HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Đặng Thị Kim Trang

PHƯƠNG PHÁP ẨN CÁC TẬP MỤC CÓ ĐỘ HỮU ÍCH CAO TRONG CƠ SỞ DỮ LIỆU GIAO TÁC LỚN

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

TP.HỒ CHÍ MINH – NĂM 2022

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Đặng Thị Kim Trang

PHƯƠNG PHÁP ẨN CÁC TẬP MỤC CÓ ĐỘ HỮU ÍCH CAO TRONG CƠ SỞ DỮ LIỆU GIAO TÁC LỚN

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC:

TS. NGUYỄN KHẮC CHIẾN

TP.HỒ CHÍ MINH - NĂM 2022

LỜI CAM ĐOAN

Tôi cam đoan luận văn: ***"Phương pháp ẩn các tập mục có độ hữu ích cao trong cơ sở dữ liệu giao tác lớn"*** là công trình nghiên cứu của chính tôi.

Các số liệu được sử dụng trong luận văn là trung thực và chính xác.

Ngoài những nội dung nghiên cứu của luận văn, các vấn đề được trình bày đều là những tìm hiểu và nghiên cứu của tôi hoặc là được trích dẫn từ các nguồn tài liệu có ghi tham khảo rõ ràng, hợp pháp.

Trong luận văn, tôi có tham khảo một số tài liệu của một số tác giả được liệt kê tại danh mục tài liệu tham khảo.

TP.HCM, Ngày 04 tháng 5 năm 2022

Học viên thực hiện luận văn

Đặng Thị Kim Trang

LỜI CẢM ƠN

Tôi chân thành cảm ơn **TS. Nguyễn Khắc Chiến** – Giảng viên của Trường Đại học Cảnh sát Nhân dân, Thầy đã chỉ bảo và hướng dẫn tận tình cho tôi trong suốt quá trình nghiên cứu khoa học và thực hiện luận văn.

Đồng thời, tôi xin cảm ơn sự giúp đỡ, tạo điều kiện và khuyến khích tôi trong quá trình nghiên cứu và học tập của các Thầy, Cô giáo của Học Viện Công nghệ Bưu chính viễn thông cơ sở tại TP.HCM.

Vì thời gian có hạn và kiến thức còn hạn hẹp, nên luận văn khó tránh khỏi những thiếu sót, rất mong nhận được ý kiến đóng góp của quý Thầy Cô, Anh Chị và các Bạn.

Xin chân thành cảm ơn!

TP.HCM, Ngày 04 tháng 5 năm 2022

Học viên thực hiện luận văn

Đặng Thị Kim Trang

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
CSDL	Database	Cơ sở dữ liệu
eu	External Utility	Độ hữu ích bên ngoài (lợi nhuận)
iu	Internal Utility	Độ hữu ích bên trong (số lượng)
HUI	High Utility Itemset	Tập mục có độ hữu ích cao
WFI	Weighted Frequent Itemset	Tập phổ biến có trọng số
HUIM	High Utility Itemset Mining	Khai thác tập mục độ hữu ích cao
PPDM	Privacy Preserving Data Mining	Khai thác dữ liệu bảo vệ tính riêng tư
PPUIM	Privacy Preserving Utility itemset Mining	Khai thác tập mục có độ hữu ích cao được bảo vệ tính riêng tư
SHUI	Sensitive High Utility Itemset	Tập mục có độ hữu ích cao nhạy cảm
NSHUI	Non Sensitive High Utility Itemset	Tập mục có độ hữu ích cao không nhạy cảm
HF	Hiding Failure	Ẩn thất bại
MC	Missing Cost	Chi phí lỗi/ẩn nhầm
ST	Sensitive Transaction	Giao tác nhạy cảm
minutil	Minimal utility threshold	Ngưỡng độ hữu ích tối thiểu
EHSHUI	An efficient algorithm for hiding sensitive high utility itemset	Một thuật toán hiệu quả để ẩn tập mục tiện ích cao nhạy cảm
IEHSHUI	An improved algorithm for hiding sensitive high utility itemsets	Một thuật toán cải tiến để ẩn các tập mục có độ hữu ích cao nhạy cảm

DANH SÁCH BẢNG

DANH SÁCH HÌNH VẼ

MỞ ĐẦU

1. Lý do chọn đề tài

Hiện nay, trong lĩnh vực kinh doanh việc tính toán doanh số và tối ưu hóa lợi nhuận bán hàng là công việc cực kỳ quan trọng, nó ảnh hưởng trực tiếp đến doanh thu và chiến lược bán hàng của các công ty, siêu thị hay các đơn vị bán lẻ. Đặc biệt, với số lượng hàng hóa lớn, giá cả khác nhau, nên việc tính toán lợi nhuận tối ưu bán hàng càng quan trọng. Với số lượng giao tác mỗi giờ có thể lên đến hàng chục nghìn giao tác, việc tính toán xem mặt hàng nào đem lại doanh số cao, mặt hàng nào kinh doanh không hiệu quả dù bán với số lượng lớn càng trở nên khó khăn do dữ liệu quá lớn, liên tục.

Khai phá tập phổ biến thường được mô tả là một quá trình lấy thông tin có giá trị từ cơ sở dữ liệu lớn, nó bắt nguồn từ dạng mẫu có sẵn tồn tại trong cơ sở dữ liệu, các mẫu này có khuynh hướng gom nhóm lại với nhau và được định nghĩa như là một mô hình khai thác. Khai phá tập mục độ hữu ích cao là một mở rộng của bài toán khai phá tập phổ biến, đã được nhiều tác giả quan tâm với mục đích đánh giá ý nghĩa của các tập mục trong khai phá luật kết hợp. Để khai phá tập mục có độ hữu ích cao, một giá trị được sử dụng đó là lợi nhuận của tập mục (Itemset), chẳng hạn tổng lợi nhuận mà doanh nghiệp thu được nếu bán tập mục ấy trong giao tác. Khác với khai phá tập phổ biến, độ hữu ích của tập mục không thỏa tính chất bao đóng giảm nên độ phức tạp của bài toán cao.

Ngoài ra, trong hợp tác kinh doanh việc muốn chia sẽ cơ sở dữ liệu với nhau để cùng có lợi, nhưng mang lại nhiều rủi ro để lộ ra các thông tin nhạy cảm như: số định danh cá nhân, số tài khoản ngân hàng,… Để giải quyết vấn đề này, các tri thức nhạy cảm có thể được ẩn bằng cách chuyển đổi cơ sở dữ liệu ban đầu thành cơ sở dữ liệu được sửa đổi theo một số chiến lược cụ thể và quá trình ẩn đó được gọi là làm sạch dữ liệu.

Bên cạnh đó, những năm gần đây, khai phá dữ liệu bảo vệ tính riêng tư đã trở thành hướng nghiên cứu quan trọng. Trong phần luận văn này, tôi xin tập trung nghiên cứu bài toán khai phá các tập mục có độ hữu ích cao được bảo vệ tính riêng tư (PPUIM - Privacy Preserving Utility itemset Mining) để ẩn các tập mục có độ hữu ích cao nhạy cảm trong cơ sở dữ liệu giao tác có kích thước lớn. Một trong những vấn đề đặt ra khi giải quyết bài toán này là làm giảm các hiệu ứng phụ như: ẩn nhầm các tập mục có độ hữu ích cao không nhạy cảm, sự khác nhau giữa CSDL ban đầu và CSDL sau khi sửa đổi,…

Vì thế, luận văn sẽ tập trung nghiên cứu thuật toán ẩn các tập mục có độ hữu ích cao nhạy cảm và đề xuất phương pháp ẩn các tập mục có độ hữu ích cao nhạy cảm hiệu quả hơn nhằm giảm thiểu các hiệu ứng phụ.

2. Mục tiêu nghiên cứu

Nghiên cứu các phương pháp ẩn tập mục độ hữu ích cao nhạy cảm hiện có dựa trên các công trình đã công bố gần đây.

Tìm hiểu những ưu điểm và hạn chế của các phương pháp ẩn từ đó đề xuất phương pháp ẩn hiệu quả hơn.

Tìm hiểu các thông số đánh giá tính hiệu quả của các phương pháp ẩn tập mục có độ hữu ích cao nhạy cảm.

Tiến hành cài đặt thử nghiệm phương pháp đề xuất, đánh giá dựa trên các thông số, so sánh với các phương pháp ẩn hiện có.

3. Tổng quan nghiên cứu của đề tài

Bài toán ẩn các tập mục độ hữu ích cao nhạy cảm đang là chủ đề được nhiều nhà nghiên cứu quan tâm. Mục tiêu của bài toán là bảo vệ các thông tin nhạy cảm không thể khai phá được bằng các phương pháp khai phá tập mục có độ hữu ích cao với cùng một ngưỡng độ hữu ích tối thiểu do người dùng quy định. Đồng thời, các phương pháp ẩn tập mục có độ hữu ích cao nhạy cảm làm giảm thiểu các hiệu ứng phụ trên các thông tin không nhạy cảm và tính toàn vẹn của cơ sở dữ liệu ban đầu.

Hiện đã có một số phương pháp ẩn hiệu quả để giải quyết vấn đề này, tuy nhiên những phương pháp này vẫn còn tạo ra các hiệu ứng phụ không mong muốn.

Đề tài đề xuất phương pháp ẩn một cách phù hợp, để ẩn các tập mục có độ hữu ích cao nhạy cảm một cách hiệu quả, làm giảm thiểu các hiệu ứng phụ trên các thông tin không nhạy cảm. Kết quả thực nghiệm cho thấy thuật toán đề xuất hiệu quả hơn các thuật toán hiện có về mặt các hiệu ứng phụ như ẩn nhầm các thông tin không nhạy cảm, chất lượng của cơ sở dữ liệu sau quá trình ẩn.

4. Đối tượng, phạm vi nghiên cứu

Phương pháp ẩn các tập mục có độ hữu ích cao nhạy cảm trong các cơ sở dữ liệu giao tác lớn.

5. Đóng góp của đề tài

Luận văn đề xuất phương pháp cải tiến thuật toán EHSHUI trong công trình của Trieu và cộng sự (2020) [4]; Vo, B và cộng sự (2013) [14]. Phương pháp được đề xuất sẽ lựa chọn tập mục nhạy cảm hợp lý và mục sửa đổi. Thực nghiệm đã chỉ ra, phương pháp đề xuất hiệu quả hơn EHSHUI [4] và thuật toán [14] về thời gian thực hiện và sử dụng bộ nhớ.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Tập mục phổ biến và khai phá tập phổ biến truyền thống

1.1.1. Tập mục phổ biến

Khai phá tập phổ biến là quá trình tìm kiếm các mục có số lần xuất hiện lớn hơn một ngưỡng do người dùng quy định và hạn chế của khai phá tập phổ biến là xử lý tất cả các mục có tầm quan trọng như nhau. Trong một giao tác mỗi mục chỉ có trạng thái xuất hiện hoặc không xuất hiện.

Do những hạn chế này làm cho bài toán khai phá tập phổ biến truyền thống không còn phù hợp với các cơ sở dữ liệu thực tế, cụ thể là: trong cơ sở dữ liệu của siêu thị, mỗi mặt hàng có tầm quan trọng và giá cả khác nhau và số lượng mua các mặt hàng trong mỗi giao tác cũng khác nhau,… Vì thế, mô hình khai phá tập phổ biến chỉ phản ánh mối tương quan giữa các mục xuất hiện trong cơ sở dữ liệu, mà không phản ánh hết ý nghĩa của từng mục dữ liệu. Để khắc phục những nhược điểm trên có hai mô hình được đưa ra là tập phổ biến có trọng số và tập mục độ hữu ích cao.

Ramkumar và cộng sự năm 1998 [22], đã đưa ra mô hình khai phá tập phổ biến có trọng số. Trong đó, mỗi mục có một trọng số khác nhau như: giá cả, số lượng,… Một mục được xem là phổ biến có trọng số khi giá trị trọng số của chúng lớn hơn một ngưỡng do người dùng quy định. Từ đó, có nhiều thuật toán khai phá tập phổ biến có trọng số được đưa ra [23], [24], [25], [26],…

Chan và cộng sự năm 2003 [27], đã đưa ra mô hình khai phá tập mục độ hữu ích cao nhằm khắc phục những hạn chế của mô hình khai phá tập phổ biến và tập phổ biến có trọng số. Mô hình này cho phép người sử dụng đánh giá tầm quan trọng của từng mục qua hai trọng số khác nhau gọi là độ hữu ích bên trong (số lượng) và độ hữu ích bên ngoài (lợi nhuận). Độ hữu ích bên trong là số lượng từng mục trong giao tác, độ hữu ích bên ngoài là lợi nhuận hoặc giá cả của các mặt hàng. Độ hữu ích của một mục là tích hai giá trị độ hữu ích bên trong và độ hữu ích bên ngoài. Tập mục độ hữu ích cao khi giá trị độ hữu ích của nó lớn hơn một ngưỡng độ hữu ích tối thiểu do

người dùng quy định, nhờ khai phá tập mục độ hữu ích cao có thể đưa ra một số quyết định quan trọng như: tối đa hóa doanh thu, giảm thiểu chi phí, hạn chế hàng tồn kho,…

1.1.2. Khám phá tri thức và khai thác dữ liệu

Khai phá tri thức là việc rút trích tri thức một cách tự động và hiệu quả từ một khối dữ liệu lớn. Tri thức đó thường ở dạng các mẫu có tính chất không tầm thường, không tường minh chưa được biết đến và có tiềm năng mang lại lợi ích. Có một số nhà nghiên cứu gọi khai phá dữ liệu là phát hiện tri thức trong cơ sở dữ liệu và có thể coi khai phá dữ liệu là cốt lõi của quá trình phát hiện tri thức. Quá trình phát hiện tri thức gồm các bước:

Bước 1: Chọn dữ liệu cần được khai phá từ các dữ liệu lớn.

Bước 2: Làm sạch dữ liệu, rút gọn dữ liệu.

Bước 3: Chuẩn hóa và làm mịn dữ liệu để đưa dữ liệu về dạng thuận lợi nhất nhằm phục vụ cho các kỹ thuật khai thác.

Bước 4: Là bước quan trọng nhất, áp dụng các kỹ thuật để khai phá, trích chọn được các mẫu thông tin, các mối liên hệ đặc biệt trong dữ liệu.

Bước 5: Đánh giá và biểu diễn tri thức theo dạng đồ thị, cây, bảng biểu, luật,…

Trong giai đoạn khai phá dữ liệu, có thể cần sự tương tác của người dùng để điều chỉnh và rút ra các tri thức cần thiết. Do đó, khai phá dữ liệu là giai đoạn duy nhất tìm ra được thông tin mới, thông tin tiềm ẩn có trong cơ sở dữ liệu chủ yếu phục vụ cho mô tả và dự đoán. Gồm các bước sau:

Bước 1: Xác định chính xác các vấn đề cần giải quyết.

Bước 2: Xác định các dữ liệu liên quan, dùng để xây dựng giải pháp.

Bước 3: Thu thập các dữ liệu liên quan và tiền xử lý chúng thành dạng, sao cho thuật toán khai phá dữ liệu có thể hiểu được.

Bước 4: Chọn thuật toán khai phá dữ liệu thích hợp và thực hiện việc khai phá dữ liệu nhằm tìm được các mẫu có ý nghĩa dưới dạng biểu diễn tương ứng.

Khai phá dữ liệu có nhiều ứng dụng trong thực tiễn như: Phân tích dữ liệu và hỗ trợ ra quyết định; Điều trị trong y học; Phân tích tình hình tài chính và dự báo giá của các cổ phiếu và bảo hiểm...v.v…

1.1.3. Khai phá tập phổ biến truyền thống

Khai phá tập phổ biến truyền thống là các thuật toán (Agrawal và đồng sự 1993; Han và đồng sự 2004) để tính toán trên các giao tác như bán hàng trong siêu thị hoặc số lượt truy cập website. Các bước thực hiện thuật toán như sau:

Bước 1: Duyệt cơ sở dữ liệu để có được độ hữu ích S của tập mục (itemset). So sánh S với ngưỡng tối thiểu do người dùng quy định để có được 1-itemset (L_1).

Bước 2: Sử dụng L_{k-1} nối với L_{k-1} để có được các mục ứng viên k-itemset. Loại bỏ các tập mục không phải là ứng viên tiềm năng thu được k-itemset.

Bước 3: Duyệt cơ sở dữ liệu để có được độ hữu ích của mỗi ứng viên k-itemset, so sánh S với k-itemset thu được ứng viên tiềm năng k-itemset (L_k).

Bước 4: Lặp lại bước 2 cho đến khi ứng viên bằng ∅.

Bước 5: Với mỗi ứng viên tiềm năng L_m, sinh các tập con của S không rỗng của I.

Cho tập các mục $I=\{i_1, i_2,..., i_n\}$. Một giao tác T là một tập con của I, $T \subseteq I$, CSDL giao tác là tập các giao tác $D =\{T_1, T_2,..., T_m\}$. Mỗi giao tác được gán một định danh TID. CSDL này thường được sử dụng trong kinh doanh thương mại hoặc các hệ thống ngân hàng. CSDL giao tác thường được biểu diễn ở dạng ngang, dạng dọc và bảng ma trận.

Biểu diễn giao tác dạng ngang là trình bày giao tác dưới dạng một danh sách, mỗi giao tác có một mã định danh riêng (TID), mỗi giao tác chứa một danh sách các mục.

Bảng 1.1: Cơ sở dữ liệu giao tác (Biểu diện dạng ngang)

TID	Mục
T1	a, b, e
T2	c, d, e, f
T3	b, c, e, f
T4	a, b, c, d, e
T5	b, c
T6	a, e, f
T7	a, c, d
T8	b, d, e
T9	a, b, c, d, e, f

Biểu diễn giao tác dạng dọc là trình bày danh sách các mục dữ liệu, mỗi mục dữ liệu chứa tất cả các mã định danh giao tác.

Bảng 1.2: Cơ sở dữ liệu giao tác (Biểu diễn dạng dọc)

Mục	TID
a	T1, T4, T6, T7, T9
b	T1, T3, T4, T5, T8, T9
c	T2, T3, T4, T5, T7, T9
d	T2, T4, T7, T8, T9
e	T1, T2, T3, T4, T6, T8, T9
f	T2, T3, T6, T9

Biểu diễn dạng ma trận Cho CSDL D = $\{T_1, T_2, \ldots, T_n\}$ trên tập các mục I = $\{I_1, I_2, \ldots, I_n\}$ được biểu diễn bởi ma trận nhị phân M = $(k_{pq})_{mxn}$

$$k_{pq} = \begin{cases} 1 \; khi \; i_q \in T_q \\ 0 \; khi \; i_q \notin T_q \end{cases}$$

Với cơ sở dữ liệu từ bảng 1.1 ta có ma trận giao tác như sau:

Bảng 1.3: Cơ sở dữ liệu giao tác (Biểu diễn dạng ma trận)

TID	a	b	c	d	e	f
T1	1	1	0	0	1	0
T2	0	0	1	1	1	1
T3	0	1	1	0	1	1
T4	1	1	1	1	1	0
T5	0	1	1	0	0	0
T6	1	0	0	0	1	1
T7	1	0	1	1	0	0
T8	0	1	0	1	1	0
T9	1	1	1	1	1	1

Ví dụ minh họa thuật toán Apriori:

Thuật toán sử dụng cơ sở dữ liệu gồm sáu giao tác T1, T2, T3, T4, T5 thể hiện trong Bảng 1.1 có sáu mặt hàng tương ứng với các mục là a, b, c, d, e, f.

Độ hỗ trợ của tập mục X trong CSDL giao tác T, ký hiệu: sup(X) là tỉ lệ số giao tác trong CSDL có chứa tập mục X trên tổng số các giao tác của T.

Sup(X) $= \frac{k}{m}$ Với k là số giao tác có chứa tập mục X; m là tổng số giao tác trong CSDL giao tác T.

Độ hỗ trợ được thể hiện trong bảng 1.1 với ngưỡng hỗ trợ nhỏ nhất (*Min_sup)* bằng *3.*

Bảng 1.4: Bảng cơ sở dữ liệu

TID	**Mục**
T1	a, b, e
T2	c, d, e, f
T3	b, c, e, f
T4	a, b, c, d, e
T5	b, c

Bảng 1.5: Duyệt CSDL lần 1

Mục	**Độ hỗ trợ**
{a}	2
{b}	4
{c}	4
{d}	2
{e}	4
{f}	2

Bảng 1.6: Lọc mục độ hỗ trợ ≥ 3

Mục	Độ hỗ trợ
{b}	4
{c}	4
{e}	4

Bảng 1.7: Kết hợp các mục từ 1.4

Mục	Độ hỗ trợ
{b, c}	3
{b, e}	3
{c, e}	3

Bảng 1.8: Lọc mục độ hỗ trợ ≥ 3

Mục	Độ hỗ trợ
{b, c}	3
{b, e}	3
{c, e}	3

Bảng 1.9: Kết hợp các mục từ 1.4

Mục	Độ hỗ trợ
{b, c, e}	2

Với CSDL cho trong Bảng 1.1 Thuật toán này thực hiện như sau:

Cho CSDL bảng 1.1

Duyệt bảng 1.4 ta có danh sách các mục và độ hỗ trợ kèm theo (Bảng 1.5)

Tiến hành loại các mục có độ hỗ trợ < *Min_sup* ta được bảng 1.6

Duyệt lần 2 bảng 1.4 bao gồm các tập mục và độ hỗ trợ (Bảng 1.7)

Tiến hành loại các mục có độ hỗ trợ < *Min_sup* ta được bảng 1.8

Duyệt lần 3 ta được bảng 1.4 bao gồm các tập mục và độ hỗ trợ (Bảng 1.9)

1.2. Tập mục độ hữu ích cao và bài toán khai phá tập mục độ hữu ích cao

Khi thực hiện khai phá tập phổ biến người ta đã bỏ qua giá trị độ hữu ích được gắn với mỗi mục. Có những tập mục không phải là tập phổ biến (có tần suất xuất hiện thấp) nhưng lại có giá trị độ hữu ích cao hơn nhiều so với tập phổ biến. Trong thực tế, việc khai phá các tập mục mang giá trị độ hữu ích cao là rất quan trọng và có ý nghĩa rất lớn trong đời sống xã hội. Từ đó dẫn đến một hướng nghiên cứu mới trong khai phá dữ liệu, đó là khai phá tập mục độ hữu ích cao.

Cụ thể, một siêu thị kinh doanh hàng trăm mặt hàng từ nhiều nhà cung cấp khác nhau. Họ bày bán các mặt hàng theo từng khu vực, việc sắp xếp các mặt hàng phụ thuộc vào chiến lược kinh doanh, kích thích khách hàng. Mỗi mặt hàng được bán

sẽ đem lại một giá trị lợi nhuận được xác định là chênh lệch giữa giá bán và giá mua. Theo đó, mỗi khách hàng vào siêu thị mua một vài mặt hàng với số lượng nhất định, tập hợp tất cả sản phẩm khách hàng mua sẽ đem lại một giá trị lợi nhuận cho siêu thị, được gọi là một giao tác. Tất cả các giao tác sẽ được siêu thị lưu trữ lại và tạo ra một cơ sở dữ liệu giao tác. Người quản lý siêu thị muốn tập hợp tất cả sản phẩm mà khách hàng đã mua đem lại lợi nhuận cho siêu thị (ví dụ: 30% tổng lợi nhuận), từ đó đưa ra các chiến lược kinh doanh, tiếp thị hoặc sắp xếp các mặt hàng cạnh nhau và đưa ra các chương trình khuyến mãi, khuyến khích khách hàng mua sản phẩm này thì sẽ mua thêm một sản phẩm khác trong các sản phẩm đã tìm ra.

Bài toán khai phá tập mục độ hữu ích cao đã được nhóm tác giả R.C. Chan, Q. Yang, Y.D. Shen đề xuất vào năm 2003 [27]. Cùng với sự phát triển của nền kinh tế, nhu cầu tính toán doanh thu, hiệu quả kinh doanh theo thời gian thực với lượng dữ liệu lớn ngày càng trở nên cấp thiết.

Khai phá tập mục độ hữu ích cao là bài toán mở rộng và tổng quát của khai phá tập phổ biến. Trong khai phá tập mục độ hữu ích cao, giá trị của mục trong giao tác được quan tâm nhiều nhất (như số lượng đã bán của mặt hàng), ngoài ra còn có bảng lợi nhuận cho biết độ hữu ích mang lại khi bán mặt hàng đó. Độ hữu ích của tập mục là số đo lợi nhuận của tập mục đóng góp trong cơ sở dữ liệu, nó có thể là tổng lợi nhuận hay tổng chi phí của tập mục.

Một trong những lý do của khai phá tập mục độ hữu ích cao là khám phá ra tất cả các tập mục có độ hữu ích không nhỏ hơn ngưỡng độ hữu ích tối thiểu do người dùng quy định. Từ đó xác định được các tập mục độ hữu ích cao, các tập mục độ hữu ích cao nhạy cảm. Sau đó xây dựng các phương pháp bảo vệ các dữ liệu nhạy cảm, làm hạn chế các thông tin nhạy cảm bị lộ ra ngoài, nhất là trong kinh doanh.

Bài toán Khai phá tập mục độ hữu ích cao được sử dụng trên cơ sở dữ liệu giao tác. Mỗi giao tác có thể là một giao tác mua hàng, một truy cập internet. Luận văn này sử dụng CSDL giao tác như sau:

Bảng 1.10: Cơ sở dữ liệu giao tác

TID	Transaction (Item, InUtility)
T1	(a,10), (b,2), (e,5)
T2	(c,4), (d,2), (e,7), (f,15)
T3	(b,15), (c,15), (e,1), (f,1)
T4	(a,5), (b,4), (c,20), (d,2), (e,5)
T5	(b,25), (c,15)
T6	(a,15), (e,7), (f,15)
T7	(a,25), (c,15), (d,40)
T8	(b,15), (d,35), (e,3)
T9	(a,5), (b,10), (c,20), (d,30), (e,2), (f,3)

Bảng 1.11: Bảng lợi nhuận

Item	a	b	c	d	e	f
Profit	7	2	1	1	5	10

Bảng 1.12: Bảng HUI

$minutil = 250$

HID	Itemset	Utility
1	ef	425
2	a	422
3	acd	372
4	aef	367
5	ae	342
6	f	340
7	af	322
8	ad	317
9	ac	300
10	cdef	281
11	cef	279
12	def	257

Một số khái niệm về khai phá tập mục độ hữu ích cao:

Cho $I = \{i_1, i_2, \dots, i_m\}$ là một tập m mục (item) phân biệt, trong đó mỗi mục $i_p \in I$ có độ hữu ích bên ngoài (được gọi là lợi nhuận) $eu(i_p), 1 \le p \le m$ và $D = \{T_1, T_2, \dots, T_n\}$ là một cơ sở dữ liệu (CSDL) giao tác, trong đó T_i là một giao tác chứa một tập các mục được chứa trong I.

Một tập gồm một hoặc nhiều mục được gọi là tập mục (itemset). Một giao tác T hỗ trợ một tập mục X nếu $X \subseteq I$. Một tập mục $X = \{i_1, i_2, \dots, i_k\}$ chứa k mục được gọi là k-itemset. Mỗi mục i_p trong giao tác T_q được kết hợp với một số lượng các mục i_p có trong giao tác T_q.

Cho CSDL giao tác như Bảng 1.10, Bảng 1.11 chứa lợi nhuận của các giao tác và Bảng 1.12 chứa các tập mục độ hữu ích cao. Luận văn sử dụng một số định nghĩa như sau:

Định nghĩa 1.1: Số lượng mục i_p trong giao tác T_q, ký hiệu là $iu(i_p, T_q)$.

Ví dụ: trong Bảng 1.10 có $iu(b, T_8) = 15$ và $iu(d, T_8) = 35$.

Định nghĩa 1.2: Lợi nhuận của mục i_p, thể hiện độ quan trọng của mục i_p, ký hiệu là $eu(i_p)$.

Ví dụ: trong Bảng 1.11 có $eu(b) = 2$ và $eu(d) = 1$.

Định nghĩa 1.3: Độ hữu ích của mục i_p trong giao tác T_q, ký hiệu là $u(i_p, T_q)$, được tính như sau: $u(i_p, T_q) = iu(i_p, T_q) * eu(i_p)$.

Ví dụ: $u(b, T_8) = iu(b, T_8) * eu(b) = 15 * 2 = 30$.

Định nghĩa 1.4: Độ hữu ích của tập mục X trong giao tác T_q, ký hiệu là $u(X, T_q)$ được tính như sau:

$$u(X, T_q) = \sum_{i_p \in X} u(i_p, T_q)$$

Ví dụ: $u(bd, T_8) = u(b, T_8) + u(d, T_8) = 15 * 2 + 35 * 1 = 65$.

Định nghĩa 1.5: Độ hữu ích của tập mục X, ký hiệu là $u(X)$, được tính như sau:

$$u(X) = \sum_{X \subseteq T_q \wedge T_q \in D} u(X, T_q)$$

Ví dụ: $u(bd) = u(bd, T_4) + u(bd, T_8) + u(bd, T_9) = 10 + 65 + 50 = 125$.

Định nghĩa 1.6: Độ hữu ích của giao tác T_q, ký hiệu là $tu(T_q)$, được tính như sau:

$$tu(T_q) = \sum_{i_p \in T_q} u(i_p, T_q)$$

Ví dụ: $tu(T_8) = u(b, T_8) + u(d, T_8) + u(e, T_8) = 15 * 2 + 35 * 1 + 3 * 5 = 80$

Định nghĩa 1.7: Bài toán khai phá tập mục độ hữu ích cao. Một tập mục X được gọi là tập mục độ hữu ích cao nếu độ hữu ích của X lớn hơn hoặc bằng ngưỡng độ hữu ích tối thiểu do người dùng quy định, ký hiệu là minutil. Gọi HUI là tập hợp các tập mục độ hữu ích cao, ta có $HUI = \{X \mid X \in I, u(X) \geq minutil\}$.

1.3. Một số thuật toán khai phá tập mục độ hữu ích cao

Bài toán khai phá tập mục độ hữu ích cao giúp giải quyết vấn đề mà bài toán khai phá tập phổ biến không giải quyết được. Trong khai phá tập mục độ hữu ích cao các mục có thể xuất hiện nhiều lần trong một giao tác, mỗi mục có một trọng số (lợi nhuận, độ hữu ích…). Kết quả của khai phá tập mục độ hữu ích cao được ứng dụng để tìm ra các tập mục trong cơ sở dữ liệu mang lại lợi nhuận cao.

Hiện có nhiều nhà nghiên cứu và đề xuất ra các thuật toán khai phá tập mục độ hữu ích cao hiệu quả. Năm 2005, Liu và các đồng sự đề xuất thuật toán Two-Phase với các khái niệm về độ hữu ích của giao tác (Transaction Utility - TU) và độ hữu ích của giao tác có trọng số (Transaction Weighted Utility -TWU) để cải tiến không gian tìm kiếm khai phá tập mục độ hữu ích cao [17]. Giá trị TWU của tập mục độ hữu ích thỏa mãn tính bao đóng giảm, do đó hoàn toàn có thể dựa vào TWU và sửa đổi các thuật toán khai phá tập phổ biến để khai phá tập mục độ hữu ích cao. Vì vậy, tác giả đã sửa đổi thuật toán Apriori để khai phá tập mục độ hữu ích cao.

Thuật toán Two-Phase bao gồm hai giai đoạn chính. Giai đoạn 1 tìm tất cả tập mục có độ hữu ích lớn hơn ngưỡng do người dùng quy định dựa trên độ hữu ích của giao tác có trọng số. Trong giai đoạn 1 chỉ có những kết hợp của những tập mục độ hữu ích cao của giao tác có trọng số mới được thêm vào tập ứng viên trong suốt quá trình tìm kiếm thông tin. Tuy các tập mục có độ hữu ích thấp có thể được đánh giá cao nhưng thuật toán lại không đánh giá thấp bất kỳ tập mục nào. Giai đoạn 2 duyệt cơ sở dữ liệu để lọc ra các tập mục có độ hữu ích cao từ tập mục độ hữu ích cao được tìm thấy trong giai đoạn 1. So với các thuật toán khai phá tập mục độ hữu ích cao hiện nay, thuật toán Two-Phase gặp vấn đề là một số lượng rất lớn các tập ứng viên được tạo ra nhưng hầu hết các ứng viên được sinh ra là có độ hữu ích không cao sau khi các giá trị độ hữu ích này được tính chính xác ở giai đoạn 2 của thuật toán. Ngoài ra, thuật toán thực hiện duyệt cơ sở dữ liệu nhiều lần sẽ gặp vấn đề về tốc độ xử lý nếu cơ sở dữ liệu có lượng giao tác lớn.

Để giải quyết các vấn đề liên quan đến việc có nhiều tập ứng viên được sinh ra làm giảm năng suất thực hiện của thuật toán Two-Phase. Tseng và các đồng sự đã đề xuất thuật toán UP-Growth vào năm 2010 [18]. Thuật toán UPGrowth gồm hai bước chính. Bước 1, xây dựng cấu trúc cây Up-Tree. Bước 2, xác định các tập mục độ hữu ích cao từ các tập mục hữu ích cao tiềm năng (PHUIs). Trong giai đoạn đầu, thuật toán duyệt cơ sở dữ liệu để tính toán TWU cho từng mục. Sau đó, ở giai đoạn hai, thuật toán duyệt cơ sở dữ liệu và loại bỏ những mục có giá trị TWU nhỏ hơn ngưỡng độ hữu ích tối thiểu do người dùng quy định ra khỏi giao tác tương ứng. Mặc dù hướng tiếp cận này của thuật toán UPGrowth sinh ra ít ứng viên hơn trong giai đoạn 1. Việc duyệt CSDL gốc vẫn rất tốn thời gian do CSDL gốc quá lớn và vẫn còn chứa nhiều mục không triển vọng.

Theo đó, một cải tiến của thuật toán Up-Growth [18] được Tseng và các đồng sự đề xuất vào năm 2013 cũng nhằm mục đích khai phá các tập mục độ hữu ích cao, và được gọi tên là UpGrowth+ [19]. Thuật toán áp dụng các kỹ thuật cắt tỉa để rút gọn các tập ứng viên. Sau khi tối ưu trên cây Up-Tree chúng ta sẽ có được các tập mục độ hữu ích cao tiềm năng (PHUIs) ít hơn so với Up-Growth. Thuật toán này được đánh giá là dễ cài đặt và có thời gian thực thi tốt hơn thuật toán Up-Growth vì chỉ thực hiện duyệt cơ sở dữ liệu hai lần.

Liu và Qu đã đề xuất thuật toán HUI-Miner (High Utility Itemset Miner) [20] để khai phá tập mục độ hữu ích cao sử dụng một cấu trúc mới, được gọi là danh sách lợi ích, để lưu trữ tất cả các thông tin hữu ích về một tập và tìm ra thông tin để cắt tỉa không gian tìm kiếm. Thuật toán HUI-Miner được xem là thuật toán tốt nhất để khai phá tập mục độ hữu ích cao cho đến khi có sự xuất hiện của thuật toán FHM [21], một thuật toán khai phá tập mục độ hữu ích cao được đề xuất bởi Phillipe và các đồng sự vào năm 2014.

Mỗi thuật toán đều phát huy hiệu quả chiến lược tỉa ứng viên của mình và đẩy nhanh tốc độ tìm kiếm tập mục độ hữu ích cao. Tuy nhiên, trong quá trình khai phá, các thuật toán vẫn quét các giao tác rỗng và chưa có phương án xử lý các dòng dữ

liệu tương đồng với nhau (giống các phần tử xuất hiện trong giao tác và chỉ khác số lượng).

Năm 2014, Philippe Fournier và cộng sự [3] xem xét thấy rằng HUI-Miner thực hiện khai phá một giai đoạn, không tạo các tập ứng viên theo mô hình hai giai đoạn. Do đó HUI-Miner tiêu tốn thời gian cho việc liên kết để tạo ra các tập và tốn thời gian để đánh giá độ hữu ích của mỗi tập. Để giảm các liên kết cần thực hiện, Philippe và cộng sự đề xuất một chiến lược cắt tỉa mới gọi là EUCP (Estimated Utility Cooccurrence Pruning). Phương pháp này cho phép cắt tỉa không cần ghép nối dựa trên ước tính độ hữu ích các cặp phần tử cùng xuất hiện. Thuật toán này có tên là FHM (Fast High-utility Minner). Thực nghiệm so sánh FHM với thuật toán HUI-Miner cho thấy giảm 95% các kết nối và nhanh hơn sáu lần.

Đồng thời, đã có nhiều thuật toán được phát triển nhằm nâng cao hiệu quả khai phá HUI, trong đó EFIM (EFfcient high utility Itemset Mining) là thuật toán mới nhất áp dụng nhiều kỹ thuật để cải thiện tốc độ và không gian tìm kiếm. Tuy nhiên, EFIM vẫn còn tốn nhiều chi phí quét các dòng dữ liệu để xác định sự liên quan đến ứng viên đang xét làm giảm hiệu quả của thuật toán, đặc biệt là đối với cơ sở dữ liệu thưa.

Năm 2017, Bảy Võ và cộng sự đề xuất một thuật toán cải tiến từ EFIM (IEFIM - Improve EFfcient high utility Itemset Mining). Thuật toán đề xuất dùng giải pháp chiếu ngược P-set để giảm số lượng giao tác cần xét trong thuật toán EFIM và làm giảm thời gian khai phá HUI. Thuật toán IEFIM làm giảm đáng kể số lượng giao tác cần xét và thời gian thực thi trên các CSDL thưa.

1.4. Kết luận Chương 1

Bài toán khai phá tập mục độ hữu ích cao đã tìm ra các giá trị hữu ích dựa trên ngưỡng tối thiểu do người dùng quy định. Tuy nhiên, trong kinh doanh dữ liệu cần được chia sẻ để cùng nhau hợp tác. Do đó, vấn đề đặt ra là làm thế nào để dữ liệu vẫn được chia sẻ giữa các doanh nghiệp mà vẫn đảm bảo được tính bảo mật trong dữ liệu. Để giải quyết vấn đề đó, bài toán ẩn tập mục có độ hữu ích cao được đề xuất.

CHƯƠNG 2: MỘT SỐ PHƯƠNG PHÁP ẨN TẬP MỤC ĐỘ HỮU ÍCH CAO

2.1. Một số khái niệm cơ bản

Phương pháp ẩn các tập mục có độ hữu ích cao nhạy cảm (gọi tắt là tập mục nhạy cảm) là nhằm bảo vệ các thông tin nhạy cảm trong các cơ sở dữ liệu giao tác, sao cho chúng không thể khám phá được bằng các phương pháp khai phá tập mục độ hữu ích cao với cùng một ngưỡng độ hữu ích tối thiểu do người dùng quy định.

Sửa đổi cơ sở dữ liệu là quá trình chuyển đổi cơ sở dữ liệu ban đầu thành một cơ sở dữ liệu đã được sửa đổi, sao cho không thể khai phá các tập mục nhạy cảm (sensitive itemset) từ cơ sở dữ liệu đã sửa đổi và giảm thiểu các hiệu ứng phụ trên các tập mục không nhạy cảm (non-sensitive itemset).

Trong luận văn này sử dụng một số định nghĩa sau được tham khảo trong công trình [3,4,13,14,15,16].

Cho các tập mục có độ hữu ích cao nhạy cảm (gọi tắt là: tập mục nhạy cảm) cần phải ẩn, ký hiệu là SHUI = $\{S_1, S_2, \ldots, S_m\}$, trong đó $S_d \in$ SHUI, $(1 \leq d \leq m)$. Bài toán ẩn tập mục nhạy cảm là việc sửa đổi CSDL D ban đầu thành CSDL D' sao cho độ hữu ích của tất cả tập mục nhạy cảm $S_d \in$ SHUI phải nhỏ hơn ngưỡng độ hữu ích tối thiểu do người dùng quy định, tức là $u(S_i) < \text{minutil}, \text{với } i = 1 \div m$.

Định nghĩa 2.1: Gọi SHUI = $\{S_1, S_2, \ldots, S_m\}$ là tập hợp các tập mục có độ hữu cao nhạy cảm (viết tắt là: tập mục nhạy cảm), trong đó S_i là tập mục nhạy cảm cần được ẩn trước khi đưa CSDL ra bên ngoài, ta có SHUI, HUI. Gọi NSHUI là tập hợp các tập mục độ hữu ích cao không nhạy cảm (gọi tắt là: tập mục không nhạy cảm), ta có SHUI $\cup$ NSHUI $=$ HUI.

Định nghĩa 2.2: Gọi ST là tập hợp các giao tác nhạy cảm mà mỗi giao tác trong ST có chứa ít nhất một tập mục nhạy cảm.

Quá trình sửa đổi dữ liệu của bài toán ẩn các tập mục nhạy cảm gồm ba bước sau:

Bước 1: Áp dụng các thuật toán khai phá độ hữu ích cao trên cơ sở dữ liệu giao tác D để có được tất cả các tập mục độ hữu ích cao (HUI);

Bước 2: Xác định tập hợp các tập mục nhạy cảm (các tập mục độ hữu ích cao nhạy cảm) SHUI dựa trên các yêu cầu của người dùng;

Bước 3: Áp dụng thuật toán ẩn các tập mục nhạy cảm để tạo ra cơ sở dữ liệu được sửa đổi D'.

Hình 2.1: Quá trình sửa đổi cơ sở dữ liệu

2.2. Một số công trình liên quan

Những năm gần đây, phương pháp khai phá tập mục có độ hữu ích bảo vệ tính riêng tư được nhiều nhà nghiên cứu quan tâm. Bài toán này trở nên quan trọng vì nó xem xét cả số lượng và lợi nhuận của mỗi mục có trong cơ sở dữ liệu giao tác để ẩn các tập mục có độ hữu ích cao nhạy cảm. Vì mục đích của khai phá tập mục có độ hữu ích cao bảo vệ tính riêng tư để ẩn các thông tin nhạy cảm trong cơ sở dữ liệu, trong khi đó vẫn đảm bảo các thông tin quan trọng khác vẫn được chia sẽ với nhau và bài toán này được xem như là bài toán tối ưu. Việc tìm ra các giao tác và mục sửa đổi trong quá trình ẩn các tập mục có độ hữu ích cao nhạy cảm một cách tối ưu là một bài toán khó và không khả thi.

Năm 2010, Yeh và cộng sự [15] là nhóm tác giả đầu tiên đưa ra hai thuật toán heuristic HHUIF và MSICF để ẩn các tập mục có độ hữu ích cao nhạy cảm. Hai thuật toán chọn mục độ hữu ích cao nhất làm mục sửa đổi cho quá trình ẩn. Thuật toán HHUIF loại bỏ các mục có độ hữu ích cao nhất. Thuật toán MSICF xem xét số lượng xung đột trong quá trình ẩn.

Sau đó, có một số tác giả khác cũng đề xuất các thuật toán nhằm cải tiến hai thuật toán trên, như Vo, B và cộng sự (2013) [14] đề xuất thuật toán nhằm cải tiến thuật toán HHUIF về mặt thời gian.

Selvaraj và cộng sự (2013) [13] đề xuất một thuật toán cải tiến MHIS, chọn mục sửa đổi trong trường hợp độ hữu ích của chúng như nhau. Kết quả cho thấy thuật toán MHIS tốt hơn thuật toán HHUIF về các hiệu ứng phụ HF (không ẩn được) và MC (ẩn nhầm).

Yun và Kim (2015) [16] đề xuất thuật toán FPUTT để cải thiện tính hiệu quả của thuật toán HHUIF bằng cách sử dụng cấu trúc cây. Kết quả nhanh hơn HHUIF khoảng 5 đến 10 lần. Tuy nhiên, các hiệu ứng phụ tạo ra cũng giống như HHUIF.

Trieu và cộng sự (2020) [4] đề xuất cải tiến thuật toán HHUIF. Thuật toán này nhằm mục đích sửa đổi số lượng các mục trong giao tác sửa đổi để ẩn các tập mục có độ hữu ích cao nhạy cảm. Kết quả cho thấy, thuật toán này hiệu quả hơn HHUIF và MSICF về các hiệu ứng phụ và thời gian chạy.

2.3. Phương pháp ẩn tập mục độ hữu ích cao nhạy cảm

Mục tiêu bài toán: Ẩn các tập mục có độ hữu ích cao nhạy cảm và giảm hiệu ứng phụ đối với tri thức không nhạy cảm do quá trình sửa đổi gây ra.

Trieu và cộng sự (2020) [4] đề xuất cải tiến thuật toán HHUIF. Thuật toán này nhằm mục đích sửa số lượng các mục trong giao tác sửa đổi, để ẩn các tập mục có độ hữu ích cao nhạy cảm. Thuật toán hiệu quả hơn HHUIF và MSICF về các hiệu ứng phục và thời gian chạy.

Thuật toán EHSHUI bao gồm ba bước heuristic:

- Giao tác chứa $S_i \in$ SHUI có độ hữu ích cao nhất được chọn là giao tác sửa đổi.

- Mục tác động đến các NSHUI ít nhất được chọn làm mục sửa đổi.

- Xác định giá trị độ hữu ích: $diffu = u(Si) - minutil + 1$ để giảm số lượng của mục i_{vic} từ giao tác T_{vic}.

Luận văn sẽ nghiên cứu và tìm hiểu thuật toán EHSHUI [4]. Sau đó luận văn sẽ đề xuất phương pháp ẩn tập mục độ hữu ích cao nhạy cảm nhằm khắc phục những hạn chế của thuật toán EHSHUI. Chi tiết thuật toán đề xuất được trình bày trong Chương 3.

Chiến lược ẩn các tập mục độ hữu ích cao nhạy cảm trong CSDL giao tác là sửa đổi CSDL, bằng cách giảm số lượng hoặc loại bỏ một số mục trong CSDL sao cho độ hữu ích của tập mục nhạy cảm giảm xuống dưới ngưỡng độ hữu ích tối thiểu (minutil). Quá trình sửa đổi tập trung vào ba nhiệm vụ sau:

- Lựa chọn giao tác để sửa đổi là giao tác có độ hữu ích cao nhất.

- Lựa chọn mục tác động đến các NSHUI ít nhất được chọn làm mục sửa đổi.

- Xác định giá trị độ hữu ích: diffu = $u(S_i)$ - minutil + 1 để giảm số lượng của mục i_{vic} từ giao tác T_{vic}.

Thuật toán [4] thực hiện như sau: Đầu tiên, thuật toán tính tần suất của tất cả SHUI: $f_{SHUIs}(S_i)$ (dòng 1). Sau đó, nó sắp xếp các SHUI theo thứ tự giảm dần của $f_{SHUIs}(S_i)$ (dòng 2). Điều này, nhằm mục đích ưu tiên đầu tiên cho tập mục có tần số cao nhất, cho ẩn để giảm sửa đổi dữ liệu.

Tiếp theo, thuật toán sẽ quét $S_i \in$ SHUIs và thực hiện quá trình ẩn (dòng 3). Đối với mỗi S_i, thuật toán thực hiện các bước sau: Tạo một phép chiếu trên cơ sở dữ liệu (DS_i) bao gồm các giao tác có chứa S_i, nhằm giảm thời gian truy cập cơ sở dữ liệu để tìm mục cần sửa đổi (dòng 4). Xác định giá trị độ hữu ích tối thiểu cần phải giảm để ẩn S_i: diffu = $u(S_i)$ - minutil + 1 (dòng 5). Trong khi, diffu > 0 (dòng 6), nó lặp lại quá trình chọn giao tác cần sửa, mục cần sửa và sửa đổi dữ liệu. Chiến lược xác định T_{vic} (dòng 7) và i_{vic} (dòng 8) quyết định đến các hiệu ứng phụ. Sau khi xác định T_{vic} và i_{vic}, thuật toán tính toán số lượng i_{vic} cần giảm để ẩn S_i, có hai trường hợp (dòng 9):

Nếu $u(i_{vic}, T_{vic})$ > diffu, thì giảm số lượng của i_{vic} trong T_{vic} một giá trị: dec = $\left\lceil \frac{\text{diffu}}{\text{eu}(i_{vic})} \right\rceil$, Tập mục S_i sẽ bị ẩn (dòng 10) và $iu(i_{vic}, T_{vic})$ được cập nhật một giá trị mới là: $iu(i_{vic},T_{vic}) = iu(i_{vic}, T_{vic}) – dec$ (dòng 11). Nếu $iu(i_{vic}, T_{vic})$ = dec thì i_{vic} bị loại khỏi

T_{vic}. Cuối cùng, khi S_i bị ẩn, diffu được gán giá trị (diffu=0) để kết thúc quá trình ẩn (dòng 12).

Nếu u(i_{vic}, T_{vic}) ≤ diffu, gán một giá trị mới cho diffu là: $diffu = diffu - u(Si, Tvic)$ và xóa i_{vic} khỏi T_{vic} (dòng 14–15). Thuật toán EHSHUI kết thúc khi mọi SHUI bị ẩn.

Input: Cơ sở dữ liệu giao tác D, tập mục có độ hữu ích cao HUI;

Tập mục có độ hữu ích cao nhạy cảm SHUI = {S_1, S_2,…. S_s}

Ngưỡng tối thiểu minutil

Output: Cơ sở dữ liệu đã sửa đổi D'

```
1   Computer f_HSUIs(S_i), 1 ≤ i ≤ |SHUIs| ∧ S_i ∈ SHUIs;
2   Sort SHUIs in decreasing order of f_HSUIs(S_i);
3   foreach (S_i ∈ SHUIs) do
4       DSi = projectData (D, S_i);
5       diffu = u(S_i) - minutil + 1;
6       while (diffu > 0) do
7           T_vic = findVictimTransaction (DSi, S_i);
8           i_vic = findVictimItem (S_i,T_vic);
9           if (u (i_vic, T_vic) > diffu) then
10              dec = ⌈diffu / eu(i_vic)⌉;
11              iu(i_vic, T_vic) = iu(i_vic, T_vic) – dec;
12              diffu = 0;
13          else
14              diffu = diffu – u(S_i, T_vic);
15              remove i_vic from T_vic ;
16  Update (D);
```

Ví dụ minh họa: Chạy thử thuật toán trên với CSDL trong bảng 1.10, bảng 1.11 và bảng 1.12, với tập mục nhạy cảm SHUI = {acd, cdef}.

Xây dựng các Bảng: I-List; HUI-Table; T-Table

Bảng 2.1: Bảng I-List thuật toán EHSHUI

TID	Giao tác (Mục, số lượng)	TU(T)	I-List
T1	(a,10),(b,2),(e,5)	99	(a,10,70) (b,2,4) (e,5,25)
T2	(c,4), (d,2), (e,7), (f,15)	191	(c,4,4) (d,2,2), (e,7,35), (f,15,150)
T3	(b,15), (c,15), (e,1), (f,1)	60	(b,15,30) (c,15,15) (e,1,5) (f,1,10)
T4	(a,5), (b,4), (c,20), (d,2), (e,5)	90	(a,5,35)(b,4,8)(c,20,20)(d,2,2)(e,5,25)
T5	(b,25), (c,15)	65	(b,25,50)(c,15,15)
T6	(a,15), (e,7), (f,15)	290	(a,15,105)(e,7,35)(f,15,150)
T7	(a,25), (c,15), (d,40)	230	(a,25,175)(c,15,15)(d,40,40)
T8	(b,15), (d,35), (e,3)	80	(b,15,30)(d,35,35)(e,3,15)
T9	(a,5),(b,10),(c,20),(d,30),(e,2),(f,3)	145	(a,5,35)(b,10,20)(c,20,20)(d,30,30)(e,2,10)(f,3,30)

Xây dựng Bảng HUI-Table

Bảng 2.2: Bảng HUI-Table thuật toán EHSHUI

HID	Itemset	Utility	TIDs
1	ef	425	T2, T3,T6,T9
2	a	422	T1,T4,T6,T7,T9
3	acd	372	T4,T7,T9
4	aef	367	T6,T9
5	ae	342	T1,T4,T6,T9
6	f	340	T2,T3,T6,T9
7	af	322	T6,T9
8	ad	317	T4,T7,T9
9	ac	300	T4,T7,T9
10	cdef	281	T2,T9
11	cef	279	T2,T9
12	def	257	T2,T9

Xây dựng Bảng T-Table

Bảng 2.3: Bảng T-Table thuật toán EHSHUI

TID	SID	NSID	I-List
T2	10	1,6,11,12	(c,4,4) (d,2,2) (e,7,35) (f,15,150)
T4	3	2,5,8,9	(a,5,35) (b,4,8) (c,20,20) (d,2,2) (e,5,25)
T7	3	2,8,9	(a,25,175)(c,15,15)(d,40,40)
T9	3,10	1,2,4,5,6,7,8,9,11,12	(a,5,35)(b,10,20)(c,20,20)(d,30,30)(e,2,10)(f,3,30)

Sắp xếp các S_i trong SHUI giảm dần theo tần suất: f(acd) = 3, f(cdef) =2

*** Chọn ngẫu nhiên: S_1 = {cdef}** để ẩn trước và có 2 giao tác được hỗ trợ là T2, T9.

Độ hữu ích của S_1: u(cdef) = 281 và munitil = 250, muốn ẩn S_1 thì độ hữu ích của S_1 phải < 250.

Tính toán diffu = 281 – 250 + 1 = 32, muốn ẩn S_1 thì phải giảm độ hữu ích $u(S_1)$ ít nhất là 32.

Bảng 2.4: Bảng CSDL chiếu trên S_1 ta được T-Table:

TID	SID	NSID	I-List
T2	10	1,6,11,12	(c,4,4) (d,2,2), (e,7,35), (f,15,150)
T9	3,10	1,2,4,5,6,7,8,9,11,12	(c,20,20)(d,30,30)(e,2,10)(f,3,30)

Tìm giao tác sửa đổi T_{vic} cần sửa đổi:

u(cdef, T2) = 191

u(cdef, T9) = 90

→ Chọn T2 làm giao tác sửa đổi

Tìm mục sửa đổi mà I_{vic} cần sửa đổi:

u(c, T2) = 4; u(d, T2) = 2; u(e, T2) = 35; u(f, T2) =150

→ Có u(e, T2) và u(f, T2) lớn hơn diffu

Tính toán các mục e phải giảm để ẩn các tập mục nhạy cảm S_1: dec = $\left\lceil \frac{\text{diffu}}{\text{eu}(i_{vic})} \right\rceil$

dec(e, T2) = |diffu/eu(e)| = 32/5 = 7 → coi như loại e ra khỏi giao tác T2

Tính độ hữu ích của tập mục SID và NSID

u(cdef) = 281 - 191 = 90 < minutil = 250 → được ẩn

→ u(ef) = 425 - 35 - 150 = 240 < minutil = 250 → ẩn nhầm

→ u(f) = 340 – 150 = 190 < minutil = 250 → ẩn nhầm

→ u(cef)= 279 – 4 – 35 - 150 = 90 < minutil = 250 → ẩn nhầm

→ u(def) = 257 – 2 – 35 - 150 = 70 < minutil = 250 → ẩn nhầm

Tính toán các mục f phải giảm để ẩn các tập mục nhạy cảm S_1: dec = $\left\lceil \frac{\text{diffu}}{\text{eu}(i_{vic})} \right\rceil$

dec(f, T2) = |diffu/eu(f)| = 32/10 = 4 → giảm f đi 4 → độ hữu ích giảm đi 40

Tính độ hữu ích của tập mục SID và NSID

u(cdef) = 281- 40 = 241 < minutil = 250 → được ẩn

→ u(ef) = 425 – 40 = 385

→ u(f) = 340 - 40 = 300

→ u(cef) = 279 - 40 = 239 < minutil = 250 → ẩn nhầm

→ u(def) = 257 – 40 = 217 < minutil = 250 → ẩn nhầm

→ Vậy mục f khi sửa sẽ tạo ra ẩn nhầm ít nhất, vậy chọn mục f làm mục sửa đổi.

Cập nhật các giá trị: bảng T-Table và HUI-Table

Bảng 2.5: cập nhật lại HUI-Table (lần 1)

HID	Itemset	Utility	ẩn cdef	TIDs
1	ef	425	***385***	T2, T3,T6,T9
2	a	422	422	T1,T4,T6,T7,T9
3	acd	372	372	T4,T7,T9
4	aef	367	367	T6,T9
5	ae	342	342	T1,T4,T6,T9
6	f	340	3***00***	T2,T3,T6,T9
7	af	322	322	T6,T9
8	ad	317	317	T4,T7,T9
9	ac	300	300	T4,T7,T9
10	***cdef***	***281***	***241***	***T2,T9***
11	cef	279	2***39***	T2,T9
12	def	257	***217***	T2,T9

Bảng 2.6: cập nhật lại T-Table (lần 1)

TID	SID	NSID	I-List
T2	10	1,6,11,12	(c,4,4) (d,2,2), (e,7,35), ***(f,11,110)***
T4	3	2,5,8,9	(a,5,35)(b,4,8)(c,20,20)(d,2,2)(e,5,25)
T7	3	2,8,9	(a,25,175)(c,15,15)(d,40,40)
T9	3,10	1,2,4,5,6,7,8,9,11,12	(a,5,35)(b,10,20)(c,20,20)(d,30,30)(e,2,10)(f,3,30)

*** Tiếp tục ẩn S_2 = {acd}** và có 3 giao tác hỗ trợ là T4, T7, T9.

Độ hữu ích của S_2: u(acd) = 372 và munitil = 250, muốn ẩn S_2 thì độ hữu ích của S_2 phải < 250.

Tính toán diffu = 372 – 250 + 1 = 123, muốn ẩn S_2 thì phải giảm độ hữu ích u(S_2) ít nhất là 123.

Bảng 2.7: Bảng CSDL chiếu trên S_2

TID	SID	NSID	I-List
T4	3	2,5,8,9	(a,5,35)(c,20,20)(d,2,2)
T7	3	2,8,9	(a,25,175)(c,15,15)(d,40,40)
T9	3,10	1,2,4,5,6,7,8,9,11,12	(a,5,35)(c,20,20)(d,30,30)

Tìm giao tác sửa đổi T_{vic} cần sửa đổi:

u(acd, T4) = 57

u(acd, T7) = 230

u(acd, T9) = 85

→ Vậy chọn T_{vic} = T7

Tìm mục sửa đổi mà I_{vic} cần sửa đổi:

u(a,T7) = 175, u(c,T7) = 15, u(d,T7) = 40

→ Vậy có u(a,T7) = 175 > diffu → Ivic = a

Tính toán các mục *a* phải giảm để ẩn các tập mục nhạy cảm S_2: dec = $\left\lceil \frac{\text{diffu}}{\text{eu}(i_{vic})} \right\rceil$

dec(a, T7) = |123/7| = 18 → vậy giảm 18 a → độ hữu ích giảm đi 126

Tính độ hữu ích của tập mục SID và NSID

u(acd) = 372 - 18*7 = 246 < minutil =250 → được ẩn

→ u(a) = 422 - 18*7 = 296

→ u(ad) = 317 - 18*7 = 191 < minutil = 250 → ẩn nhầm

→ u(ac) = 300 - 18*7 = 174 < minutil = 250 → ẩn nhầm

Cập nhật các giá trị: bảng T-Table và HUI-Table

Bảng 2.8: cập nhật lại HUI-Table (lần 2)

HID	Itemset	Utility	ẩn cdef	ẩn acd	TIDs
1	ef	425	***385***	385	T2, T3,T6,T9
2	a	422	422	***296***	T1,T4,T6,T7,T9
3	***acd***	372	***372***	***246***	***T4,T7,T9***
4	aef	367	367	367	T6,T9
5	ae	342	342	342	T1,T4,T6,T9
6	f	340	***300***	300	T2,T3,T6,T9
7	af	322	322	322	T6,T9
8	ad	317	317	***191***	T4,T7,T9
9	ac	300	300	***174***	T4,T7,T9
10	***cdef***	281	***241***	241	***T2,T9***
11	cef	279	2***39***	239	T2,T9
12	def	257	***217***	217	T2,T9

Bảng 2.9: cập nhật lại T-Table (lần 2)

TID	SID	NSID	I-List
T2	10	1,6,11,12	(c,4,4) (d,2,2), (e,7,35), ***(f, 11, 110)***
T4	3	2,5,8,9	(a,5,35)(b,4,8)(c,20,20)(d,2,2)(e,5,25)
T7	3	2,8,9	***(a, 7, 35)***(c,15,15)(d,40,40)
T9	3,10	1,2,4,5,6,7,8,9,11,12	(a,5,35)(b,10,20)(c,20,20)(d,30,30)(e,2,10)(f,3,30)

2.4. Kết luận Chương 2

Thuật toán EHSHUI đã hoàn thành được việc ẩn các tập mục độ hữu ích cao nhạy cảm. Tuy nhiên, Thời gian chạy của thuật toán tăng lên cùng với sự phát triển của HUI. Lý do là tốn thời gian bằng cách tính toán để chỉ định số lượng không phải SHUI bị ảnh hưởng cho mỗi mục nhạy cảm. Sau khi nhận thấy những tồn tại của thuật toán này. Luận văn đề xuất thuật toán để cải tiến thuật toán trên. Phương pháp đề xuất thuật toán ẩn tập mục nhạy cảm hiệu quả được trình bày ở Chương 3.

CHƯƠNG 3: ĐỀ XUẤT PHƯƠNG PHÁP ẨN TẬP MỤC ĐỘ HỮU ÍCH CAO

3.1. Cơ sở để đề xuất thuật toán

Phương pháp ẩn các tập mục nhạy cảm để bảo vệ quyền riêng tư không chỉ để ẩn tất cả các tập mục nhạy cảm mà còn để giảm thiểu hiệu ứng phụ đối với thông tin không nhạy cảm và tính toàn vẹn của cơ sở dữ liệu gốc. Trên cơ sở phương pháp ẩn tổng quát này, bài toán ẩn các tập mục nhạy cảm là sửa đổi cơ sở dữ liệu ban đầu bằng cách xóa hoặc giảm số lượng các mục để độ hữu ích của các tập mục nhạy cảm này giảm xuống dưới ngưỡng độ hữu ích tối thiểu.

Hầu hết các công trình đã tập trung vào việc xác định: giao tác nào được chọn để sửa đổi T_{vic} và mục nào được chọn để sửa đổi I_{vic} trong giao tác sửa đổi T_{vic}. Trong luận văn này, tập trung vào:

(i) Khi ẩn các tập mục nhạy cảm, thứ tự chọn ẩn tập mục nhạy cảm nào trước tiên sẽ ảnh hưởng đến quá trình ẩn và gây ra các hiệu ứng phụ không mong muốn. Luận văn chọn tập mục có độ hữu ích cao nhạy cảm lớn nhất cần được ẩn trước. Vì khi ẩn các tập mục nhạy cảm này thì có thể ẩn các tập mục nhạy cảm khác, khi đó chúng ta không cần ẩn tập mục nhạy cảm đó nữa. Điều này được chứng minh trong ví dụ minh họa. Do đó, có thể tăng hiệu quả của quá trình ẩn.

(ii) Luận văn chọn mục cần sửa đổi (i_{vic}) nằm trong số các tập mục nhạy cảm nhiều nhất để sửa đổi trước. Nếu có nhiều mục như vậy, luận văn chọn mục nằm trong số các tập mục không nhạy cảm ít nhất để sửa đổi. Điều này giảm thiểu các hiệu ứng phụ đối với các tập mục có độ hữu ích cao không nhạy cảm.

(iii) Trong hầu hết các thuật toán đã xuất bản như [4, 13, 14, 15, 16], chúng chỉ sửa đổi từng giao tác một. Điều này có thể làm tăng thời gian xử lý. Trong luận văn sử dụng hệ số α được đề xuất trong [14] để tính tỷ lệ giảm số lượng của mục cần sửa đổi i_{vic} trong tất cả các giao tác nhạy cảm hỗ trợ tập mục nhạy cảm S_i cần ẩn. Sau đó, thuật toán được đề xuất sẽ sửa đổi tất cả các giao tác nhạy cảm cùng một lúc. Điều

này làm giảm số lần quét cơ sở dữ liệu cũng như thời gian cần thiết để ẩn các tập mục nhạy cảm.

Đặt Sj là tập mục có độ hữu ích cao nhạy cảm. Để ẩn Sj, độ hữu ích của Sj phải giảm ít nhất một lượng theo công thức sau:

$$diffu = u(S_j) - minutil + 1$$

Trong đó, u(Sj) là độ hữu ích của tập mục nhạy cảm Sj và minutil là ngưỡng độ hữu ích tối thiểu.

Hệ số α được tính như sau:

$$\alpha = diu \times \frac{eu(i_p)}{sum(i_p)}$$

Trong đó, $diu = \left\lceil \frac{diffu}{eu(i_p)} \right\rceil$, $sum(i_p)$ là tổng độ hữu ích của mục i_p trong tất cả các giao tác nhạy cảm hỗ trợ S_j.

Định nghĩa 3.1: Xác định mục cần sửa đổi (i_{vic}): mục nằm trong số các tập mục nhạy cảm nhiều nhất. Nếu có nhiều mục thỏa mãn, luận văn chọn mục nằm trong số tập mục không nhạy cảm ít nhất.

Đối với các thuật toán ẩn tập mục nhạy cảm đã có, thường phải quét cơ sở dữ liệu nhiều lần, trong luận văn này tôi sẽ sử dụng cấu trúc dữ liệu được trình bày trong [4] để giảm số lần quét cơ sở dữ liệu, cấu trúc dữ liệu được giới thiệu trong các Định nghĩa 3.2, Định nghĩa 3.3 và Định nghĩa 3.4

Định nghĩa 3.2: Cho một giao tác T, danh sách mục (I-list) lưu trữ thông tin của các mục trong T. Mỗi mục i trong I-list gồm ba thành phần: $i = \langle Item, InUtility, Utility \rangle$. Trong đó Item là mục i, InUtility là số lượng của i trong T, Utility là độ hữu ích của i trong T.

Ví dụ trong Bảng 2.1, I-list của T1 là (a,10,70) (b,2,4) (e,5,25).

Định nghĩa 3.3: Cho một cơ sở dữ liệu D, một tập hợp các tập mục độ hữu ích cao $HUI = \{X \mid X \in I, u(X) \geq minutil\}$, một Bảng tập mục độ hữu ích cao (HUI-table) chứa thông tin về các tập mục độ hữu ích cao được khai thác từ D. Mỗi tập mục độ hữu ích cao X trong bảng HUI-table có bốn thành phần: X =<HID, Items, HUI-utility, TIDs>. Trong đó HID là định danh duy nhất của X, Items là danh sách các mục có trong X, HUI-utility là độ hữu ích của X, TIDs cho biết các giao tác hỗ trợ X trong D.

Với cơ sở dữ liệu giao tác cho trong Bảng 1.10 và Bảng 1.11, ngưỡng độ hữu ích tối thiểu là minutil = 250. Chúng ta xây dựng được bảng HUI-Table như trong Bảng 1.12.

Định nghĩa 3.4: Cho cơ sở dữ liệu D, một tập hợp các tập mục nhạy cảm $SHUI = \{S_1, S_2, \ldots, S_k\}$, Bảng giao tác (T-table) chứa thông tin của các giao tác nhạy cảm trong D. Mỗi giao tác T trong bảng T-table có bốn thành phần: T = <TID, SID, NSID, I-list>. Trong đó TID là mã định danh duy nhất của T, SID và NSID lần lượt là mã định danh các tập mục nhạy cảm và tập mục không nhạy cảm được hỗ trợ bởi T. I-list là danh sách mục của T.

3.2. Thuật toán đề xuất

Dựa vào thuật toán EHSHUI [4] và hệ số α [14], luận văn đề xuất một thuật toán ẩn các tập mục nhạy cảm trong CSDL giao tác lớn như trong bảng 1.10, bảng 1.11 và bảng 1.12.

Thuật toán được thực hiện như sau: Đầu tiên, các tập mục nhạy cảm được sắp xếp giảm dần theo độ hữu ích (dòng 1). Tiếp theo, thực hiện lặp để ẩn các tập mục nhạy cảm (dòng 2-13). Tính toán độ hữu ích được giảm bớt cho mỗi tập mục nhạy cảm (dòng 3). Dòng 4, tìm tập hợp các giao tác nhạy cảm hỗ trợ S_i. Xác định mục cần được sửa đổi I_{vic} theo Định nghĩa 3.1 (dòng 6). Tính số lượng các mục phải giảm để ẩn tập mục nhạy cảm S_i: tính α (dòng 7-8). (Dòng 9-12) tiến hành sửa đổi số lượng của mục cần sửa đổi I_{vic} trong các giao tác nhạy cảm ST. Nếu $\alpha < 1$, số lượng mục I_{vic} trong mỗi giao tác ST sẽ giảm theo tỷ lệ α. Nếu $\alpha \geq 1$, nó sẽ làm giảm số lượng

mục I_{vic} thành 1, để tránh sai lệch quá nhiều trong cấu trúc cơ sở dữ liệu sau khi sửa đổi. Thực hiện cho đến khi, các tập mục nhạy cảm bị ẩn đi thì thuật toán kết thúc. Thuật toán được đề xuất đảm bảo rằng, các tập mục nhạy cảm đều được ẩn.

Thuật toán IEHSHUI

Input: Cơ sở dữ liệu giao tác D, tập mục có độ hữu ích cao HUI;

Tập mục có độ hữu ích cao nhạy cảm SHUI = {S_1, S_2,…. S_s}

Ngưỡng tối thiểu minutil

Output: Cơ sở dữ liệu đã sửa đổi D'

1 Sort $SHUI$ in decreasing order of $u(S_i)$;

2 **foreach** ($S_j \in SHUI$) **do**

3 $diffu = u(S_j) - minutil + 1.$

4 Find set of sensitive transaction ST support S_i.

5 **while** ($diffu > 0$) **do**

6 Find i_{vic} by *Definition 3.1*

7 $diu = \left\lceil \frac{diffu}{eu(i_{vic})} \right\rceil$

8 Calculate factor $\alpha = diu \times \frac{eu(i_{vic})}{sum(i_{vic})}$,
where $sum(i_{vic}) = \sum_{T_q \in ST} u(i_{vic}, T_q)$

9 **for each** $T_q \in ST$ **do**

10 Modify the quantity of i_{vic}.

11 $iu(i_{vic}) = \begin{cases} iu(i_{vic}) - iu(i_{vic}) \times \alpha \; if \; \alpha < 1 \\ 1 \; if \; \alpha \geq 1 \end{cases}$

12 Modify $diffu$.

13 Update (D);

Ví dụ minh họa: Với cơ sở dữ liệu được đưa ra trong bảng 1.10, bảng 1.11 và ngưỡng độ hữu ích tối thiểu là minutil = 250, chúng ta có thể khai thác tất cả các tập mục có độ hữu ích cao HUI được trình bày trong bảng 1.12.

Bảng 1.10: Cơ sở dữ liệu giao tác

TID	Transaction (Item, InUtility)
T1	(a,10),(b,2),(e,5)
T2	(c,4), (d,2), (e,7), (f,15)
T3	(b,15), (c,15), (e,1), (f,1)
T4	(a,5), (b,4), (c,20), (d,2), (e,5)
T5	(b,25), (c,15)
T6	(a,15), (e,7), (f,15)
T7	(a,25), (c,15), (d,40)
T8	(b,15), (d,35), (e,3)
T9	(a,5),(b,10),(c,20),(d,30),(e,2),(f,3)

Bảng 1.11: Bảng lợi nhuận

Item	a	b	c	d	e	f
Profit	7	2	1	1	5	10

Bảng 1.12: Bảng HUI

$minutil = 250$

HID	Itemset	Utility
1	ef	425
2	a	422
3	acd	372
4	aef	367
5	ae	342
6	f	340
7	af	322
8	ad	317
9	ac	300
10	cdef	281
11	cef	279
12	def	257

Giả sử các tập mục nhạy cảm cần ẩn là SHUI = {ae, ef, aef}

Dòng 1: Sắp xếp theo thứ tự giảm dần của $u(S_j)$: SHUI = {ef (425), aef (367), ae (342)}

Dòng 2: chọn ẩn S_1 = {ef}

Dòng 3: tính toán $diffu = u(\text{ef}) - minutil + 1 = 425 - 250 + 1 = 176$

Dòng 4: Tìm tập các giao tác nhạy cảm hỗ trợ S_1 như: ST = {T2, T3, T6, T9}.

Dòng 5: diffu > 0

Dòng 6: Tìm mục i_{vic} cần sửa đổi: Có 2 mục e và f.

Mục e có trong các tập mục nhạy cảm: {ae}, {ef} và {aef}.

Mục f có trong các tập mục nhạy cảm: {ef} và {aef}.

Vì vậy, chọn mục e để sửa đổi vì nó nằm trong số các tập mục nhạy cảm nhiều nhất.

Dòng 7: Tính toán các mục e phải giảm để ẩn các tập mục nhạy cảm S_1 = {ef} như:

$$diu = \left\lceil \frac{diffu}{eu(e)} \right\rceil = \left\lceil \frac{176}{5} \right\rceil = 36$$

Dòng 8: Tính hệ số α cho mục e.

$$sum(e) = u(e,T2) + u(e,T3) + u(e,T6) + u(e,T9)$$
$$= 35 + 5 + 35 + 10 = 85$$

$$\alpha = diu \times \frac{eu(e)}{sum(e)} = 36 \times \frac{5}{85} = 2.12 > 1$$

Dòng 11: Vì α > 1, thuật toán IEHSHUI điều chỉnh số lượng mục e trong giao tác T2, T3, T6, T9 đến giá trị 1 (không cho về 0)

Số mục e trong T2 là 7 --> giảm mất 6 (Còn lại 1)

Số mục e trong T3 là 1 --> giữ nguyên

Số mục e trong T6 là 7 --> giảm đi 6 (Còn lại 1)

Số mục e trong T9 là 2 --> giảm 1 (Còn 1)

Vậy số lượng mục e đã giảm là: 6 + 0 + 6 + 1 = 13, Mà profit(e) = 5 Vậy đã giảm độ hữu ích đi: 13 x 5 = 65

Dòng 12: cập nhật các giá trị: Độ hữu ích của tập mục nhạy cảm S_1 = {ef}, giảm xuống còn lại là: u(ef) = 425 – 65 = 360

Cập nhật: diffu = 360 – 250 + 1 = 111

Dòng 13: Cập nhật lại cơ sở dữ liệu

Vì diffu > 0 tiếp tục quay lại dòng 5, 6. Thuật toán IEHSHUI chọn mục f để sửa đổi.

Dòng 7: Tính toán số lượng mục f cần phải giảm để ẩn tập mục nhạy cảm S_1 = {ef} thì:

$$diu = \left\lceil \frac{diffu}{eu(F)} \right\rceil = \left\lceil \frac{111}{10} \right\rceil = 12$$

Dòng 8: Tính hệ số α cho mục f

$$sum(F) = u(F,T2) + u(F,T3) + u(F,T6) + u(F,T9)$$
$$= 150 + 10 + 150 + 30 = 340$$

$$\alpha = diu \times \frac{eu(F)}{sum(F)} = 12 \times \frac{10}{340} = 0.35$$

Vì α = 0.35 < 1. Tính số mục f phải giảm trong các giao tác T2, T3, T6, T9 như sau:

Số mục f phải giảm trong T2 là 15 * 0.35 = 5

Số mục f phải giảm trong T6 là 15 * 0.35 = 5

Số mục f phải giảm lại trong T9 là 3 * 0.35 = 1

Tổng số mục f cần phải giảm để ẩn {ef} là: 12

Vậy số mục f phải giảm trong T3 là 12 – 5 – 5 - 1 = 1

Nhưng số mục f trong T3 chỉ là 1, vì vậy khi giảm một mục f khỏi giao tác T3, nó được coi là loại bỏ mục f khỏi giao tác T3. Do đó, T3 sẽ không hỗ trợ tập mục nhạy cảm {ef}. Độ hữu ích của tập mục {ef} phải giảm đi u(ef, T3) khi f bị loại bỏ khỏi giao tác T3.

Cập nhật lại các giá trị: u(ef) = 360 - 5*10 – 5*10 – 1*10 – u(ef,T3) = 360 - 110 - 15= 235 < $minutil$ = 250

Như vậy mục tập mục S_1 = <ef> đã được ẩn thành công.

Diffu = 235 – 250 + 1= -14 < 0

Dòng 13: cập nhật lại cơ sở dữ liệu.

Làm tương tự để ẩn các tập mục nhạy cảm S_2 = <ae> và S_3 = <aef>. cuối cùng Thuật toán đề xuất của IEHSHUI ẩn tất cả các tập mục nhạy cảm và ẩn nhầm 5 tập mục không nhạy cảm, đó là < f >, < af >, < cdef>, < cef > và < def >.

Do đó, trong thuật toán đề xuất IEHSHUI, có thể sửa đổi nhiều giao tác tại cùng thời điểm và nhanh chóng ẩn các tập mục nhạy cảm. Trong phần 4, thực nghiệm sẽ so sánh và đánh giá thuật toán đề xuất IEHSHUI với thuật toán EHSHUI trong [4].

3.3. Kết luận Chương 3

Như vậy, với thuật toán đề xuất, có thể ẩn nhầm tập mục không nhạy cảm ít hơn, sự thay đổi về cơ sở dữ liệu trước và sau khi sửa đổi cũng có thể ít hơn. Về giá trị độ hữu ích của toàn bộ cơ sở dữ liệu có thể ít hơn so với thuật toán EHSHUI. Để có cơ sở đánh giá khách quan hơn, thuật toán đề xuất được chạy thực nghiệm trên cơ sở dữ liệu thực tế và được trình bày trong Chương 4.

CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1. Môi trường thực nghiệm và dữ liệu sử dụng

Thực nghiệm được thực hiện trên máy tính Intel ® Core™ i7 CPU 2.00 GHz, RAM 8GB chạy trên Windows 10. Các thuật toán được thực hiện bằng ngôn ngữ Java. Cơ sở dữ liệu thử nghiệm thu được trên trang web http://www.philippefournier-viger.com/spmf/index.php?link=datasets.php có các đặc điểm sau trong Bảng 4.1:

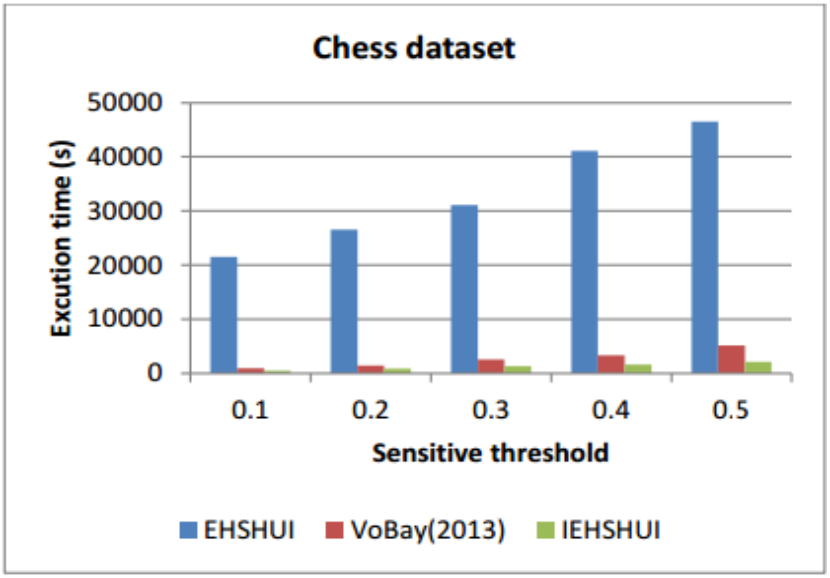
Bảng 4.1: Cơ sở dữ liệu dùng cho thực nghiệm

Cơ sở dữ liệu giao tác	Số giao tác	Số lượng mục
Chess	3196	75
Mushroom	8124	120

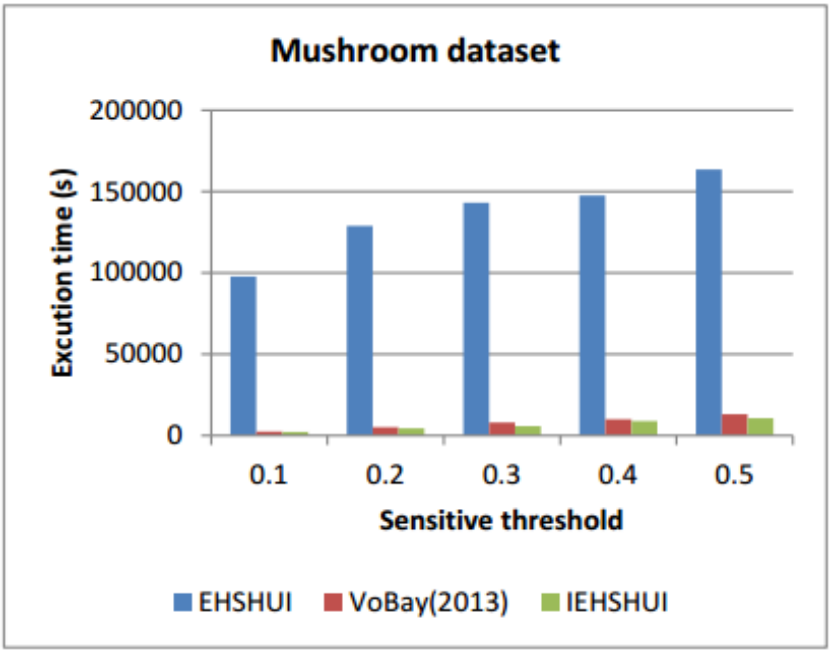
Luận văn thêm ngẫu nhiên số lượng cho các mục trong mỗi giao tác các giá trị trong phạm vi [1-10] bằng cách sử dụng phân phối đồng đều và giá trị lợi nhuận của mỗi mặt hàng trong cơ sở dữ liệu cũng được tạo ngẫu nhiên.

4.2. Kết quả thực nghiệm

Trong phần này, luận văn đã so sánh thuật toán đề xuất IEHSHUI với các thuật toán EHSHUI [4] và thuật toán (VoBay2013) [14] về thời gian thực hiện và sử dụng bộ nhớ. Thực nghiệm được chạy 50 lần, sau đó lấy giá trị trung bình. Số lượng các tập mục nhạy cảm được chọn ngẫu nhiên lần lượt là 0.1, 0.2, 0.3, 0.4 và 0.5 trên số tập mục có độ hữu ích cao (HUI).



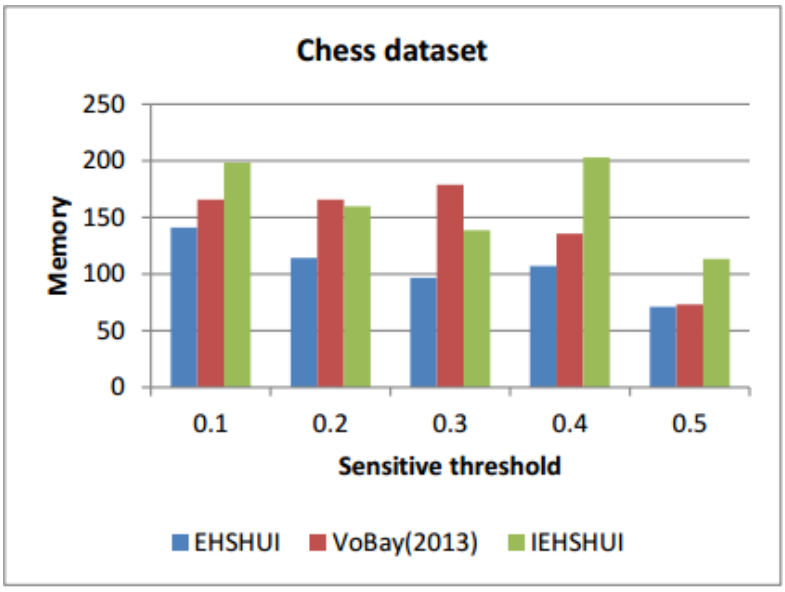
Hình 4.1: So sánh thời gian thực hiện trên tập dữ liệu Chess



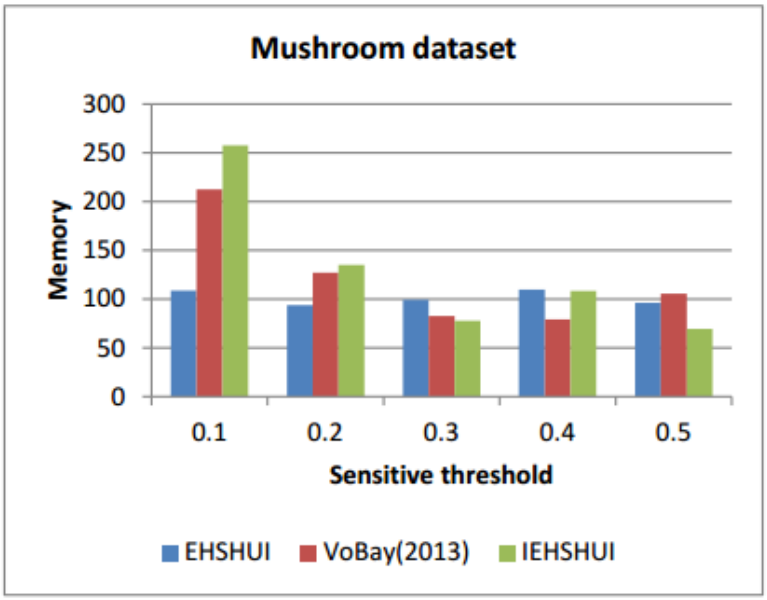
Hình 4.2: So sánh thời gian thực hiện trên tập dữ liệu Mushroom

Hình 4.1 và Hình 4.2 cho thấy rằng thuật toán đề xuất IEHSHUI là hiệu quả nhất về mặt thời gian thực hiện trên cả cơ sở dữ liệu Chess và Mushroom. Thuật toán IEHSHUI nhanh hơn thuật toán EHSHUI trong [4] nhiều lần vì thuật toán IEHSHUI sửa đổi nhiều giao tác cùng một lúc để ẩn thông tin nhạy cảm. Thuật toán EHSHUI trong [4] sửa đổi mỗi lần một giao tác.

Hình 4.3 và Hình 4.4 cho thấy việc sử dụng bộ nhớ của thuật toán đề xuất IEHSHUI nhiều hơn các thuật toán khác. Điều này là do thuật toán đề xuất phải lựa chọn mục cần sửa đổi.



Hình 4.3: So sánh việc sử dụng bộ nhớ trên tập dữ liệu Chess



Hình 4.4: So sánh việc sử dụng bộ nhớ trên tập dữ liệu Mushroom

4.3. Kết luận Chương 4

Luận văn đã đề xuất được một thuật toán IEHSHUI để bảo vệ các tập mục nhạy cảm một cách hiệu quả dựa trên chiến lược lựa chọn tập mục nhạy cảm hợp lý và mục sửa đổi. Kết quả thử nghiệm cho thấy thuật toán IEHSHUI hiệu quả hơn EHSHUI [4] và thuật toán [14] về thời gian thực hiện. Hướng nghiên cứu tiếp theo, tác giả tiếp tục cải tiến thuật toán và thử nghiệm thuật toán đề xuất trên các cơ sở dữ liệu giao tác khác và so sánh với các thuật toán khác để đánh giá hiệu quả và hiệu suất trên các phép đo khác.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Luận văn đã đề xuất được một thuật toán IEHSHUI để bảo vệ các tập mục nhạy cảm một cách hiệu quả dựa trên chiến lược lựa chọn tập mục nhạy cảm hợp lý và mục sửa đổi. Kết quả thử nghiệm cho thấy thuật toán IEHSHUI hiệu quả hơn EHSHUI [4] và thuật toán [14] về thời gian thực hiện.

Trong tương lai, tiếp tục nghiên cứu, cải tiến và thử nghiệm thuật toán đề xuất trên các cơ sở dữ liệu giao tác khác và so sánh với các thuật toán khác để đánh giá hiệu quả và hiệu suất trên các phép đo khác.

CÔNG TRÌNH ĐÃ CÔNG BỐ

[1] Chien, N.K. and D.T.K. Trang. *An Improved Algorithm to Protect Sensitive High Utility Itemsets in Transaction Database*. in *International Conference on Nature of Computation and Communication*. 2021. Springer. https://doi.org/10.1007/978-3-030-92942-8_9.

DANH MỤC TÀI LIỆU THAM KHẢO

[1]. Agrawal, R. and R. Srikant. Privacy-preserving data mining. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data. 2000.

[2]. Atallah, M., et al. Disclosure limitation of sensitive rules. in Proceedings 1999 Workshop on Knowledge and Data Engineering Exchange (KDEX'99)(Cat. No. PR00453). 1999. IEEE.

[3]. Fournier-Viger, P., et al., A survey of tập mục mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2017. **7**(4): p. e1207.

[4]. Huynh Trieu, V., H. Le Quoc, and C. Truong Ngoc, An efficient algorithm for hiding sensitive-high utility itemsets. Intelligent Data Analysis, 2020. 24(4): p. 831-845.

[5]. Krishnamoorthy, S., Pruning strategies for mining high utility itemsets. Expert Systems with Applications, 2015. 42(5): p. 2371-2381.

[6]. Lin, C.-W., et al., A GA-based approach to hide sensitive high utility itemsets. The Scientific World Journal, 2014. 2014.

[7]. Lin, J.C.-W., et al., Fast algorithms for hiding sensitive high-utility itemsets in privacy-preserving utility mining. Engineering Applications of Artificial Intelligence, 2016. 55: p. 269-284.

[8]. Liu, X., S. Wen, and W. Zuo, Effective sanitization approaches to protect sensitive knowledge in high-utility tập mục mining. Applied Intelligence, 2020. 50(1): p. 169-191.

[9]. Mendes, R. and J.P. Vilela, Privacy-preserving data mining: methods, metrics, and applications. IEEE Access, 2017. 5: p. 10562-10582.

[10]. O'Leary, D.E., Knowledge Discovery as a Threat to Database Security. Knowledge discovery in databases, 1991. 9: p. 507-516.

[11]. Rajalaxmi, R. and A. Natarajan, Effective sanitization approaches to hide sensitive utility and frequent itemsets. Intelligent Data Analysis, 2012. 16(6): p. 933-951.

[12]. Saravanabhavan, C. and R. Parvathi, PRIVACY PRESERVING SENSITIVE UTILITY PATTERN MINING. Journal of Theoretical & Applied Information Technology, 2013. 49(2).

[13]. Selvaraj, R. and V.M. Kuthadi, A modified hiding high utility mục first algorithm (HHUIF) with mục selector (MHIS) for hiding sensitive itemsets. 2013.

[14]. Vo, B., et al. An Efficient Method for Hiding High Utility Itemsets. in KESAMSTA. 2013.

[15]. Yeh, J.-S. and P.-C. Hsu, HHUIF and MSICF: Novel algorithms for privacy preserving utility mining. Expert Systems with Applications, 2010. 37(7): p. 4779-4786.

[16]. Yun, U. and J. Kim, A fast perturbation algorithm using tree structure for privacy preserving utility mining. Expert Systems with Applications, 2015. 42(3): p. 1149-1165. S

[17]. Y. Liu, W. Liao, and A. Choudhary, "A Two-Phase algorithm for fast discovery of high utility itemsets.," in Proceedings of the 9th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining, 2005, pp. 689-695.

[18]. S. V. Tseng, C. W. Wu, B. E. Shie, and P. S. Yu, "UP-Growth: an efficient algorithm for high utility itemset mining," in Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 2010, pp. 253-262.

[19]. V.S. Tseng, C Wu, B Shie, and P.S. Yu, "Efficient algorithms for mining high utility itemsets from transactional databases," IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 8, pp. 1772–1786, 2013.

[20]. M. Liu and J. Qu, "Mining high utility itemsets without candidate generation.," in Proceedings of the 21st ACM international conference on Information and knowledge management, 2012, pp. 55-64.

[21]. P. Fournier-Viger, C. Wu, S. Zida, and V.S. Tseng, "Faster high utility itemset mining using estimated utility cooccurrence pruning," in Proceedings 21st International Symposium on Methodologies for Intelligent Systems, 2014, pp. 83-92.

[22]. Ramkumar G.D., Sanjay R., and Tsur S. (1998). Weighted Association Rules: Model and Algorithm. Proc. Fourth ACM Int'l Conf. Knowledge Discovery and Data Mining.

[23]. Cai C.H., Fu A.W.C., Cheng C.H. et al. (1998). Mining Association Rules with Weighted Items. Proceedings of the 1998 International Symposium on Database Engineering & Applications, Washington, DC, USA, IEEE Computer Society, 68–.

[24]. Kumar P. and S A.V. (2009). Parallel Method for Discovering Frequent Itemsets Using Weighted Tree Approach. 2009 International Conference on Computer Engineering and Technology, 124–128.

[25]. Tao F., Murtagh F., and Farid M. (2003). Weighted Association Rule Mining Using Weighted Support and Significance Framework. Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, ACM, 661–666.

[26]. Vo B., Coenen F., and Le B. (2013). A New Method for Mining Frequent Weighted Itemsets Based on WIT-trees. Expert Syst Appl, 40(4), 1256–1264.

[27]. Chan R., Yang Q., and Shen Y.-D. (2003). Mining High Utility Itemsets. IEEE Computer Society, 19.

BẢN CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung luận văn/luận án qua phần mềm Kiểm tra tài liệu (https://kiemtratailieu.vn) một cách trung thực và đạt kết quả mức độ tương đồng **19%** toàn bộ nội dung luận văn/luận án. Bản luận văn/luận án kiểm tra qua phần mềm là bản cứng luận văn/luận án đã nộp bảo vệ trước hội đồng. Nếu sai sót tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học viện.

TP.HCM, ngày 04 tháng 5 năm 2022

Học viên thực hiện luận văn

Đặng Thị Kim Trang

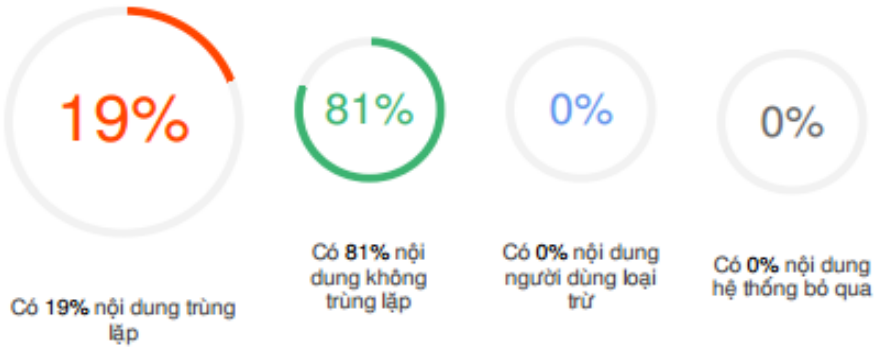
KiểmTraTàiLiệu

BÁO CÁO KIỂM TRA TRÙNG LẶP

Thông tin tài liệu

Tên tài liệu:	Phương pháp ấn các tập mục độ hữu ích cao trong cơ sở dữ liệu giao tác lớn
Tác giả:	Đặng Thị Kim Trang
Điểm trùng lặp:	19
Thời gian tải lên:	09:43 04/05/2022
Thời gian sinh báo cáo:	04:38 05/05/2022
Các trang kiểm tra:	37/37 trang

Kết quả kiểm tra trùng lặp



Nguồn trùng lặp tiêu biểu

tailieu.vn 123docz.net vjol.info.vn

Học viên **Người hướng dẫn khoa học**

Đặng Thị Kim Trang **TS. Nguyễn Khắc Chiến**