Khai thác hiệu quả các luật kết hợp súc tích có lợi ích và tương quan cao

Trần Thống[1, 2, 3], Hoàng Minh Tiến[1, 2, 4], Dương Văn Hải[4*], Trương Chí Tín[4]

[1]Khoa Công nghệ thông tin, Trường Đại học Khoa học Tự nhiên, TP. Hồ Chí Minh, 227 đường Nguyễn Văn Cừ, phường Chợ Quán, TP. Hồ Chí Minh, Việt Nam
[2]Đại học Quốc gia TP. Hồ Chí Minh, phường Linh Xuân, TP. Hồ Chí Minh, Việt Nam
[3]Khoa Công nghệ thông tin, Trường Đại học Đà Lạt, Đường Phù Đổng Thiên Vương, Phường Lâm Viên, TP. Đà Lạt, tỉnh Lâm Đồng, Việt Nam
[4]Khoa Toán - Tin học, Trường Đại học Đà Lạt, Đường Phù Đổng Thiên Vương, Phường Lâm Viên, TP. Đà Lạt, tỉnh Lâm Đồng, Việt Nam

Ngày nhận 13/1/2025; ngày gửi phản biện 28/1/2025; ngày nhận phản biện 24/2/2025

Tóm tắt:

Bài báo này xét bài toán khai thác các luật kết hợp lợi ích cao từ cơ sở dữ liệu định lượng. Các tiếp cận truyền thống chủ yếu dựa trên độ tin cậy - lợi ích thông thường mà không xem xét mức độ gắn kết thực sự giữa các mục. Do thiếu một độ đo tương quan, nhiều luật được phát hiện có lợi ích cao nhưng các mục trong luật lại ít đồng xuất hiện, khó diễn giải và kém ý nghĩa thực tiễn; đồng thời số lượng luật sinh ra thường rất lớn. Để khắc phục các hạn chế này, bài báo đề xuất một độ đo tin cậy dựa trên lợi ích mới, đồng thời tích hợp thêm độ đo tương quan Bond nhằm bảo đảm các mục trong luật vừa mang lại lợi ích cao vừa có mối liên kết chặt chẽ. Trên nền tảng đó, bài báo đề xuất lớp luật kết hợp súc tích có lợi ích cao và tương quan mạnh, gọi là SCoHUAR, trong đó mỗi luật được biểu diễn dưới dạng rút gọn, đại diện cho một nhóm luật tương đương về độ hỗ trợ. Ngoài ra, bài báo còn trình bày phương pháp phân hoạch tập các SCoHUAR giúp bảo đảm sinh đầy đủ và không trùng lặp các luật SCoHUAR và các kỹ thuật tối ưu để tăng tốc quá trình khai thác. Cuối cùng, thuật toán M-SCoHUAR được thiết kế để khai thác hiệu quả các SCoHUAR, trong đó tích hợp các kết quả lý thuyết được đề xuất. Các kết quả thực nghiệm chỉ ra rằng thuật toán được đề xuất có nhiều ưu điểm nổi bật hơn so với các phương pháp trước đây về thời gian thực thi, mức bộ nhớ sử dụng và chất lượng của các mẫu được khám phá.

Từ khóa: **độ đo tương quan Bond; hàm tin cậy - lợi ích; luật kết hợp lợi ích cao; luật kết hợp súc tích; luật kết hợp tương quan cao; mẫu đóng; mẫu sinh.**

Efficient mining of succinct correlated high utility association rules

Thong Tran[1, 2, 3], Tien Hoàng Minh[1, 2, 4], Hai Dương Van[4*], Tin Truong Chi[4]

[1]Falcuty of Information Technology, VNUHCM-University of Science, Ho Chi Minh City, Vietnam
[2]Vietnam National University Ho Chi Minh City, Linh Xuan Ward, Hochiminh City, Vietnam
[3]Falcuty, of Information Technology, Da Lat University, Phu Dong Thien Vuong street, Lam Vien Ward, Da Lat City, Lam Dong province, Vietnam
[4]Falcuty of Mathematics and Computer Science, Da Lat University, Phu Dong Thien Vuong Street, Lam Vien Ward, Da Lat City, Lam Dong province, Vietnam

Received 13 January 2025; revised 20 February 2025; accepted 24 February 2025

Abstract:

This paper investigates the problem of mining high utility association rules from quantitative databases. Traditional approaches mainly rely on conventional utility-confidence measures without considering the actual strength of association among items. Due to the absence of a correlation measure, many discovered rules exhibit high utility while the items within the rules co-occur infrequently, making them difficult to interpret and of limited practical significance; moreover, the number of generated rules is often very large. To address these limitations, this paper proposes a novel utility-confidence measure and integrates the Bond correlation measure to ensure that the items in a rule not only yield high utility but also exhibit strong correlations. On this basis, the paper introduces a class of succinct correlated high utility association rules, called SCoHUARs, in which each rule is represented in a compact form that serves as a representative of an equivalence class of rules sharing the same support. In addition, a partitioning method for the set of SCoHUARs is presented to guarantee complete and non-redundant generation of all SCoHUARs, together with several optimization techniques to accelerate the mining process. Finally, the M-SCoHUAR algorithm is designed to efficiently mine SCoHUARs by integrating the proposed theoretical results. Experimental results demonstrate that the proposed algorithm significantly outperforms existing methods in terms of execution time, memory consumption, and the quality of the discovered patterns.

Keywords: **Bond correlation; closed pattern; correlated association rule; generator pattern; high utility association rule; succinct association rule; utility-confidence measure.**

[*]Tác giả liên hệ: Email: haidv@dlu.edu.vn

1. Giới thiệu

Khai thác luật kết hợp là một trong những hướng nghiên cứu nền tảng của khai phá dữ liệu và đã thu hút sự quan tâm của nhiều nhà nghiên cứu trong những năm qua [1]. Các phương pháp truyền thống dựa chủ yếu trên độ hỗ trợ và độ tin cậy. Cách tiếp cận này xem mọi mục là như nhau và chỉ quan tâm đến tần suất xuất hiện, nên không phản ánh được tầm quan trọng hay lợi ích của từng mục trong giao dịch. Điều này dẫn đến khả năng thu được các luật có độ tin cậy cao nhưng đóng góp lợi ích thấp, kém giá trị trong các bài toán ra quyết định. Để khắc phục hạn chế đó, các nghiên cứu về khai thác tập mục lợi ích cao (high-utility itemset (HUI) mining - HUIM) đã được đề xuất [2], trong đó mỗi mục được gán một trọng số lợi ích và một tập mục được xem là lợi ích cao nếu tổng lợi ích của nó vượt một ngưỡng do người dùng chỉ định [2]. Bài toán này có nhiều ứng dụng trong thực tế, chẳng hạn như khám phá các sản phẩm có lợi ích cao khi được mua đồng thời, phân tích hành vi người dùng trong các cửa hàng trực tuyến, nghiên cứu dữ liệu y-sinh, và tiếp thị chéo [2-4]. Các thuật toán như Two-Phase [5], HUI-Miner [6], FHM [7] và EFIM [8] đã được phát triển để xử lý những khó khăn do độ đo lợi ích không thỏa tính đơn điệu, thông qua việc thiết kế các chặn trên (UB) và cấu trúc dữ liệu đặc thù nhằm giảm số lượng ứng viên và chi phí tính toán.

Dựa trên các HUI, bài toán khai thác các luật kết hợp lợi ích cao (high-utility association rules - HUAR) tiếp tục được quan tâm với mục tiêu cung cấp thông tin hỗ trợ ra quyết định trong các bối cảnh như thương mại điện tử, tài chính hay Internet vạn vật (IoT). Sahoo và cs [9] đã đề xuất khung đo lường độ tin cậy - lợi ích (utility-confidence) dựa trên hàm tin cậy - lợi ích , định nghĩa lại luật kết hợp trong không gian lợi ích, và đưa ra các khái niệm tập mục đóng lợi ích cao, tập mục sinh lợi ích cao cùng thuật toán HUCI-Miner để khai thác đồng thời hai lớp tập mục này. Từ đó, nhóm tác giả xây dựng được một cơ sở luật lợi ích cao không dư thừa, có tiền đề (vế trái của luật) tối tiểu và hậu đề (vế phải của luật) tối đa. Các thuật toán sau đó sử dụng cấu trúc dàn của tập mục lợi ích cao để sinh toàn bộ hoặc một tập con không dư thừa các HUAR với hiệu quả tốt hơn [10-12].

Tuy đạt được nhiều kết quả quan trọng, các hướng tiếp cận trên chủ yếu đánh giá luật dựa trên lợi ích và độ tin cậy theo lợi ích, chưa tích hợp trực tiếp các độ đo tương quan để kiểm soát mức độ gắn kết giữa các mục trong luật. Trong thực tế, một luật có lợi ích cao nhưng tương quan yếu (chẳng hạn khi vế phải của luật xuất hiện rất phổ biến, ít phụ thuộc vào vế trái) có thể ít mang lại giá trị giải thích và hỗ trợ ra quyết định. Đồng thời, các cơ sở luật không dư thừa hiện có mới chủ yếu khai thác khái niệm đóng/sinh trong không gian lợi ích, chưa kết hợp rõ ràng cả hai yếu tố: lợi ích cao và tương quan mạnh.

Xuất phát từ những nhận định đó, bài báo này nghiên cứu bài toán khai thác các luật kết hợp súc tích có lợi ích và tương quan cao (Succinct Correlated High-Utility Association Rules - SCoHUAR) trên cơ sở dữ liệu định lượng. Trước hết, bài báo sử dụng khái niệm độ tin cậy - lợi ích trong [9] nhưng đề xuất một hàm độ tin cậy - lợi ích mới gọi là , nhấn mạnh phần lợi ích mà vế phải thực sự đóng góp khi vế trái xảy ra. Đồng thời, bài báo tích hợp thêm độ đo tương quan bond [13], vốn phản ánh tỉ lệ giữa độ hỗ trợ đồng xuất hiện và độ hỗ trợ rời rạc của tập mục, để bảo đảm các luật thu được không chỉ có lợi ích cao mà còn thể hiện mối liên hệ chặt chẽ giữa các mục. Trong khai thác các tập mục phổ biến, nhiều độ đo tương quan đã được đề xuất nhằm đánh giá mức độ gắn kết giữa các mục, tiêu biểu như bond [13], lift [14], all-confidence [15], kulc [16], v.v. Mỗi độ đo phản ánh mối quan hệ giữa các mục dưới những góc nhìn khác nhau. Cụ thể, lift đánh giá mức độ phụ thuộc thống kê bằng cách so sánh tần suất đồng xuất hiện quan sát được với tần suất kỳ vọng dưới giả định độc lập, qua đó cho phép phát hiện tương quan dương hoặc âm giữa các thuộc tính. Trong khi đó, kulc tập trung đo lường mức độ gắn kết hai chiều, thông qua trung bình xác suất có điều kiện theo cả hai hướng giữa các tập mục. So với các độ đo trên, bond, được định nghĩa là tỷ lệ giữa support của giao và support của hợp của các tập mục, sở hữu những ưu điểm nổi bật. Cụ thể, bond là một độ đo có tính chất đơn điệu giảm nên đặc biệt phù hợp để tích hợp vào các chiến lược cắt tỉa hiệu quả trong quá trình khai thác. Hơn nữa, như đã được chỉ ra trong phần thực nghiệm của [13], việc sử dụng bond mang lại chất lượng tập kết quả tốt hơn và thời gian thực thi nhanh hơn so với all-confidence trong bài toán khai thác các tập mục có lợi ích và tương quan cao. Nhờ những đặc tính này, bond cho phép loại bỏ sớm các mẫu không tương quan, đồng thời nâng cao khả năng mở rộng của thuật toán. Vì các lý do nêu trên, bài báo này lựa chọn bond làm độ đo tương quan để khai thác các SCoHUAR.

Trên cơ sở đó, bài báo đồng thời giới thiệu lớp luật SCoHUAR, trong đó vế trái là tập mục sinh lợi ích cao, hợp của hai vế của luật là tập mục đóng lợi ích cao [17], và cả hai đồng thời thỏa các ràng buộc về lợi ích, độ tin cậy - lợi ích và tương quan mạnh.

Để thấy được ý nghĩa của hàm tin cậy - lợi ích mới và độ đo tương quan bond, bài báo trình bày một ví dụ về cơ sở dữ liệu thực tế minh họa việc bán lẻ phụ kiện công nghệ trong một cửa hàng online, trong đó mỗi giá trị là lợi ích thu được từ mặt hàng trong một giao dịch. Cơ sở dữ liệu được trình bày trong bảng 1, gồm các thuộc tính: *Smartphone*, *ốp lưng*, *kính cường lực*, *bảo hiểm rơi vỡ*, *gói data* và *tai nghe*. Giả sử ngưỡng độ tin cậy-lợi ích tối thiểu và ngưỡng tương quan tối thiểu, xét hàm tin cậy - lợi ích được giới thiệu trong [13] (xem định nghĩa 16), hàm tin cậy - lợi ích mới được trình bày trong định nghĩa 17, và độ đo tương quan bond được trình bày trong định nghĩa 8.

Xét luật $R1$: Smartphone → Ốp lưng, ta có $uconf(R1) = 0{,}616 > muconf$, $cuconf(R1) = 0{,}03 < muconf$ và $bond(\{Smartphone\ ,\ ốp\ lưng\}) = 0{,}375 < mb$. Luật này minh họa trường hợp $uconf$ đủ lớn (vế trái "giữ lợi ích" khi đi kèm vế phải), nhưng $cuconf$ rất nhỏ, tức Ốp lưng chỉ đóng góp lợi ích không đáng kể. Do đó, nếu chỉ dùng $uconf$, ta có thể giữ lại các luật mà vế phải mang ít giá trị kinh tế. Tuy nhiên, nếu sử dụng thêm $cuconf$, ta có thể nhận diện và hạ ưu tiên các luật loại này. Ngoài ra, ta cũng có thể loại bỏ luật này do mức tương quan không đạt ngưỡng. Đáng chú ý hơn, luật $R2$: Ốp lưng → Tai nghe cho kết quả $uconf(R2) = 0.4 < muconf$, $cuconf(R2) = 1{,}467 > muconf$ và $bond(\{\text{Ốp lưng, Tai nghe}\}) = 0.5 > mb$, cho thấy Tai nghe đóng góp lợi ích vượt trội trong các giao dịch kết hợp Ốp lưng, mặc dù $uconf$ không đạt ngưỡng. Ví dụ này minh họa rõ rằng $cuconf$ có khả năng phát hiện các luật bán kèm sinh lợi cao mà $uconf$ có thể loại bỏ, đồng thời việc xét thêm bond giúp đảm bảo các luật thu được phản ánh những mối liên kết đồng xuất hiện có ý nghĩa. Ngoài ra, $cuconf$ có thể hỗ trợ $uconf$ trong việc loại bỏ các luật mặc dù có độ tin cậy - lợi ích cao (theo $uconf$) nhưng vế phải mang ít giá trị kinh tế.

Bảng 1. Cơ sở dữ liệu định lượng minh họa bán lẻ phụ kiện công nghệ.

ID	Các thuộc tính (với lợi ích) trong giao tác
	Smartphone (900), Ốp lưng (40), Kính cường lực (25), Bảo hiểm rơi vỡ (200)
	Smartphone (850), Bảo hiểm rơi vỡ (200), Gói data (150)
	Smartphone (820), Ốp lưng (45), Kính cường lực (30)
	Ốp lưng (35), Kính cường lực (20), Tai nghe (120)
	Smartphone (880), Bảo hiểm rơi vỡ (200)
	Ốp lưng (30), Kính cường lực (20), Tai nghe (110)
	Smartphone (950), Ốp lưng (50), Bảo hiểm rơi vỡ (200)
	Ốp lưng (25), Kính cường lực (15), Tai nghe (100)

Ví dụ trên cho thấy rằng việc khai thác các luật kết hợp có lợi ích và tương quan cao, dựa trên độ tin cậy - lợi ích mới *cuconf* kết hợp với độ đo tương quan *bond*, mang lại giá trị rõ rệt và có ý nghĩa thực tiễn trong các ứng dụng thực tế. Trên cơ sở đó, các đóng góp chính của bài báo được tóm tắt như sau:

(1) Một hàm tin cậy - lợi ích mới được đề xuất, thay thế hoặc hỗ trợ (tùy vào ứng dụng thực tế) cho hàm tin cậy - lợi ích đã được sử dụng bởi những công trình trước đây.

(2) Đề xuất bài toán khai thác các luật kết hợp súc tích có lợi ích và tương quan cao (SCoHUAR) sử dụng hàm tin cậy - lợi ích mới , độ đo tương quan *bond* và các khái niệm tập mục đóng và sinh lợi ích cao và tương quan mạnh.

(3) Giới thiệu một phương pháp phân hoạch mới để sinh ra đầy đủ và không trùng lặp tất cả các SCoHUAR.

(4) Thiết kế một thuật toán hiệu quả M-SCoHUAR nhằm khai thác nhanh các SCoHUAR, trong đó tích hợp các kết quả lý thuyết đã đề xuất, kết hợp với các kỹ thuật tối ưu nhằm giảm thiểu tối đa số phép kiểm tra quan hệ bao hàm giữa các tập mục, qua đó tăng tốc quá trình khai thác.

(5) Cung cấp các kết quả thực nghiệm thuyết phục nhằm xác nhận tính đúng đắn và ý nghĩa của các kết quả lý thuyết được đề xuất, đồng thời chứng minh hiệu quả của thuật toán M-SCoHUAR trong việc khai thác các luật SCoHUAR.

Phần còn lại của bài báo được tổ chức như sau: (i) Mục 2 trình bày tổng quan các công trình nghiên cứu liên quan đến khai thác các tập mục lợi ích cao và các dạng biểu diễn súc tích của chúng, cũng như các công trình liên quan đến khai thác luật kết hợp lợi ích cao; (ii) Mục 3 giới thiệu các khái niệm và ký hiệu cần thiết cho việc khám phá các luật kết hợp súc tích có lợi ích và tương quan cao; (iii) Mục 4 trình bày quy trình tổng quát để khai thác các luật kết hợp hữu ích, cùng với phương pháp và thuật toán nhằm sinh nhanh các luật này; (iv) Kết quả thực nghiệm được thảo luận trong (v) mục 5; Cuối cùng, những kết luận và công việc tương lai được trình bày trong mục 6.

2. Những công việc liên quan

1. Khai thác các tập mục lợi ích cao

Khai thác các tập mục lợi ích cao (HUIM) đã là một chủ đề nghiên cứu phổ biến trong khai thác dữ liệu, với nhiều thuật toán được phát triển để khám phá hiệu quả những tập mục như vậy [18, 19, 20-23]. Các thuật toán này thường dựa vào các chặn trên (UBs) để giảm không gian tìm kiếm. Việc thiết kế các UBs chặt có thể dẫn đến cải thiện trực tiếp hiệu quả của HUIM. Do đó, một số UBs đã được thiết kế. Chặn trên đầu tiên là TWU (transaction weight utility), được sử dụng trong thuật toán Two-Phase [5]. Thuật toán này trước tiên xác định các tập mục có giá trị TWU cao, và sau đó lọc bỏ các tập mục có lợi ích thấp (LUIs). Cách tiếp cận này cũng được áp dụng trong các thuật toán khác như GPA và UP-Growth+ [4, 24]. UB thứ hai, gọi là, được trình bày trong [2], và chặt hơn TWU. Để tính toán hiệu quả, các tác giả của [2] đã giới thiệu cấu trúc utility-list để lưu trữ thông tin cần thiết của các tập mục. Cách tiếp cận này cũng được sử dụng trong các thuật toán FHM [7] và HMiner, với các chiến lược bổ sung như U-Prune, EUCP, và LA-Prune. Bên cạnh đó, các phương pháp khác cũng được phát triển để cải thiện hiệu quả của HUIM, dựa trên các cấu trúc dữ liệu mới [25, 26] [27, 28]. Gần đây, một chặn trên yếu (WUB) gọi là được

đề xuất trong [29], chặt hơn và được sử dụng trong các thuật toán C-HUIM và MaxC-HUIM để loại bỏ sớm hơn nhiều tập mục có lợi ích thấp.

2. Khai thác các tập mục đóng và sinh có lợi ích cao

Để khắc phục vấn đề sinh quá nhiều kết quả trong các thuật toán HUIM, nhiều dạng biểu diễn súc tích của tập mục lợi ích cao đã được đề xuất. Tuy nhiên, việc thiết kế các thuật toán hiệu quả cho các biểu diễn này không đơn giản, vì đòi hỏi phải loại bỏ sớm các tập mục lợi ích thấp (LUI), cắt tỉa sớm các tập mục lợi ích cao không là đóng và/hoặc không là sinh, đồng thời thường xuyên kiểm tra quan hệ bao hàm giữa các tập mục. Một số thuật toán tiêu biểu theo hướng này gồm CHUD [30], CHUI-Miner [31], EFIM-Closed [32], HMiner-Closed [33], HUG-Miner [34], HUCI-Miner [9,35], C-HUIM [29] và MaxC-HUIM [29]. Các thuật toán này nhìn chung sử dụng các chặn trên TWU và để loại bỏ sớm các LUI; CHUD và CHUI-Miner sử dụng hai tập PrevSet và PostSet để cắt tỉa nhanh các HUI nhưng không là đóng; EFIM-Closed dùng cấu trúc utility-bin, kỹ thuật closure jumping và gộp giao dịch; HMiner-Closed dùng cấu trúc MCUL mở rộng từ CUL cùng khái niệm lợi ích đóng và không đóng để giảm số lần quét dữ liệu. Gần đây, trong [17], các tác giả đã đề xuất thuật toán FCGHUI-Miner để khai thác đồng thời các tập mục đóng và sinh có lợi ích cao, sử dụng chặn trên yếu của hàm lợi ích, chiến lược cắt tỉa địa phương theo hai mức liên tiếp của cây tiền tố, tránh kiểm tra quan hệ bao hàm, qua đó giảm mạnh ứng viên dư thừa và cải thiện đáng kể hiệu quả của quá trình khai thác.

3. Khai thác các luật kết hợp lợi ích cao

J. Sahoo và cs (2015) [9] là một trong những tác giả đầu tiên đề xuất bài toán khai thác luật kết hợp lợi ích cao. Họ đã đề xuất độ đo tin cậy dựa trên lợi ích (utility-confidence) để đánh giá luật trong không gian lợi ích, đồng thời giới thiệu các khái niệm tập mục đóng và tập mục sinh lợi ích cao, cùng thuật toán HUCI-Miner để khai thác đồng thời các loại tập mục này từ cơ sở dữ liệu định lượng. Từ các tập mục đóng và sinh có lợi ích cao, nhóm tác giả xây dựng một cơ sở luật kết hợp lợi ích cao không dư thừa, trong đó vế trái của luật là tập sinh và hợp hai vế của luật là tập đóng. Dựa trên hướng tiếp cận này, T. Mai và cs [12] đã đề xuất thuật toán LNR-HAR, sử dụng dàn của các tập mục lợi ích cao để sinh trực tiếp các luật kết hợp lợi ích cao không dư thừa, trong khi L.T.T. Nguyen và cs (2020) [11] giới thiệu các thuật toán HGB-HAR và LARM, kết hợp giữa tập đóng và tập sinh lợi ích cao, và cấu trúc dàn để cải thiện hiệu quả về thời gian và bộ nhớ của quá trình khai thác. Điểm chung của các công trình trên là tập trung vào việc giảm số lượng luật được khai thác và loại bỏ dư thừa sử dụng hàm tin cậy - lợi ích , nhưng chưa tích hợp một độ đo tương quan để bảo đảm mức độ gắn kết giữa các mục. Điều này dẫn đến khả năng sinh ra các luật có lợi ích cao nhưng sự tương quan giữa các thuộc tính trong luật còn yếu.

Khoảng trống nghiên cứu trên là động lực để bài báo này đề xuất bài toán khai thác các luật kết hợp súc tích có lợi ích và tương quan cao (SCoHUAR). Khác với các hướng tiếp cận trước đây, bài báo đề xuất một hàm lợi ích - tin cậy mới gọi là *cuconf* và tích hợp độ đo tương quan bond vào quy trình khai thác, đồng thời sử dụng các tập mục đóng và sinh lợi ích cao để bảo đảm tính súc tích và không dư thừa của các luật được khai thác.

3. Các khái niệm và định nghĩa cơ sở

Phần này trình bày bài toán khai thác các luật kết hợp súc tích có lợi ích và tương quan cao trên các cơ sở dữ liệu giao dịch định lượng. Định dạng dữ liệu đầu vào sử dụng trong bài toán này được xác định như sau.

Định nghĩa 1 (Cơ sở dữ liệu định lượng). Cho $A \stackrel{def}{=} \{a_j, j \in J \stackrel{def}{=} \{1,2,\ldots,M\}\}$ là tập hữu hạn các thuộc tính riêng biệt và $A \subseteq A$ là tập con của A. Nếu A chứa k thuộc tính, nó được gọi là một k-itemset, nghĩa là, k=|A|. Không mất tính tổng quát, giả sử rằng các mục trong mỗi tập mục được sắp xếp theo một thứ tự toàn phần $\prec$ (ví dụ thứ tự từ điển). Vector lợi ích $P \stackrel{def}{=} (p(a_j), j \in J$ và $p(a_j) > 0)$ biểu diễn lợi ích ngoài của các mục trong A, trong đó $p(a_j)$ thể hiện sự quan trọng tương đối của mục a_j, chẳng hạn như lợi ích đơn vị, giá, trọng số hay các thừa số khác. Một q-item là một cặp (a,q), trong đó a là một mục trong A và q là một số không âm thể hiện số lượng mua hay lợi ích trong của a. Một cơ sở dữ liệu định lượng (QDB) D là tập tất cả các giao tác $D \stackrel{def}{=} \{T_i, i \in I \stackrel{def}{=} \{1,2,\ldots,N\}\}$, trong đó mỗi giao tác T_i, $T_i \stackrel{def}{=} \{(a_j, q_i j), j \in J_i\}$ gồm tập các q-items, được xác định duy nhất bởi TID (ví dụ, TID= i) và J_i là tập con của J, $J_i \subseteq J$. Lợi ích của q-item (a,q), ký hiệu là u((a,q)), được định nghĩa là $u((a,q)) \stackrel{def}{=} q \times p(a)$. Bảng 2 chỉ ra ví dụ về QDB và vector lợi ích tương ứng (bảng 3).

Bảng 2. Cơ sở dữ liệu định lượng.

TID \ Item						
T_1	5	0	3	0	7	3
T_2	0	0	0	3	5	6
T_3	0	5	0	2	6	3
T_4	0	0	0	2	4	7
T_5	5	0	2	0	2	0
T_6	0	2	0	0	0	3
T_7	0	0	0	3	2	5
T_8	0	0	0	8	4	0

Bảng 3. Vector lợi ích.

a_j	*a*	*b*	*c*	*d*	*e*	*f*
p(aj)	4	6	4	3	2	3

Định nghĩa 2 (Cơ sở dữ liệu định lượng tích hợp). Để tránh tính toán lặp lại lợi ích u của các q-items (a_j, q_{ij}) trong các giao tác T_i trong $\mathcal{D}$, các giá trị này được tính toán 1 lần và được lưu trữ trong QDB tương ứng gọi là $\mathcal{D}'$, $\mathcal{D}' \triangleq \{T'_i \triangleq \{(a_j, q'_{ij}), j \in J_i\}, J_i \subseteq J, i \in I\}$, trong đó $q'_{ij} = q_{ij} * p(a_j)$ nếu $(a_j, q_{ij}) \in T_i$ (ngược lại, $q'_{ij} = 0$), $\forall i \in I, j \in J$. Khi đó, $\mathcal{D}'$ được gọi là một QDB tích hợp của $\mathcal{D}$.

Ví dụ minh họa. Bảng 2 và bảng 3 lần lượt biểu diễn QDB $\mathcal{D}$ và vector lợi nhuận $\mathcal{P}$, trong khi Bảng 4 minh họa QDB tích hợp $\mathcal{D}'$ thu được bằng cách kết hợp $\mathcal{D}$ và $\mathcal{P}$. QDB $\mathcal{D}'$ sẽ được sử dụng làm ví dụ minh họa xuyên suốt bài báo.

Bảng 4. Cơ sở dữ liệu định lượng tích hợp D'.

TID \ Item						
T_1	20	0	12	0	14	9
T2	0	0	0	9	10	18
T3	0	30	0	6	12	9
T4	0	0	0	6	8	21
T5	20	0	8	0	4	0
T6	0	12	0	0	0	9
T7	0	0	0	9	4	15
T8	0	0	0	24	8	0

Định nghĩa 3 (*Độ hỗ trợ của tập mục*). Cho bất kỳ tập mục $A \subseteq \mathcal{A}, A \neq \emptyset$, đặt $\rho(A) \triangleq \{T_i \in \mathcal{D}' \mid q'_{ij} > 0, \ \forall a_j \in A\}$ là tập các giao dịch trong $\mathcal{D}'$ chứa A, và theo quy ước $\rho(\emptyset) \triangleq \mathcal{D}'$. Độ hỗ trợ của A, ký hiệu $supp(A)$, được xác định là số lượng giao dịch trong QDB $\mathcal{D}'$ chứa A, nghĩa là $supp(A)$ chính là lực lượng (cardinality) của tập $\rho(A)$. Lưu ý rằng ta chỉ xét các tập mục khác rỗng A và xuất hiện ít nhất một lần trong $\mathcal{D}'$, nghĩa là $A \in \mathcal{IS} \triangleq \{A \subseteq \mathcal{A} \mid A \neq \emptyset \wedge \rho(A) \neq \emptyset\}$.

Ví dụ 1. Cho tập mục $A = ac$, vì $A \subseteq T_1$ và $A \subseteq T_5$, ta có $\rho(A) = \{T_1, T_5\}$ và $supp(A) = 2$.

Định nghĩa 4 (*Lợi ích của tập mục trong một giao tác*). Lợi ích của một mục a_j ($j \in J$) trong giao tác $T_i \in \mathcal{D}'$ được ký hiệu và định nghĩa là $u(a_j, T_i) \triangleq q'_{ij}$. Lợi ích của tập mục $A \in \mathcal{IS}$ trong T_i chứa A được định nghĩa như sau:

$$u(A, T_i) \triangleq \sum_{a_j \in A} u(a_j, T_i). \quad (1)$$

Định nghĩa 5 (*Lợi ích của tập mục trong cơ sở dữ liệu*). Lợi ích của tập mục trong được định nghĩa là

$$u(A) \triangleq \sum_{T_i \in \rho(A)} u(A, T_i), \quad (2)$$

trong đó $A \in \mathcal{IS}$ và $u(A) \in R_+ \triangleq \{x \in R \mid x > 0\}$.

Định nghĩa 6 (*Lợi ích giao tác và lợi ích cơ sở dữ liệu*). Lợi ích của giao tác T_i được xác định là $tu(T_i) \triangleq \sum_{q'_{ij} > 0} q'_{ij}$. Khi đó, lợi ích tổng cộng TU của $\mathcal{D}'$ là tổng lợi ích của tất cả các giao tác trong $\mathcal{D}'$, nghĩa là, $TU \triangleq \sum_{T_i \in \mathcal{D}} tu(T_i)$.

Ví dụ 2. Cho tập mục $A = ac$, khi đó $\rho(A) = \{T_1, T_5\}$, $u(A, T_1) = 20 + 12 = 32$ và $u(A, T_5) = 20 + 8 = 28$. Khi đó, $u(A) = u(A, T_1) + u(A, T_5) = 32 + 28 = 60$. Ta có $tu(T_1) = 20 + 12 + 14 + 9 = 55$ và $TU = 297$.

Định nghĩa 7 (*Tập mục lợi ích cao*). Với một ngưỡng lợi ích tối thiểu do người dùng chỉ định, ký hiệu là mu, một tập mục A được gọi là tập mục lợi ích cao (HUI) nếu lợi ích của nó trong QDB $\mathcal{D}'$ không nhỏ hơn ngưỡng này, nghĩa là $u(A) \geq mu$. Ngược lại, nếu $u(A) < mu$, thì A được xem là tập mục lợi ích thấp (LUI). Mục tiêu của bài toán khai thác tập mục lợi ích cao (HUIM) là khám phá tập $\mathcal{HUI}$ gồm tất cả các HUI, được xác định cụ thể như sau: $\mathcal{HUI} \triangleq \{A \in \mathcal{IS} \mid u(A) \geq mu\}$.

Định nghĩa 8 (*Độ đo tương quan bond [36]*). Độ đo bond đánh giá mức độ gắn kết giữa các mục trong một tập mục dựa trên tần suất đồng xuất hiện của chúng trong cơ sở dữ liệu. Với một tập mục A, giá trị bond của A được định nghĩa như sau:

$$bond(A) = \frac{supp(A)}{dissup(A)}, \quad (3)$$

trong đó $dissup(A) = |\{T \in \mathcal{D} \mid A \cap T \neq \emptyset\}|$ chỉ ra số giao tác trong $\mathcal{D}$ chứa ít nhất một mục của A.

Độ đo bond của tập mục A thỏa điều kiện: $0 < bond(A) \leq 1$. Giá trị bond càng lớn cho thấy mức độ tương quan giữa các mục trong A càng mạnh, trong khi giá trị nhỏ phản ánh sự gắn kết yếu.

Định nghĩa 9 (*Tập mục lợi ích và tương quan cao*). Ký hiệu mu là ngưỡng lợi ích tối thiểu do người dùng đặt ra và mb là ngưỡng bond tối thiểu. Một tập mục $A \subseteq \mathcal{A}$ được gọi là tập mục lợi ích và tương quan cao (CoHUI) nếu nó đồng thời thỏa mãn hai ràng buộc về lợi ích và tương quan, nghĩa là $u(A) \geq mu$ và $bond(A) \geq mb$. Tập tất cả các CoHUI trong $\mathcal{D}'$ được xác định như sau:

$$\mathcal{CoHUI} = \{A \in \mathcal{IS} \mid u(A) \geq mu \wedge bond(A) \geq mb\}. \quad (4)$$

trong đó, $\mathcal{IS}$ là danh sách tất cả các tập mục có thể trong không gian tìm kiếm. Định nghĩa này bảo đảm rằng các mẫu được phát hiện không chỉ có lợi ích cao mà còn thể hiện sự tương quan mạnh giữa các mục, từ đó nâng cao khả năng diễn giải và giá trị ứng dụng trong thực tiễn.

Ví dụ 3. Cho tập mục $A = ce$, ta có $supp(A) = 2$, và $dissup(A) = 7$. Do đó, $bond(A) = \frac{supp(A)}{dissup(A)} = \frac{2}{7} = 0.2857$. Ngoài ra, $u(A) = 38$. Nếu $mu = 30$ và $mb = 0.2$, thì $u(A) > mu$ and $bond(A) > mb$. Vì vậy, A là một CoHUI.

Định nghĩa 10 (*Lớp tương đương trên các tập mục*). Dựa trên định nghĩa của toán tử ρ, quan hệ tương đương trên các tập mục trong $\mathcal{IS}$ được ký hiệu là $\sim$ và được định nghĩa như sau: $\forall A, B \in \mathcal{IS}$, $A \sim B \Leftrightarrow \rho(A) = \rho(B)$. Khi đó, lớp tương đương của A là $[A] \triangleq \{B \in \mathcal{IS} \mid \rho(B) =$

Định nghĩa 11 (*Tập mục lợi ích cao đóng*). Một HUI A được gọi là *tập mục lợi ích cao đóng* (CHUI) nếu không tồn tại một HUI nào khác là tập cha thật sự của A và có cùng độ hỗ trợ với A. Tập đầy đủ tất cả các CHUI được ký hiệu và định nghĩa hình thức như sau:

$$\mathcal{CHUI} \stackrel{\text{def}}{=} \{A \in \mathcal{HUI} \mid \nexists B \in \mathcal{HUI}: B \supset A \wedge supp(B) = supp(A)\}. \quad (5)$$

Định nghĩa 12 (*Bao đóng của tập mục*). Bao đóng của một tập mục A, ký hiệu $h(A)$, được định nghĩa là một tập mục C sao cho C là tập cha lớn nhất của A và có độ hỗ trợ bằng độ hỗ trợ của A, nghĩa là $supp(A) = supp(C)$. Nếu A là CHUI, thì $A = h(A)$.

Định nghĩa 13 (*Tập mục sinh lợi ích cao*). Một HUI A được gọi là tập sinh lợi ích cao (GHUI) nếu không tồn tại một HUI nào khác là tập con thật sự của A và có cùng độ hỗ trợ với A. Tập tất cả các GHUI được ký hiệu và được định nghĩa như sau:

$$\mathcal{GHUI} \stackrel{\text{def}}{=} \{A \in \mathcal{HUI} \mid \nexists B \in \mathcal{HUI}: B \subset A \wedge supp(B) = supp(A)\}. \quad (6)$$

Cho bất kỳ $A \in \mathcal{CHUI}$ và $B \in \mathcal{GHUI}$, B được gọi là tập mục sinh của A nếu $B \subseteq A$ và $supp(B) = supp(A)$. Ký hiệu $\mathcal{G}(A)$ là tập tất cả các GHUI của A.

Dựa trên định nghĩa của 2 tập $\mathcal{GHUI}$, $\mathcal{CHUI}$ và độ đo $bond$, ta đạt được hai tập $\mathcal{CoGHUI}$ và $\mathcal{CoCHUI}$ được định nghĩa hình thức như sau:

$$\mathcal{CoGHUI} \stackrel{\text{def}}{=} \{A \in \mathcal{GHUI} \mid bond(A) \geq mb\}, \text{ and}$$

$$\mathcal{CoCHUI} \stackrel{\text{def}}{=} \{A \in \mathcal{CHUI} \mid bond(A) \geq mb\}. \quad (7)$$

Định nghĩa 14 (*Lợi ích cục bộ của một mục [9]*). Cho bất kỳ mục a_j và tập mục A, lợi ích cục bộ của a_j trong A, $luv(a_j, A)$, là tổng lợi ích của a_j trong các giao tác chứa A, cụ thể:

$$luv(a_j, A) \stackrel{\text{def}}{=} \sum_{T_i \in \rho(A)} u(a_j, T_i). \quad (8)$$

Ví dụ 4. Xét tập mục $A = ce$. Ta có $\rho(A) = \{T_1, T_5\}$, $u(e, T_1) = 14$ và $u(e, T_2) = 4$. Vì vậy, $luv(e, A) = 14 + 4 = 18$.

Định nghĩa 15 (*Lợi ích cục bộ của một tập mục [9]*). Cho bất kỳ 2 tập mục A và B sao cho $A \subseteq B$, lợi ích cục bộ của A trong B được định nghĩa như sau:

$$luv(A, B) \stackrel{\text{def}}{=} \sum_{a_j \in A} luv(a_j, B). \quad (9)$$

Ví dụ 5. Xét hai tập mục $A = de$ và $B = def$. Khi đó, $\rho(B) = \{T_2, T_3, T_4, T_7\}$ và $luv(A, B) = luv(d, B) + luv(e, B) = 30 + 34 = 64$.

Định nghĩa 16 (*Độ tin cậy lợi ích của một luật [9]*). Với hai tập mục khác rỗng A và B, trong đó $A, B \in \mathcal{IS}$ và $A \cap B = \emptyset$, một luật kết hợp R có dạng $R: A \rightarrow B$, trong đó A và B lần lượt được gọi là vế trái (antecedent) và vế phải (consequent) của luật. Độ tin cậy - lợi ích của R được ký hiệu và định nghĩa như sau:

$$uconf(R) \stackrel{\text{def}}{=} \frac{luv(A,\ A \cup B)}{u(A)}. \quad (10)$$

Độ tin cậy-lợi ích $uconf(R)$ phản ánh mức độ đóng góp lợi ích của các mục trong A khi chúng xuất hiện cùng với B.

Ví dụ 6. Xét luật $R: e \rightarrow f$. Ta có, $luv(e, ef) = 48$ và $u(e) = 60$. Do đó, $uconf(R) = \frac{48}{60} = 0.8$.

Lưu ý rằng, mặc dù hàm tin cậy - lợi ích $uconf$ truyền thống phản ánh mức độ lợi ích của A trong các giao tác mà A cùng xuất hiện với B, nhưng nó chỉ cung cấp một góc nhìn một chiều: $uconf$ chỉ xem xét đóng góp lợi ích của A mà bỏ qua phần lợi ích thực sự do B tạo ra trong các giao tác đó. Trong nhiều tình huống thực tế, như đánh giá gói sản phẩm hoặc chiến lược bán kèm, việc biết mức lợi nhuận mà B đóng góp khi xuất hiện cùng với A là thông tin quan trọng không kém, thậm chí quan trọng hơn. Hạn chế này dẫn đến nhu cầu xây dựng một hàm tin cậy - lợi ích mới gọi là $cuconf$, so sánh lợi ích của B trong các giao tác chứa $A \cup B$ với tổng lợi ích của A. Nhờ đó, $cuconf$ cung cấp cái nhìn rõ ràng hơn về hiệu quả kết hợp giữa hai tập mục A và B.

Định nghĩa 17 (*Độ tin cậy - lợi ích chéo của luật kết hợp*). Cho bất kỳ luật $R: A \rightarrow B$, độ tin cậy-lợi ích chéo của luật kết hợp R được ký hiệu và định nghĩa như sau:

$$cuconf(R) \stackrel{\text{def}}{=} \frac{luv(B,\ A \cup B)}{u(A)}. \quad (11)$$

Độ tin cậy-lợi ích chéo $cuconf(R)$ đo lường mức đóng góp lợi ích của tập mục B trong các giao tác chứa cả A và B, và so sánh phần lợi ích này với tổng lợi ích của A. Nếu $cuconf(R)$ cao, điều đó cho thấy rằng khi A xuất hiện, việc bổ sung B mang lại phần lợi ích đáng kể, phản ánh sự gắn kết mạnh của cặp tập mục này.

Khác với $uconf$, vốn chỉ xem xét đóng góp lợi ích của A, độ đo $cuconf$ trực tiếp đánh giá giá trị mà B tạo ra trong ngữ cảnh của luật, từ đó cung cấp góc nhìn đầy đủ hơn về tính hiệu quả của cặp tập mục A và B.

Xét bối cảnh trong siêu thị với A là *mì gói* và B là *xúc xích*. Khi phân tích dữ liệu giao dịch, giả sử rằng trong những giao tác chứa cả mì gói và xúc xích, xúc xích tạo ra phần lớn doanh thu, trong khi mì gói chỉ đóng góp mức lợi nhuận nhỏ. Khi đó, $cuconf(A \rightarrow B)$ sẽ cao và cho nhà quản lý biết rằng: *"Khi khách mua mì gói, họ thường mua xúc xích và phần lợi nhuận đáng kể đến từ xúc xích"*. Điều này giúp doanh nghiệp đưa ra quyết định về chiến lược bán kèm, đặt kệ hàng, hoặc gói combo để tối đa hóa lợi nhuận.

Ví dụ 7. Xét luật $R: e \rightarrow f$ trên cơ sở dữ liệu minh họa. Ta có $u(e) = 60$. Trong các giao tác chứa đồng thời e và

f, tổng lợi ích của e là $luv(e, ef) = 48$ và tổng lợi ích của f là $luv(f, ef) = 72$. Do đó, $uconf(R) = 48/60 = 0{,}8$, $cuconf(R) = 72/60 = 1.2$. Giá trị $uconf(R)$ cho thấy mức lợi ích của e được duy trì trong các giao tác có chứa cả e và f. Trong khi đó, $cuconf(R) > 1$ phản ánh rằng f đóng góp lợi ích lớn hơn tổng lợi ích của e, cho thấy việc mở rộng từ e sang f mang lại hiệu quả mạnh hơn nhiều. Điều này minh họa rõ sự khác biệt trong góc nhìn của hai độ đo và vai trò bổ sung của $cuconf$ khi đánh giá các luật kết hợp theo lợi ích.

Định nghĩa 18 (Luật kết hợp lợi ích và tương quan cao). Luật R: $A \rightarrow B$ được gọi là một luật kết hợp lợi ích và tương quan cao (CoHUAR) nếu R thỏa những điều kiện sau: $u(A) \geq mu$, $u(A \cup B) \geq mu$, $bond(A \cup B) \geq mb$ và $cuconf(R) \geq muconf$, trong đó $muconf$ là ngưỡng tin cậy - lợi ích tối thiểu được định nghĩa bởi người dùng. Tập $\mathcal{CoHUAR}$ gồm tất cả các CoHUARs được định nghĩa hình thức như sau:

$$\mathcal{CoHUAR} = \{R: \ A \rightarrow B \mid u(A) \geq mu, \ u(A \cup B) \geq mu, \ bond(A \cup B) \geq mb \wedge cuconf(R) \geq muconf\}. \quad (12)$$

Ví dụ 8. Xét cơ sở dữ liệu minh họa trong Bảng 3 với các ngưỡng $mu = 30$, $mb = 0.2$ và $muconf = 0{,}6$. Với luật $R: e \rightarrow f$, ta có $u(e) = 60$, $u(ef) = 120$, và $bond(ef) = \frac{5}{8} = 0.625$. Ngoài ra, độ tin cậy-lợi ích chéo của luật là $cuconf(R) = 1.2$. Do đó, luật thỏa đồng thời bốn điều kiện: $u(e) \geq mu$, $u(ef) \geq mu$, $bond(ef) \geq mb$ và $cuconf(R) \geq muconf$. Vì vậy, R là một CoHUAR.

Dựa trên các định nghĩa về CHUI, GHUI và độ đo bond, bài báo này giới thiệu khái niệm luật kết hợp súc tích có lợi ích và tương quan cao (SCoHUAR), trong đó vế trái của mỗi luật là một GHUI và hợp của vế trái với vế phải tạo thành một CHUI. Định nghĩa hình thức của SCoHUAR được trình bày như sau.

Định nghĩa 19 (*Luật kết hợp súc tích có lợi ích và tương quan cao*). Cho trước một CoHUAR R: $A \rightarrow B$, R được gọi là luật kết hợp súc tích có lợi ích và tương quan cao (SCoHUAR) nếu $A \in \mathcal{CoGHUI}$ và $A \cup B \in \mathcal{CoCHUI}$. Tập $\mathcal{SCoHUAR}$ gồm tất cả các SCoHUAR trong $\mathcal{D}'$ được định nghĩa hình thức như sau:

$$\mathcal{SCoHUAR} = \{R: A \rightarrow B \mid R \in \mathcal{CoHUAR} \wedge A \in \mathcal{CoGHUI} \wedge A \cup B \in \mathcal{CoCHUI}\}. \quad (13)$$

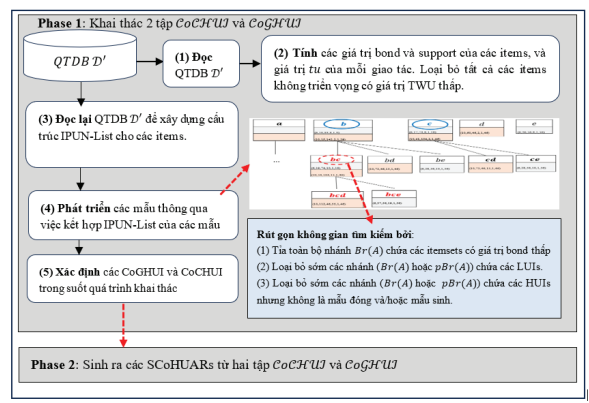
Phát biểu bài toán. Cho SDB $\mathcal{D}'$ và ba ngưỡng dương do người dùng chỉ định, gồm lợi ích tối thiểu (mu), độ tin cậy-lợi ích tối thiểu ($muconf$), và mức tương quan bond tối thiểu (mb), mục tiêu của bài toán đặt ra trong bài báo này là khám phá tập $\mathcal{SCoHUAR}$ gồm tất cả các SCoHUAR có trong $\mathcal{D}'$.

4. Khai thác các luật kết hợp súc tích có lợi ích và tương quan cao

4.1. Quá trình khai thác tổng thể các SCoHUAR

Mục này trình bày chi tiết quy trình khai thác của phương pháp được đề xuất, được minh họa trong hình 1. Quy trình tổng thể gồm hai giai đoạn. Giai đoạn thứ nhất dùng để khám phá hai tập $\mathcal{CoCHUI}$ và $\mathcal{CoGHUI}$ từ SDB $\mathcal{D}'$, trong khi giai đoạn thứ hai dùng để sinh tất cả các SCoHUAR từ hai tập này. Giai đoạn thứ nhất bắt đầu bằng việc đọc QDB $\mathcal{D}'$ để tính các giá trị bond và support của các mục, cũng như tổng lợi ích của các giao dịch (tu). Các mục có giá trị TWU thấp bị loại bỏ khỏi $\mathcal{D}'$, và các mục còn lại được sắp xếp theo thứ tự tăng dần của support. QDB $\mathcal{D}'$ sau đó được quét lại để mỗi giao dịch được sắp xếp theo thứ tự support. Cấu trúc dữ liệu IPUN-list, được cải tiến từ MPUN-list trong [29], sau đó được xây dựng cho các mục còn lại. Sau khi quét xong tất cả các giao dịch, các IPUN-list của các 1-itemset được kết hợp để hình thành các 2-itemset và IPUN-list tương ứng của chúng. Tiếp theo, các 3-itemset được sinh ra bằng cách sử dụng IPUN-list của các 2-itemset. Quy trình tiếp tục được lặp lại cho đến khi hai tập $\mathcal{CoCHUI}$ và $\mathcal{CoGHUI}$ được xác định.

Lưu ý rằng, để khai thác đồng thời hai tập $\mathcal{GHUI}$ và $\mathcal{CHUI}$ trong giai đoạn 1, nghiên cứu [17] đã đề xuất thuật toán FCGHUI-Miner. Bài báo này kế thừa và áp dụng trực tiếp thuật toán đó để khám phá hai tập $\mathcal{GHUI}$ và $\mathcal{CHUI}$. Vì vậy, các kết quả lý thuyết liên quan đến việc thiết kế các chiến lược loại bỏ các tập mục có lợi ích thấp hoặc các tập mục có lợi ích cao nhưng không phải là mẫu đóng và/hoặc mẫu sinh sẽ không được trình bày lại trong. Tuy nhiên, do bài báo còn tích hợp thêm độ đo tương quan *bond* nhằm khai thác các mẫu vừa có lợi ích cao vừa có mức độ tương quan mạnh, chúng tôi mở rộng FCGHUI-Miner bằng cách tích hợp tính chất trong mệnh đề 1 (liên quan đến độ đo



Hình 1. Quá trình khai thác tổng thể các SCoHUAR.

bond) vào quá trình khai thác để tìm 2 tập $CoCHUI$ và $CoGHUI$. Thuật toán sau khi sửa đổi được gọi là CoFCGHUI-Miner. Mệnh đề 1 được phát biểu như sau:

Mệnh đề 1. **(***Chiến lược tỉa dựa trên bond***)**. Xét tập mục A bất kỳ, nếu giá trị *bond* của A nhỏ hơn ngưỡng tối thiểu mb, nghĩa là $bond(A) < mb$, thì A và tất cả tập cha của A được gọi là tương quan yếu, và có thể được loại bỏ.

Một thách thức quan trọng khi tích hợp độ đo *bond* vào thuật toán CoFCGHUI-Miner là chi phí tính toán *bond* cho các tập mục trong suốt quá trình khai thác thường rất lớn. Để giải quyết hạn chế này, bài báo tích hợp *bit vector* (*bv*) vào cấu trúc dữ liệu IPUN-list (được sử dụng trong thuật toán CoFCGHUI-Miner) để giảm thời gian tính toán giá trị *bond* của các tập mục. Với sự mở rộng này, bit vector trong cấu trúc IPUN-list của mỗi tập mục được tính toán như sau.

Cách xác định *bv* trong IPUN-List. Việc tính toán giá trị bond của tập mục A phụ thuộc vào hai giá trị $supp(A)$ và $dissup(A)$. Giá trị $supp(A)$ được tính toán trực tiếp từ các thông tin có sẵn trong IPUN-List. Tuy nhiên, $dissup(A)$ thường được tính bằng cách quét lại toàn bộ QDB, một thao tác tốn kém, đặc biệt đối với các QDB lớn. Để khắc phục hạn chế này, IPUN-List bổ sung thêm trường bv (bit vector). Mỗi bit trong bv tương ứng với một giao tác trong cơ sở dữ liệu: bit có giá trị 1 nếu giao tác chứa ít nhất một mục thuộc A, và 0 nếu ngược lại. Đối với các 1-itemset (tập mục chỉ có 1 thuộc tính), bv được khởi tạo trực tiếp trong lần quét QDB đầu tiên. Đối với các k-itemsets ($k > 1$), bv được cập nhật hiệu quả trong quá trình thực hiện phép mở rộng A bằng cách thực hiện phép toán OR trên các bit vector của hai tập cha: $bv(Pxy) = bv(Px)\ OR\ bv(Py)$. Ví dụ, giả sử $bv(a) = 10011$ và $bv(b) = 11101$, ta có $bv(ab) = 11111$. Khi đó, $dissup(A)$ của tập mục A được xác định như sau: $dissup(X) = |bv(A)|$, tức là số lượng bit 1 trong bit vector.

4.2. Sinh ra nhanh các SCoHUAR

Nhắc lại rằng việc sinh ra các SCoHUAR dạng $R: A \rightarrow B$, trong đó $A \in CoGHUI$ và $A \cup B \in CoCHUI$ giúp đảm bảo rằng các luật SCoHUAR được biểu diễn ở dạng ngắn gọn nhất và không dư thừa. Các luật này không chỉ giảm số lượng luật được sinh ra mà còn tăng khả năng diễn giải, bởi mỗi luật đều được xem như đại diện cho một lớp tương đương các luật có cùng độ hỗ trợ. Điều này giúp các nhà phân tích dễ hiểu và ứng dụng các luật vào các bài toán ra quyết định dựa trên dữ liệu.

Để bảo đảm việc sinh ra đầy đủ và không trùng lặp tất cả các luật trong tập $SCoHUAR$, bài báo đề xuất một cách tiếp cận dựa trên phân hoạch (partitioning) của không gian luật. Phân hoạch này cho phép chia tập $SCoHUAR$ thành các tập con rời nhau, mỗi tập con được đại diện bởi một cặp tập đóng lồng nhau. Ý tưởng cốt lõi là: đối với mỗi tập mục $A \in CoGHUI$, ta chỉ cần xem xét các mở rộng $A \cup B \in CoCHUI$, và từ đó sinh ra tất cả các luật $R: A \rightarrow B$ thuộc lớp tương ứng. Với cách sinh luật này, toàn bộ không gian của tập $SCoHUAR$ được chia thành các phân hoạch độc lập dựa trên các CoGHUI và CoCHUI. Cách tiếp cận này không chỉ đảm bảo rằng mỗi luật được sinh ra đúng một lần mà còn giúp giảm đáng kể chi phí khai thác bằng cách hạn chế vùng tìm kiếm cho mỗi phân hoạch. Phương pháp phân hoạch được trình bày trong định lý sau.

Định lý 1 (*Phân hoạch tập* $SCoHUAR$).

$$SCoHUAR = \sum_{(L,S) \in NCHUI} SCoHUAR(L,S)\ , \quad (14)$$

trong đó $NCHUI = \{(L,S) \mid \emptyset \neq L \subseteq S \wedge L,S \in CoCHUI\}$ và $SCoHUAR(L,S)$ được xác định như sau:

a) $SCoHUAR(L,S) = \{R: L_i \rightarrow S \backslash L_i \mid L_i \in G(L) \wedge cuconf(R) \geq muconf\}, if\ L \subset S$.

b) $SCoHUAR(L,L) = \{R: L_i \rightarrow L \backslash L_i \mid L_i \in G(L) \wedge cuconf(R) \geq muconf\}, if\ L \notin G(L)$.

Tất cả các luật trong tập $SCoHUAR$ được sinh ra đầy đủ và không trùng lặp.

Chứng minh.

1) *Các luật trong tập* $SCoHUAR$ *được sinh ra đầy đủ.*

+ Ta sẽ Chứng minh $SCoHUAR \subseteq \sum_{(L,S) \in NCHUI} SCoHUAR(L,S)$. Lấy tùy ý $R: A \rightarrow B \in SCoHUAR$. Khi đó $A \in CoGHUI$, $A \cup B \in CoCHUI$ và $cuconf(R) \geq muconf$. Đặt $L = h(A)$, $S = h(A \cup B)$, trong đó $h(A)$ là bao đóng của A. Vì $A \cup B \in CoCHUI$ nên $S = A \cup B$. Do $A \in CoGHUI$ nên $A \in G(L)$. Suy ra, $\emptyset \neq L \subseteq S$, $S \in CoCHUI$, nên $(L,S) \in NCHUI$. Ta lại có $S \backslash A = (A \cup B) \backslash A = B$.

Nếu $L \subset S$ thì $R: A \rightarrow B$ có dạng $L_i \rightarrow S \setminus L_i$ với $L_i = A \in G(L)$, nên $R \in SCoHUAR(L, S)$ theo (a).

Nếu $L = S$ và $L \notin G(L)$ thì $R: A \rightarrow B$ có dạng $L_i \rightarrow L \setminus L_i$ với $L_i = A \in G(L)$, nên $R \in SCoHUAR(L, L)$ theo (b). Vậy $SCoHUAR \subseteq \sum_{(L,S) \in NCHUI} SCoHUAR(L,S)$.

+ Chiều ngược lại là hiển nhiên từ định nghĩa của $SCoHUAR(L,S)$.

2) *Các luật trong* $SCoHUAR$ *được sinh không trùng lặp*.

Giả sử tồn tại $R: A \rightarrow B$ sao cho $R \in SCoHUAR(L_1,S_1) \cap SCoHUAR(L_2,S_2)$, $(L_1,S_1) \neq (L_2,S_2)$. Theo định nghĩa, vế trái của R là một GHUI của mỗi L_j, nên $A \in G(L_1), A \in G(L_2) \Rightarrow h(A) = L_1 = L_2$. Mặt khác, vế phải của R lần lượt là $S_1 \setminus A$ và $S_2 \setminus A$, nên $S_1 \backslash A =$

$B = S_2 \backslash A \Rightarrow S_1 = A \cup B = S_2$. Suy ra $(L_1, S_1) = (L_2, S_2)$, mâu thuẫn với giả thiết. Do đó, nếu $(L_1, S_1) \neq (L_2, S_2)$ thì $\mathcal{SCoHUAR}(L_1, S_1) \cap \mathcal{SCoHUAR}(L_2, S_2) = \emptyset$.

Từ (1) và (2), ta có các luật SCoHUAR được sinh ra đầy đủ và không trùng lặp.

4.3. Kỹ thuật tối ưu hóa

(a) Giảm số lần kiểm tra quan hệ bao hàm giữa các tập mục

Để tăng tốc việc kiểm tra quan hệ bao hàm giữa các tập mục khi áp dụng Định lý 1 để sinh ra các SCoHUAR trong tập $\mathcal{SCoHUAR}$, bài báo áp dụng kỹ thuật băm dựa trên giá trị TWU. Mỗi tập mục A được gán một khóa băm $TWU(A) = \sum_{T_i \in \rho(A)} tu(T_i)$ (xem [17]). Nếu hai tập mục xuất hiện trong cùng tập giao tác của QDB thì chúng có cùng TWU. So với việc dùng $supp(A)$ làm khóa băm, số tập mục có cùng TWU thường ít hơn đáng kể, nên số lần trùng khóa băm và số lần kiểm tra quan hệ bao hàm cũng được giảm đáng kể, góp phần cải thiện hiệu quả tính toán của thuật toán M-SCoHUAR được đề xuất trong Mục 4.3

Ví dụ 9. Xét QDB tích hợp trong Bảng 4 và tập mục $A = \{b\}$ với $supp(A) = 2$ và $TWU(A) = 78$. Kết quả cho thấy có 8 tập mục có cùng độ hỗ trợ bằng 2, trong khi chỉ có 2 tập mục có cùng giá trị TWU bằng 78. Sự chênh lệch này cho thấy việc sử dụng TWU làm khóa băm giúp giảm đáng kể số lượng tập mục trùng khóa so với sử dụng độ hỗ trợ.

(b) Xác định nhanh tập $\mathcal{G}(L)$ và cuconf

Dựa trên ý tưởng khóa băm theo TWU, sau khi khai thác 2 tập $\mathcal{CoCHUI}$ và $\mathcal{CoGHUI}$ sử dụng thuật toán CoCGHUI-Miner, với mỗi tập mục $L \in \mathcal{CoCHUI}$, ta xác định bản ghi $\langle L, supp(L), u(L), TWU(L), \{luv(x, L)\}_{\{x \in L\}}, \mathcal{G}(L)\rangle$ và chèn bản ghi này vào bảng băm HT_{closed} với khóa là $TWU(L)$. Trường $luv(x, L)$ lưu sẵn giá trị $luv(x, L) = \sum_{T_i \in \rho(L)} u(x, T_i)$ cho từng mục $x \in L$; trường $\mathcal{G}(L)$ là danh sách các CoGHUI có bao đóng là L. Với mỗi CoGHUI $X \in \mathcal{CoGHUI}$, ta truy cập danh sách $HT_{closed}[TWU(X)]$, xác định tập đóng L sao cho $X \subset L$ và $supp(X) = supp(L)$, và đưa X vào $\mathcal{G}(L)$. Chú ý rằng, các thông tin $supp$, u và TWU của L và X được tính sẵn trong thuật toán CoFCGHUI-Miner.

Nhờ cách tổ chức này, khi sinh một luật $R: A \rightarrow B$ với $A \in \mathcal{G}(L)$ và $A \cup B = L$, giá trị $luv(B, A \cup B) = luv(B, L) = \sum_{b \in B} luv(b, L)$ được tính rất nhanh bằng cách cộng các giá trị $luv(b, L)$ đã lưu sẵn trong bản ghi của L. Do đó, $cuconf(R) = luv(B, A \cup B)/u(A)$ được tính trong thời gian tuyến tính theo $|B|$, giúp toàn bộ quá trình sinh các SCoHUAR theo Định lý 1 diễn ra hiệu quả.

4.4. Thuật toán đề xuất M-SCoHUAR

Dựa vào phương pháp sinh các SCoHUAR trong Định lý 1 và các kỹ thuật tối ưu hóa trong Nhận xét 1, bài báo đề xuất thuật toán *M-SCoHUAR* để sinh ra nhanh tất cả các luật trong tập $\mathcal{SCoHUAR}$. Mã giả của *M-SCoHUAR* được trình bày trong Hình 2.

Thuật toán M-SCoHUAR hoạt động như sau. Trước hết, thuật toán xây dựng bảng băm các tập mục đóng có lợi ích và tương quan cao theo khóa $TWU(L)$ (dòng 1–6): với mỗi $L \in \mathcal{CoCHUI}$, bản ghi $rec(L) = \langle L, supp(L), u(L), TWU(L), \{luv(x, L)\}_{\{x \in L\}}, G(L) := \emptyset\rangle$ được tạo ra và chèn vào trong $HT_{closed}[TWU(L)]$. Tiếp theo, đối với mỗi tập sinh $A \in \mathcal{CoGHUI}$, thuật toán tính $TWU(A)$, duyệt các bản ghi trong $HT_{closed}[TWU(A)]$ để tìm tập đóng L thỏa $h(A) = L$ (hay $A \subseteq L$ và $supp(A) = supp(L)$); khi điều kiện này được thỏa, A được thêm vào danh sách $\mathcal{G}(L)$ (dòng 7–15). Dựa trên cấu trúc này, thuật toán hình thành tập $NCHUI = \{(L, S) \mid \emptyset \neq L \subseteq S \wedge L, S \in \mathcal{CoCHUI}\}$ theo Định lý 1 (dòng 16), trong đó mỗi cặp (L, S) xác định một phân hoạch con của tập kết quả $\mathcal{SCoHUAR}$. Cuối cùng, với mỗi $(L, S) \in NCHUI$, thuật toán duyệt từng $L_i \in \mathcal{G}(L)$, xác định vế phải của luật là $B = S \setminus L_i$, sau đó tính nhanh $luv(B, S) = \sum_{b \in B} luv(b, S)$ nhờ các giá trị $luv(b, S)$ đã lưu sẵn trong $rec(S)$, và tính $cuconf(R) = luv(B, S)/u(L_i)$; nếu $cuconf(R) \geq muconf$ thì luật $R: L_i \rightarrow B$ được đưa vào tập $\mathcal{SCoHUAR}$ (dòng 17–28). Nhờ việc kết hợp băm theo TWU, ánh xạ trực tiếp từ mỗi tập đóng L sang danh sách các tập mục sinh trong $\mathcal{G}(L)$ và phân hoạch $NCHUI$, thuật toán M-SCoHUAR sinh đầy đủ và không trùng lặp tất cả các SCoHUAR trong tập $\mathcal{SCoHUAR}$ một cách hiệu quả.

Ví dụ 10. Với QDB cho trong Bảng 4, $mu = 30$, $mb = 0.2$ và $muconf = 0{,}6$, ta có $\mathcal{CoCHUI} = \{e, f, ef, ace, def, de, bf\}$ và $\mathcal{CoGHUI} = \{e, f, ef, a, ce, df, d, b\}$. Khi đó, ta có tập kết quả $\mathcal{SCoHUAR}$ gồm 9 luật với thông tin chi tiết cho trong bảng 5.

Bảng 5. Danh sách các luật kết hợp súc tích có lợi ích và tương quan cao.

TT	Luật R:A→B	u(A)	*u* (A∪B)	Bond (A∪B)	Cuconf (R)
1	e→f	60	120	0.625	1.2
2	e→ac	60	78	0.286	1.0
3	e→df	60	127	0.5	1.55
4	e→d	60	96	0.714	0.9
5	a→ce	40	78	0.286	0.95
6	ce→a	38	78	0.286	1.05
7	f→de	81	127	0.5	0.79
8	d→ef	54	127	0.5	1.796
9	d→e	54	96	0.71	0.78

Độ phức tạp của thuật toán. Trong Mục IV.1, bài báo chỉ ra rằng quá trình tổng thể sinh các SCoHUAR bao gồm hai giai đoạn. Giai đoạn thứ nhất sử dụng thuật toán CoFCGHUI-Miner, được cải tiến từ FCGHUI-Miner [17] thông qua việc tích hợp độ đo bond, để khai thác hai tập $\mathcal{CoCHUI}$ và $\mathcal{CoGHUI}$. Giai đoạn thứ hai tiến hành sinh toàn bộ các SCoHUAR từ hai tập kết quả này. Dựa trên các quan sát thực nghiệm được thực hiện trên những cơ sở dữ liệu lớn với ngưỡng mu, ms và mb nhỏ, giai đoạn thứ nhất thường tiêu tốn thời gian xử lý nhiều hơn đáng kể so với giai đoạn thứ hai. Do đó, trong bài báo này, độ phức tạp thời gian trong trường hợp xấu nhất của thuật toán CoFCGHUI-Miner được phân tích như một ví dụ minh họa. Cần lưu ý rằng độ phức tạp tính toán của thuật toán chủ yếu phụ thuộc vào số lượng các phép mở rộng có thể xảy ra. Gọi M là số lượng các mục trong tập $\mathcal{A}$, và $L = maximum\{|t|: t \in \mathcal{D}\}$ biểu thị độ dài giao dịch lớn nhất trong cơ sở dữ liệu $\mathcal{D}$. Xét tập mục A bất kỳ với chiều dài $K = |A|$, và đặt $a_k = lastItem(A)$ với $K \le k \le M$. Thủ tục đệ quy Find-FCloGenHUI(A, mu, ms, mb, $\mathcal{CoCHUI}$, $\mathcal{CoGHUI}$) (hay ngắn gọn Find-FCloGenHUI(A)) có thể sinh ra nhiều nhất $E(k) \stackrel{\text{def}}{=} M - k$ các tập mở rộng (xem dòng 15 của Hình 4 trong [17]). Tại mức K ($1 \le K \le L$), gọi $C(K,k)$ và $\mathcal{C}(K)$ là độ phức tạp thời gian của Find-FCloGenHUI(A) để xử lý tương ứng một tập mục A và tất cả các tập mục A với $|A| = K$. Gọi $HUS(K)$ là tập tất cả các tập mục lợi ích cao có chiều dài K, và quy ước $|HUS(0)| = 1$. Khi đó, số lượng tập mở rộng tối đa của tất cả các tập mục có độ dài K là $ME(K) \stackrel{\text{def}}{=} \sum_{k=K..M} E(k) = \frac{(M-K)(M-K+1)}{2} = O(M^2)$. Do đó, ta có $C(K,k) = E(k) + \sum_{k'=k+1..M} C(K+1,k')$ với $K = 1,\dots,L-1$, $\mathcal{C}(K) = |HUS(K-1)| \times \sum_{k=K..M} C(K,k)$, và các điều kiện biên $C(L,k) \stackrel{\text{def}}{=} 0$, $\mathcal{C}(L) \stackrel{\text{def}}{=} 0$. Vì vậy, ta đạt được kết quả $C(L-1,k) = E(k) = M - k = O(M-k)$ và $\mathcal{C}(L-1) = |HUS(L-2)| \times ME(L-1) = |HUS(L-2)| \times \frac{(M-L+1)(M-L+2)}{2} = |HUS(L-2)| \times O((M-L+2)^2)$. Tương tự, $C(L-2,k) = \frac{(M-k)(M-k+1)}{2} = O((M-k)^2)$ và $\mathcal{C}(L-2) = |HUS(L-3)| \times \left[\frac{(M-L+2)(M-L+3)(M-L+4)}{6}\right] = |HUS(L-3)| \times O((M-L+2)^3)$. Bằng phương pháp quy nạp, ta thu được dạng tổng quát sau: $C(K,k) = O((M-k)^{L-K})$ và $\mathcal{C}(K) = |HUS(K-1)| \times O((M-L+2)^{L-K+1})$. Cụ thể, độ phức tạp thời gian trong trường hợp xấu nhất của thủ tục Find-FCloGenHUI là $\mathcal{C}(1) = O((M-L+2)^L)$. Khi $L = M$, $\mathcal{C}(1) = 2^M - 1 = O(2^M)$, tương ứng với số lượng tất cả các tập con khác rỗng của tập $\mathcal{A}$ gồm M thuộc tính. Tuy nhiên, khi các điều kiện cắt tỉa sớm được kích hoạt (xem các dòng 1 và 12 trong Hình 4, các dòng 1–5 trong Hình 5, và các dòng 1–6 trong Hình 6 trong [17]), toàn bộ các mở rộng tiến thật sự của A có thể được loại bỏ. Ngoài ra, khi tích hợp độ đo tương quan bond vào thuật toán, nhờ tính chất đơn điệu giảm của bond như đã trình bày trong Mệnh đề 1, nếu điều kiện $bond(A) < mb$ được thỏa mãn, thì tập mục A và tất cả các mở rộng tiến của nó đều có thể bị cắt tỉa. Khi đó, số lượng tối đa các tập mục bị loại bỏ là $C(K,k) = O((M-k)^{L-K})$.

Bảng 6. Đặc trưng của các cơ sở dữ liệu.

Cơ sở dữ liệu	Số giao tác (D)	Số thuộc tính (N)	Chiều dài trung bình của giao tác (T)	Mật độ (T*100/N %)
Chess	3,196	75	37	49.33
Foodmart	4,141	1,559	4.42	0.28
Mushroom	8,124	119	23	19.33
Retail	88,162	16,470	10,30	0.06
C20D10K1	10,000	192	20	10.42

Phân tích trên cho thấy rằng, trong thủ tục Find-FCloGenHUI (và thuật toán CoFCGHUI-Miner), các điều kiện cắt tỉa được thỏa mãn càng sớm và càng thường xuyên thì số lượng các tập ứng viên bị loại bỏ càng lớn. Nói cách khác, các điều kiện cắt tỉa giúp thu hẹp đáng kể không gian tìm kiếm, từ đó giảm độ phức tạp thời gian thực thi và nâng cao hiệu quả của thuật toán CoFCGHUI-Miner. Trên cơ sở đó, hiệu quả của toàn bộ quá trình sinh các SCoHUAR (bao gồm cả thuật toán M-SCoHUAR) cũng được cải thiện đáng kể.

5. Đánh giá thực nghiệm

Mục này đánh giá hiệu quả của phương pháp đề xuất so với các thuật toán trước đây để khai thác các luật kết hợp súc tích có lợi ích cao. Các thực nghiệm được thực hiện trên máy tính chạy hệ điều hành Windows 10, với bộ vi xử lý Intel(R) Core(TM) i5-2400 và bộ nhớ RAM 16 GB. Các thuật toán được xem xét để so sánh gồm HGB [9] và LNR-HAR [12], và được cài đặt bằng ngôn ngữ Java 8 SE. Mã nguồn của thuật toán HGB được lấy từ thư viện SPMF [37], trong khi chúng tôi cài đặt thuật toán LNR-HAR dựa trên phương pháp được mô tả trong [12]. Các bộ dữ liệu thực được lấy từ [37] và các bộ dữ liệu tổng hợp được sinh bằng công cụ tạo dữ liệu IBM Quest (lấy từ [37]), cho phép kiểm soát nhiều tham số như độ dài giao tác trung bình (T), kích thước trung bình của tập mục phổ biến tối đại (I), số lượng mục khác nhau (N), và số lượng giao tác (D) trong QDB. Đặc trưng của các QDB được trình bày trong Bảng 6.

M-SCoHUAR($\mathcal{CoCHUI}$, $\mathcal{CoGHUI}$, mu, mb, $muconf$)
Đầu vào: $\mathcal{CoCHUI}$, $\mathcal{CoGHUI}$ được khám phá bởi thuật toán CoFCGHUI-Miner, các ngưỡng mu, mb, $muconf$.
Đầu ra: tập $\mathcal{SCoHUAR}$ gồm tất cả các SCoHUAR.
//Xây dựng bảng băm các tập đóng theo TWU
1. $\mathcal{SCoHUAR} := \emptyset$;
2. Khởi tạo bảng băm rỗng HT_{closed}; // khóa: TWU(L)
3. **for each** tập đóng L ∈ $\mathcal{CoCHUI}$ **do** {
4. **Tạo** bản ghi $rec(L) = \langle L, supp(L), u(L), TWU(L), \{luv(x,L)\}_{\{x \in L\}}, \mathcal{G}(L) := \emptyset \rangle$;
5. **Chèn** $rec(L)$ vào $HT_{closed}[TWU(L)]$;
6. }
// Gán mỗi CoGHUI A với tập đóng $L = h(A)$ thông //qua TWU hashing
7. **for each** tập sinh $A \in \mathcal{CoGHUI}$ **do** {
8. $t := TWU(A)$;
9. **for each** $rec(L)$ trong $HT_{closed}[t]$ **do** {
10. **if** $h(A) = L$ **then** {
11. Thêm A vào $\mathcal{G}(L)$;
12. **break**;
13. }
14. }
15. }
16. Xác định tập $NCHUI = \{(L,S) \mid \emptyset \neq L \subseteq S \wedge L, S \in \mathcal{CoCHUI}\}$ dựa vào HT_{closed};
17. **for each** (L, S) ∈ $NCHUI$ **do** {
18. **for each** $L_i \in \mathcal{G}(L)$ **do** { // L_i là CoGHUI của L
19. $B := S \setminus L_i$; // vế phải duy nhất theo phân hoạch
// Tính nhanh $cuconf(R)$ nhờ $luv(b,S)$
20. $luvB := 0$;
21. **for each** $b \in B$ **do** $luvB := luvB + luv(b,S)$;
22. $cuconf := luvB \,/\, u(L_i)$;
23. **if** $cuconf \geq muconf$ **then** {
24. $R := (L_i \rightarrow B)$;
25. **Thêm** R vào $\mathcal{SCoHUAR}$;
26. }
27. }
28. }
29. **return** $\mathcal{SCoHUAR}$;

Hình 2. Thuật toán M-SCoHUAR.

5.1. So sánh kích thước của hai tập CoCHUI và CoGHUI với hai tập CHUI và GHUI

Để đánh giá tác động của việc áp dụng độ đo tương quan bond đến số lượng các tập mục đóng và sinh có lợi ích cao, trong thực nghiệm này, chúng tôi tiến hành so sánh kích thước của hai tập *CHUI* và *GHUI* với hai tập *CoCHUI* và *CoGHUI* dưới các ngưỡng lợi ích tối thiểu khác nhau.

Nhắc lại rằng, việc khai thác hai tập *CHUI* và *GHUI* được thực hiện bằng thuật toán *FCGHUI*-Miner trong [17], trong khi hai tập *CoCHUI* và *CoGHUI* được khai thác bằng thuật toán *CoFCGHUI*-Miner, là phiên bản được điều chỉnh từ *FCGHUI*-Miner thông qua việc tích hợp độ đo bond. Kết quả thực nghiệm được minh họa trong Hình 3.

Dựa trên kết quả thực nghiệm trong hình 3, ta có thể nhận thấy rằng khi ngưỡng lợi ích tối thiểu giảm, kích thước của các tập *CHUI*, *CHUI*, *CoCHUI* và *CoGHUI* đều tăng một cách nhất quán, phản ánh đặc điểm phổ biến của các thuật toán khai thác mẫu lợi ích cao.

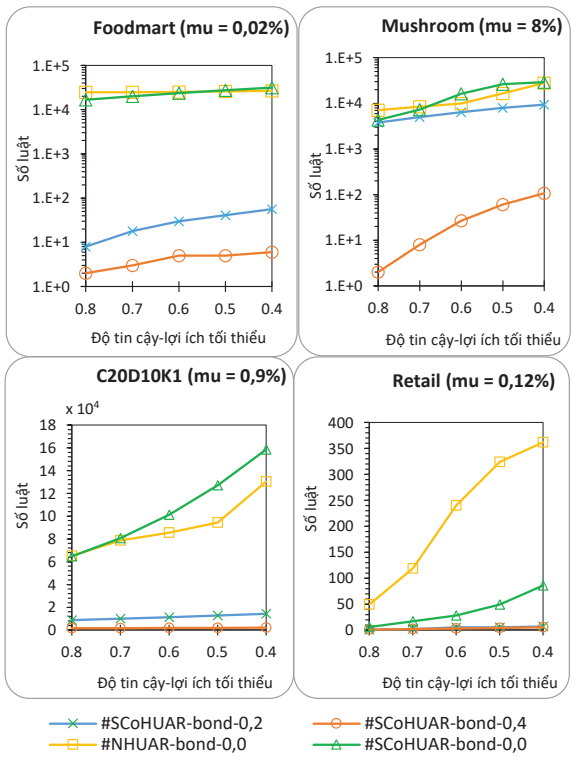
Tuy nhiên, tại mọi mức ngưỡng lợi ích, kích thước của hai tập *CoCHUI* và *CoGHUI* luôn nhỏ hơn đáng kể so với các tập *CHUI* và *GHUI* tương ứng, qua đó cho thấy tác động rõ rệt của việc tích hợp độ đo tương quan bond vào quá trình khai thác. Cụ thể, độ đo bond đã loại bỏ hiệu quả các tập mục mặc dù có lợi ích cao nhưng có mức độ tương quan yếu, dẫn đến sự giảm đáng kể số lượng mẫu được sinh ra. Những kết quả này khẳng định rằng độ đo bond không chỉ đóng vai trò như một ràng buộc tương quan bổ sung, mà còn hoạt động như một cơ chế lọc mạnh, giúp thu gọn không gian kết quả, nâng cao tính cô đọng và cải thiện khả năng diễn giải của các mẫu lợi ích cao được khai thác.

5.2. So sánh số luật được khai thác khi sử dụng uconf và cuconf

Trong thực nghiệm này, bài báo so sánh số lượng luật được sinh ra trên các QDB khi sử dụng các hàm tin cậy - lợi ích $uconf$ và $cuconf$ kết hợp với các giá trị ngưỡng bond khác nhau. Cụ thể, #NHUAR-bond-0,0 biểu thị số luật được khai thác khi sử dụng hàm tin cậy - lợi ích $uconf$ được trình bày trong Định nghĩa 16 và đề xuất trong [9], với giá trị mb = 0. Trong khi đó, #SCoHUAR-bond-0,0, #SCoHUAR-bond-0,2 và #SCoHUAR-bond-0,4 lần lượt là số luật được sinh ra khi sử dụng hàm tin cậy - lợi ích mới $cuconf$ được trình bày trong Định nghĩa 17, tương ứng với các giá trị ngưỡng mb = 0, mb = 0,2 và mb = 0,4.

Dựa trên các kết quả thực nghiệm được trình bày trong Hình 4, ta có thể quan sát thấy rằng khi mb = 0, số lượng luật #SCoHUAR-bond-0.0 thu được khi sử dụng hàm tin cậy - lợi ích $cuconf$ khác biệt rõ ràng so với #NHUAR-bond-0.0 sử dụng $uconf$. Cụ thể, tùy thuộc vào đặc trưng của từng cơ sở dữ liệu, $cuconf$ có thể sinh ra nhiều luật hơn hoặc ít luật hơn so với $uconf$. Hiện tượng này phản ánh sự khác biệt về bản chất trong cách hai độ đo đánh giá mức độ tin cậy - lợi ích của luật, trong đó $cuconf$ nhấn mạnh hơn vào đóng góp lợi ích tương đối của vế phải, từ đó có thể làm nổi bật các luật mà $uconf$

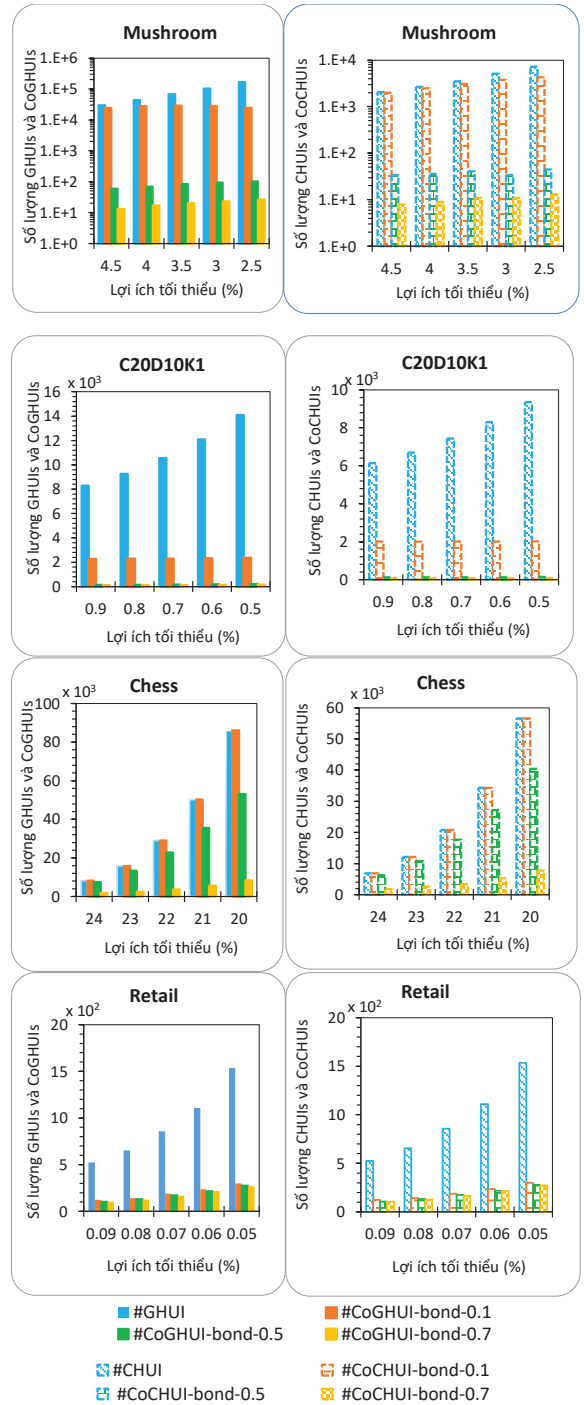
chưa phản ánh đầy đủ. Khi ngưỡng bond tăng lên *mb* = 0,2 và *mb* = 0,4, số lượng luật tương ứng (#SCoHUAR-bond-0,2 và #SCoHUAR-bond-0,4) giảm mạnh so với cả #SCoHUAR-bond-0,0 và #NHUAR-bond-0,0. Xu hướng này cho thấy độ đo bond đóng vai trò như một ràng buộc tương quan hiệu quả, giúp loại bỏ các luật tuy có lợi ích cao nhưng mức độ liên kết giữa các thuộc tính trong luật còn yếu. Đặc biệt, với *mb* = 0.4, tập luật thu được là gọn nhất, chỉ giữ lại các luật vừa thỏa mãn tiêu chí lợi ích và tin cậy cao, vừa có tương quan chặt chẽ, qua đó nâng cao đáng kể tính cô đọng và khả năng diễn giải của tập luật được khai thác.



Hình 4. So sánh số luật được sinh ra khi sử dụng các hàm tin cậy - lợi ích *uconf* và *cuconf*.

5.3. Đánh giá hiệu quả của thuật toán đề xuất

Trong mục này, bài báo tiến hành so sánh hiệu quả của thuật toán đề xuất M-SCoHUAR với các thuật toán trước đây dựa trên hai tiêu chí chính là thời gian chạy và mức sử dụng bộ nhớ cực đại. Theo hiểu biết của chúng tôi, cho đến nay hai thuật toán tiêu biểu được sử dụng để khai thác các luật kết hợp lợi ích cao không dư thừa là HGB [9] và LNR-HAR [12], đều dựa trên hàm độ tin cậy - lợi ích uconf được định nghĩa trong định nghĩa 16. Tuy nhiên, như đã chỉ ra trong các thực nghiệm của [12], thuật toán LNR-HAR cho thấy hiệu quả vượt trội hơn HGB cả về thời gian thực thi lẫn mức tiêu thụ bộ nhớ. Do đó, trong bài báo này, chúng tôi chỉ tiến hành so sánh thuật toán đề



Hình 3. So sánh kích thước của *CHUI* và *GHUI* với *CoCHUI* và *CoGHUI*.

xuất M-SCoHUAR với LNR-HAR. Để bảo đảm tính công bằng, hai thuật toán được so sánh trong cùng điều kiện với ngưỡng *mb*=0. Sau đó, nhằm đánh giá tác động của tham số mb đến hiệu quả của thuật toán đề xuất, M-SCoHUAR tiếp tục được thực thi với ngưỡng *mb*=0,2 trên các cơ sở dữ liệu đã sử dụng trong thực nghiệm.

Dựa trên các kết quả thực nghiệm được trình bày trong hình 5 và hình 6, ta có thể đưa ra các nhận xét và thảo luận chi tiết hơn về hiệu quả của thuật toán đề xuất M-SCoHUAR so với LNR-HAR, cũng như tác động của tham số bond đến thời gian chạy và mức sử dụng bộ nhớ.

Trước hết, về thời gian chạy, các đồ thị cho thấy M-SCoHUAR-bond-0,0 luôn đạt thời gian thực thi thấp hơn so với LNR-HAR-bond-0,0 trên tất cả các QDB được khảo sát. Sự khác biệt này xuất phát chủ yếu từ cách tiếp cận của các thuật toán. Cụ thể, LNR-HAR được xây dựng dựa trên phương pháp xây dựng và duyệt dàn (lattice) các mẫu lợi ích cao, trong đó các mối quan hệ bao hàm giữa các tập mục phải được kiểm tra thường xuyên khi xây dựng dàn. Cách tiếp cận dựa trên dàn này làm phát sinh chi phí tính toán lớn, đặc biệt khi số lượng các mẫu lợi ích cao tăng nhanh, dẫn đến thời gian thực thi kéo dài. Ngược lại, M-SCoHUAR áp dụng chiến lược phân hoạch để chia không gian kết quả thành các lớp rời nhau, nhờ đó tránh được việc sinh trùng lặp cũng như giảm đáng kể chi phí kiểm tra trùng lặp giữa các tập CoCHUI và CoGHUI. Bên cạnh đó, thuật toán còn khai thác các cấu trúc dữ liệu hiệu quả và các kỹ thuật tối ưu đã nêu trong Nhận xét 1, trong đó các mẫu được lưu trữ trong bảng băm với khóa băm là giá trị TWU, giúp hạn chế mạnh việc kiểm tra quan hệ bao hàm giữa các mẫu. Việc sử dụng cấu trúc bit vector để tính toán nhanh giá trị *bond* cũng góp phần làm giảm đáng kể chi phí xử lý.

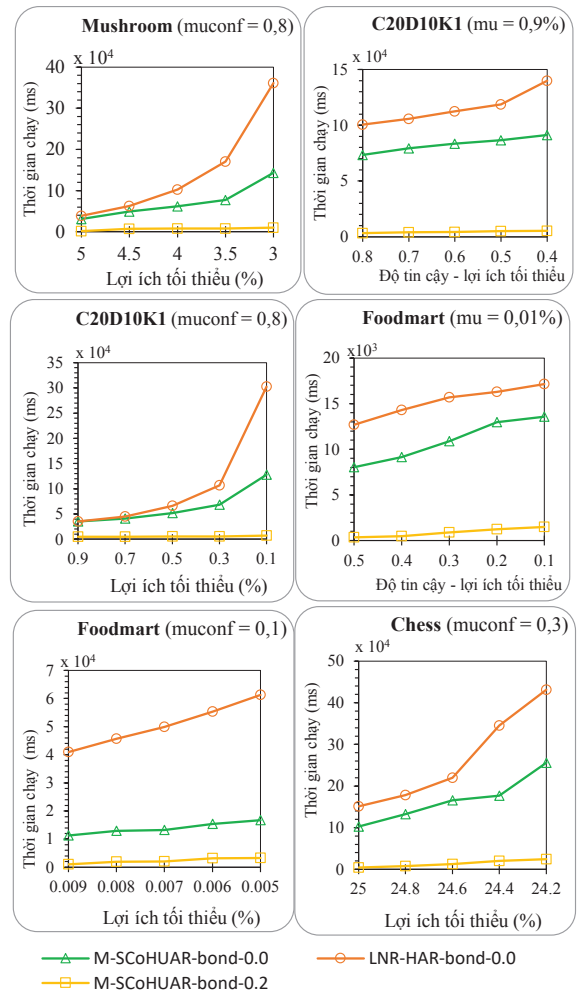
Tiếp theo, khi so sánh M-SCoHUAR-bond-0.2 với M-SCoHUAR-bond-0,0, ta có thể thấy rằng, thời gian chạy của thuật toán với *mb*=0,2 luôn nhỏ hơn so với trường hợp *mb*=0,0. Điều này cho thấy tác động tích cực của ngưỡng bond đến hiệu quả của thuật toán: khi giá trị *bond* tăng, các ràng buộc tương quan trở nên chặt chẽ hơn, cho phép loại bỏ sớm nhiều mẫu và luật có mức độ liên kết yếu. Hệ quả là không gian tìm kiếm được thu hẹp đáng kể, số lượng mẫu trung gian cần xử lý giảm, dẫn đến thời gian thực thi ngắn hơn. Ngược lại, với *mb*=0,0, việc không áp đặt ràng buộc tương quan khiến thuật toán phải xử lý nhiều mẫu hơn, làm gia tăng chi phí tính toán.

Cuối cùng, về mức sử dụng bộ nhớ, các kết quả trong Hình 6 cho thấy LNR-HAR thường tiêu thụ bộ nhớ lớn hơn so với M-SCoHUAR. Nguyên nhân chính là do phương pháp dựa trên dàn của LNR-HAR đòi hỏi phải lưu trữ một số lượng lớn các mẫu lợi ích cao. Trong khi đó, M-SCoHUAR nhờ cơ chế phân hoạch, tổ chức mẫu bằng bảng băm theo TWU và các cấu trúc dữ liệu gọn nhẹ, đã kiểm soát tốt bộ nhớ sử dụng ngay cả khi tích hợp thêm độ đo *bond*.

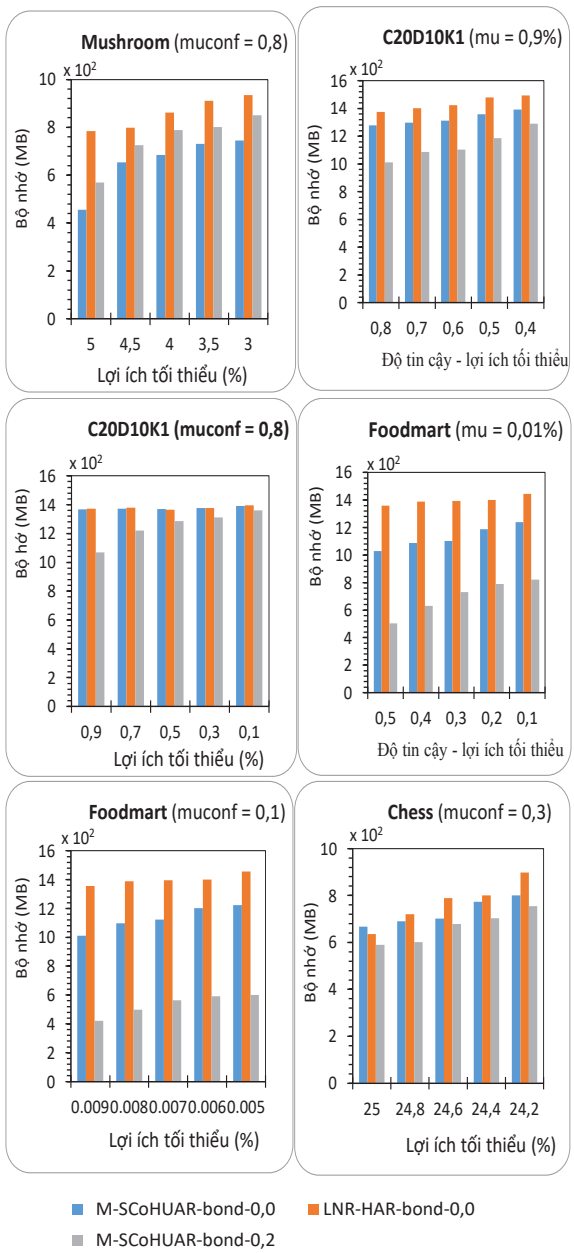
Tóm lại, các kết quả thực nghiệm cho thấy M-SCoHUAR không chỉ vượt trội hơn LNR-HAR về thời gian chạy và mức sử dụng bộ nhớ, mà còn có ưu điểm rõ rệt từ việc tích hợp tham số bond, qua đó nâng cao khả năng mở rộng, hiệu quả tính toán và tính thực tiễn trong khai thác các luật kết hợp súc tích có lợi ích và tương quan cao.

6. Kết luận và công việc tương lai

Bài báo này nghiên cứu bài toán khai thác các luật kết hợp súc tích, có lợi ích cao và tương quan mạnh, nhằm khắc phục tình trạng dư thừa cũng như hạn chế về khả



Hình 5. So sánh thời gian chạy của các thuật toán.



Hình 6. So sánh mức sử dụng bộ nhớ của các thuật toán.

năng diễn giải của các luật lợi ích cao truyền thống, đồng thời cho phép đánh giá toàn diện hơn cả khía cạnh lợi ích và mức độ liên kết giữa các thành phần trong luật. Trên cơ sở đó, thuật toán M-SCoHUAR được đề xuất để khai thác hiệu quả tất cả các SCoHUAR sử dụng hàm tin cậy - lợi ích mới kết hợp với độ đo tương quan bond. Ngoài ra, thuật toán M-SCoHUAR áp dụng phương pháp phân hoạch không gian của tập các SCoHUAR và cấu trúc dữ liệu hiệu quả dựa trên bảng băm theo TWU, cũng như các kỹ thuật tối ưu nhằm giảm chi phí kiểm tra trùng lặp và quan hệ bao hàm giữa các tập mục. Kết quả thực nghiệm trên các QDB cho thấy việc tích hợp độ đo bond có tác động rõ rệt trong việc kiểm soát kích thước tập mẫu và tập luật, qua đó loại bỏ hiệu quả các luật có mức độ tương quan yếu. So với thuật toán tiêu biểu trước đây LNR-HAR, thuật toán đề xuất M-SCoHUAR đạt hiệu quả vượt trội về thời gian chạy và mức sử dụng bộ nhớ. Những kết quả này khẳng định tính hiệu quả và khả năng mở rộng của phương pháp đề xuất trong khai thác các SCoHUAR.

Trong tương lai, hướng nghiên cứu có thể được mở rộng sang các độ đo tương quan khác, các dạng cơ sở dữ liệu phức tạp hơn như dữ liệu chuỗi hoặc dữ liệu luồng, cũng như tích hợp các cơ chế giải thích (XAI) nhằm nâng cao hơn nữa khả năng ứng dụng thực tiễn của các luật được khai thác.

LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi Trường Đại học Đà Lạt, Việt Nam theo Quyết định số 198/QĐ-ĐHĐL ngày 06/03/2025.

TÀI LIỆU THAM KHẢO

[1] R. Agrawal, R. Srikant (1994), "Fast algorithms for mining association rules in large databases", *Proc. of the 20th International Conference on Very Large Data Bases*, pp.487-499.

[2] M. Liu, J. Qu (2012), "Mining high utility itemsets without candidate generation", *Proceedings of ACM International Conference on Information and Knowledge Management*, pp.55-64.

[3] B.E. Shie, P.S. Yu, V.S. Tseng (2012), "Mining interesting user behavior patterns in mobile commerce environments", *Applied Intelligence*, **38**, pp.418-435.

[4] B.E. Shie, C.W. Wu, V.S. Tseng, et al. (2013), "Efficient algorithms for mining high utility itemsets from transactional databases", *EEE Transactions on Knowledge and Data Engineering*, **25(8)**, pp.1772-1786, DOI: 10.1109/TKDE.2012.59.

[5] Y. Liu, L. Wang, L. Feng (2020), "Mining high utility itemsets based on pattern growth without candidate generation", *Mathmetics*, pp.1-22, DOI:10.3390/math9010035.

[6] M. Liu, J. Qu (2012), "Mining high utility itemsets without candidate generation", *Proceedings of The 21st ACM International Conference on Information and Knowledge Management*, pp.55-64, DOI: 10.1145/2396761.2396773

[7] P.F. Viger, C.W. Wu, S. Zida, et al. (2014), "FHM: Faster high-utility itemset mining using estimated utility cooccurrence pruning", *Lecture Notes in Computer Science*, **8502**, pp.83-92.

[8] S. Zida, P.F. Viger, J.C.W. Lin, et al. (2015), "A highly efficient algorithm for high-utility itemset mining", *Proceedings of Mexican International Conference on Artificial Intelligence*, pp.530-546.

[9] J. Sahoo, A. Kumar, A. Goswami (2015), "An efficient approach for mining association rules from high utility itemsets", *Expert Syst. Appl.* **42(13)**, pp.5754-5778, DOI: 10.1016/j.eswa.2015.02.051.

[10] T. Mai, B. Vo, L.T.T. Nguyen (2017), "A lattice-based approach for mining high utility association rules", *Inf. Sci.*, **399**, pp.81-97, DOI: 10.1016/j.ins.2017.02.058.

[11] L.T.T. Nguyen, T. Mai, B. Vo (2019), *High Utility Association Rule Mining, High-Utility Pattern Mining*, Springer, pp.161-174.

[12] T. Mai, L.T.T. Nguyen, B. Vo, et al. (2020), "Efficient algorithm for mining non-redundant high-utility association rules", *Sensors* (Switzerland), **20(4)**, pp.1-17, DOI: 10.3390/s20041078..

[13] P.F. Viger, J.C.W. Lin, T. Dinh, et al. (2016), *Mining Correlated High-Utility Itemsets Using The Bond Measure*, Springer, pp.53-65.

[14] C.C. Aggarwal, P.S. Yu (1998), "New framework for itemset generation, Proc", ACM Symp. Principles of Database Systems, pp.18-24, DOI: 10.1145/275487.275490.

[15] E.R. Omiecinski (2023), "Alternative interest measures for mining associations in databases", *IEEE Trans. Knowl. Data Eng.*, pp.57-69.

[16] T. Wu, Y. Chen, J. Han (2010), "Re-examination of interestingness measures in pattern mining: A unified framework", *Data Min. Knowl. Discov.*, pp.371-397.

[17] T. Tran, H. Duong, T. Truong, et al. (2023), "Efficient mining of concise and informative representations of frequent high utility itemsets", *Eng. Appl. Artif. Intell*, **126**, DOI: 10.1016/j.engappai.2023.107111.

[18] L.T.T. Nguyen, P. Nguyen, T.D.D. Nguyen, et al. (2019), "Mining high-utility itemsets in dynamic profit databases", *Knowl. Based. Syst.*, **175**, pp.130-144.

[19] T. Truong, H. Duong, B. Le, et al. (2019), "Efficient vertical mining of high average-utility itemsets based on novel upper-bounds", *IEEE Trans. Knowl. Data Eng.*, **31**, pp.301-314.

[20] J.F. Qu, P.F. Viger, M. Liu, et al. (2023), "Mining high utility itemsets using prefix trees and utility vectors", *IEEE Trans. Knowl. Data Eng.*, **35**, pp.10224-10236.

[21] Z. Cheng, W. Fang, W. Shen, et al. (2023), "An efficient utility-list based high-utility itemset mining algorithm", *Applied Intelligence*, **53**, pp.6992-7006.

[22] R.U. Kiran, P. Veena, P. Ravikumar, et al. (2023), "HDSHUI-miner: a novel algorithm for discovering spatial high-utility itemsets in high-dimensional spatiotemporal databases", *Applied Intelligence*, **53**, pp.8536-8561.

[23] J.M. Luna, R.U. Kiran, P.F. Viger (2023), "Efficient mining of top-k high utility itemsets through genetic algorithms", *Inf. Sci.*, **624**, pp.529-553.

[24] G.C. Lan, T.P. Hong, V.S. Tseng (2014), "An efficient projection-based indexing approach for mining high utility itemsets", *Knowl. Inf. Syst.*, **38**, pp.85-107.

[25] S. Dawar, V. Goyal, D. Bera (2017), "A hybrid framework for mining high-utility itemsets in a sparse transaction database", *Applied Intelligence*, **47**, pp.809-827.

[26] D.Q. Huy, P.F. Viger, H. Ramampiaro, et al. (2018), "Efficient high utility itemset mining using buffered utility-lists", *Applied Intelligence*, **48**, pp.1859-1877.

[27] Z. Deng (2018), "An efficient structure for fast mining high utility itemsets", *Applied Intelligence*, **48**, pp.3161-3177.

[28] T. Truong, H. Duong, B. Le, et al. (2019), "Efficient high average-utility itemset mining using novel vertical weak upper-bounds", *Knowl. Based. Syst.*, **183**, DOI: 1016/j.knosys.2019.07.018.

[29] H. Duong, T. Hoang, T. Tran, et al. (2022), "Efficient algorithms for mining closed and maximal high utility itemsets", *Knowl. Based. Syst.*, **257**, DOI: 10.1016/j.knosys.2022.109921.

[30] V.S. Tseng, C. Wu, P.F. Viger (2015), "Efficient algorithms for mining the concise and lossless representation of high utility itemsets", *IEEE Trans. Knowl. Data Eng.*, **27**, pp.726-739.

[31] C.W. Wu, P.F. Viger, J.Y. Gu, et al. (2015), "Mining closed + high utility itemsets without candidate generation", *Conference on Technologies and Applications of Artificial Intelligence*, pp.187-194.

[32] P.F. Viger, S. Zida, J.C.W. Lin, et al. (2016), "EFIM-Closed: Fast and memory efficient discovery of closed high-utility itemsets", *International Conference on Machine Learning and Data Mining in Pattern Recognition*, pp.199-213.

[33] L.T.T. Nguyen, V.V. Vu, M.T.H. Lam, et al. (2019), "An efficient method for mining high utility closed itemsets", *Inf. Sci.*, **495**, pp.78-99.

[34] P.F. Viger, C.W. Wu, V.S. Tseng (2014), *Novel Concise Representations of High Utility Itemsets Using Generator Patterns*, International Conference on Advanced Data Mining and Applications, pp.30-43.

[35] J. Sahoo, A.K. Das, A. Goswami (2014), "An algorithm for mining high utility closed itemsets and generators", *Proceedings of The 2014 IEEE International Conference on Data Science and Advanced Analytics*, pp.1-9.

[36] S. Bouasker, S. Yahia (2015), "Key correlation mining by simultaneous monotone and anti-monotone constraints checking", *Proceedings of The ACM Symposium on Applied Computing*, pp.851-856.

[37] P.F. Viger, A. Gomariz, A. Soltani, et al. (2014), "SPMF: A java open-source pattern mining library", *J. Mach. Learn. Res.*, **15**, pp.3569-3573.