ỨNG DỤNG THUẬT GIẢI TỐI ƯU BẦY ĐÀN ẨN TẬP HỮU ÍCH CAO NHẠY CẢM

Lâm Thị Họa Mi[1], Vũ Văn Vinh[1]

Title: *Private preserving High Utility Itemsets using Particle Swarm Optimization*

Từ khóa: *ẩn tập hữu ích cao, bảo vệ tính riêng tư, tập hữu ích cao nhạy cảm, thuật toán tối ưu bầy đàn*

Keywords: *hiding high utility itemsets, protect privacy, sensitive high utility itemsets, Particle Swarm Optimization algorithm*

Lịch sử bài báo

Ngày nhận bài: 8/9/2023

Ngày nhận kết quả bình duyệt: 01/10/2023

Ngày chấp nhận đăng bài: 19/10/2023

Tác giả: *[1]Trường Đại học Công Thương Thành phố Hồ Chí Minh*

Email liên hệ: *milth@hufi.edu.vn*

TÓM TẮT

Bài toán khai thác tập hữu ích cao (HUI) đã thu hút sự quan tâm của các nhà khoa học. Tuy nhiên trong quá trình khai thác HUI, nhiều thông tin nhạy cảm cũng bị phát hiện. Để giải quyết vấn đề này, các thuật toán ẩn HUI bằng phương pháp tối ưu cục bộ: HHUIF, HHUIF cải tiến và MSCIF đã được giới thiệu. Tiếp đó, với mục tiêu tối ưu toàn cục dựa trên giải thuật di truyền (GA), các thuật toán thực hiện thêm và xóa các giao dịch trong cơ sở dữ liệu (CSDL) gốc được đưa ra. Nhưng các phương pháp này khiến cho CSDL bị mất hoặc thêm các giao dịch ảo không cần thiết. Để giải quyết hạn chế này, chúng tôi đã kết hợp việc thay đổi số lượng các hạng mục trong CSDL gốc và kỹ thuật tối ưu bầy đàn (PSO) vừa ẩn được các thông tin nhạy cảm vừa tối tiểu sự thay đổi của CSDL. Kết quả thực nghiệm cho thấy thuật toán đề xuất (PSO-PPUM) ẩn được các thông tin nhạy cảm và tối ưu hóa độ lệch của CSDL chỉnh sửa so với CSDL gốc.

ABSTRACT

High utility itemset mining (HUI) has attracted the attention of researchers. During exploiting HUI, however, a lot of sensitive information was also discovered. To solve this problem, algorithms to protect HUI by local optimization methods such as HHUIF, improved HHUIF, and MSCIF have been introduced. Then, with the global optimization goal of using a Genetic Algorithm (GA), HUI mining approaches that add new transactions or delete some inherent transactions in the original database are proposed. However, these methods cause the database to be lost or add unnecessary virtual transactions. To solve this limitation, we have combined changing the number of items in the original database and the Particle Swarm Optimization (PSO) technique to hide sensitive information and minimize the variation of the original database. Experimental results showed that the proposed algorithm (PSO-PPUM) could hide sensitive information and optimize the deviation of the sanitized database compared to the original database.

1. Đặt vấn đề

Khai thác tập phổ biến (Agrawal et al., 1993) là tìm ra các tập mục có tần số xuất hiện lớn hơn hay bằng một giá trị được gọi là ngưỡng độ phổ biến tối thiểu (minsup). Hạn chế của phương pháp này là bỏ qua lợi ích của các tập mục. Tập phổ biến chỉ phản ánh mối tương quan về mặt thống kê nhưng không phản ánh được tầm quan trọng về mặt ngữ nghĩa giữa các hạng mục. Do đó, bài toán khai thác tập hữu ích cao (HUI) được ra đời vào năm 2004 (Yao et

al., 2004), là tìm ra các tập mục có giá trị hữu ích không nhỏ hơn một ngưỡng tối thiểu (minutil) được xác định trước.

Khai thác HUI giúp cho nhà quản trị nắm bắt được doanh thu thu về khi khách hàng mua các món hàng cùng nhau. Điều đó rất hữu ích cho nhà kinh doanh nhưng lại rất nguy hiểm khi dữ liệu này bị lộ ra ngoài. Vì vậy, họ cần phải bảo vệ tính riêng tư trên tập dữ liệu của mình, nhằm hạn chế việc tiết lộ các thông tin tiềm ẩn của người dùng. Vấn đề này trong khai thác tập phổ biến đã được đề cập nhiều trong các công trình nghiên cứu của nhiều tác giả (Dasseni et al., 2001; Evfimievski et al., 2004; C. W. Lin et al., 2013; J. L. Lin & Cheng, 2009). Tuy nhiên, các công trình về bảo vệ tính riêng tư liên quan đến khai thác tập hữu ích cao lại rất hạn chế.

Với kỹ thuật khai thác dữ liệu ngày càng phát triển, con người ngày càng nắm bắt được nhiều thông tin hơn. Và với sự phát triển kinh doanh như hiện nay, doanh nghiệp không thể chỉ khăng khăng giữ lấy thông tin của riêng mình. Việc trao đổi thông tin giữa các doanh nghiệp là điều cần thiết. Thông qua sự trao đổi này, doanh nghiệp có thể rút trích được các thông tin quan trọng phục vụ cho yêu cầu kinh doanh của mình. Tuy nhiên, khi những thông tin chung liên quan nhiều hơn đến sự riêng tư, người dùng không sẵn lòng cung cấp dữ liệu thật của cá nhân khi được yêu cầu. Bên cạnh đó, các công ty có thể sử dụng dữ liệu của khách hàng cho việc khai thác dữ liệu tuy không dễ dàng vì điều này làm tổn hại tính riêng tư của khách hàng. Vì vậy, làm thế nào có thể đảm bảo được tính riêng tư của tập dữ liệu là vấn đề nghiên cứu thu hút đáng kể sự chú ý trong nhiều năm nay.

Nghiên cứu này tập trung vào vấn đề bảo toàn tính riêng tư trong khai thác tập hữu ích cao. Cụ thể là ẩn các thông tin nhạy cảm trên tập dữ liệu để người dùng không khai thác được các tập hữu ích cao này mà việc thay đổi CSDL ban đầu là nhỏ nhất.

2. Các công trình liên quan

2.1. Khai thác tập hữu ích cao

Vào năm 1993, Agrawal và cộng sự đề xuất phương pháp khai thác luật kết hợp (Agrawal et al., 1993). Nhóm tác giả chia bài toán làm hai giai đoạn: i) Khai thác các tập mục có độ hỗ trợ lớn hơn hoặc bằng ngưỡng độ hỗ trợ tối thiểu cho trước (tập phổ biến); ii) Sinh luật kết hợp từ các tập phổ biến. Năm 1994, nhóm tác giả trong (Agrawal & Srikant, 1994) đưa ra tính chất Apriori (mọi tập con của một tập phổ biến phải phổ biến) và thuật toán Apriori để tỉa các ứng viên. Sau đó, một số thuật toán khai thác nhanh tập phổ biến dựa trên tính chất Apriori được phát triển như Eclat (Zaki & Hsiao, 2005), FP-Growth (Han et al., 2004).

Khác với khai thác tập phổ biến, khai thác HUI từ CSDL quan tâm đến lợi ích mang lại hay còn gọi là độ hữu ích của các tập mục. Do độ hữu ích của các tập mục không thỏa tính chất Apriori nên không thể áp dụng trực tiếp các thuật toán khai thác tập phổ biến vào khai thác tập HUI. Vì vậy, các nghiên cứu về HUIM chủ yếu tập trung vào việc làm thế nào để tỉa được các ứng viên không liên quan càng nhiều càng tốt. Bài toán khai thác tập hữu ích cao cũng đã nhận được sự quan tâm lớn từ cộng đồng nghiên cứu khoa học và một loạt các thuật toán hiệu quả đã được đề xuất. Năm 2005, Liu và cộng sự (Y. Liu et al., 2005) kế thừa thuật toán Apriori và đề xuất thuật toán Two-phase để khai thác HUI trong CSDL giao dịch. Năm 2012, Tseng và cộng sự (Tseng et al., 2013) đã giới thiệu thuật toán UP-Growth và UP-Growth+ bằng cách sử dụng cấu trúc cây để giảm số lần duyệt CSDL. Tiếp đó, Liu và đồng sự (M. Liu & Qu, 2012) đề xuất thuật toán duyệt theo chiều

sâu để tìm các tập HUI chỉ với hai lần duyệt CSDL. Các tác giả đã giới thiệu cấu trúc mang tính đột phá tên là Utility List và cận trên chặt chẽ hơn giúp loại bỏ giới hạn của phương pháp Two-phase. Với thuật toán HMiner được đề xuất năm 2017 (Krishnamoorthy, 2017), tác giả Srikumar đã đề xuất cấu trúc CUL (Compact Utility List) để lưu trữ thông tin liên quan tới một tập mục, giới thiệu khái niệm đóng (Closed) và không đóng (Non-Closed) của một giao dịch đối với tập mục X đồng thời sử dụng tốt các chiến lược tỉa C-Prune, U-Prune, LA-Prune và EUCP giúp cho việc tìm HUI rút ngắn được cả thời gian thực thi và không gian lưu trữ. Từ những yêu cầu khác nhau của người dùng và các ứng dụng trong cuộc sống, nhiều phiên bản khác nhau đã được đề xuất để mở rộng khái niệm khai thác tập HUI như khai thác K tập hữu ích cao nhất (top-K HUI) (Ashraf et al., 2022; Krishnamoorthy, 2019; J. Liu et al., 2018; Pham et al., 2022) và tập hữu ích cao đóng HUI (Closed-HUI) (Duong et al., 2022; Nguyen et al., 2019; Zida et al., 2017).

2.2 Ẩn tập hữu ích nhạy cảm

Năm 2004, Evfimievski và các đồng sự (Evfimievski et al., 2004) đưa ra một ví dụ rất sinh động về vấn đề bảo toàn tính riêng tư. Giả sử, máy chủ có nhiều máy khách và mỗi máy khách có dữ liệu riêng. Các máy khách mong chờ máy chủ thu thập thông tin thống kê dữ liệu từ toàn bộ máy khách về mối kết hợp giữa các mục để cung cấp những đề nghị cho khách hàng của họ. Tuy nhiên, các máy khách không muốn máy chủ lấy các tập mục chứa các tri thức nhạy cảm cao. Vì vậy, khi một máy khách giao CSDL cho máy chủ, một số tập nhạy cảm bị ẩn từ CSDL theo các chính sách bảo mật riêng tư. Máy chủ chỉ tập hợp thông tin thống kê từ CSDL có chỉnh sửa. Chính vì vậy, bài toán khai thác dữ liệu bảo vệ tính riêng tư thu hút được nhiều sự chú ý của các nhà nghiên cứu. Một số nghiên cứu điển hình như: Ẩn các luật kết hợp dựa trên độ phổ biến và độ tin cậy (Dasseni et al., 2001); Lin và Cheng đã thêm nhiễu vào các giao dịch (thêm các mục vào các giao dịch) để tạo ra dữ liệu giả (J. L. Lin & Cheng, 2009); Các thuật toán sử dụng cấu trúc cây để tăng tốc độ ẩn các tập nhạy cảm có độ hữu ích cao do Yun và Kim đề xuất (Vo et al., 2013).

Yeh và Hsu trình bày hai thuật toán có tên là HHUIF (Hiding High Utility Item First) và MSICF (Maximum Sensitive Itemsets Conflict First) (Yeh & Hsu, 2010), với mục tiêu ẩn các tập nhạy cảm trong CSDL. Thủ tục chuyển đổi CSDL gốc thành CSDL chỉnh sửa được gọi là qui trình sửa đổi. Qui trình sửa đổi hoạt động trên dữ liệu để loại bỏ số lượng nhỏ các mục trong một số giao dịch chứa các tập nhạy cảm. Gần đây một số thuật toán về PPUM (Privacy Preserving Utility Mining) cũng được đề xuất dựa trên việc thêm và xóa các giao dịch liên quan tới tập nhạy cảm có độ hữu ích cao bằng cách áp dụng thuật giải di truyền (GA- Genetic Algorithm) (C. W. Lin et al., 2013, 2014).

3. Cơ sở lý thuyết

3.1. Khái niệm và định nghĩa

Gọi $I = \{x_1, x_2, x_3, \dots, x_m\}$ là tập các mục trong CSDL giao dịch. Một giao dịch $T_j = \{x_l | \, l = 1,2, \dots, N_j, x_l \in I\}$ với N_j là số lượng mục trong giao dịch T_j. CSDL D là tập hợp các giao dịch $D = \{T_1, T_2, \dots, T_n\}$, trong đó n là số lượng giao dịch trong D. Ví dụ: D là CSDL được cho trong Bảng 1 có $I = \{a, b, c, d, e\}$ với $n = 12$.

Định nghĩa 1. Lợi ích của một mục $x_i \in I$, ký hiệu là $EU(x_i)$. Trong CSDL D được cho trong Bảng 2, ta có $EU(a) = 6$ và $EU(b) = 2$.

Định nghĩa 2. Số lượng của một mục $x_i \in T_j$, ký hiệu là $IU(x_i, T_j)$. Ví dụ: $IU(a, T_1) = 6$ và $IU(b, T_3) = 6$.

Bảng 1. Cơ sở dữ liệu giao dịch *D*

Mục / Tid	a	b	c	d	e
T_1	6	0	0	0	0
T_2	0	1	0	5	0
T_3	0	6	0	0	0
T_4	0	0	5	0	0
T_5	0	0	0	0	8
T_6	2	0	2	0	3

Mục / Tid	a	b	c	d	e
T_7	0	1	0	4	0
T_8	0	4	0	0	0
T_9	0	2	3	7	0
T_{10}	0	0	0	0	1
T_{11}	0	5	2	5	0
T_{12}	0	3	0	3	0

Bảng 2. Lợi ích của các mục trong cơ sở dữ liệu

Mục	a	b	c	d	e
Lợi nhuận	6	2	15	7	10

Định nghĩa 3. Độ hữu ích của mục $x_i \in T_j$, ký hiệu là $U(x_i, T_j)$, được xác định là tích của số lượng với lợi ích của x_i trong T_j, tức là $U(x_i, T_j) = EU(x_i) * IU(x_i, T_j)$. Ví dụ: $(a, T_1) = 6 * 6 = 36, U(b, T_3) = 2 * 6 = 12$.

Định nghĩa 4. Độ hữu ích của một tập mục X trong giao dịch T_j ($X \subseteq T_j$), ký hiệu là $U(X, T_j)$ và xác định như sau: $U(X, T_j) = \sum_{x_i \in X} U(x_I, T_j)$.

Ví dụ: $U(bc, T_9) = 2 * 2 + 15 * 3 = 49$, $U(bc, T_{11}) = 2 * 5 + 15 * 2 = 40$.

Định nghĩa 5. Độ hữu ích của một tập mục X trong CSDL D được kí hiệu là $U(X)$ và định nghĩa như sau: $U(X) = \sum_{X \subseteq T_j \in D} U(X, T_j)$.

Ví dụ: $U(a) = U(a, T_1) + U(a, T_6) = 36 + 12 = 48$,

$$U(bc) = U(bc, T_9) + U(bc, T_{11}) = 49 + 40 = 89.$$

Định nghĩa 6. Tập mục X được gọi là một tập hữu ích cao (HUI) nếu $U(X) \geq minutil$ ($minutil$ là ngưỡng độ hữu ích tối thiểu do người dùng xác định).

Giả sử $minutil = 168$, $\{bd\}$ là tập hữu ích cao vì $U(bd) = 192 \geq minutil$ và $\{bc\}$ không là tập hữu ích cao vì $U(bc) = 89 \leq minutil$.

Định nghĩa 7. Khai thác tập hữu ích cao là khám phá một tập hợp chứa toàn bộ các tập mục X thỏa ngưỡng $minutil$ cho trước, nghĩa là: $HUI = \{X \subseteq I | U(X) \geq minutil\}$

Ví dụ: Với $minutil = 168$, các tập hữu ích cao của D được liệt kê trong Bảng 3.

Bảng 3. HUI và Độ hữu ích (lợi nhuận) của các HUI trong CSDL *D*

HUI	bcd	bd	c	d
Độ hữu ích	173	192	180	168

Định nghĩa 8. Một tập hữu ích cao X được gọi là là tập nhạy cảm nếu sự xuất hiện của X trong tập HUI có thể dẫn đến tiết lộ một số thông tin bí mật. Loại tập mục này nên được loại bỏ khỏi tập HUI để người quản lý có thể chia sẻ và công khai dữ liệu đến với người dùng.

Định nghĩa 9. Gọi $SHUI = \{S_1, S_2, S_3, \dots, S_l\} \subset HUI$, với S_i là một tập nhạy cảm cần phải ẩn để bảo toàn tính riêng tư.

Định nghĩa 10. Số lượng đụng độ của mục x_p trong $SHUI$, ký hiệu $Icount_{x_p}(SHUI)$, là số tập nhạy cảm có chứa x_p. Nghĩa là, $Icount_{x_p}(SHUI) = |\{S_i \in SHUI | x_p \in S_i\}|$.

3.2. Qui trình ẩn tập hữu ích cao nhạy cảm

Một qui trình ẩn các tập hữu ích cao nhạy cảm gồm 3 bước:

(1) Ứng dụng thuật toán khai thác tập hữu ích cao trên CSDL để tìm HUI;

(2) Xác định các tập nhạy cảm dựa trên các yêu cầu nghiệp vụ;

(3) Áp dụng thuật toán sửa đổi để sinh CSDL sửa đổi.

Bước 1: Khai thác HUI, đầu tiên người dùng chọn ngưỡng hữu ích tối thiểu *minutil* và áp dụng một thuật toán khai thác tập mục hữu ích cao tùy chọn trên CSDL được chọn.

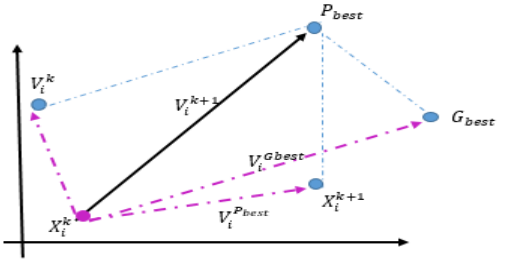
Bước 2: Dựa trên yêu cầu nghiệp vụ, người dùng xác định tập nhạy cảm trong số các tập hữu ích cao. Tập nhạy cảm chứa tất cả các mẫu giới hạn mà người dùng không muốn công khai.

Bước 3: Áp dụng thuật toán sửa đổi trên CSDL đã cho. Mục tiêu chính của thuật toán sửa đổi là giảm giá trị hữu ích của mỗi tập nhạy cảm bằng cách chỉnh sửa số lượng của các mục bên trong nó.

3.3. Thuật giải tối ưu bầy đàn (PSO - Particle Swarm Optimization)

Particle Swarm Optimization (PSO) được giới thiệu bởi Eberhart vào năm 1995 (Eberhart, 1995) và sau đó được Mandapati cùng cộng sự ứng dụng trong khai phá dữ liệu bảo vệ tính riêng tư vào năm 2013 (Mandapati et al., 2013). PSO là một trong những thuật giải xây dựng dựa trên khái niệm trí tuệ bầy đàn để tìm kiếm lời giải cho các bài toán tối ưu trên một không gian tìm kiếm nào đó, là một dạng của thuật giải tiến hóa quần thể như thuật giải GA. Nhưng khác với GA, PSO sử dụng sự tương tác giữa các cá thể trong một quần thể để khám phá ra không gian tìm kiếm.

Trong PSO, mỗi giải pháp đơn là một phần tử (particle) trong quần thể. Mỗi phần tử được đặc trưng bởi hai tham số là vị trí hiện tại của phần tử X và vận tốc hiện tại V. Đây là hai vectơ trên trường số R_n (n là tham số được xác định từ bài toán cụ thể). Đồng thời mỗi phần tử có một giá trị thích nghi (fitness value), được đánh giá bằng hàm đo độ thích nghi (fitness function). Tại thời điểm xuất phát, vị trí của mỗi phần tử được khởi tạo một cách ngẫu nhiên (hoặc theo một cách thức nào đó dựa vào tri thức biết trước về bài toán). Trong quá trình chuyển động, mỗi phần tử chịu ảnh hưởng bởi hai thông tin: thông tin thứ nhất, gọi là P_{best}, là vị trí tốt nhất mà phần tử đó đã đạt được trong quá khứ; thông tin thứ hai, gọi là G_{best}, là vị trí tốt nhất mà cả bầy đàn đã đạt được trong quá khứ.



Hình 1. Sơ đồ di chuyển của cá thể trong PSO

Trong đó:

X_i^k: Vị trí cá thể thứ i tại bước k;

V_i^k Vận tốc cá thể thứ i tại bước k;

X_i^{k+1}: Vị trí cá thể thứ i tại bước $k+1$;

V_i^{k+1} Vận tốc cá thể thứ i tại bước $k+1$;

P_{best}: Vị trí tốt nhất của cá thể thứ i;

G_{best}: Vị trí tốt nhất trong quần thể.

Khi đó, vận tốc và vị trí của cá thể trong quần thể được tính như sau:

$$V_i^{k+1} = \omega * V_i^k + c_1 * r_1 * \left(P_{best} - X_i^k\right) + c_2 * r_2 * (G_{best} - X_i^k)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1}$$

Ý nghĩa của các thông tin trong công thức trên là: ω: là hệ số quán tính; c_1, c_2: các hệ số gia tốc và r_1, r_2: các số ngẫu nhiên nhận giá trị trong khoảng [0,1].

Giá trị của trọng số quán tính ω sẽ giảm tuyến tính từ 1 đến 0 tuỳ thuộc vào số

lần lặp xác định trước. Thêm nữa, giá trị ω lớn cho phép cá thể thực hiện mở rộng phạm vi tìm kiếm, giá trị ω nhỏ làm tăng sự thay đổi để nhận được giá trị tối ưu địa phương. Bởi vậy, khi sử dụng ω có giá trị lớn ($\omega = 0{,}9$) ở thời điểm bắt đầu và sau đó giảm dần giá trị ω sẽ cho hiệu năng tìm kiếm tốt nhất. Các hệ số gia tốc nhận giá trị từ 1,5 đến 2,5 và $c_1 + c_2 = 4$.

Trong đó, điều kiện dừng phổ biến là: số lần cập nhật. Số lần cập nhật bầy đàn mà không đưa lại kết quả tốt hơn, số lần cập nhật mà lượng thay đổi giữa hai lần cập nhật liên tiếp nhỏ hơn một ngưỡng nào đó.

4. Ẩn tập hữu ích cao nhạy cảm sử dụng PSO

Thuật toán HHUIF, HHUIF cải tiến và MSICF đã ẩn đi các tập nhạy cảm hữu ích cao tuy nhiên các thuật toán này chưa quan tâm tới việc số lượng của một số mục trong giao dịch bị chỉnh sửa trở thành 0. Do đó các mục này sẽ không xuất hiện trong giao dịch nữa. Đặc biệt, thuật toán HHUIF và MSICF lại lựa chọn giảm trên các mục có giá trị lớn nhất trong các giao dịch nên sự thay đổi đó sẽ làm cho CSDL có sự đột biến rất lớn tại giao dịch bị chỉnh sửa. Hoặc khi sử dụng thuật toán chèn hoặc xóa các giao dịch trong CSDL ban đầu cũng sẽ tăng độ chênh lệch về tổng độ hữu ích trong CSDL ban đầu và sau khi đã chỉnh sửa.

Để hạn chế những vấn đề đã nêu, nghiên cứu này đề xuất phương pháp ẩn các tập nhạy cảm có độ hữu ích cao bằng cách sử dụng thuật giải PSO để có thể đạt tới sự tối ưu toàn cục. Đồng thời, cũng để hạn chế việc thay đổi CSDL ban đầu. Phương pháp này sẽ không thêm hay bớt các giao dịch có sẵn mà chỉ tính toán để chỉnh sửa lại số lượng của các mục trong mỗi giao dịch. Thuật toán cũng hạn chế sửa đổi số lượng của các mục về 0 để tránh thay đổi bản chất của giao dịch đó trong những trường hợp không cần thiết.

Định nghĩa 11. MDL là tổng độ hữu ích cần giảm của tất cả các tập nhạy cảm có độ hữu ích cao; S_{items} là tập hợp các mục có trong ít nhất một tập nhạy cảm có độ hữu ích cao; n là số lượng các mục trong S_{items} và $n = |S_{items}|$; S_{tids} là tập hợp các giao dịch có trong ít nhất một tập nhạy cảm có độ hữu ích cao; và m là số lượng các giao dịch có S_{tids}, $m = |S_{tids}|$.

Ví dụ: Nếu $minutil = 168$ thì MDL được tính như sau (Bảng 4):

Bảng 4. Minh họa cách tính MDL

Tập mục	Độ hữu ích	Cần giảm
bcd	173	5
bd	192	24
c	180	12
d	168	0
MDL		41

Mô hình hóa một cá thể (giải pháp của bài toán)

Một cá thể chứa thông tin ẩn của các mục trong từng giao dịch, vận tốc di chuyển cá thể và độ thích nghi của cá thể.

Thông tin ẩn của các mục trong từng giao dịch

Số lượng của mỗi mục trong một giao dịch đều ảnh hưởng tới độ hữu ích của các tập nhạy cảm, vì vậy số lượng thay đổi của các mục trong mỗi giao dịch đều cần phải lưu trữ lại. Thông tin của mỗi cá thể sẽ được lưu trữ dưới dạng một mảng hai chiều a có m dòng (số lượng giao dịch ảnh hưởng tới tập nhạy cảm có độ hữu ích cao) và n cột (số lượng các mục chứa trong tập nhạy cảm). Mỗi phần tử $a[i, j]$ là số lượng giảm của mục i trong S_{items} trong giao dịch j của S_{tids}. Giá trị của $a[i, j]$ được giới hạn từ 0 đến số lượng giao dịch tối đa của các mục trong CSDL giao dịch ban đầu.

Ví dụ: Với $S_{items} = \{b, c, d\}$ và $S_{tids} = \{2, 4, 6, 7, 9, 11, 12\}$, khi đó một cá thể là một mảng hai chiều có 7 dòng và 3 cột, cụ thể như sau:

Bảng 5. Biểu diễn của một cá thể

1	0	0
0	1	0
0	0	0
0	0	0
1	0	2
0	0	0
1	0	1

Trong cá thể này, CSDL sẽ giảm số lượng của các mục b, c, d trong giao dịch T_2 lần lượt là 1, 0 và 0. Số lượng giảm của b, c, d trong giao dịch T_{12} lần lượt là 1, 0 và 1.

Số lượng giảm của các mục trong mỗi giao dịch phải nhỏ hơn số lượng của các mục trong giao dịch đang xét và nhỏ hơn số lượng tối đa cần phải giảm. Vì số lượng tối đa trong của c trong giao dịch T_4 là 5 nhưng vì $MDL = 41$, suy ra số lượng tối đa giảm là $41/15 = 3$. Do đó số lượng giảm của c trong giao dịch T_4 tối đa là 3. Tương tự như vậy số lượng của c tối đa cần sửa trong giao dịch T_{11} là 2 vì trong giao dịch T_{11} chỉ có 2 mục c.

Vận tốc của một cá thể

Vận tốc của mỗi cá thể sẽ đại diện cho sự biến đổi của cá thể tại một thời điểm được phát sinh ngẫu nhiên và được lưu trong mảng hai chiều các số thực có m dòng và n cột. Khi di chuyển, giá trị của vận tốc sẽ bị giới hạn bởi giá trị vận tốc tối đa (V_{max}), và giá trị vận tốc tối thiểu (V_{min}) để đảm bảo kết quả tối ưu cho bài toán.

Tính độ thích nghi của một cá thể

Độ thích nghi của một cá thể được tính theo công thức $f = A + \delta * FS + \beta * FT$

Trong đó:

A: là độ lệch của CSDL ban đầu so với CSLD đã chỉnh sửa.

FS: là số lượng các tập nhạy cảm có độ hữu ích cao chưa được ẩn.

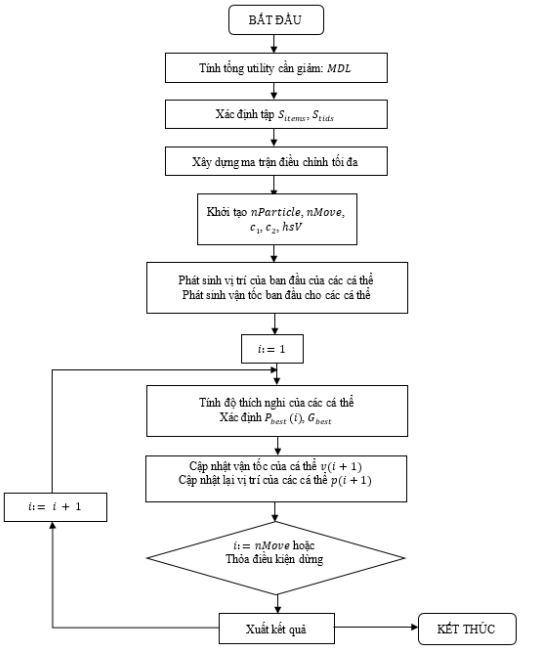
δ: Hệ số thích nghi. Do mục tiêu của thuật toán là ẩn các tập nhạy cảm có độ hữu ích cao nên chọn δ rất lớn so với độ lệch A (ví dụ chọn $\delta = minutil = 168$).

β: Hệ số mất giao dịch. Để hạn chế việc mất các giao dịch hệ số β sẽ phải được chọn phù hợp.

FT: là số lượng giao dịch bị mất do số lượng mục trong giao dịch bị điều chỉnh về 0.

Ví dụ: Với cá thể ở trên, $A = 6 + 15 + 21 = 42$ và $FS = 0$ (vì tất cả các tập nhạy cảm đều được ẩn) và $FT = 0$ (vì không có giao dịch nào có số lượng mục bị chỉnh về 0) nên độ thích nghi của cá thể là $f = 42 + \delta * 0 + \beta * 0 = 42$.

Thuật toán



Hình 2. Lưu đồ mô tả thuật toán PSO-PPUM

Ví dụ: Với CSDL trong Bảng 1 và $minutil = 168$ thuật toán sẽ thực hiện như sau:

Bước 1: Với $SHUI$ như trong Bảng 4 thì $MDL = 41$.

Bước 2: Xác định tập $S_{items} = \{b, c, d\}$ và $S_{tids} = \{2, 4, 6, 7, 9, 11, 12\}$ nên $m = 7$ và $n = 3$.

Bước 3: Tính ma trận điều chỉnh tối đa của các mục trong quần thể (Bảng 6).

Bảng 6. Ma trận điều chỉnh tối đa

	b	c	d
T_2	1	0	5
T_4	0	5	0
T_6	0	2	0
T_7	1	0	4
T_9	2	3	7
T_{11}	5	2	5
T_{12}	3	0	3

Bảng 7. Bảng vận tốc tối đa

	b	c	d
T_2	0,5	0	2,5
T_4	0	2,5	0
T_6	0	1	0
T_7	0,5	0	2
T_9	1	1,5	3,5
T_{11}	2,5	1	2,5
T_{12}	1,5	0	1,5

Bước 4: Giả sử chọn $nParticle = 10$, $c_1 = 2, c_2 = 2$, $nMove = 40$ và hệ số vận tốc tối đa $hsV = 0{,}5$, xây dựng bảng vận tốc tối đa (Bảng 7).

Bước 5: Khởi tạo quần thể ban đầu.

Bước 6: Tính độ thích nghi của tất cả các cá thể đã phát sinh.

Tìm vị trí tốt nhất của các cá thể mà nó từng đi qua $P_{best}(i)$. Vì bước đầu nên vị trí của nó cũng là vị trí tốt nhất.

Tìm vị trí tốt nhất của cả quần thể đã đi qua G_{best}. Do đây là bước đầu nên cá thể có độ thích nghi lớn nhất chính là G_{best}.

Bảng 8. Quần thể ban đầu

Cá thể thứ 1

						P_{best} thứ 1		
0	0	4	1,00	0,00	1,02	0	0	4
0	0	0	0,00	1,78	0,00	0	0	0
0	0	0	0,00	0,60	0,00	0	0	0
0	0	3	1,00	0,00	1,85	0	0	3
0	2	5	0,35	1,15	1,98	0	2	5
0	0	0	0,07	0,77	0,65	0	0	0
2	0	0	1,41	0,00	1,06	2	0	0
$fitness$: 286						$fitnes$		286

Cá thể thứ 2

						P_{best} thứ 2		
0	0	3	1,00	0,00	0,65	0	0	3
0	2	0	0,00	0,21	0,00	0	2	0
0	1	0	0,00	0,58	0,00	0	1	0
0	0	2	1,00	0,00	0,27	0	0	2
0	1	3	0,67	1,24	1,28	0	1	3
3	1	2	1,31	0,65	1,36	3	1	2
1	0	1	0,90	0,00	0,55	1	0	1
$fitness$: 160						$fitnes$		160

Các cá thể từ 3 đến 10 được phát sinh tương tự với $fitness$ lần lượt là 87, 129, 117, 132, 111, 219, 134 và 100. Do đó, G_{best} tại bước khởi tạo là:

G_{best} tại bước khởi tạo		
0	0	1
0	1	0
0	0	0
0	0	2
0	0	0
4	1	3
0	0	1
$fitness$		87

Bước 7: Cập nhật trạng thái mới của quần thể (cập nhật vận tốc $v(i)$ của mỗi cá thể và vị trí mới của từng cá thể $X(i + 1)$).

Tương ứng $r = 0{,}85$, $\omega = 0{,}89$, trạng thái mới của quần thể được cập nhật lại như sau:

Bảng 9. Quần thể sau khi cập nhật

Cá thể thứ 1

0	0	1		0,50	0,00	-2,50
0	2	0		0,00	2,50	0,00
0	0	0		0,00	0,53	0,00
0	0	2		0,50	0,00	-0,06
0	0	1		0,31	-1,50	-3,50
2	1	2		2,50	1,00	2,50
0	0	1		-1,50	0,00	1,50
$fitness$: 98						

P_{best} thứ 1

0	0	1
0	2	0
0	0	0
0	0	2
0	0	1
2	1	2
0	0	1
$fitnes$		98

Cá thể thứ 9

0	0	1		0,50	0,00	0,37
0	0	0		0,00	-2,48	0,00
0	0	0		0,00	0,14	0,00
0	0	2		0,50	0,00	2,00
0	0	0		0,19	-1,50	-3,50
3	1	2		2,50	1,00	-1,32
0	0	0		-1,50	0,00	-1,50
$fitness$: 56						

P_{best} thứ 9

0	0	1
0	0	0
0	0	0
0	0	2
0	0	0
3	1	2
0	0	0
$fitnes$		56

Các cá thể từ 2, 3, 4, 5, 6, 7, 8 và 10 trong quần thể sau khi cập nhật tương tự với $fitness$ lần lượt là 74, 125, 82, 82, 71, 99, 104 và 131. Vì vậy, giá trị P_{best} của các cá thể trong quần thể mới có $fitness$ là 98, 74, 87, 82, 82, 71, 99, 104, 56 và 131. Do đó, G_{best} tại bước di chuyển thứ 1 là:

G_{best} tại bước di chuyển thứ 1		
0	0	1
0	0	0
0	0	0
0	0	2
0	0	0
3	1	2
0	0	0
$fitness$		56

Lặp lại quá trình di chuyển của quần thể theo tiến trình trên, quá trình lặp dừng lại khi độ thích nghi của các thể (fitness của P_{best}) không đổi quá 5 lần tại bước thứ 11. Kết quả cuối cùng của G_{best} như sau:

Bảng 10. Cá thể G_{best}

0	0	0
0	0	0
0	0	0
0	0	3
0	0	0
2	1	0
0	0	0
$fitness$		40

Tương ứng G_{best} ở trên, thuật toán đã đưa ra cách chỉnh sửa CSDL ban đầu thành CSDL trong Bảng 11 và đã ẩn được tất cả các tập nhạy cảm có độ hữu ích cao với độ lệch so với CSDL ban đầu là 40.

Bảng 11. Cơ sở dữ liệu chỉnh sửa D'

Mục / Tid	a	b	c	d	e		Mục / Tid	a	b	c	d	e
T_1	6	0	0	0	0		T_7	0	1	0	**1**	0
T_2	0	1	0	5	0		T_8	0	4	0	0	0
T_3	0	6	0	0	0		T_9	0	2	3	7	0
T_4	0	0	5	0	0		T_{10}	0	0	0	0	1
T_5	0	0	0	0	8		T_{11}	0	**3**	**1**	5	0
T_6	2	0	2	0	3		T_{12}	0	3	0	3	0

Sử dụng thuật giải PSO vào bài toán ẩn tập nhạy cảm có độ hữu ích cao, thuật toán PSO-PPUM đã ẩn được các tập mục nhạy cảm có độ hữu ích cao mà sự khác biệt của CSDL gốc và CSDL sau khi chỉnh sửa là thấp hơn và tốc độ hội tụ nhanh hơn so với các thuật toán khác. Tuy nhiên không gian lưu trữ lại tương đối lớn vì phải lưu trữ lại các thông tin về các cá thể cũng như vận tốc của chúng tại mỗi bước thực thi.

5. Thực nghiệm

5.1. Cơ sở dữ liệu thực nghiệm

Thuật toán thực nghiệm trên máy tính có cấu hình như sau: Intel core i5 (5×2.53 GHz), 8GB RAM memory, Windows 10 và sử dụng ngôn ngữ C#.

Thực nghiệm trên ba CSDL thực là Retail, Mushroom, BMS-POS từ website http://www.cs.rpi.edu/~zaki/Workshops/FIMI/data/. Đặc trưng của các CSDL này được mô tả trong Bảng 12. Với tất cả các những CSDL đã có độ hữu ích của các mục được phát sinh ngẫu nhiên là các số từ 1 tới 10 và số lượng được phát sinh của các mục trong từng giao dịch cũng được phát sinh từ 1 tới 10. Tỉ lệ phát sinh ngẫu nhiên là từ 0.1 tới 1.

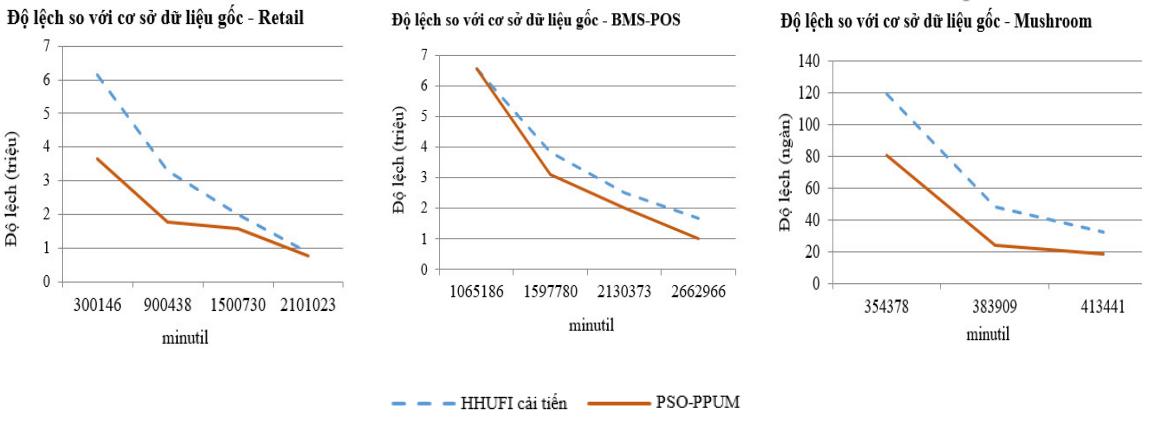
Bảng 12. Thông tin về các cơ sở dữ liệu

Cơ sở dữ liệu	Số giao dịch	Số item	Độ dài trung bình	Độ dài
Retail	88.162	16.470	10,3	76
BMS-POS	515.597	1.657	6,4	46
Mushroom	8.124	119	23	23

5.2. Kết quả thực nghiệm

Trong phần thực nghiệm, thuật toán đề xuất PSO-PPUM được so sánh với thuật toán HHUIF cải tiến, trình bày trong Hình 3. Nghiên cứu giả sử tất cả các HUI đều là tập nhạy cảm hữu ích cao cần được ẩn. Các thuật toán đã ẩn được tất cả HUI nhạy cảm của CSDL thực nghiệm. Thực nghiệm quan tâm tới độ lệch (độ lệch được xác định bằng hiệu tổng độ hữu ích của CSDL gốc với tổng độ hữu ích của CSDL đã chỉnh sửa) sau khi tất cả các tập HUI nhạy cảm đã được ẩn.

Các thông số được chọn như sau: Số lượng cá thể của quần thể: 500, 1000; Số thế hệ tối đa của quần thể: $n = 1000$; Điều kiện dừng: Khi giá trị tối ưu của thuật toán không thay đổi trong 5 lần gần nhất hoặc số thế hệ đạt giá trị n; Giá trị $\beta = \delta = minutil$.



Hình 3. Biểu đồ độ lệch trên các cơ sở dữ liệu

Kết quả thu được cho thấy, cả hai thuật toán đều ẩn được tất cả các tập hữu ích cao trong các CSDL thử nghiệm. Khi giá trị *minutil* càng cao, số lượng HUI càng nhỏ thì độ lệch sau khi chỉnh sửa càng nhỏ. Điều này hiển nhiên vì giá trị độ hữu ích cần giảm của mỗi tập HUI và cả số lượng tập HUI nhạy cảm đều nhỏ hơn.

Kết quả thực nghiệm cũng cho thấy độ lệch khi sử dụng phương pháp tối ưu toàn cục bằng PSO đều cho kết quả tốt hơn so với sử dụng phương pháp tối ưu cục bộ của thuật toán HHUIF cải tiến. Trong phương pháp tối ưu cục bộ, thuật toán ẩn lần lượt các tập HUI nhạy cảm và tập trung vào việc chỉnh sửa CSDL để đảm bảo HUI đang xét mà không quan tâm tới ảnh hưởng của việc điều chỉnh tới các HUI khác. Do đó, việc điều chỉnh không chọn được các giá trị điều chỉnh mà giúp giảm

được nhiều HUI nhạy cảm một cách đồng thời như các thuật toán tối ưu toàn cục. Chính vì vậy, độ lệch do HHUIF cải tiến tạo ra lớn hơn nhiều (tới 2–4 lần) so với thuật toán so với PSO-PPUM.

Kết luận

Ẩn tập hữu ích cao nhạy cảm là chủ đề nghiên cứu thú vị và có nhiều ứng dụng trong thực tế. Nghiên cứu đã tìm hiểu được qui trình ẩn các tập hữu ích cao nhạy cảm để bảo toàn tính riêng tư của dữ liệu. Nghiên cứu cũng đã ứng dụng được thuật giải Heuristic (thuật giải PSO) và đề xuất thuật toán PSO-PPUM áp dụng trong bài toán này nhằm tối tiểu hóa sự thay đổi của CSDL khi tiến hành chỉnh sửa. Thuật toán PSO-PPUM đã phát huy được các ưu điểm của thuật giải tối ưu bầy đàn, ẩn được tất cả các tập nhạy cảm hữu ích cao với độ lệch ít hơn so với thuật toán HHUIF cải tiến.

Do số lượng của hữu ích cao là rất lớn nên việc tính toán độ thích nghi của mỗi cá thể trong quần thể (hay bầy đàn) tiêu tốn rất nhiều thời gian cho việc kiểm tra các tập hữu ích cao đã được ẩn hay không. Hơn nữa, trong các CSDL giao dịch, số lượng các tập hữu ích cao đóng (Closed HUI) sẽ chứa tất cả các tập hữu ích cao và số lượng ít hơn rất nhiều lần so với số lượng tập HUI. Vì vậy để ẩn tất cả các tập nhạy cảm hữu ích cao, trong thời gian tới nhóm tác giả sẽ thực hiện trên các tập hữu ích cao đóng để giảm bớt chi phí về thời gian tính độ thích nghi, giúp cải thiện tốc độ thực thi của thuật toán. Nghiên cứu sẽ tìm cách giảm chi phí về không gian và thời gian thực thi của bài toán bằng cách xem xét việc gộp cách giao dịch có chung tập mục.

TÀI LIỆU THAM KHẢO

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, *22*(2), 207–216. https://doi.org/10.1145/170036.170072

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. *Proceedings of the 20th International Conference on Very Large Data Bases*, 478–499.

Ashraf, M., Abdelkader, T., Rady, S., & Gharib, T. F. (2022). TKN: An efficient approach for discovering top-k high utility itemsets with positive or negative profits. *Information Sciences*, *587*, 654–678. https://doi.org/10.1016/J.INS.2021.12.024

Dasseni, E., Verykios, V. S., Elmagarmid, A. K., & Bertino, E. (2001). Hiding association rules by using confidence and support. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2137*, 369–383. https://doi.org/10.1007/3-540-45496-9_27/COVER

Duong, H., Hoang, T., Tran, T., Truong, T., Le, B., & Fournier-Viger, P. (2022). Efficient algorithms for mining closed and maximal high utility itemsets. *Knowledge-Based Systems*, *257*, 109921. https://doi.org/10.1016/J.KNOSYS.2022.109921

Eberhart, J. K. and R. (1995). Proceedings of ICNN'95 - International Conference on Neural Networks. *Particle Swarm Optimization*, 1942–1948.

Evfimievski, A., Srikant, R., Agrawal, R., & Gehrke, J. (2004). Privacy preserving mining of association rules. *Information Systems*, *29*(4), 343–364. https://doi.org/10.1016/j.is.2003.09.001

Han, J., Pei, J., Yin, Y., & Mao, R. (2004). Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, *8*(1), 53–87. https://doi.org/10.1023/B:DAMI.0000005258.31418.83/METRICS

Krishnamoorthy, S. (2017). HMiner: Efficiently mining high utility itemsets. *Expert Systems with Applications*, *90*, 168–183. https://doi.org/10.1016/j.eswa.2017.08.028

Krishnamoorthy, S. (2019). Mining top-k high utility itemsets with effective threshold raising strategies. *Expert Systems with Applications*, *117*, 148–165. https://doi.org/10.1016/J.ESWA.2018.09.051

Lin, C. W., Hong, T. P., Wong, J. W., & Lan, G. C. (2013). Privacy preserving high utility mining based on genetic algorithms. *Proceedings - 2013 IEEE International Conference on Granular Computing, GrC 2013*, 191–195. https://doi.org/10.1109/GrC.2013.6740406

Lin, C. W., Hong, T. P., Wong, J. W., Lan, G. C., & Lin, W. Y. (2014). A GA-based approach to hide sensitive high utility Itemsets. *The Scientific World Journal*, *2014*. https://doi.org/10.1155/2014/804629

Lin, J. L., & Cheng, Y. W. (2009). Privacy preserving itemset mining through noisy items. *Expert Systems with Applications*, *36*(3 PART 1), 5711–5717. https://doi.org/10.1016/j.eswa.2008.06.052

Liu, J., Zhang, X., Fung, B. C. M., Li, J., & Iqbal, F. (2018). Opportunistic mining of top-n high utility patterns. *Information Sciences*, *441*, 171–186. https://doi.org/10.1016/J.INS.2018.02.035

Liu, M., & Qu, J. (2012). Mining high utility itemsets without candidate generation. *Proceedings of the 21st ACM International Conference on Information and Knowledge Management - CIKM '12*, 55. https://doi.org/10.1145/2396761.2396773

Liu, Y., Liao, W. K., & Choudhary, A. (2005). A two-phase algorithm for fast discovery of high utility itemsets. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3518 LNAI*, 689–695. https://doi.org/10.1007/11430919_79/COVER

Mandapati, S., Bhogapathi, R. B., & Rao, M. V. P. C. S. (2013). *Swarm Optimization Algorithm for Privacy Preserving in Data Mining*. *10*(2), 46–51.

Nguyen, L. T. T., Vu, V. V., Lam, M. T. H., Duong, T. T. M., Manh, L. T., Nguyen, T. T. T., Vo, B., & Fujita, H. (2019). An efficient method for mining high utility closed itemsets. *Information Sciences*, *495*, 78–99. https://doi.org/10.1016/J.INS.2019.05.006

Pham, N. N., Kominkova Oplatkova, Z., Huynh, H. M., & Vo, B. (2022). Mining Top-K High Utility Itemset Using Bio-Inspired Algorithms. *2022 IEEE Workshop on Complexity in Engineering (COMPENG)*, 1–5. https://doi.org/10.1109/COMPENG50184.2022.9905433

Tseng, V. S., Shie, B.-E., Wu, C.-W., & Yu, P. S. (2013). Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases. *IEEE Transactions on Knowledge and Data Engineering*, *25*(8), 1772–1786. https://doi.org/10.1109/TKDE.2012.59

Vo, B., Lin, C. W., Hong, T. P., Vu, V. V., Nguyen, M., & Le, B. (2013). An Efficient Method for Hiding High Utility Itemsets. *Frontiers in Artificial Intelligence and Applications*, *252*, 356–363. https://doi.org/10.3233/978-1-61499-254-7-356

Yao, H., Hamilton, H. J., & Butz, G. J. (2004). A foundational approach to mining itemset utilities from databases. *Proceedings*, 482–486. https://doi.org/10.1137/1.9781611972740.51

Yeh, J. S., & Hsu, P. C. (2010). HHUIF and MSICF: Novel algorithms for privacy preserving utility mining. *Expert Systems with Applications*, *37*(7), 4779–4786. https://doi.org/10.1016/J.ESWA.2009.12.038

Zaki, M. J., & Hsiao, C. J. (2005). Efficient algorithms for mining closed itemsets and their lattice structure. *IEEE Transactions on Knowledge and Data Engineering*, *17*(4), 462–478. https://doi.org/10.1109/TKDE.2005.60

Zida, S., Fournier-Viger, P., Lin, J. C.-W., Wu, C.-W., & Tseng, V. S. (2017). EFIM: a fast and memory efficient algorithm for high-utility itemset mining. *Knowledge and Information Systems*, *51*(2), 595–625. https://doi.org/10.1007/s10115-016-0986-0