thác sẽ được tiến hành trên cơ sở dữ liệu đó.

Tập hợp dữ liệu có thể được xem như là một ma trận Boolean. Với T là ma trận dữ liệu ban đầu, và D là ma trận sau khi được làm nhiễu với xác suất p. Các số

'1' và '0' bao gồm trong ith(cột thứ i) của T được định nghĩa tương ứng là c_1^T, c_0^T. Và ma trận dữ liệu D cũng được định nghĩa tương tự c_1^D, c_0^D.

Qua trình biến đổi:

$$c_1^T \times p + c_0^T \times (1-p) = c_1^D;\ c_0^T \times p + c_1^T \times (1-p) = c_0^D$$

Công thức sau được dùng để ước lượng độ hỗ trợ của item i trong T:

$$C^T = M^{-1} C^D$$

Trong đó:

$$M = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix} \quad C^D = \begin{bmatrix} c_1^D \\ c_0^D \end{bmatrix} \quad C^T = \begin{bmatrix} c_1^T \\ c_0^T \end{bmatrix}$$

Khi mỗi mục bị xáo trộn theo cách thức tương tự, dễ dàng ta có công thức tính độ hỗ trợ của n-itemset:

$$C^T = M^{-1} C^D$$

Trong đó:

$$M = \begin{bmatrix} m_{0,0} & m_{0,1} & \cdots & m_{0,2^n-1} \\ m_{1,0} & m_{1,1} & \cdots & m_{1,2^n-1} \\ \vdots & \vdots & \cdots & \vdots \\ m_{2^n-1,0} & m_{2^n-1,1} & \cdots & m_{2^n-1,2^n-1} \end{bmatrix}$$

$$C^D = \begin{bmatrix} c_{2^n-1}^D \\ . \\ . \\ . \\ c_1^D \\ c_0^D \end{bmatrix} \qquad C^T = \begin{bmatrix} c_{2^n-1}^T \\ . \\ . \\ . \\ c_1^T \\ c_0^T \end{bmatrix}$$

Số lượng các bộ trong T được thể hiện là c_k^T, trong đó T có dạng nhị phân của k đối với các tập phổ biến nhất định. Ví dụ, đối với 2-itemset, c_1^T có nghĩ là số '01', c_2^T có nghĩa là số '10',… Trong ma trận được biến đổi D, định nghĩa cho c_k^D tương tự như c_k^T.

Trong ma trận M, m_{ij} =xác suất 1 bộ có dạng c_j^T trong T, tương ứng thành một bộ có dạng c_i^D trong D . Xét đối với 2-itemset, M là ma trận thứ tư, $m_{1,3}$ là xác suất mà bộ '11' thay đổi sang bộ '01'. Được xác định bằng $p \times (1-p)$.

2.3.3 Một số biến thể của thuật toán MASK và hạn chế

EMASK(Efficient MASK), nâng cao hiệu quả về thơi gian so với thuật toán MASK, được đề xuất bởi Agrawal và cá đồng tác giả năm 2004 [7]. Nhưng EMASK không phá vỡ được sự phức tạp hàm mũ trong việc tái xây dựng hỗ trợ ban đầu.

MMASK (Modified MASK), được được ra bởi Andruszkiewcz vào năm 2007 [4], giảm thời gian và độ phức tạp hơn EMASK.

2.4 Lý thuyết giàn và ứng dụng trong thuật toán ẩn tập mục nhạy cảm [2]

2.4.1 Phát biểu bài toán

Cho một bảng trị T 0/1 gồm N dòng và M cột. Các cột được gán tên lần lượt A, B, C,… lấy từ một tập hữu hạn các phần tử U. Mỗi phần tử trong U gọi là một mục, mỗi tập con X của U gọi là một tập mục. Mỗi dòng t của bảng T dược gọi là một giao tác. Theo truyền thống của lý thuyết khai thác tri thức ta ký hiệu tập mục như một dãy các kí tự viết liền nhau, hợp của hai tập mục X và Y được kí hiệu là XY. Với mỗi giao tác $t \in T$ và mỗi mục $A \in U$ ta kí hiệu t.A là trị tương ứng xuất hiện trên giao của dòng t và cột A trong bảng T. Như vậy t.A $\in \{0,1\}$. Ta định nghĩa Set(t) là tập mục tại đó t nhận trị 1, $Set(t) = \{A \in U \mid t.A = 1\}$. Nếu $X \subseteq Set(t)$ thì ta nói giao tác t chứa tập mục X. Với mỗi tập mục $X \subseteq U$ ta xác định $\alpha(X)$ là số lượng giao tác chứa X, $\alpha(X) = ||\{t \in T \mid X \subseteq Set(t)\}||$, trong đó kí hiệu $||M||$ cho biết lực lượng (số phần tử) của tập M. Tỷ số $\alpha(X)/N$ được gọi là độ hỗ trợ của tập mục X. Với N cho trước và cố định, ta có thể coi $\alpha(X)$ là độ hỗ trợ của tập mục X. Cho trước giá trị σ và gọi là *ngưỡng hỗ trợ.* Các tập mục X thỏa tính chất $\alpha(X) > \sigma$ được gọi là các *tập mục thường xuyên*.

Ví dụ 2.1: Cho **Bảng 2.2** tâp giao tác T gồm 22 giao tác và ngưỡng hỗ trợ $\sigma = 4$. Khi đó họ các tập mục thường xuyên trong **Bảng 2.3** sẽ bao gồm $P = \{A/10, B/11, C/8, D/12, E/18, AB/4, AD/4, AE/10, BE/8, CE/7, DE/9, ABE/4, ADE/4\}$, trong đó số viết kèm tập mục là độ hỗ trợ của mục đó.

Bảng 2.2 Tập giao tác T

ABCDE	ABCDE
1. 10101	12. 01011
2. 10101	13. 00110
3. 11001	14. 10011
4. 11001	15. 01101
5. 01010	16. 11001
6. 00111	17. 11001
7. 10011	18. 01010
8. 01101	19. 00111
9. 00011	20. 10011
10. 01000	21. 01101
11. 10011	22. 00011

Bảng 2.3 Bảng dữ liệu P

	α
A	10
B	11
C	8
D	12
E	18
AB	4
AD	4
AE	10
BE	8
CE	7
DE	9
ABE	4
ADE	4

Bài toán ẩn tập mục nhạy cảm Cho bảng T gồm N giao tác trên M mục. Cho ngưỡng hỗ trợ σ và danh sách P các tập mục thường xuyên theo ngưỡng σ. Cho tập

mục nhạy cảm H ∈ P. Yêu cầu: Ẩn tập mục nhạy cảm H theo nghĩa sau: Cần chỉ ra các vị trí cần sửa dữ liệu trên bảng T sao cho $\alpha(X) < \sigma$ và các tập mục thường xuyên khác bị ảnh hưởng ít nhất.

Để ẩn một tập mục nhạy cảm H ta cần tìm cách giảm đỗ hỗ trợ của H xuống dưới ngưỡng σ, chẳng hạn ta sẽ sửa bảng T để $\alpha(H) = \sigma - 1$. Để sửa một trị trong bảng *T* ta cần chỉ ra giao tác *t* (dòng) và mục *C* (cột) và sửa giá trị tại đó từ 1 thành 0. Điều đó có nghĩa là giảm độ hỗ trợ (số lần xuất hiện) của mục C 1 đơn vị. Việc làm này kéo theo hệ quả là giảm độ hỗ trợ của tập mục *H* 1 đơn vị. Tổng quát, nếu $A \in X$, ta kí hiệu *Update*(*A,X,d*) là thao tác sửa *d* lần (từ 1 thành 0) tại mục (cột) *A* trên các giao tác (dòng) chứa tập mục *X* . Việc chọn mục cần sửa là điều quan trọng. Giả sử với ngưỡng hỗ trợ $\sigma = 4$ và cần ẩn tập mục *ADE*. Ta thấy $\alpha(ADE) = 4$ nên *ADE* sẽ bị ẩn nếu ta giảm độ hỗ trợ của *ADE* xuống dưới ngưỡng σ, cụ thể là ta sẽ sửa bảng *T* để $\alpha(ADE) = \sigma - 1 = 3$. Ta chọn giao tác 7 và sửa vị trí A trên dòng này từ 1 thành 0 Ta có ngay $\alpha(ADE) = 3$. Tuy nhiên, khi đó $\alpha(AD) = 3$, tức là *AD* đang là tập mục thường xuyên trở thành tập mục không thường xuyên.

Phần tiếp theo sẽ nêu cơ sở lý thuyết và giải trình thuật toán nhằm chỉ ra rằng nếu sửa mục E trên giao tác 7 thì ADE sẽ bị ẩn và các tập mục thườg xuyên còn lại sẽ được bảo lưu.

2.4.2 Lý thuyết giàn giao

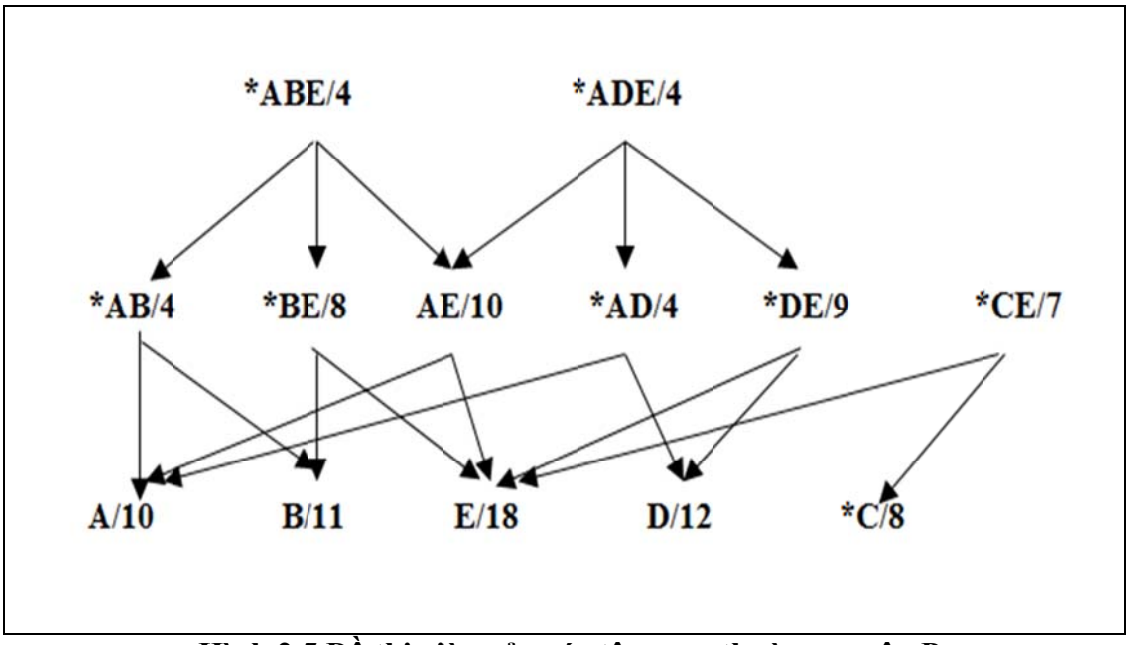
Trong phần này sẽ nhắc đến một số khái niệm và kết quả lien quan đến giàn giao.

Cho tập hữu hạn U gọi là tập nền, ta kí hiệu Poset(U) là họ toàn thể các tập con của U với thứ tự bộ phận là phép bao hàm ⊆, Poset'(U) = Poset(U) – {U}. Một giàn giao G là một họ các tập con của U đóng với phép giao, cụ thể là, nếu G = $\{V_1, V_2, \ldots, V_k \mid V_i \in \text{Poset}(U), i = 1,2,\ldots,k\}$ thì $\forall V_i, V_j \in G: V_i \cap V_j \in G$. Khi đó G chứa duy nhất một họ con S sao cho mọi phần tử của G đều được biểu diễn qua giao của các phần tử trong S, cụ thể là, S là tập con nhỏ nhất của G thỏa tính chất G = $\{Y \mid Y = X_1 \cap \ldots \cap X_k, k \geq 0, X_1, \ldots, X_k \in S\}$. S được gọi là tập sinh của giàn G và được ký hiệu là Gen(G). Theo quy ước, giao của một họ rỗng các tập con chính là U, do đó mọi Gen đều không chứa U.

```
Algroithm Gen
Input     - Giàn giao G
Output    - Tập sinh Gen(G)
Method
            1. Xây dựng  đồ thị có hướng          trong đó G là tập đỉnh.
               Cung         khi và chỉ khi u là cha trực tiếp của v.
            2. Return                 \\ d(g) là bậc của đỉnh g, tức là
               số cung  đến đỉnh g.
End Gen
```

Hình 2.4 Thuật toán tìm tập sinh của giàn giao [3]

Ta có ví dụ từ **Hình 2.5,** đồ thị của giàn các tập mục thường xuyên P = {A/10, B/11, C/8, D/12, E/18, AB/4, AD/4, AE/10, BE/8, CE/7, DE/9, ABE/4, DE/4}. Các phần tử Gen có dấu *: Gen = {ABE, ADE, AB, BE, AD, DE, CE, C} .



Hình 2.5 Đồ thị giàn của các tập mục thường xuyên P

Ta có **Bảng 2.4** là bảng dữ liệu của tập mục thường xuyên P.

Bảng 2.4 Bảng dữ liệu của tập mục thường xuyên P

	α	d
A	10	5
B	11	3
**C*	8	1
**D*	12	3
E	18	6
**AB*	4	1
**AD*	4	1
AE	10	2
**BE*	8	1
**CE*	7	0
**DE*	9	1
**ABE*	4	0
**ADE*	4	0

Cho (M, ≤) là một tập hữu hạn có thứ tự bộ phận. Phần tử m trong M được gọi là cực đại nếu từ m ≤ x và x∈M ta luôn có m=x. Ta ký hiệu MAX(M) là tập các phần tử cực đại của M. Dễ thấy rằng, với mỗi phần tử x trong M, luôn tồn tại một phần tử m trong MAX(M) thỏa x ≤ m. Với mỗi họ các tập con của một tập hữu hạn U cho trước ta xét thứ tự bộ phận ⊆. Cho G là một giàn giao trên tập hữu hạn U. Ta ký hiệu Coatom(G) = MAX(G – {U}) và gọi các phần tử trong Coatom(G) là đối nguyên tử của giàn giao G.Trong [2] phát biểu và chứng minh các kết quả sau.

Mệnh đề 1: Với mọi giàn giao G trên tập hữu hạn U, ta có:

$$MAX(Gen(G)) = MAX(G - \{U\}) = Coatom(G)$$

2.4.3 Các tính chất của tập mục thường xuyên

Cho bảng *T* gồm *N* giao tác trên tập mục nền *U*, *P* là họ các tập mục thường xuyên theo ngưỡng σ cho trước. Trước hết ta nhận xét rằng, nếu $X \subseteq Y \subseteq U$ thì $\alpha(X) \geq \alpha(Y)$. Hệ thức này thể hiện tính nghịch biến của hàm đo độ hỗ trợ α.

Mệnh đề 2: *P là một giàn giao.*

Chứng minh

Giả sử $X, Y \in P$, $Z = X \cap Y$. Ta có $Z \subseteq X$, do đó $\alpha(Z) \geq \alpha(X) \geq \sigma$. Vậy $Z \in P$

Mệnh đề 2 cho phép chúng ta vận dụng các tính chất của giàn giao trong xử lý các tập mục thường xuyên. Cụ thể là khi cần ẩn tập mục nhạy cảm H ta sẽ sửa các tập mục lớn nhất chứa H trong giàn giao P, tức là các Coatom chứa H.

Mệnh đề 3: Với mỗi tập thường xuyên X trong P, $Poset(X) \subseteq P$ và là một giàn giao đầy đủ với tập Gen gồm các phần tử trên hàng thứ 2.

Ta có thuật toán thuật toán xác định tập sinh trong giàn giao đầy đủ Gen(X):

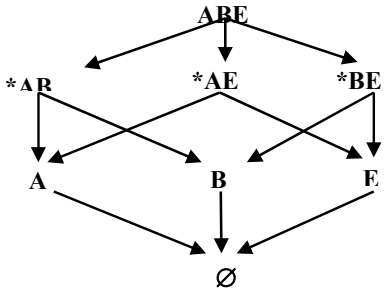
```
Algorithm Gen
Input:            Tập mục X ⊆ U
Output:     Họ các tập sinh G của giàn giao đầy đủ X
Method
      G = ∅
      For each item A in X do
            Add X – {A} to G;
      End for
      Return G;
End Gen
```

Hình 2.6 Thuật toán xác định tập sinh trong giàn giao đầy đủ Gen(X)

Chứng minh:

Giả sử $X \in P$ và $Y \subseteq X$. Ta có ngay $\alpha(Y) \geq \alpha(X) \geq \sigma$. Từ đây suy ra $Y \in P$, nghĩa là mọi tập con của X đều là tập mục thường xuyên. Do *Poset*(X) chứa mọi tập con của X nên *Poset*(X) là đầy đủ và đương nhiên đóng với phép giao. Theo mệnh đề 1 ta thấy với mọi mục $A \in X$, $X-\{A\}$ chỉ khuyết duy nhất một phần tử, do đó chúng có duy nhất một cha. Mọi tập con còn lại trong *Poset*(X) đều khuyết từ hai phần tử trở lên do đó chúng có ít nhất là hai cha. Vậy *Gen*(X) bao gồm các phần tử đứng trên hàng thứ hai trên đồ thị biểu diễn giàn đã cho.

Do *Gen*(X) chỉ gồm các tập con khuyết 1 phần tử của X nên thuật toán xác định *Gen*(X) khá đơn giản. Ta có ví dụ trong **Hình 2.7**



Hình 2.7 Giàn giao đầy đủ Poset(ABE)

Mệnh đề 3 và tính chất nghịch biến của hàm α cho ta thấy rằng các phần tử trong *Gen*(*X*) có độ hỗ trợ nhỏ nhất trong *Poset*(*X*)–{*X*}. Nếu $X \in P$, với mỗi mục *A* trong *X* ta xét hàm *L*(*A,X*) cho giá trị là cặp *Y*/λ trong đó λ là giá trị nhỏ nhất trong số các độ hỗ trợ của các tập con đúng *Y* chứa *A* của *X* (tức là $Y \subseteq X$, $Y \neq X$ và $A \in Y$),

$$L(A,X) = Y/\lambda\ ,\ \lambda = min\ \{\alpha(Y) \mid A \in Y,\ Y \subset X\}$$

Dựa vào nhận xét trên ta thấy có thể tính *L*(*A,X*) thông qua các tập chứa *A* trong *Gen*(*X*).

Mệnh đề 4: Nếu tập mục thường xuyên X bị ẩn thì mọi tập mục thường xuyên Y chứa X cũng bị ẩn theo.

Chứng minh:

Nếu *X* bị ẩn thì $\alpha(X) < \sigma$. Nếu $X \subseteq Y$, thì $\alpha(Y) \leq \alpha(X) < \sigma$, nghĩa là *Y* cũng bị ẩn theo.

Mệnh đề 5: Nếu $X \in P$ thì mọi Update(A,X,d), $A \in X$ đều kéo theo Update(A,Y,d), $Y \subseteq X$, $A \in Y$. tức là độ hỗ trợ của mọi tập con chứa A của X đều bị giảm d đơn vị.

Chứng minh:

Chúng ta thấy rằng thao tác *Update*(*A,X,1*) sẽ xóa một xuất hiện của mục *A* trong tập mục *X*, nghĩa là thay *X* bằng *X*–{*A*}. Từ đó suy ra rằng mọi tập mục con chứa *A* của *X* cũng sẽ giảm số lần xuất hiện 1 đơn vị

Các tập mục *Y* như mô tả trong mệnh đề 5 được gọi là các tập mục *chịu hiệu ứng phụ* khi cập nhật (xóa) mục *A* tập mục *X*. Mệnh đề này cho thấy nếu cập nhật mục *A* trong tập mục *X* thì cần chú ý đến các tập mục con đúng và chứa *A* của *X*.

Nếu độ hỗ trợ của chúng lớn hơn ngưỡng σ không nhiều thì chúng sẽ có nguy cơ bị ẩn theo.

2.4.4 Thuật toán ẩn tập mục nhạy cảm

Từ các mệnh đề từ 1 đến 5, trong phần này trình bày thuật toán ẩn một tập mục nhạy cảm H.

Cho bảng T gồm N giao tác trên M mục, cho ngưỡng hỗ trợ σ và giả thiết rằng ta đã xây dựng được họ các tập mục thường xuyên P. Cho tập mục nhạy cảm $H \in P$. Khi đó thuật toán ẩn tập mục H được thực hiện qua các bước sau đây.

Bước 1: Xác định họ V các tập mục chứa H trong $Coatom(P)$,

$$V = \{X \in Coatom(P) \mid H \subseteq X\}$$

Bước 2: Với mỗi mục $A \in H$ và với mỗi tập mục X trong V lượng giá xem có nên sửa mục A trong X không? Tiêu chuẩn đặt ra là việc sửa mục A trong X không gây hiệu ứng phụ đến các tập con đúng chứa A của X. Gọi $M(H)$ là hàm cho giá trị là bộ tứ (A, X, Z, μ) trong đó μ là giá trị lớn nhất trong số các độ hỗ trợ λ tìm được qua các hàm $L(A,X)$, cụ thể là, $M(H) = (A, X, Z, \mu)$, $\mu = max\ \{\lambda \mid L(A,X) = Z/\lambda, A \in H, X \in V\}$. Ta gọi thủ tục $Update(A, X, d)$, với $d = min\ \{\alpha(H) - (\sigma - 1), \alpha(Z) - \sigma\}$

Bước 2 sẽ được lặp đến khi $\alpha(H) = \sigma - 1$.

```
Algorithm Itemhide
Input:
   -  Bảng trị 0/1 thể hiện các giao tác trên tập mục U.
   -  σ:  ngưỡng hỗ trợ.
   -  P: Họ các tập mục thường xuyên theo ngưỡng σ.
   -  H: Tập mục nhạy cảm cần ẩn, H ∈ P.
Output: Bảng kết quả T.
     s = α(H);
     while (s > σ − 1) do
               compute  V = {X ∈ Coatom(P)|H ⊆ X};
               let (A,X,μ) = M(H);
               update (A,X,d);
               s = s-d;
     End While
End Itemhide.
```

Hình 2.8 Thuật toán ẩn tập mục nhạy cảm.

Ví dụ 2.2: Với dữ liệu trong ví dụ 2.1, ta chọn ngưỡng hỗ trợ $\sigma = 4$, họ các tập mục thường xuyên với độ hỗ trợ tương ứng tạo thành giàn P = {ABE/4, ADE/4, AB/4, BE/8, AE/10, AD/4, DE/9, CE/7, A/10, B/11, C/8, D12, E/18} trên tập nền U = ABCDE.

Giả sử ta cần ẩn tập mục nhạy cảm *H = AE/10.*

Qua **Hình 2.5** bên trên , ta thấy rằng sau khi ẩn AE thì hai tập mục ABE và ADE sẽ bị ẩn theo là đúng theo mong muốn của thuật toán..

Như vậy kết quả dự kiến sẽ phải là P = {AB, BE, AD, DE, CE, A, B, C, D, E}. Ta có, Gen(P)={ABE, ADE, AB, BE, AD, DE, CE, AB, C}, Coatom(P) = MAX(Gen(P)) = {ABE, ADE, CE}, V = {ABE, ADE}, s = α(H) = 10. Ta lần lượt xét các tập mục trong V.

Ta xét tập mục *ABE:*

Với $A \in H$ ta có *L*(*A, ABE*) = *min*{*AB/4, AE/10*} = *AB/4*,

Với $E \in H$ ta có *L*(*E, ABE*) = *min*{*AE/10, BE/8*} = *BE/8*.

Tương tự, xét tập mục *ADE:*

L(*A, ADE*) = *min*{*AE/10, AD/4*} = *AD/4*,

L(*E, ADE*) = *min*{*AE/10, DE/9*} = *DE/9.*

Tổng hợp 4 trường hợp ta thu được *max* {*AB/4, BE/8, AD/4, DE/9*} = *DE/9.* Như vậy *M*(*H*) =*M*(*AE*) = (*E, ADE,DE,9*) .

Kết luận: cần cập nhật mục *E* trong tập mục *ADE.* Ta có *d* = *min* {7, 4, 5} = 4.

Thủ tục Update(E,ADE,4) sẽ cho ta kết quả ADE/0 (bị ẩn), AE/6, DE/5.

Tiếp tục, do α(*ADE*) = 0 < σ = 4 nên tập mục này bị ẩn, ta loại ra khỏi *P.* Ta tính lại *Coatom*(*P*) = {*ABE/4*, *AD/4, DE/5, CE/7*}, do đó *V* = {*ABE*}. Ta có

L(*A, ABE*) = *min*{*AB/4, AE/6*} = *AB/4; L*(*E, ABE*) = *min*{*AE/6, BE/8*} = *AE/6.*

max {*AB/4, AE/6*} = *AE/6*, do đó *M*(*H*) = *M*(*AE*) = (*E, ABE, AE, 6*). Ta cập nhật *E* trong *ABE.* Thủ tục *Update*(*E,ABE,3*) cho ta thêm hai tập mục bị ẩn và do đó chúng bị loại khỏi danh mục các tập mục thường xuyên là ABE/1, AE/3. Ta nhận được P = {AB/4, BE/5, AD/4, DE/5, CE/7, A/10, B/11, C/8, D/12, E/11}.

Qua thuật toán ẩn tập mục nhạy cảm, chúng ta thấy phát sinh một vấn đề, khi ẩn một tập mục nhạy cảm, trong qua trình khai thác sẽ dẫn đến ẩn các tập mục có liên quan đến tập mục cần ẩn, như trong Hình 2.5 khi chúng ta tiến hành ẩn tập mục AE sẽ dẫn đến ẩn tập mục ADE và ABE. Nhưng trong một số trường hợp, việc ẩn kéo theo sẽ dẫn đến làm mất đi các tập mục có ích không cần ẩn đi, và sẽ dẫn đến mất các luật quan trọng trong quá trình khai thác.

Do đó, trong phần tiếp theo sẽ giới thiệu một thuật toán, chỉ tiến hành giấu (ẩn đi) các tập luật nhạy cảm, hạn chế việc ẩn các tập mục không cần thiết phải ẩn đi.

CHƯƠNG 3 THUẬT TOÁN BẢO TOÀN TÍNH RIÊNG TƯ TRONG KHAI THÁC LUẬT KẾT HỢP

3.1 Giới thiệu

Xáo trộn dữ liệu và hạn chế truy vấn, giấu các tập luật nhạy cảm,… là các phương pháp cơ bản trong quá trình khai thác dữ liệu bảo toàn tính riêng tư. Với sự hỗ trợ của phương pháp gây nhiễu dữ liệu, dữ liệu thô trước hết bị xáo trộn thông qua việc chuyển đổi dữ liệu, rồi rạc hoá dữ liệu, và tăng thêm sự nhiễu trong dữ liệu, thông tin nhạy cảm được bảo vệ; sau đó sẽ có mô hình mong muốn, các kiến thức, các quy tắc tìm ra được từ những dữ liệu bị nhiễu loạn [12]

Ví dụ như thuật toán MASK và các thuật toán tối ưu hoá khác chỉ sử dụng phương pháp gây nhiễu dữ liệu. Tuy nhiên, dữ liệu bị nhiễu loạn vẫn còn tồn tại phần liên quan đến dữ liệu thô ban đầu. Che dấu dữ liệu, phân vùng dữ liệu và lấy mẫu dữ liệu được áp dụng trong phương pháp hạn chế truy vấn để tránh việc các dữ liệu thô bị lộ. Sau đó, kết quả khai thác các dữ liệu mong muốn có thể thu được bằng phương pháp xác suất thống kê hoặc phương pháp tính toán phân tán.

Tuy nhiên, nếu chỉ áp dụng đơn lẻ các phương pháp trên sẽ rơi vào tình huống xấu rằng mức độ bảo mật sẽ thấp. Trong phần này sẽ giới thiệu một thuật toán mới, trong đó là sự kết hợp hai phương pháp xáo trộn và giấu tập luật nhạy cảm để cải thiện hiệu suất bảo mật trong khai thác dữ liệu có tính riêng tư.

3.2 Thuật toán

3.2.1 Mô tả bài toán

Trong thực tế khai thác dữ liệu, có các thông tin và tập luật cần phải dấu đi trong quá trình khai thác cũng như trong quá trình công bố kết quả, trong thuật toán này, dữ liệu gốc ban đầu sẽ được mã hoà làm nhiễu, và trong quá trình khai thác sẽ tiến hành phương pháp làm ẩn các tập luật nhạy cảm cần dấu, để đảm bảo các kết quả khai thác được không làm tiết lộ các thông tin nhạy cảm.

- Dữ liệu đầu vào:
  + Tập dữ liệu gốc.
  + Khoá K xác thực đăng nhập.

+ Độ hỗ trợ tối thiểu (Minsup), độ phổ biến tối thiểu (*Minconf*), tập luật nhạy cảm cần dấu (Rh).

- Kết quả:

Tập phổ biến và độ hỗ trợ của các phần tử phổ biền. Trong tập phổ biến này, khi khái thác luật kết hợp sẽ không xuất hiện các tập luật nhạy cảm cần dấu.

3.2.2 Thuật toán

Bước 1: Kiểm tra xác thực truy cập.

- Nhập khoá K xác thực đăng nhập.
- Nếu đúng, được quyền truy cập tập dữ liệu.
- Nếu khoá sai, thoát không được truy cập dữ liệu.

Bước 2: Mã hoá làm nhiễu dữ liệu.

- Từ tập dữ liệu gốc D ban đầu, mã hoá làm nhiễu thành tập dữ liệu D'.
- Tiến hành khai thác trên tập D'.

Bước 3: Xác định tập phổ biến L (không xuất hiện tập luật nhạy cảm cần giấu). Để dấu các luật nhạy cảm trong tập phổ biến khai thác được, ta tiến hành giảm độ hộ trợ của các phần tử phổ biến nếu thuộc luật nhảy cảm cần giấu

- Áp dụng thuật toán Apriori tìm tập phổ biến 1 phần từ từ tập dữ liệu D'. $L_1 = \{Apriori[D']\}$
- Tìm các tập phổ biến có từ 2 phần tử trờ lên $L_2 = \{Apriori[D']\}$

(a) Xét phần từ C_k, nếu là C_k phổ biến, ta tiến hành kiểm tra C_k có thuộc các phần tử của luật nhạy cảm H($H \in Rh$) cần dấu hay không. Nếu C_k không thuộc các phần tử của luật nhạy cảm H thì C_k là phần tử phổ biến của tập L_2, sang bước (d). Ngược lại ta sang bước (b).

(b) Nếu C_k thuộc H, ta xét nếu $C_k.conf \geq Minconf$ thì tiến hành giảm độ hỗ trợ của C_k ($C_k.\sup -1$). Lập lại bước (b) cho đến khi $C_k.conf < Minconf$ (để C_k không hình thành luật thuộc tập luật nhạy cảm Rh).

(c) Sau khi xác định $C_k.conf < Minconf$, nếu độ hỗ trợ của C_k lớn hơn hoăc bằng Minsup thì C_k là phần tử phổ biến của tập L_2 . Ngược lại C_k không là tập phổ biến. Sang bước (d)

(d) Lập lại bước (a) cho đến hết tập D'.

- Tập phổ biến cần tìm là $L = L_1 \cup L_2$

3.2.3 Mã giã thuật toán.

```
INPUT
- Cơ sờ dữ liệu gốc D.
- Khoá K, xác thực truy cập.
- Độ hỗ trợ tối thiểu Minsup, độ phổ biến tối thiểu Minconf.
- Tập luật nhạy cảm cần giấu Rh.
OUTPUT
-Tập phổ biến L (Không bao gồm các luật của Rh)
Begin
0:(1) Check the Authentication   //chứng  thực đăng nhập
a.    Enter uid & pw
b.    if(uid==udb && pw==pdb)
      {
c.    Welcome in the databse
      SIPM(DB)
                  User(entry)
                  {
                        Log(id)
                  }
      }
d.    else
      {
            Not an authorized user
      }
      Exit
(2) IPPM(DB)   //quá trình mã hoá xáo trộn dữ liệu => D'
a.While(object.read() != -1)
{
      [start reading]
      [generate tokens]
            TK1,TK2,…TKn
      [token is generated according to the alphabet entered]
      If(,)
      {
            TK1,TK2,…TKn
      }
      Else
      {
            [enter the character]
```

```
            String a=object.nextLine();
            STK1,STK2,…,STKn
      }
}
(3) Generate Frequent Itemsets   //xác định tập phổ biến L
{
      C_1 = find_candidate_itemset(D')
      L_1 = {c ∈ C_1 | c.sup ≥ Minsup} //tập phổ biến 1 phần từ
      For (k = 1; L_{k-1} ≠ 0; k++)
      {
            C_k = apriori(L_{k-1});    //sinh tập phổ biến có từ 2 phần tử trở
lên.
            If (c ∈ Rh)           //nếu thuộc các luật trong tập Rh
            {
                  while(c.conf ≥ Minconf)
                  {     c.sup--        }        //giảm độ hỗ trợ
                  if(c.sup ≥ Minsup)
                  {     c ∈ L_k        }
            }
            Else c ∈ L_k
      }
      Return L = ∪L_K     //tập phổ biến không bao gồm các tập Rh
}
```

Hình 3.1 Mã giả thuật toán ẩn tập luật nhạy cảm.

3.2.4 Ví dụ

- Cho CSDL mẫu D như trong **Bảng 3.1**
- Giả sử ta có : + *Minsup*=3,*Minconf*=75%.

  + Tập luật cần giấu Rh={b->a; b->d; c->d}.

Bảng 3.1 Cơ sở dữ liệu gốc D

TID	Item
1	a b c d e
2	a c d
3	a b d f g
4	b c d e
5	a b d
6	c d e f h
7	a b c g
8	a c d e
9	a c d h

- Áp dụng thuật toán cải tiến, sau bước 2 ta có **Bảng 3.2** là cơ sở dữ liệu D' được mã hoá làm nhiễu, và các luật cần dấu Rh'={2->1, 2->4, 3->4}

Bảng 3.2 Cơ sở dữ liệu được mã hoá D'

TID	Encrypted Item
1	1 2 3 4 5
2	1 3 4
3	1 2 4 6 7
4	2 3 4 5
5	1 2 4
6	3 4 5 6 8
7	1 2 3 7
8	1 3 4 5
9	1 3 4 8

- Áp dụng thuật toán Apriori trên CSDL D, ta có **Bảng 3.3** thể hiện tập phổ biến L(D) và độ hỗ trợ cho các phần tử phổ biến

Bảng 3.3 Bảng dữ liệu cho tập phổ biến L(D)

Item	Supp
a	7
b	5
c	7
d	8
e	4
ab	4
ac	5
ad	6
bc	3
bd	4
cd	6
ce	4
de	4
abd	3
acd	4
cde	4

- Sau khi có CSDL D', ta áp dụng bước 3 của thuật toán:

  + Áp dụng thuật toán Apriori ta có **Bảng 3.4** là bảng dữ liệu của tập phổ biến 1 phần tử $L_1 = \{1, 2, 3, 4, 5\}$ và độ hỗ trợ của chúng. (1)

Bảng 3.4 Bảng dữ liệu tập phổ biến 1 phần tử

Item	Supp
1	7
2	5
3	7
4	8
5	4

+ Tìm tập phổ biến từ 2 phần tử trở lên, tập L_2:

- Xét phần tử (12)

Ta có supp(12)=4 ≥ *Minsup*=3,vậy (12) phổ biến.

(12) là phổ biến, ta kiểm tra (12) và tập luật Rh, ta có (12) thuộc luật (2->1), ta xét:

$$conf(12) = \frac{\text{supp}(12)}{\text{supp}(2)} = \frac{4}{5} = 0.8 > Minconf = 0.75$$

Theo thuật toán ta giảm độ hỗ trợ của (12), supp(12)=4-1=3, tiếp tục xét

$$conf(12) = \frac{\text{supp}(12)}{\text{supp}(2)} = \frac{3}{5} = 0.6 < Minconf = 0.75$$

Lúc này conf(12) < *Minconf* , dừng bước xét (12) và Rh, kiểm tra:

$$\text{supp}(12) = 3 \geq Minsup = 3$$

Vậy (12) là phổ biến, ta có $L_2 = \{12\}$

- Xét phần tử (24):

Ta có supp(24)=4 ≥ *Minsup*=3, vậy (24) phổ biến.

(24) là phổ biến, ta kiểm tra (24) và tập luật Rh, ta có (24) thuộc luật (2->4), ta xét:

$$conf(24) = \frac{\text{supp}(24)}{\text{supp}(2)} = \frac{4}{5} = 0.8 > Minconf = 0.75$$

Theo thuật toán ta giảm độ hỗ trợ của (24), supp(12)=4-1=3, tiếp tục xét

$$conf(24) = \frac{\text{supp}(24)}{\text{supp}(2)} = \frac{3}{5} = 0.6 < Minconf = 0.75$$

Lúc này conf(24) < *Minconf* , dừng bước xét (24) và Rh, kiểm tra:

$$\text{supp}(24) = 3 \geq Minsup = 3$$

Vậy (24) là phổ biến, ta có $L_2 = \{12,24\}$.

- Xét phần tử (34):

Ta có supp(34)=6 ≥ *Minsup*=3, vậy (34) phổ biến.

(34) là phổ biến, ta kiểm tra (34) và tập luật Rh, ta có (34) thuộc luật (3->4), ta xét:

$$conf(34) = \frac{\text{supp}(34)}{\text{supp}(3)} = \frac{6}{7} = 0.85 > Minconf = 0.75$$

Theo thuật toán ta giảm độ hỗ trợ của (24), supp(34)=6-1=5, tiếp tục xét

$$conf(34) = \frac{\text{supp}(34)}{\text{supp}(3)} = \frac{5}{7} = 0.71 < Minconf = 0.75$$

Lúc này conf(34) < *Minconf* , dừng bước xét (34) và Rh, kiểm tra:

$$\text{supp}(34) = 5 \geq Minsup = 3$$

Vậy (34) là phổ biến, ta có $L_2 = \{12, 24, 34\}$.

- Xét phần tử (124):

Ta có supp(124)=3 ≥ *Minsup*=3, vậy (124) phổ biến.

(124) là phổ biến, ta kiểm tra (124) và tập luật Rh, ta có (124) thuộc luật (2->1), ta xét:

$$conf(124) = \frac{\text{supp}(124)}{\text{supp}(2)} = \frac{3}{5} = 0.6 < Minconf = 0.75$$

Theo thuật toán ta không giảm độ hỗ trợ của (124),kiểm tra:

$$\text{supp}(124) = 3 \geq Minsup = 3$$

Vậy (124) là phổ biến, ta có $L_2 = \{12, 24, 34,124\}$.

- Xét phần tử (134):

Ta có supp(134)=4 ≥ Minsup=3, vậy (124) phổ biến.

(134) là phổ biến, ta kiểm tra (134) và tập luật Rh, ta có (134) thuộc luật (3->4), ta xét:

$$conf(134) = \frac{\text{supp}(134)}{\text{supp}(3)} = \frac{4}{7} = 0.57 < Minconf = 0.75$$

Theo thuật toán ta không giảm độ hỗ trợ của (134),kiểm tra:

$$\text{supp}(134) = 4 \geq Minsup = 3$$

Vậy (134) là phổ biến, ta có $L_2 = \{12, 24, 34,124,134\}$.

- Xét phần tử (345):

Ta có supp(345)=4 ≥ *Minsup*=3, vậy (345) phổ biến.

(345) là phổ biến, ta kiểm tra (345) và tập luật Rh, ta có (345) thuộc luật (3->4), ta xét:

$$conf(345) = \frac{\text{supp}(345)}{\text{supp}(3)} = \frac{4}{7} = 0.57 < Minconf = 0.75$$

Theo thuật toán ta không giảm độ hỗ trợ của (345),kiểm tra:

$$\text{supp}(345) = 4 \geq Minsup = 3$$

Vậy (345) là phổ biến, ta có $L_2 = \{12, 24, 34,124,134,345\}$.

Ngoài ra, trong CSDL D' ta còn có các phần tử thoả mãn điều kiện $supp \geq Minsup$, và không thuộc luật của tập luật Rh, nên ta có tập phổ biến 2 phần trở lên sẽ là $L_2 = \{12, 13,14, 23, 24, 34, 45, 124, 134, 345\}$. (2)

Từ (1) và (2) ta có tập phổ biến $L(D') = L_1 \cup L_2$, và **Bảng 3.5** độ hỗ trợ của các phần tử phổ biến

$$L = \{1, 2, 3, 4, 5,12, 13,14, 23, 24, 34, 45, 124, 134, 345\}$$

Bảng 3.5 Bảng dữ liệu tập phổ biến L(D')

Item	Supp
1	7
2	5
3	7
4	8
5	4
12	3
13	5
14	6
23	3
24	3
34	5
35	4
45	4
124	3
134	4
345	4

Từ dữ liệu của **Bảng 3.5** ta có thể kiểm tra tính đúng đắn của thuật toán, xem có thế khai thác được các luật nhạy cảm yêu cầu đề bài đưa ra là cần giấu.

Kiểm tra từ **Bảng 3.5** có thể khai thác đươc luật 2->4) hay không tương đương b->d?

+ Từ **Bảng 3.5** ta có: supp(24)=3, supp(2)=5;

+ *Minconf*=0.75(dữ liệu ban đầu).

+Xét $conf(24) = \frac{\text{supp}(24)}{\text{supp}(2)} = \frac{3}{5} = 0.6 < Minconf = 0.75$ (*)

⇨ Từ (*) ta có thể kết luận, không thể khai thác được luật(2->4) tương đương khô thể suy ra (b->d), đảm bào yêu cầu thuật toán đề ra.

3.3 Chương trình minh hoạ cho thuật toán

3.3.1 Giới thiệu

Chương trình minh họa được thực hiện trên máy tính xách tay Vaio Z(VGN-SZ640) có cấu hình như sau:

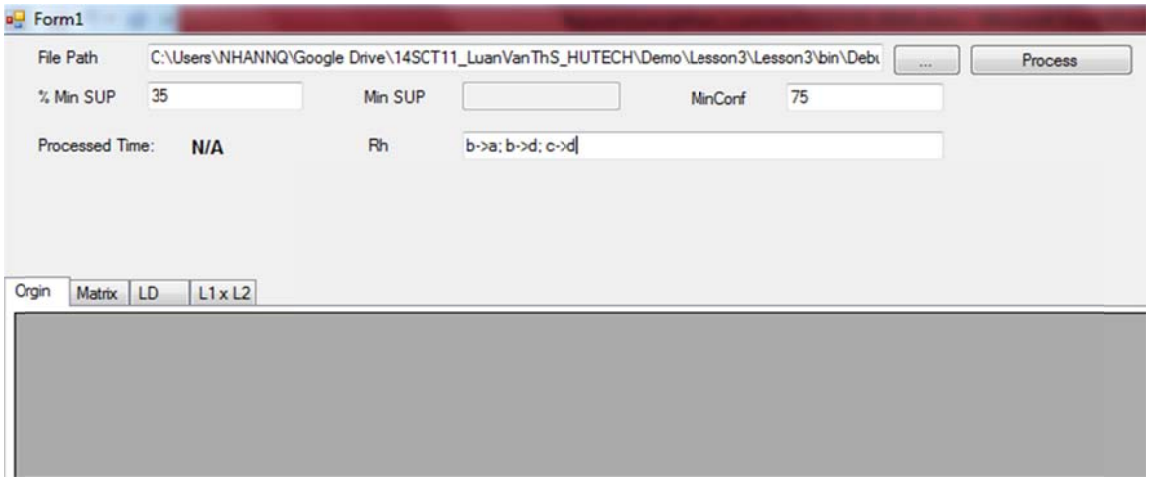
- Phần cứng : CPU Intel Core 2 Duo T7200, 2.2GHz ; RAM : 4GB ; Ổ cứng : 120GB.
- Phần mềm : Hệ điều hành Windows 7(SP 1) ; DotNet FrameWork 4.5, Microsoft Visual Studio 2015.

Chương trình minh họa được xây dựng dựa trên ví dụ từ mục 3.2.4, với các dữ liệu đầu vào là:

- Cơ sở dữ liệu gốc D.
- Độ hỗ trợ tối thiểu *Minsup*.
- Độ phổ biến tối tối thiểu *Minconf*
- Tập luật cần giấu Rh.

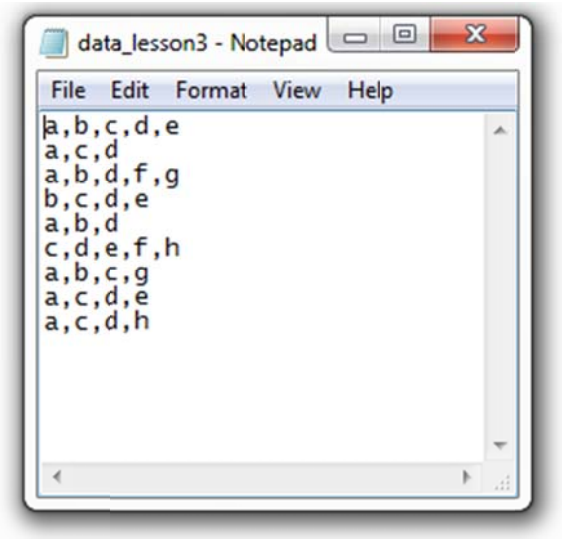
Dữ liệu đầu ra bao gồm tập các phần tử phổ biến và độ hỗ trợ của chúng, từ các tập phổ biến này khôn thể khai thác được các luật cần giấu thuộc tập luật Rh.

3.3.2 Một số giao diện chính của chương trình



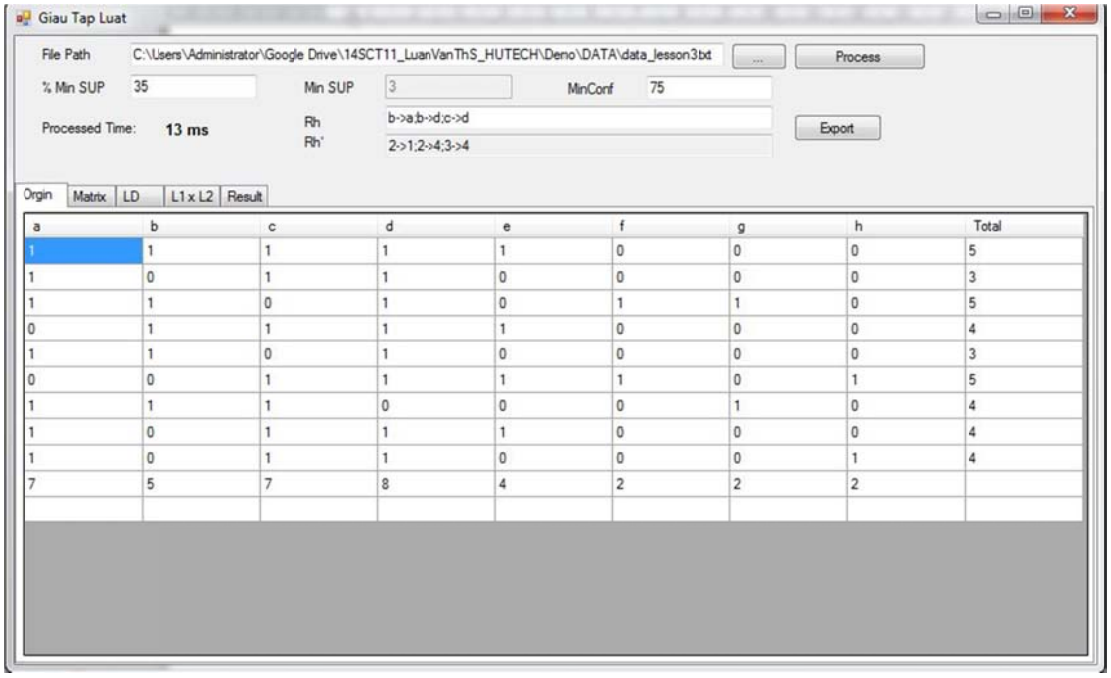
Hình 3.2 Giao diện chính

Hình 3.2 là giao diện chính của chương trinh mình họa, chọn đường dẫn lưu CSDL gốc D **Hình 3.3**, nhập độ hỗ trợ tối thiểu Minsup, độ phổ biến tối thiểu *Minconf*, tập luật cần dấu Rh.



Hình 3.3 CSDL gốc D

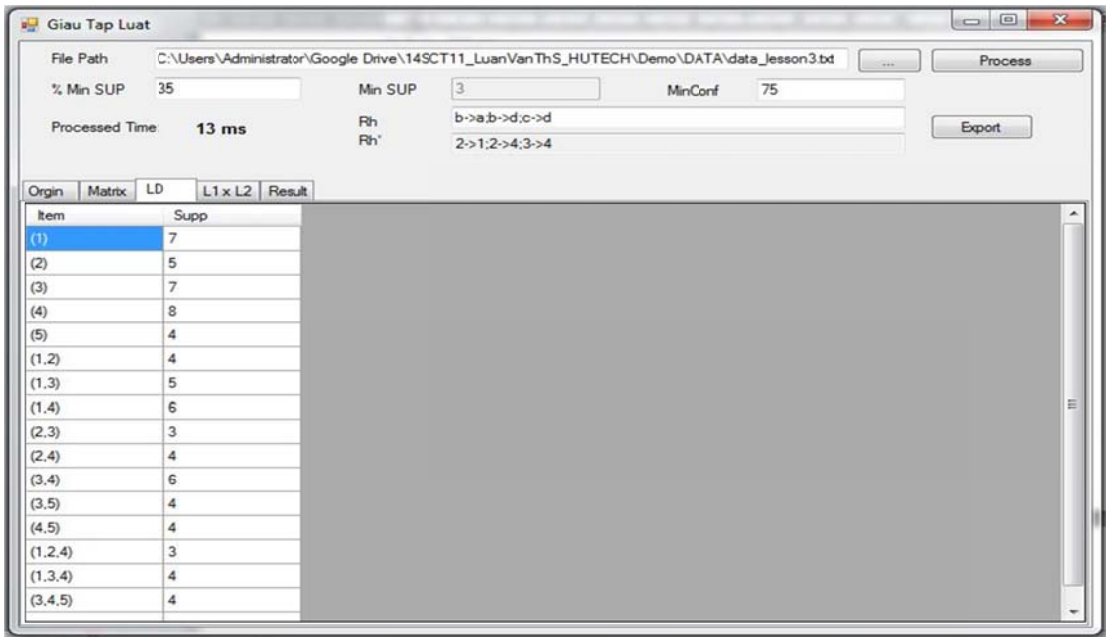
Sau khi nhập đầy đủ các dữ liệu đầu vào, ta chọn Process, chương trình sẽ tiến hành bước đầu tiên là mã hóa làm nhiễu dữ liệu gốc ban đầu theo thuật toán trong mục 3.1



Hình 3.4 Cơ sở dữ liệu D'

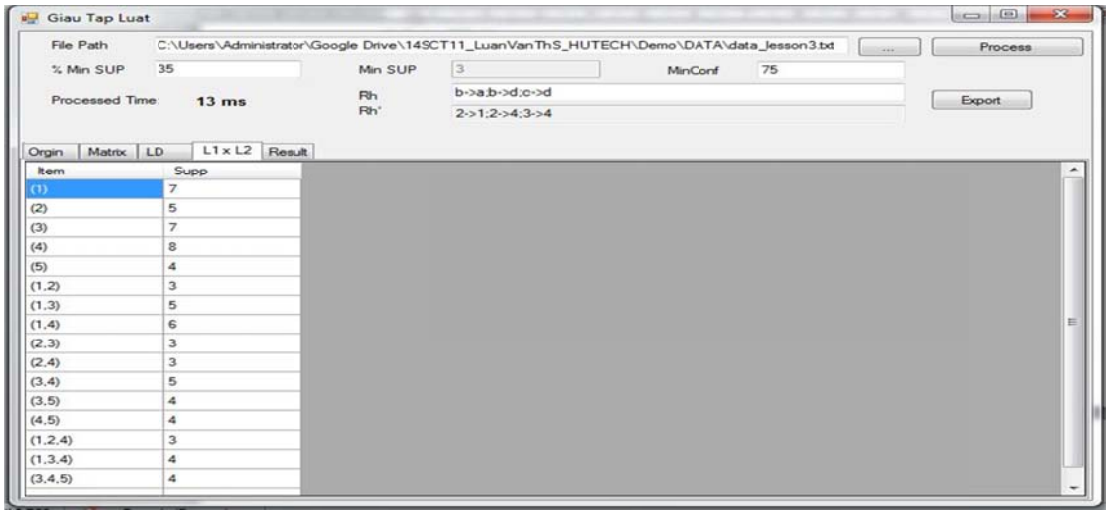
Sau bước đầu tiên ta sẽ có **Hình 3.3,** trong tab Orgin là CSDL được mã hóa làm nhiễu D'.

Chương trình minh họa sẽ cho ra hai kết quả là tập phổ biến của CSDL gốc D và tập phổ biến của CSDL được mã hóa làm nhiễu D' để có được sự so sánh có tìm được luật cần dấu hay không.



Hình 3.5 Tập phổ biến cho CSDL D'

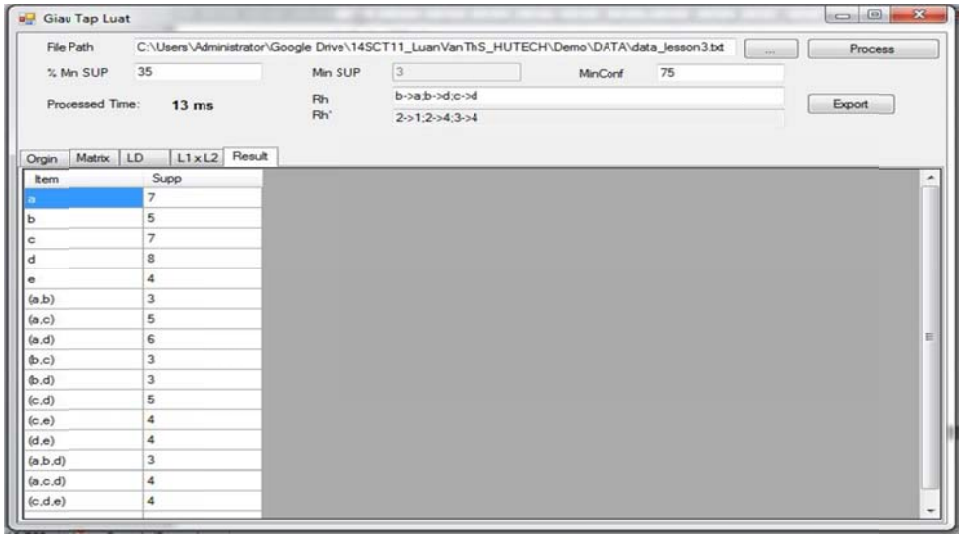
Tab LD’ trong **Hình 3.5** cho ta tập phổ biến của CSDL gốc D’ khi không có giấu luật, và độ hỗ trợ của các phần tử phổ biến.



Hình 3.6 Tập phổ biến cho CSDL D'

Tab L1xL2 trong **Hình 3.6** cho ta tập phổ biến của CSDL D’ khi có áp dụng giấu luật, và độ hỗ trợ của các phần tử phổ biến.

Từ **Hình 3.5** và **Hình 3.6** ta thấy được sự khác nhau của độ hỗ trợ của các tập phổ biến và các luật cần giấu đã không thể khai thác được trong kết quả cuối cùng tại **Hình 3.6**.



Hình 3.7 Kết quả cần tìm

Hình 3.7 hiển thị cho kết quả cuối cùng là tập phổ biến của CSDL D ban đầu, và độ hổ trợ của các phần từ phổ biến. Từ **Hình 3.7** ta không thể tìm được các luật cần giấu trong tập Rh.

3.3.3 Nhận xét

Chương trình đã thực hiện được việc minh họa cho thuật toán tiến hành mã hóa làm nhiễu dữ liệu, tiến hành ẩn các tập luật nhạy cảm theo yêu cầu. Đảm bảo các yêu cầu của thuật toán đề ra.

Nhưng chương trinh chỉ mới dừng lại ở bước minh họa cho thuật toán trong luận văn, đáp ứng các tiêu chí đưa ra đúng kết quả theo các ví dụ, đây là mặt hạn chế của chương trình. Trong thời gian tới, tác giả sẽ tiếp tục cải tiến chương trình để có thể áp dụng cho các dữ liệu thực tế.

KẾT LUẬN

Sau thời gian nghiên cứu và tiến hành thực hiện, Luận văn đạt được:

1. **Kết quả:**

Giới thiệu tổng quan về Khai thác dữ liệu, qua đó giúp hiểu rõ hơn về các khái niệm cũng, tầm quan trọng của Khai thác dữ liệu trong thời đại bùng nỗ thông tin hiện nay.

Các phương pháp Khai thác dữ liệu, Khai thác dữ liệu bảo toàn tính riêng tư, từ đó nắm được các kỹ thuật khai thác dữ liệu, cũng như sự cần thiết đảm báo tính riêng tư trong quá trình khai thác dữ liệu.

Khái niệm về luật kết hợp, các thuật toán khai thác luật kết hợp. Nắm được vai trò của khai thác luật kết hợp trong khai thác dữ liệu, và tầm quan trọng của khai thác luật kết hợp bảo toàn tính riêng tư.

Nghiên cứu giới thiệu thuật toán nâng cao hiệu quả thực thi trong khai thác luật kết hợp bảo toàn tín riêng tư. Cụ thể thông qua việc mã hoá nhiễu loạn dữ liệu và giấu các luật nhậy cảm, tăng hiệu quả thực thi trong khai thác luật kết hợp bảo toàn tính riêng tư.

Xây dựng chương demo minh hoạ cho việc nâng hiệu quả thực thi trong khai thác luật kết hợp bảo toàn tín riêng tư.

2. **Hạn chế:**

Thuật toán chỉ sử dụng thuật toán mã hoá nhiễu loạn đơn giản để làm nhiễu CSDL gốc

Chương trình chỉ dừng ở bước minh hoạ cho thuật toán, chưa được áp dụng được cho thực tế.

3. **Kiến nghị:**

Với các mục hạn chế như trên, đó chính là hướng phát triển trong tương lai, có thể áp dụng các kỹ thuật làm nhiệu mà hoá phức tạp hơn cho CSDL gốc như SHA, và áp dụng thuật toán vào môi trường thực tế.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Cao Tùng Anh (2014), *Khai Thác Dữ Liệu Phân Tán Bảo Toàn Tính Riêng Tư*, Luận Án Tiến Sĩ Toán Học, Viện Hàn Lâm Khoa Học Và Công Nghệ Việt Nam - Viện Công Nghệ Thông Tin.

[2] Nguyễn Xuân Huy, Lê Quốc Hải, Nguyễn Gia Như, Cao Tùng Anh, Bùi Đức Minh(2009), *Lý thuyết giàn và ứng dụng trong thuật toán ẩn tập mục nhạy cảm*, Báo cáo tại Hội thảo Quốc gia " Một số vấn đề chọn lọc của CNTT và truyền thông, Đồng Nai.

[3] Nguyễn Thị Thuỳ (2014), *Một số kỹ thuật khai thác luật kết hợp có đảm bảo tính riêng tư trong các tập giao dịch phân tán ngang*, Luận văn Thạc sĩ, Trường Đại học Thái Nguyên – Trường Đại học CNTT và Truyền thông.

Tiếng Anh

[4] Andruszkiewicz (2007), *Optimization for MASK scheme in privacy preserving data mining for association rules*, International Conference on Rough Sets and Intelligent Systems Paradigms, Warsaw, pp. 465 - 474.

[5] H. Lou, Y. Ma, F. Zhang, M. Liu, W. Shen (2014), *Data Mining for Privacy Preserving Association Rules Based on Improved MASK Algorithm*, Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design, Hsinchu, pp.265 - 270.

[6] R. Agrawal, T. Imielinski, and A. Swami (1993), *Mining association rules between sets of items in large database*s, International Conference on Management of Data, Washington D.C, pp.207 - 216.

[7] S. Agrawal, V. Krishnan, and J. R. Haritsa (2004), *On addressing efficiency concerns in privacy-preserving mining*, International Conference on Database Systems for Advanced Applications, Jeju Island, pp.113 - 114.

[8] S. Geng, Y. Li, và L. Zhen, (2013), *An approach to association rules mining using inclusion degree of soft sets*, Tien Tzu Hsueh Pao/Acta Electronica Sinica, Volume 41, pp.804 - 809.

[9] S. Verykios, A. K. Elmagarmid, B.Elisa, Y. Saygin, and D.Elena (2004), *Association rule hiding*, IEEE Transactions on Knowledge and Data Engineering, Volume 16, Issue 4, pp.434 - 447.

[10] S.J.Rizvi , J.R.Haritsa(2002), *Maintaining data privacy in association rule mining*, Proceedings of the 28th international conference on Very Large Data Bases, pp. 682 – 693, Hong Kong, China.

[11] V. Nebot, R. Berlang (2010)a, *Mining association rules from semantic web data*, International Conference on Industrial Engineering and Other Applications of Applied Intelligence Systems, Cordoba, pp.504-513.

[12] W. Li, J. Liu(2010), *Privacy Preserving Association Rules Mining Based on Data Disturbance and Inquiry Limitation*, International Conference on Internet Computing for Science and Engineering, Harbin, pp.24 - 29.

[13] Xuan Canh Nguyen, Tung Cao Anh, Hoai Bac Le (2012), *An Enhanced Scheme for Privacy-Preserving Association Rules Mining on Horizontally Distributed Databases*, IEEE RIVF International Conference on Computing & Communication Technologies, research, Innovation, pp.1 - 4.

[14] Y.Saygin, V.S.Verykios, C.Clifton (2001), *Using unknowns to prevent discovery of association rules*, ACM SIGMOD Record, Volume 30, Issue 4 , pp.45-54 ISSN:0163-5808.