TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

TÔ NGỌC HUYỀN - 242805005

CÁC THUẬT TOÁN KHAI THÁC TOP-K TẬP MỤC THƯỜNG XUYÊN TỪ CƠ SỞ DỮ LIỆU KHÔNG CHẮC CHẮN

CHUYÊN ĐỀ NGHIÊN CỨU 3

KHOA HỌC MÁY TÍNH

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

TÔ NGỌC HUYỀN - 242805005

CÁC THUẬT TOÁN KHAI THÁC TOP-K TẬP MỤC THƯỜNG XUYÊN TỪ CƠ SỞ DỮ LIỆU KHÔNG CHẮC CHẮN

CHUYÊN ĐỀ NGHIÊN CỨU 3

KHOA HỌC MÁY TÍNH

Người hướng dẫn
TS. Nguyễn Chí Thiện

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến Trường Đại học Tôn Đức Thắng, nơi đã tạo ra một môi trường học tập hiện đại, năng động, giúp em có điều kiện thuận lợi để học tập và phát triển toàn diện trong suốt quá trình nghiên cứu.

Em cũng xin gửi lời cảm ơn sâu sắc đến Khoa Công nghệ Thông tin đã tổ chức và triển khai môn học Chuyên đề nghiên cứu 3, đồng thời luôn đồng hành, định hướng và cung cấp những kiến thức chuyên sâu cùng tài liệu tham khảo hữu ích, giúp chúng em tiếp cận một cách bài bản và hiệu quả.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Chí Thiện – giảng viên hướng dẫn chuyên đề – người đã tận tình chỉ dẫn, hỗ trợ và góp ý chuyên môn trong suốt quá trình em thực hiện đề tài. Sự hướng dẫn tận tâm và kinh nghiệm quý báu của thầy chính là nguồn động lực lớn giúp em hoàn thành báo cáo này một cách tốt nhất.

Do hạn chế về kiến thức và kinh nghiệm, báo cáo chắc chắn còn nhiều thiếu sót. Em rất mong nhận được những ý kiến đóng góp quý báu từ quý Thầy Cô để có thể hoàn thiện hơn trong các nghiên cứu tiếp theo.

TP. Hồ Chí Minh, ngày 08 tháng 12 năm 2025

Tác giả

(Ký tên và ghi rõ họ tên)

Tô Ngọc Huyền

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH
TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng tôi xin cam đoan đây là công trình nghiên cứu của riêng chúng tôi và được sự hướng dẫn khoa học của TS. Nguyễn Chí Thiện. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Chuyên đề nghiên cứu còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung Chuyên đề nghiên cứu của mình. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do chúng tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 08 tháng 12 năm 2025

Tác giả

(Ký tên và ghi rõ họ tên)

Tô Ngọc Huyền

CÁC THUẬT TOÁN KHAI THÁC TOP-K TẬP MỤC THƯỜNG XUYÊN TỪ CƠ SỞ DỮ LIỆU KHÔNG CHẮC CHẮN

TÓM TẮT

Khai thác Top-K tập mục thường xuyên từ cơ sở dữ liệu không chắc chắn là một hướng nghiên cứu quan trọng trong khai phá dữ liệu hiện đại, thông tin thu thập thường kèm theo mức độ sai lệch nhất định do giới hạn của thiết bị đo, nhiễu môi trường hoặc sự không hoàn hảo trong quá trình lấy mẫu, sự không đầy đủ của giao dịch hoặc nhiễu trong môi trường cảm biến. Trong mô hình dữ liệu này, mỗi item được gắn với xác suất xuất hiện, khiến việc xác định mức độ phổ biến của tập mục không thể áp dụng trực tiếp các thuật toán FIM truyền thống.

Nghiên cứu tập trung khảo sát ba thuật toán tiêu biểu trên dữ liệu không chắc chắn: UApriori, UFP-Growth và UH-Mine, đại diện cho ba hướng tiếp cận chính gồm sinh - kiểm tra, khai thác dựa trên cấu trúc FP-tree và khai thác theo mô hình liên kết dọc. Ngoài ra, nghiên cứu đề xuất mô hình kết hợp Hybrid-UFHM, tận dụng ưu điểm của FP-tree trong môi trường dữ liệu dày và của H-struct khi dữ liệu thưa. Các thuật toán được đánh giá dựa trên thời gian thực thi, mức sử dụng bộ nhớ, khả năng cắt tỉa thông qua threshold raising và upper-bound pruning, cùng tính thích ứng với các đặc trưng dữ liệu khác nhau. Kết quả thực nghiệm trên các bộ dữ liệu chuẩn cho thấy UFP-Growth đạt hiệu năng cao trên dữ liệu có mật độ lớn, UH-Mine phù hợp hơn với dữ liệu rời rạc, trong khi UApriori bị hạn chế bởi số lượng ứng viên sinh ra quá nhiều. Thuật toán Hybrid-UFHM thể hiện khả năng cân bằng hiệu năng trên cả hai nhóm dữ liệu và là hướng tiếp cận tiềm năng cho các ứng dụng khai thác Top-K trong môi trường dữ liệu không chắc chắn. Những phân tích và kết quả thu được góp phần làm rõ đặc tính của các thuật toán, hỗ trợ việc lựa chọn giải pháp phù hợp trong thực tiễn.

ALGORITHMS FOR MINING TOP-K FREQUENT ITEMSETS FROM UNCERTAIN DATABASES

ABSTRACT

Mining Top-K frequent itemsets from uncertain databases has emerged as an important research direction in modern data mining, where collected information often contains inherent inaccuracies caused by sensor limitations, measurement noise, incomplete transactions, and various imperfections during the data acquisition process. In this data model, each item is associated with a probability of occurrence, making traditional frequent itemset mining (FIM) techniques inapplicable without substantial modification.

This study examines three representative algorithms for mining from uncertain data—UApriori, UFP-Growth, and UH-Mine—which correspond to three primary methodological families: the generate-and-test approach, FP-tree–based mining, and vertical mining using linked structures. Furthermore, the study introduces a hybrid model, Hybrid-UFHM, which leverages the strengths of FP-tree structures for dense datasets and H-struct–based mining for sparse datasets. The algorithms are evaluated in terms of execution time, memory consumption, pruning effectiveness through threshold raising and upper-bound pruning, and adaptability to varying data characteristics.

Experimental results on several benchmark datasets indicate that UFP-Growth performs efficiently on dense data, UH-Mine is better suited for sparse data, while UApriori suffers from excessive candidate generation. The proposed Hybrid-UFHM demonstrates balanced performance across both dense and sparse scenarios, making it a promising approach for Top-K mining in uncertain environments. The analysis and results contribute to a deeper understanding of the behavior of these algorithms and provide practical guidance for selecting appropriate techniques in real-world applications.

MỤC LỤC

DANH MỤC HÌNH VẼ ..... **vii**

DANH MỤC BẢNG BIỂU ..... **vii**

DANH MỤC CÁC CHỮ VIẾT TẮT ..... **viii**

CHƯƠNG 1. GIỚI THIỆU **1**

CHƯƠNG 2. CÔNG TRÌNH LIÊN QUAN **3**

CHƯƠNG 3. ĐỊNH NGHĨA VÀ PHÁT BIỂU CỦA VẤN ĐỀ **5**

3.1 Định nghĩa 5

3.1.1 Mô hình dữ liệu không chắc chắn *5*

3.1.2 Xác suất xuất hiện của một itemset *6*

3.1.3 Hỗ trợ kỳ vọng (Expected Support – ES) *6*

3.1.4 Định nghĩa bài toán Top-K Frequent Itemsets *7*

CHƯƠNG 4. PHƯƠNG PHÁP **9**

4.1 Thuật toán UApriori cho Top-K Frequent Itemsets 9

4.2 Thuật toán UFP-Growth trong khai thác Top-K 9

4.3 Thuật toán UH-Mine trong khai thác Top-K 10

4.4 Thuật toán kết hợp: Hybrid-UFHM (UFP-Growth + UH-Mine) 11

CHƯƠNG 5. THIẾT LẬP THỰC NGHIỆM **13**

5.1 Mục tiêu thực nghiệm 13

5.2 Môi trường thực nghiệm 13

5.3 Dữ liệu thực nghiệm 13

CHƯƠNG 6. KẾT QUẢ THỰC NGHIỆM VÀ THẢO LUẬN **15**

6.1 Hiệu suất về thời gian chạy 15

6.2 Hiệu suất về mức tiêu thụ bộ nhớ 18

CHƯƠNG 7. KẾT LUẬN.....22

7.1 Kết luận 22

7.2 Hướng phát triển 22

TÀI LIỆU THAM KHẢO 24

DANH MỤC HÌNH VẼ

Hình 6.1 Hiệu suất về thời gian chạy của các thuật toán trên bộ Chess 16

Hình 6.2 Hiệu suất về thời gian chạy của các thuật toán trên bộ Foodmart 17

Hình 6.3 Hiệu suất về thời gian chạy của các thuật toán trên bộ Retail 17

Hình 6.4 Hiệu suất về thời gian chạy của các thuật toán trên bộ t20i6d100k.....18

Hình 6.5 Hiệu suất về mức tiêu thụ bộ nhớ của các thuật toán trên bộ Chess.....19

Hình 6.6 Hiệu suất về mức tiêu thụ bộ nhớ của các thuật toán trên bộ Foodmart....20

Hình 6.7 Hiệu suất về mức tiêu thụ bộ nhớ của các thuật toán trên bộ Retail.....20

Hình 6.8 Hiệu suất về mức tiêu thụ bộ nhớ của các thuật toán trên bộ t20i6d100k .21

DANH MỤC BẢNG BIỂU

Bảng 3.1 Ví dụ về cơ sở dữ liệu giao dịch không chắc chắn 5

Bảng 3.2 Minh họa tính Expected Support của {A} và {A,C,D}.....6

Bảng 3.3 Top-5 item phổ biến nhất theo Expected Support 8

Bảng 5.1 Một số thông số về các bộ dữ liệu dùng để thử nghiệm.....14

DANH MỤC CÁC CHỮ VIẾT TẮT

FIM	Frequent Itemset Mining
UDB	Uncertain Database
ES	Expected Support
UApriori	Uncertain Apriori Algorithm
UFP-Growth	Uncertain FP-Growth Algorithm
UH-Mine	Uncertain H-Mine Algorithm
Hybrid-UFHM	Hybrid Uncertain FP-Growth & H-Mine
FP-tree	Frequent Pattern Tree
IoT	Internet of Things
RFID	Radio Frequency Identification
TID	Transaction Identifier

CHƯƠNG 1. GIỚI THIỆU

Trong những năm gần đây, khai thác dữ liệu trong môi trường không chắc chắn (Uncertain Data Mining) đã trở thành một hướng nghiên cứu quan trọng, đặc biệt khi dữ liệu được thu thập từ các hệ thống cảm biến, thiết bị IoT, RFID, dữ liệu sinh học và các nền tảng mạng xã hội ngày càng gia tăng về quy mô và mức độ đa dạng (Aggarwal & Yu, 2009) [1]. Các nguồn dữ liệu này thường chứa mức độ không chính xác nhất định do nhiễu đo lường, sai số mô hình, lỗi phần cứng hoặc tính ngẫu nhiên trong quá trình thu thập. Khác với dữ liệu truyền thống ở dạng nhị phân chắc chắn, trong dữ liệu không chắc chắn, mỗi mục (item) được gắn với một giá trị xác suất xuất hiện trong mỗi giao dịch, làm thay đổi bản chất của việc tính toán hỗ trợ và đòi hỏi các thuật toán mới có khả năng mô hình hóa xác suất.

Trong bối cảnh đó, khai thác tập mục thường xuyên (Frequent Itemset Mining – FIM) [2-4] trên dữ liệu không chắc chắn giữ vai trò nền tảng trong nhiều ứng dụng như phát hiện bất thường, phân tích thị trường, tối ưu hóa quy trình vận hành, dự đoán hành vi và phân tích sở thích người dùng. Tuy nhiên, các thuật toán FIM truyền thống như Apriori hay FP-Growth được thiết kế cho dữ liệu chắc chắn và phụ thuộc mạnh vào ngưỡng hỗ trợ tối thiểu (minSUP). Việc áp dụng chúng trực tiếp cho dữ liệu không chắc chắn dẫn đến nhiều khó khăn, bởi xác suất xuất hiện của các mục khiến phân bố hỗ trợ thay đổi, và người dùng khó xác định được một ngưỡng minSUP phù hợp. Một giá trị minSUP quá cao có thể loại bỏ nhiều mẫu giá trị, trong khi minSUP quá thấp lại khiến không gian ứng viên bùng nổ và làm tăng đáng kể chi phí tính toán.

Để khắc phục những hạn chế của phương pháp dựa trên ngưỡng hỗ trợ tối thiểu, nhiều nghiên cứu gần đây tập trung vào hướng tiếp cận Top-K Frequent Itemsets, trong đó thuật toán trực tiếp truy xuất K tập mục có hỗ trợ kỳ vọng (Expected Support – ES) cao nhất, thay vì yêu cầu người dùng đặt trước giá trị minSUP. Hướng tiếp cận này đã được chứng minh hiệu quả trong các công trình của Chui & Kao (2007) [5], nơi mô hình hỗ trợ kỳ vọng và kỹ thuật tỉa dựa trên bound được giới thiệu lần đầu cho dữ liệu không chắc chắn, Bernecker et al. (2010, 2012) [6-7] với phương pháp mô hình hoá phân phối xác suất để giảm chi phí tính toán ES

và Aggarwal & Yu (2009) [8] là những người phát triển khung thống kê cho việc khai thác tập mục trong cơ sở dữ liệu xác suất quy mô lớn. Tuy nhiên, việc khai thác Top-K trên dữ liệu không chắc chắn vẫn đối mặt nhiều thách thức do tính chất bùng nổ theo hàm mũ của không gian tập mục, sự phức tạp trong việc tính toán Expected Support, yêu cầu về các cấu trúc dữ liệu tối ưu, cũng như nhu cầu thiết kế các cơ chế loại bỏ ứng viên dựa trên ngưỡng động và các dạng giới hạn mạnh.

Xuất phát từ những vấn đề nêu trên, nghiên cứu này hướng tới việc hệ thống hóa các mô hình toán học, khái niệm nền tảng và thuật toán tiêu biểu trong khai thác Top-K tập mục thường xuyên trên dữ liệu không chắc chắn, bao gồm các thuật toán UApriori, UFP-Growth, UH-Mine và các phương pháp Top-K dựa trên Expected Support. Đồng thời, nghiên cứu đề xuất một thuật toán kết hợp mới Hybrid Top-K Expected Support Mining – tận dụng ưu điểm của cấu trúc FP-tree khi xử lý cơ sở dữ liệu dày (dense) và cấu trúc H-Mine khi xử lý cơ sở dữ liệu thưa (sparse). Thuật toán được triển khai thực nghiệm nhằm đánh giá hiệu năng, so sánh với các phương pháp truyền thống và phân tích kết quả thu được, từ đó cung cấp một cái nhìn toàn diện và sâu sắc về bài toán khai thác Top-K tập mục trong môi trường dữ liệu không chắc chắn.

CHƯƠNG 2. CÔNG TRÌNH LIÊN QUAN

Khai thác tập mục thường xuyên (Frequent Itemset Mining – FIM) được xem là một hướng nghiên cứu cốt lõi trong khai phá dữ liệu, với thuật toán kinh điển đầu tiên là Apriori (Agrawal & Srikant, 1994) [9], dựa trên tính chất suy giảm, nếu một tập mục không thường xuyên thì mọi tập mục mở rộng của nó cũng không thể thường xuyên. Tính chất này cho phép Apriori cắt giảm đáng kể số lượng ứng viên, song chi phí quét cơ sở dữ liệu nhiều lần làm giảm khả năng mở rộng đối với tập dữ liệu lớn. Để khắc phục nhược điểm này, Han et al. (2000) [3] đề xuất FP-Growth – đánh dấu một bước tiến đáng kể bằng cấu trúc FP-Tree nhằm nén dữ liệu và khai thác theo chiến lược chia để trị thông qua các cơ sở dữ liệu điều kiện. Trong bối cảnh dữ liệu thưa, Pei et al. (2001) [10] phát triển H-Mine – thuật toán sử dụng H-struct với khả năng quét dữ liệu đúng một lần và lưu vết giao dịch theo dạng liên kết dọc cho thấy hiệu quả rõ rệt. Nhìn chung, các thuật toán truyền thống đều tập trung tối ưu hóa cấu trúc dữ liệu và giảm chi phí sinh ứng viên để cải thiện hiệu năng.

Trong môi trường dữ liệu không chắc chắn, mỗi giao dịch hoặc mục kèm theo một xác suất xuất hiện, khiến định nghĩa "tập mục thường xuyên" đòi hỏi cách định nghĩa lại khái niệm hỗ trợ. Chui & Kao (2007) [5] giới thiệu khái niệm Expected Support (ES), trong đó độ hỗ trợ kỳ vọng được tính bằng tổng xác suất xuất hiện của tập mục trên tất cả giao dịch, một tập mục được xem là thường xuyên khi ES vượt ngưỡng quy định. Dựa trên khái niệm này, nhiều hướng tiếp cận đã được mở rộng từ FIM truyền thống. Các thuật toán UApriori, UFP-Growth và UH-Mine là những thuật toán mở rộng quan trọng của các thuật toán khai thác tập mục thường xuyên truyền thống, được phát triển nhằm xử lý dữ liệu không chắc chắn. Mặc dù mỗi thuật toán xuất phát từ một hướng tiếp cận khác nhau nhưng chúng đều được điều chỉnh để phù hợp với đặc trưng xác suất của dữ liệu, đặc biệt trong cách tính hỗ trợ kỳ vọng và cơ chế cắt tỉa. Trong số đó, UFP-Growth cho hiệu năng cao trên dữ liệu dày nhờ khả năng nén cây mạnh, trong khi UH-Mine thể hiện ưu thế trên dữ liệu thưa nhờ cấu trúc liên kết dọc gọn nhẹ. Nhìn chung, khai thác tập mục trong dữ liệu không chắc chắn phải giải quyết với hai thách thức chính: chi phí tính toán Expected Support đáng kể

và không gian tìm kiếm tăng đáng kể do các tính chất suy giảm, vốn đóng vai trò trung tâm trong việc cắt tỉa ứng viên trong dữ liệu truyền thống, bị suy yếu đáng kể trong bối cảnh dữ liệu xác suất.

Đối với mô hình Top-K, thay vì dựa vào ngưỡng hỗ trợ tối thiểu, các thuật toán hướng đến việc trích xuất trực tiếp K tập mục mạnh nhất, giúp giảm sự phụ thuộc vào ngưỡng minSup vốn khó xác định tối ưu trong thực tiễn. Trên dữ liệu không chắc chắn, nhiều hướng tiếp cận Top-K được đề xuất, trong đó các chiến lược chủ đạo tập trung vào cơ chế nâng ngưỡng động và cắt tỉa nhánh dựa trên upper-bound. Một số công trình có ảnh hưởng lớn trong lĩnh vực này bao gồm Han et al. (2002) [11], đặt nền móng cho việc khai thác Top-K frequent closed patterns thông qua cấu trúc FP-tree và kỹ thuật threshold raising; Soulet & Raïssi (2006) với mô hình Top-K itemsets tổng quát sử dụng hệ thống bộ lọc upper-bound mạnh để loại bỏ sớm các nhánh không tiềm năng và Bernecker et al. (2009) [6] là những người mở rộng hướng tiếp cận sang môi trường dữ liệu không chắc chắn bằng cách mô hình hóa phân phối xác suất của hỗ trợ, từ đó cho phép áp dụng cơ chế branch-and-bound hiệu quả hơn. Một số nghiên cứu khác mở rộng theo hướng xác suất, trong đó các ràng buộc dựa trên xác suất vượt ngưỡng (như probabilistic frequent itemsets) được áp dụng để định nghĩa lại tập mục trong môi trường dữ liệu không chắc chắn. Gần đây, các hướng tiếp cận xấp xỉ như Riondato & Upfal (2010) [12] cho thấy hiệu năng cải thiện đáng kể nhờ sử dụng approximate upper-bounds dựa trên lý thuyết mẫu hóa, giúp loại trừ sớm các nhánh kém triển vọng mà vẫn duy trì sai lệch kết quả rất nhỏ. Nhìn tổng thể, khai thác Top-K từ dữ liệu không chắc chắn tập trung vào hai cơ chế cốt lõi là nâng ngưỡng động nhằm thu hẹp nhanh không gian tìm kiếm và cắt tỉa nhánh dựa trên upper-bound theo cơ chế branch-and-bound nhằm loại trừ các tập mục không có khả năng cạnh tranh vào Top-K.

CHƯƠNG 3. ĐỊNH NGHĨA VÀ PHÁT BIỂU CỦA VẤN ĐỀ

3.1 Định nghĩa

3.1.1 Mô hình dữ liệu không chắc chắn

Một cơ sở dữ liệu giao dịch không chắc chắn được ký hiệu là $D = \{T_1, T_2, \dots, T_n\}$, trong đó mỗi giao dịch T_j bao gồm một tập các cặp (i_k, p_{kj}), với $i_k$là một item thuộc tập item $I = \{i_1, i_2, \dots, i_m\}$ và $p_{kj} \in (0,1]$ là xác suất xuất hiện của item i_k trong giao dịch T_j. Mô hình biểu diễn dưới dạng xác suất này cho phép thể hiện độ không chắc chắn vốn có của dữ liệu, thường bắt nguồn từ sai số đo lường, tổng hợp dữ liệu cảm biến, giao dịch dự đoán hoặc dữ liệu thu thập từ môi trường thực không hoàn hảo (Chui & Kao, 2007; Aggarwal & Yu, 2009; Zhou, Zhao, Da Yan & Wilfred Ng (2012)) [5, 13, 14].

Một ví dụ minh họa được thể hiện trong Bảng 3.1. Mỗi item được gán một giá trị xác suất thể hiện mức độ tin cậy rằng item đó có xuất hiện trong giao dịch. Các giá trị trống ("–") biểu thị item không xuất hiện.

Bảng 3.1 Ví dụ về cơ sở dữ liệu giao dịch không chắc chắn

TID	A	B	C	D	E
T1	0.8	–	0.6	0.7	–
T2	0.4	0.9	–	0.1	–
T3	0.6	–	0.8	0.5	0.7
T4	–	0.7	0.4	0.9	–
T5	0.7	0.6	–	0.5	0.3
T6	–	–	0.7	0.6	0.4
T7	0.9	0.5	–	–	0.2
T8	0.5	0.8	0.6	0.7	–

Ví dụ này làm rõ rằng sự xuất hiện của mỗi item được gắn với một mức độ tin cậy định lượng, từ đó tạo nên đặc thù của các thuật toán khai thác dữ liệu xác suất.

3.1.2 Xác suất xuất hiện của một itemset

Với một tập mục $X = \{i_1, i_2, \dots, i_m\}$, xác suất để toàn bộ itemset xuất hiện trong một giao dịch T_j được ký hiệu là $P(X \subseteq T_j)$và được tính bằng tích các xác suất thành phần:

$$P(X \subseteq T_j) = \prod_{i \in X} p_{ij} \qquad (1)$$

Trong đó, p_{ij} là xác suất xuất hiện của item $i \in X$trong giao dịch T_j.

Ví dụ: Từ dữ liệu trong Bảng 3.1, ta có thể minh họa như sau:

$P(\{A\} \subseteq T_1) = 0.8$

$P(\{A, C, D\} \subseteq T_1) = 0.8 \times 0.6 \times 0.7 = 0.336$

Ví dụ này cho thấy với những itemset càng lớn, xác suất xuất hiện càng giảm—một đặc điểm quan trọng quyết định chiến lược pruning trong các thuật toán Top-K.

3.1.3 Hỗ trợ kỳ vọng (Expected Support – ES)

Hỗ trợ kỳ vọng (Expected Support) của một itemset X trong toàn bộ cơ sở dữ liệu $D = \{T_1, T_2, \dots, T_n\}$ được định nghĩa là tổng các xác suất xuất hiện của X trong từng giao dịch:

$$ES(X) = \sum_{j=1}^{n} P(X \subseteq T_j) \qquad (2)$$

Khác với hỗ trợ truyền thống dựa trên tần suất đếm số lần xuất hiện, $ES(X)$ đo lường kỳ vọng xuất hiện của itemset, phản ánh bản chất xác suất của dữ liệu. Trong khai thác dữ liệu không chắc chắn, các thuật toán thường dựa vào $ES(X)$ để đánh giá độ phổ biến của các itemset.

Bảng 3.2 Minh họa tính Expected Support của {A} và {A,C,D}

TID	**P(A)**	**P(A,C,D)**
T1	0.8	0.8 × 0.6 × 0.7 = 0.336

T2	0.4	0.4 × 0 × 0.1 = 0
T3	0.6	0.6 × 0.8 × 0.5 = 0.24
T4	0	0 × 0.4 × 0.9 = 0
T5	0.7	0.7 × 0 × 0.5 = 0
T6	0	0 × 0.7 × 0.6 = 0
T7	0.9	0.9 × 0 × 0 = 0
T8	0.5	0.5 × 0.6 × 0.7 = 0.21

Từ đó:

$ES(A) = 0.8 + 0.4 + 0.6 + 0 + 0.7 + 0 + 0.9 + 0.5 = 3.9$

$ES(ACD) = 0.336 + 0 + 0.24 + 0 + 0 + 0 + 0 + 0.21 = 0.786$

Kết quả này cho thấy item {A} có kỳ vọng xuất hiện cao và khả năng cao nằm trong các tập mục phổ biến nhất, trong khi itemset {A,C,D} có mức phổ biến thấp hơn đáng kể.

3.1.4 Định nghĩa bài toán Top-K Frequent Itemsets

Cho một cơ sở dữ liệu không chắc chắn D và một giá trị nguyên dương K, mục tiêu là xác định tập S gồm đúng K itemset có Expected Support cao nhất. Một itemset X được xem là thuộc tập nghiệm S khi và chỉ khi:

$$ES(X) \geq ES(Y), \forall Y \notin S \qquad (4)$$

Trong đó, Y biểu thị mọi itemset không thuộc tập Top-K. Điều kiện này đảm bảo rằng mỗi itemset trong S đều có Expected Support không thấp hơn bất kỳ itemset nào nằm ngoài S, tương ứng với việc S chính là K itemset đứng đầu khi sắp xếp toàn bộ không gian itemset theo Expected Support giảm dần.

Khác với các phương pháp khai thác tập mục thường xuyên trong dữ liệu truyền thống vốn yêu cầu người dùng chỉ định trước một ngưỡng hỗ trợ (minsup), bài toán Top-K không cần thiết lập ngưỡng này. Thay vào đó, thuật toán áp dụng cơ chế threshold raising, tức tự động điều chỉnh và nâng dần ngưỡng hỗ trợ trong quá trình khai thác nhằm thu hẹp không gian tìm kiếm và tăng hiệu quả tính toán.

Ví dụ: Top-5 item phổ biến nhất từ dữ liệu Bảng 3.1

Ta tính ES cho từng item đơn lẻ:

$ES(A) = 0.8 + 0.4 + 0.6 + 0 + 0.7 + 0 + 0.9 + 0.5 = 3.9$

$ES(B) = 0 + 0.9 + 0 + 0.7 + 0.6 + 0 + 0.5 + 0.8 = 3.5$

$ES(C) = 0.6 + 0 + 0.8 + 0.4 + 0 + 0.7 + 0 + 0.6 = 3.1$

$ES(D) = 0.7 + 0.1 + 0.5 + 0.9 + 0.5 + 0.6 + 0 + 0.7 = 4.0$

$ES(E) = 0 + 0 + 0.7 + 0 + 0.3 + 0.4 + 0.2 + 0 = 1.6$

Kết quả Top-5 được trình bày trong bảng sau.

Bảng 3.3 Top-5 item phổ biến nhất theo Expected Support

Xếp hạng	**Item**	**ES(X)**
1	D	4.0
2	A	3.9
3	B	3.5
4	C	3.1
5	E	1.6

CHƯƠNG 4. PHƯƠNG PHÁP

4.1 Thuật toán UApriori cho Top-K Frequent Itemsets

Thuật toán UApriori mở rộng từ Apriori truyền thống và hoạt động dựa trên cơ chế sinh–kiểm tra (generate-and-test), trong đó độ phổ biến của itemset được đánh giá thông qua hỗ trợ kỳ vọng. Với một itemset X, hỗ trợ kỳ vọng được tính bằng tổng xác suất xuất hiện của itemset trong toàn bộ giao dịch:

$$ES(X) = \sum_{T_i \in D} P(X \subseteq T_i)$$

Xác suất xuất hiện trong từng giao dịch được xác định bằng tích xác suất của các item thành phần:

$$P(X \subseteq T_i) = \prod_{x \in X} P(x \in T_i)$$

Trong quá trình khai thác Top-K, UApriori duy trì một danh sách tạm thời chứa K itemset có ES cao nhất, và khi danh sách đã đủ K phần tử, thuật toán cập nhật ngưỡng động theo công thức:

$$threshold = \min_{X \in TopK} ES(X)$$

Ngưỡng này giúp loại bỏ sớm các ứng viên có khả năng thấp. Ưu điểm của UApriori là đơn giản, dễ triển khai và kế thừa trực tiếp tính chất suy giảm của Apriori. Tuy nhiên, nhược điểm lớn của thuật toán là tạo ra số lượng ứng viên rất lớn, đặc biệt khi dữ liệu có số lượng item cao hoặc phân bố thưa, làm cho chi phí tính toán hỗ trợ kỳ vọng trở nên rất tốn kém. Do đó, UApriori hiếm khi được dùng cho dữ liệu lớn hoặc dữ liệu xác suất dày đặc.

4.2 Thuật toán UFP-Growth trong khai thác Top-K

Thuật toán UFP-Growth kế thừa cơ chế của FP-Growth nhưng được điều chỉnh để xử lý dữ liệu xác suất thông qua cấu trúc UFP-tree. Cây này được xây dựng bằng cách sắp xếp các item theo giá trị hỗ trợ kỳ vọng giảm dần và chèn từng giao dịch vào cây cùng với xác suất xuất hiện tương ứng. Trong quá trình khai thác Top-K, mỗi

item được mở rộng thành một cơ sở dữ liệu điều kiện (conditional database), nhưng chỉ tiếp tục mở rộng nếu giá trị upper-bound của itemset không nhỏ hơn ngưỡng hiện tại, trong đó upper-bound được tính bằng:

$$UB(X) = \sum_{T_i \in D} \min_{x \in X} P(x \in T_i)$$

Khi phát hiện một itemset mới có hỗ trợ kỳ vọng vượt ngưỡng, danh sách Top-K được cập nhật và ngưỡng mới được xác định theo:

$$threshold = \min_{X \in TopK} ES(X)$$

Nhờ cấu trúc nén của FP-tree, thuật toán hạn chế được việc sinh ứng viên và khai thác theo cấu trúc prefix giúp hiệu quả đặc biệt cao trên dữ liệu dày, nơi nhiều giao dịch chia sẻ các tiền tố (prefix) giống nhau. Ưu điểm của UFP-Growth là nén mạnh dữ liệu, hạn chế sinh ứng viên và đạt hiệu năng rất tốt khi dữ liệu có mật độ lớn. Tuy nhiên, nhược điểm của thuật toán là tiêu tốn bộ nhớ khi dữ liệu thưa (ít trùng lặp), dẫn đến FP-tree phân nhánh rộng và mất hiệu quả. Điều này cho thấy UFP-Growth chỉ thực sự phù hợp trong môi trường dữ liệu dày đặc có cấu trúc lặp lại.

4.3 Thuật toán UH-Mine trong khai thác Top-K

Thuật toán UH-Mine, một phiên bản mở rộng theo hướng xác suất của H-Mine, được thiết kế đặc biệt nhằm khai thác dữ liệu thưa thông qua cấu trúc H-struct theo dạng vertical, tối ưu hóa hiệu quả lưu trữ và xử lý trong môi trường dữ liệu không chắc chắn. Với mỗi item, thuật toán lưu trữ danh sách các cặp $(T_i, P(x \in T_i))$, cho phép mở rộng itemset theo chiều sâu mà không cần xây dựng cây. Trong quá trình khai thác, hỗ trợ kỳ vọng của itemset được tính theo công thức:

$$ES(X) = \sum_{T_i \in D} P(X \subseteq T_i)$$

Để cắt tỉa nhánh, UH-Mine sử dụng upper-bound để ước lượng giá trị tối đa mà itemset có thể đạt:

$$UB(X) < threshold \Rightarrow prune(X)$$

Do sử dụng cấu trúc liên kết dọc và chiến lược duyệt theo chiều sâu, UH-Mine đặc biệt hiệu quả trong trường hợp dữ liệu thưa, nơi mỗi giao dịch chỉ chứa ít item và độ trùng lặp thấp. Ưu điểm của UH-Mine là không tốn bộ nhớ để xây dựng cây, xử lý rất tốt các giao dịch ngắn và dữ liệu rời rạc. Tuy nhiên, nhược điểm xuất hiện khi dữ liệu dày danh sách vertical trở nên lớn, chi phí cập nhật trong quá trình mở rộng tăng đáng kể, và hiệu năng giảm rõ rệt so với UFP-Growth.

4.4 Thuật toán kết hợp: Hybrid-UFHM (UFP-Growth + UH-Mine)

Từ phân tích ưu và nhược điểm của ba thuật toán UApriori, UFP-Growth và UH-Mine, có thể thấy rõ rằng UFP-Growth và UH-Mine mang tính chất bổ sung cho nhau trong khai thác Top-K Frequent Itemsets từ dữ liệu không chắc chắn. Cụ thể, UFP-tree hoạt động đặc biệt hiệu quả trên dữ liệu dày, nơi nhiều giao dịch chia sẻ cấu trúc tương đồng, nhờ khả năng nén mạnh và khai thác dựa trên các prefix chung. Ngược lại, H-struct trong UH-Mine lại tỏ ra vượt trội trên dữ liệu thưa, nơi FP-tree không thể nén tốt và các giao dịch có ít trùng lặp, nhờ cấu trúc vertical gọn nhẹ giúp duyệt nhanh và hạn chế sử dụng bộ nhớ. Từ đó, dễ dàng rút ra nhận định rằng không một thuật toán đơn lẻ nào có thể tối ưu trên mọi loại phân bố dữ liệu nên việc kết hợp hai mô hình này sẽ cho phép xử lý hiệu quả cả dữ liệu dày lẫn dữ liệu thưa. Dựa trên định hướng này, thuật toán kết hợp Hybrid-UFHM được đề xuất nhằm lựa chọn cấu trúc phù hợp theo mật độ trung bình của cơ sở dữ liệu D, được đo bằng:

$$density = \frac{\sum_{T_i \in D} | T_i |}{| D |}$$

Nếu $density > \theta$(ví dụ 40–50%), dữ liệu được xem là dày và thuật toán chọn nhánh UFP-Growth và ngược lại khi mật độ thấp nhánh UH-Mine được ưu tiên. Dù sử dụng cấu trúc nào, Hybrid-UFHM vẫn tuân theo các cơ chế chung của khai thác Top-K trong dữ liệu không chắc chắn là tính hỗ trợ kỳ vọng cho từng itemset, duy trì danh sách Top-K hiện thời, áp dụng chiến lược threshold raising với ngưỡng:

$$threshold = \min_{X \in TopK} ES(X)$$

Và thực hiện upper-bound pruning để loại bỏ các nhánh không tiềm năng:

$$UB(X) < threshold \Rightarrow prune(X)$$

Quá trình khai thác được tiến hành theo chiều sâu, mở rộng các prefix dựa trên cấu trúc tương ứng (FP-tree hoặc H-struct), nhằm đạt hiệu quả tối ưu trên từng loại dữ liệu. Nhờ kết hợp linh hoạt, thuật toán sở hữu nhiều ưu điểm như tối ưu hiệu năng trên cả dữ liệu dày và dữ liệu thưa, giảm chi phí bộ nhớ so với việc chỉ sử dụng UFP-tree, đồng thời tận dụng được tốc độ xử lý của FP-tree và sự gọn nhẹ của H-struct. Tuy nhiên, bên cạnh các ưu điểm nổi bật, thuật toán vẫn tồn tại một số hạn chế như cần lựa chọn ngưỡng phân loại mật độ phù hợp và có thể phát sinh chi phí khi dữ liệu không đồng nhất, đòi hỏi cân nhắc kỹ lưỡng trong triển khai thực tế.

CHƯƠNG 5. THIẾT LẬP THỰC NGHIỆM

5.1 Mục tiêu thực nghiệm

Mục tiêu của chương thực nghiệm là đánh giá hiệu năng của các thuật toán UApriori, UFP-Growth, UH-Mine và thuật toán kết hợp Hybrid-UFHM trong việc khai thác Top-K tập mục thường xuyên từ cơ sở dữ liệu không chắc chắn. Thực nghiệm nhằm phân tích thời gian chạy, mức độ tiêu thụ bộ nhớ, khả năng mở rộng khi kích thước dữ liệu tăng lên và mức độ hiệu quả của các cơ chế cắt tỉa như threshold raising và upper-bound pruning. Ngoài ra, thực nghiệm còn hướng đến việc kiểm chứng tính thích ứng của thuật toán kết hợp đối với dữ liệu có mật độ phân bố khác nhau, từ dữ liệu thưa cho đến dữ liệu dày, qua đó đánh giá tính ưu việt của mô hình kết hợp so với từng thuật toán đơn lẻ.

5.2 Môi trường thực nghiệm

Các thử nghiệm được thực hiện trên máy tính cá nhân sử dụng bộ vi xử lý Intel Core i5-10300H (2.50 GHz) cùng 8 GB RAM, chạy hệ điều hành Windows 11 64-bit. Toàn bộ thuật toán được cài đặt và thực thi bằng OpenJDK 11.0.29 trong môi trường IntelliJ IDEA. Thời gian chạy được đo bằng hàm System.nanoTime(), và mỗi phép thử được lặp lại nhiều lần để lấy giá trị trung bình nhằm đảm bảo kết quả ổn định và đáng tin cậy.

5.3 Dữ liệu thực nghiệm

Trong thực nghiệm này, bốn bộ dữ liệu phổ biến thường được sử dụng trong lĩnh vực khai thác tập mục thường xuyên được lựa chọn, bao gồm Chess, Foodmart, Retail và t20i6d100k. Đây là những bộ dữ liệu có kích thước, số lượng giao dịch và mức độ dày và thưa rất khác nhau, giúp đánh giá đầy đủ hiệu năng của từng thuật toán trong nhiều bối cảnh dữ liệu. Vì các bộ dữ liệu gốc là dữ liệu cố định, xác suất xuất hiện của từng item trong mỗi giao dịch được gán ngẫu nhiên trong khoảng $(0,1]$ nhằm chuyển đổi thành dữ liệu không chắc chắn.

Bảng 5.1 Một số thông số về các bộ dữ liệu dùng để thử nghiệm

Bộ dữ liệu	**Số lượng giao dịch**	**Số lượng items (I)**	**Độ dài trung bình của giao dịch (A)**	**Mật độ (A / I) * 100**
Chess	3196	75	37	49.33 %
Foodmart	4141	1559	4.42	0.28 %
Retail	88162	16470	10.30	0.06 %
t20i6d100k	99922	893	19.90	2.23 %

Toàn bộ mã nguồn của các giải thuật Uapriori, UFP-Growth, UH-Mine và Hybrid-UFHM được lưu trữ trên GitHub:

https://github.com/Huynneh/DataMiningTopKUncertain_CD3.git

CHƯƠNG 6. KẾT QUẢ THỰC NGHIỆM VÀ THẢO LUẬN

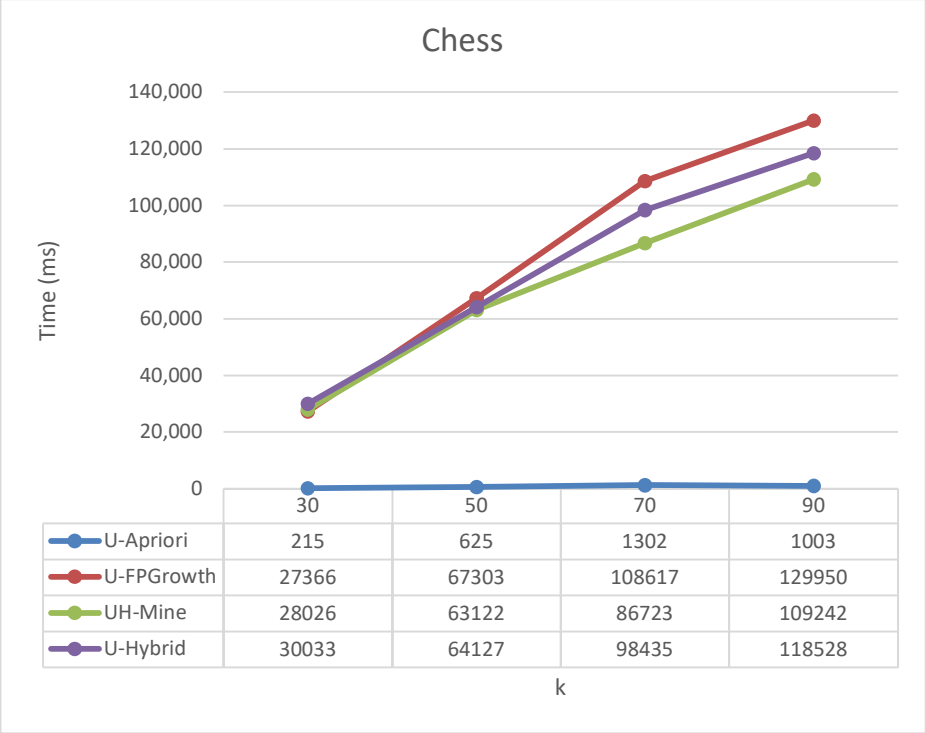
Chương này trình bày kết quả thực nghiệm và phân tích hiệu suất của bốn thuật toán: UApriori, UFP-Growth, UH-Mine và Hybrid-UFHM trên các bộ dữ liệu không chắc chắn Chess, Foodmart, Retail và t20i6d100k. Các thuật toán được đánh giá dựa trên hai tiêu chí chính: (1) thời gian chạy, (2) mức sử dụng bộ nhớ, cùng với việc phân tích khả năng mở rộng và ảnh hưởng của mật độ dữ liệu. Các kết quả thu được cho phép so sánh ưu – nhược điểm của từng thuật toán cũng như xác minh tính hiệu quả của mô hình kết hợp Hybrid-UFHM.

6.1 Hiệu suất về thời gian chạy

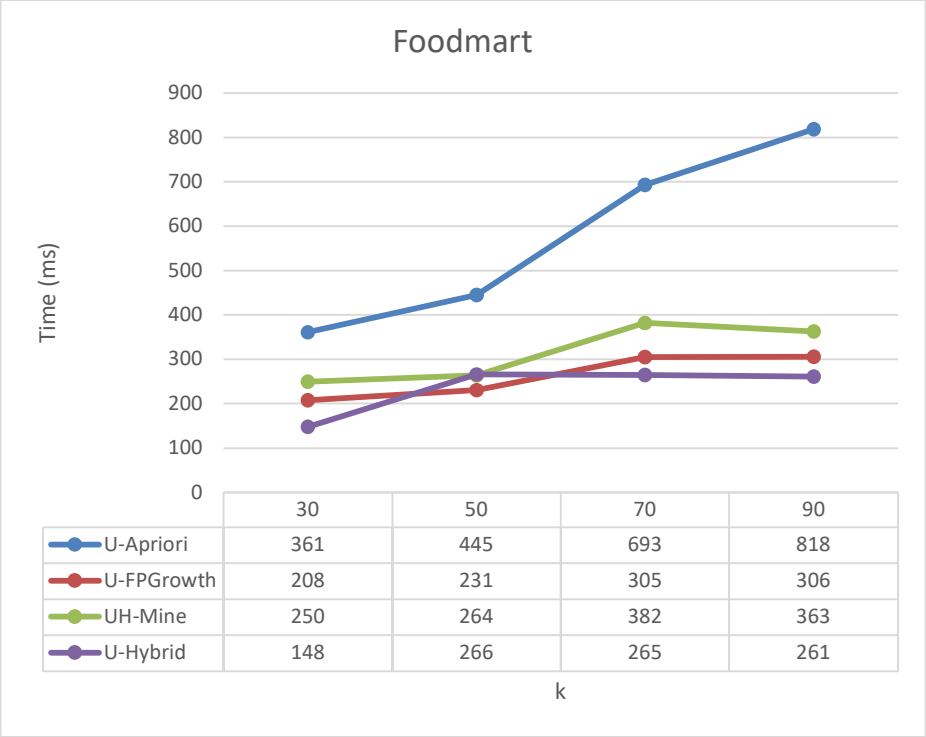
Kết quả thực nghiệm trên bốn bộ dữ liệu Chess, Foodmart, Retail và t20i6d100k cho thấy hiệu năng của các thuật toán khai thác tập mục thường xuyên phụ thuộc mạnh vào đặc tính cấu trúc, mật độ và độ dài giao dịch của dữ liệu. Trên bộ dữ liệu Chess, với đặc trưng dày và giao dịch ngắn, U-Apriori đạt hiệu năng cao nhất, với thời gian chạy dao động từ 215 ms đến 1302 ms, trong khi UFP-Growth và UH-Mine mất hàng chục nghìn mili-giây do chi phí xây dựng cây FP hoặc xử lý cấu trúc vertical lớn. Hybrid-UFHM, mặc dù kết hợp các nhánh UFP-Growth và UH-Mine, vẫn không nhanh hơn U-Apriori trong trường hợp này. Ngược lại, trên bộ dữ liệu Foodmart, đặc trưng thưa với giao dịch ngắn, UFP-Growth và UH-Mine đạt hiệu năng tốt hơn nhờ khả năng nén cây FP và cấu trúc vertical nhẹ, trong khi U-Apriori có thời gian chạy cao hơn. Hybrid-UFHM tiếp tục duy trì hiệu năng tốt nhất nhờ chọn nhánh UH-Mine phù hợp, đặc biệt ở các ngưỡng minsup cao, với thời gian chạy từ 148 ms đến 266 ms. Trên bộ dữ liệu Retail, vốn cực thưa và có số lượng giao dịch lớn, U-Apriori là thuật toán hiệu quả nhất, với thời gian chạy từ 108146 ms đến 165014 ms, nhanh hơn đáng kể so với UFP-Growth, UH-Mine và Hybrid-UFHM, do hầu hết các tập mục kích thước lớn bị loại bỏ sớm, giảm thiểu chi phí xử lý. Trên bộ dữ liệu t20i6d100k, với giao dịch dài, U-Apriori tiếp tục duy trì thời gian chạy thấp hơn đáng kể so với UFP-Growth và UH-Mine, trong khi Hybrid-UFHM giữ hiệu

năng ổn định nhờ khả năng lựa chọn nhánh thuật toán phù hợp, với thời gian dao động từ 645094 ms đến 1813064 ms.

Nhìn chung, kết quả chỉ ra rằng U-Apriori hiệu quả nhất trong dữ liệu thưa hoặc cực thưa, UFP-Growth phù hợp với dữ liệu dày, UH-Mine tối ưu trong dữ liệu thưa – giao dịch ngắn, và Hybrid-UFHM là phương pháp tương đối ổn định nhờ khả năng thích ứng động với đặc tính từng bộ dữ liệu, luôn nằm trong nhóm nhanh nhất.

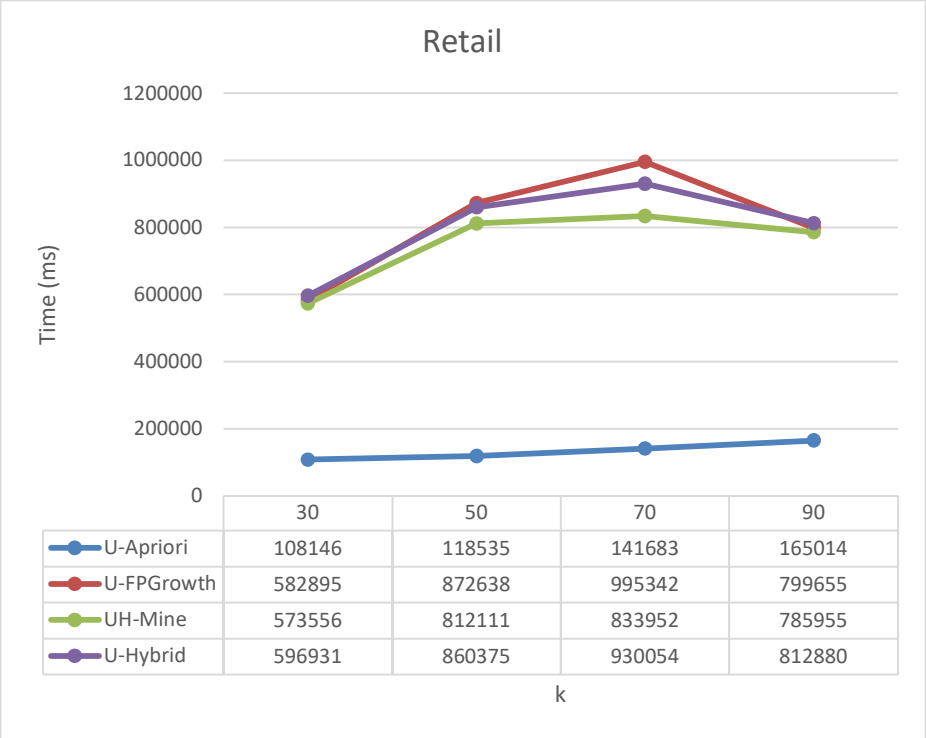


Hình 6.1 Hiệu suất về thời gian chạy của các thuật toán trên bộ Chess



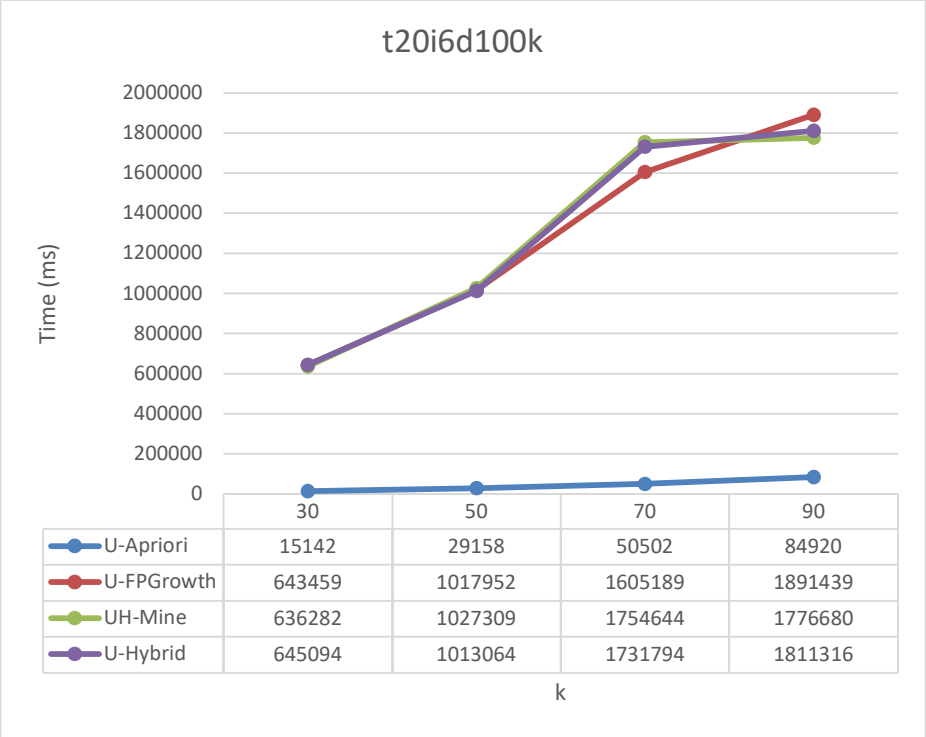
	30	50	70	90
U-Apriori	361	445	693	818
U-FPGrowth	208	231	305	306
UH-Mine	250	264	382	363
U-Hybrid	148	266	265	261

Hình 6.2 Hiệu suất về thời gian chạy của các thuật toán trên bộ Foodmart



	30	50	70	90
U-Apriori	108146	118535	141683	165014
U-FPGrowth	582895	872638	995342	799655
UH-Mine	573556	812111	833952	785955
U-Hybrid	596931	860375	930054	812880

Hình 6.3 Hiệu suất về thời gian chạy của các thuật toán trên bộ Retail



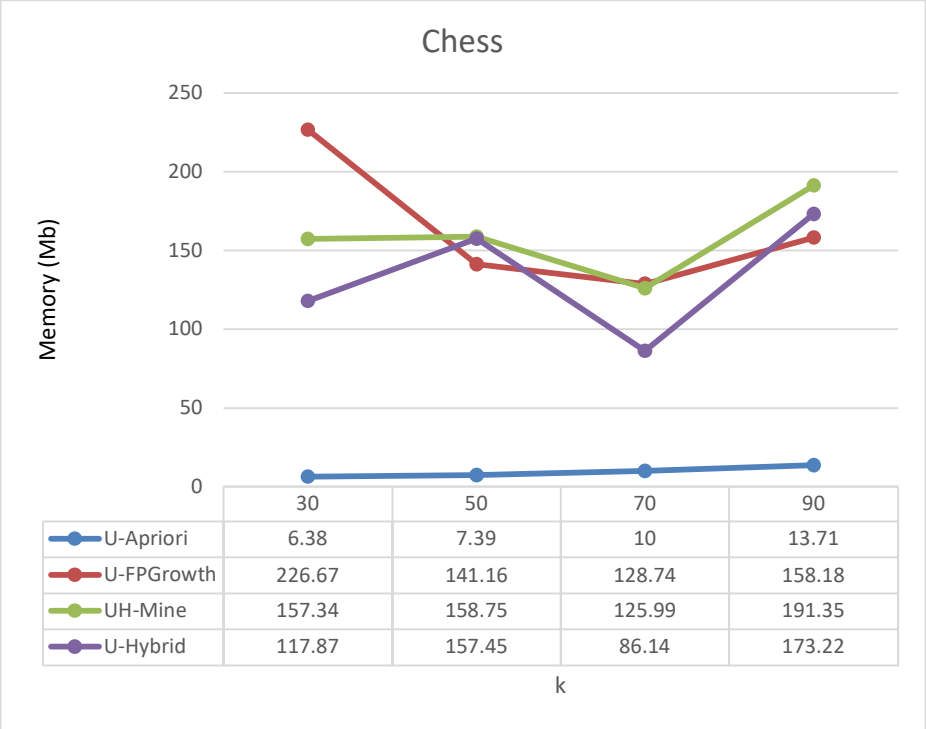
Hình 6.4 Hiệu suất về thời gian chạy của các thuật toán trên bộ t20i6d100k

6.2 Hiệu suất về mức tiêu thụ bộ nhớ

Hiệu suất tiêu thụ bộ nhớ của các thuật toán khai thác tập mục thường xuyên phụ thuộc rõ rệt vào đặc tính cấu trúc và mật độ dữ liệu. Trên bộ dữ liệu Chess, dày với giao dịch ngắn, U-Apriori sử dụng bộ nhớ thấp nhất, dao động từ 6.38 MB đến 13.71 MB, trong khi UFP-Growth tiêu tốn nhiều bộ nhớ hơn (128–226 MB) do chi phí lưu trữ cây FP lớn. UH-Mine và Hybrid-UFHM sử dụng bộ nhớ ở mức trung bình, từ 86 MB đến 191 MB, với Hybrid-UFHM giữ mức tiêu thụ hợp lý nhờ lựa chọn nhánh thuật toán thích hợp. Trên bộ dữ liệu Foodmart, thưa với giao dịch ngắn, U-Apriori vẫn duy trì mức sử dụng bộ nhớ thấp (0.53–21.04 MB), trong khi UH-Mine tiêu tốn nhiều hơn ở một số ngưỡng (12–17 MB). UFP-Growth và Hybrid-UFHM có mức biến động, nhưng Hybrid-UFHM thường ổn định nhờ chọn nhánh hiệu quả. Trên bộ dữ liệu Retail, cực thưa và có nhiều giao dịch, UH-Mine và UFP-Growth tiêu tốn bộ nhớ thấp hoặc tương đương U-Apriori tùy ngưỡng, trong khi Hybrid-UFHM dao động trung bình, phản ánh việc kết hợp các nhánh thuật toán duy trì mức sử dụng bộ

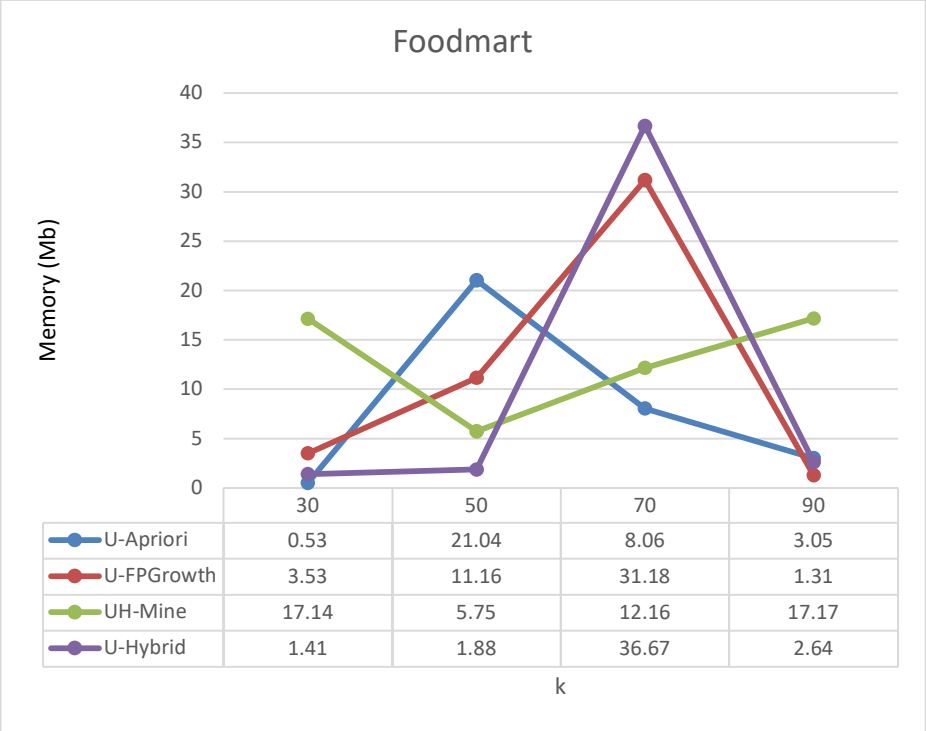
nhớ ổn định. Trên bộ dữ liệu t20i6d100k, với giao dịch dài, mức tiêu thụ bộ nhớ của các thuật toán biến động lớn; U-Apriori và UFP-Growth có lúc sử dụng bộ nhớ cao (234–283 MB), UH-Mine dao động từ 44 đến 256 MB, trong khi Hybrid-UFHM thể hiện khả năng kiểm soát bộ nhớ tốt hơn ở ngưỡng thấp (10.44 MB tại minsup 30), nhưng tăng lên ở ngưỡng cao do lựa chọn nhánh thuật toán.

Nhìn chung, U-Apriori tiêu thụ bộ nhớ thấp trên dữ liệu nhỏ hoặc thưa, UFP-Growth sử dụng bộ nhớ nhiều hơn trên dữ liệu dày do cấu trúc cây FP, UH-Mine duy trì mức sử dụng ổn định trong dữ liệu thưa – giao dịch ngắn nhờ cấu trúc vertical, và Hybrid-UFHM giữ khả năng kiểm soát bộ nhớ hợp lý nhờ lựa chọn nhánh thuật toán thích ứng với đặc tính từng bộ dữ liệu.



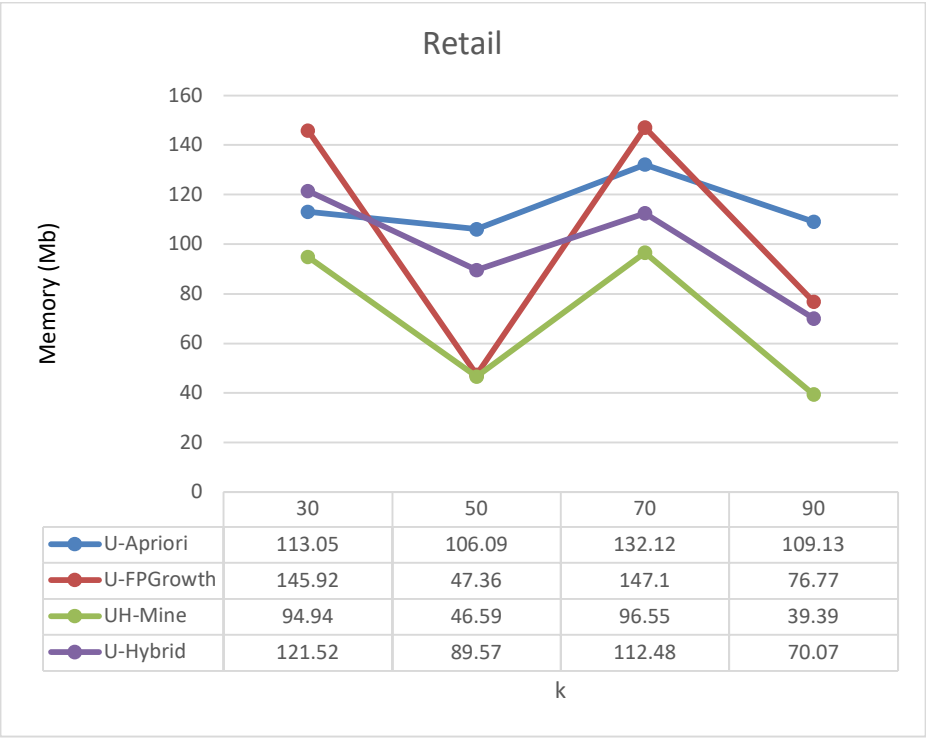
	30	50	70	90
U-Apriori	6.38	7.39	10	13.71
U-FPGrowth	226.67	141.16	128.74	158.18
UH-Mine	157.34	158.75	125.99	191.35
U-Hybrid	117.87	157.45	86.14	173.22

Hình 6.5 Hiệu suất về mức tiêu thụ bộ nhớ của các thuật toán trên bộ Chess



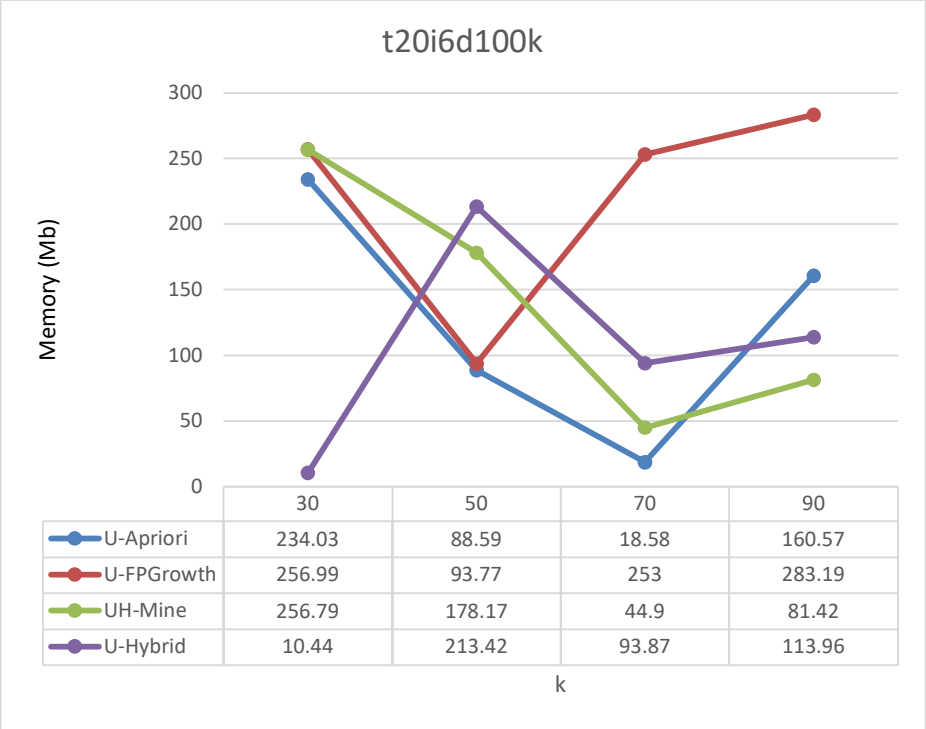
	30	50	70	90
U-Apriori	0.53	21.04	8.06	3.05
U-FPGrowth	3.53	11.16	31.18	1.31
UH-Mine	17.14	5.75	12.16	17.17
U-Hybrid	1.41	1.88	36.67	2.64

Hình 6.6 Hiệu suất về mức tiêu thụ bộ nhớ của các thuật toán trên bộ Foodmart



	30	50	70	90
U-Apriori	113.05	106.09	132.12	109.13
U-FPGrowth	145.92	47.36	147.1	76.77
UH-Mine	94.94	46.59	96.55	39.39
U-Hybrid	121.52	89.57	112.48	70.07

Hình 6.7 Hiệu suất về mức tiêu thụ bộ nhớ của các thuật toán trên bộ Retail



Hình 6.8 Hiệu suất về mức tiêu thụ bộ nhớ của các thuật toán trên bộ t20i6d100k

CHƯƠNG 7. KẾT LUẬN

7.1 Kết luận

Nghiên cứu này đã khảo sát và đánh giá hiệu năng của các thuật toán khai thác Top-K tập mục thường xuyên trong môi trường dữ liệu không chắc chắn, bao gồm UApriori, UFP-Growth, UH-Mine và mô hình kết hợp Hybrid-UFHM. Thông qua phân tích lý thuyết, đặc tính hoạt động và thực nghiệm trên các bộ dữ liệu Chess, Foodmart, Retail và t20i6d100k với các mức độ dày – thưa và độ dài giao dịch khác nhau, nghiên cứu đã làm rõ ưu điểm, hạn chế và phạm vi ứng dụng của từng thuật toán.

Kết quả thực nghiệm cho thấy hiệu năng của các thuật toán phụ thuộc đáng kể vào đặc tính dữ liệu. UFP-Growth hiệu quả với dữ liệu dày nhờ khả năng nén FP-tree, trong khi UH-Mine ưu thế trên dữ liệu thưa nhờ cấu trúc vertical gọn nhẹ. UApriori tỏ ra phù hợp với dữ liệu cực thưa và số lượng giao dịch lớn, nhờ khả năng loại bỏ nhanh các ứng viên mở rộng. Hybrid-UFHM duy trì thời gian chạy ổn định và thấp hơn hoặc tương đương với thuật toán nhanh nhất trên hầu hết bộ dữ liệu, đồng thời kiểm soát bộ nhớ tốt nhờ cơ chế lựa chọn động giữa FP-tree và H-struct.

Nhìn chung, nghiên cứu đã hoàn thành mục tiêu đề ra: hệ thống hóa cơ sở lý thuyết, phân tích chi tiết các thuật toán hiện có, và đề xuất một giải pháp mới có tính ứng dụng cao. Kết quả này không chỉ hỗ trợ lựa chọn thuật toán phù hợp trong thực tiễn mà còn mở rộng khả năng khai thác Top-K trong môi trường dữ liệu không chắc chắn.

7.2 Hướng phát triển

Trên cơ sở kết quả thực nghiệm và những hạn chế nhận thấy, một hướng phát triển quan trọng của Hybrid-UFHM là hoàn thiện cơ chế lựa chọn động giữa FP-tree và H-struct. Việc này có thể thực hiện thông qua xây dựng các mô hình đánh giá đặc trưng dữ liệu chính xác và toàn diện hơn, kết hợp các chỉ số thống kê như mức độ phân tán của các mục, phân bố độ dài giao dịch, cũng như mối tương quan giữa các mục trong giao dịch. Mục tiêu là giảm thiểu số lần lựa chọn sai cấu trúc khai thác và

nâng cao tính ổn định của thuật toán trên các bộ dữ liệu có đặc tính đa dạng hoặc phân bố không đồng nhất.

Bên cạnh đó, việc tối ưu hóa cơ chế cắt tỉa và cấu trúc dữ liệu cũng là một hướng phát triển quan trọng. Có thể phát triển các dạng upper-bound chặt chẽ hơn, áp dụng kỹ thuật ước lượng xác suất tiên tiến, nén FP-tree bằng mã hóa tuyến tính hoặc rút gọn đường dẫn, và tối ưu hóa tid-list của UH-Mine. Những cải tiến này sẽ giúp giảm lượng phép tính không cần thiết, tăng hiệu quả khai thác, đồng thời tiết kiệm bộ nhớ trong quá trình xử lý các bộ dữ liệu lớn hoặc giao dịch dài.

Mở rộng khả năng xử lý song song và phân tán cũng là hướng triển vọng, nhằm đáp ứng nhu cầu khai thác dữ liệu quy mô lớn. Hybrid-UFHM có thể phân tách không gian tìm kiếm và khai thác trên nhiều lõi hoặc cụm máy, kết hợp các mô hình học máy để dự đoán cấu trúc khai thác tối ưu dựa trên đặc trưng dữ liệu. Cách tiếp cận này giúp giảm chi phí đánh giá trong quá trình khai thác, đồng thời nâng cao khả năng mở rộng và tốc độ xử lý của thuật toán.

Cuối cùng, Hybrid-UFHM có thể được mở rộng để xử lý các dạng dữ liệu phức tạp hơn như dữ liệu luồng, dữ liệu thời gian thực hoặc dữ liệu nhiều chiều. Ngoài bài toán Top-K, thuật toán cũng có thể áp dụng cho khai thác các loại mẫu nâng cao như tập mục đóng, tập mục cực đại hoặc mẫu tuần tự xác suất. Những hướng phát triển này kỳ vọng sẽ củng cố hiệu quả, tăng tính ổn định và mở rộng phạm vi ứng dụng của Hybrid-UFHM trong các hệ thống khai phá dữ liệu hiện đại.

TÀI LIỆU THAM KHẢO

[1] C. C. Aggarwal and P. S. Yu, "A Survey of Uncertain Data Algorithms and Applications," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 5, pp. 609-623, May 2009, doi: 10.1109/TKDE.2008.190

[2] Han, J., Pei, J., Yin, Y. *et al.* Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Mining and Knowledge Discovery* 8, 53–87 (2004). https://doi.org/10.1023/B:DAMI.0000005258.31418.83

[3] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. SIGMOD Rec. 29, 2 (June 2000), 1–12. https://doi.org/10.1145/335191.335372

[4] Mohammed J. Zaki. 2000. Scalable Algorithms for Association Mining. IEEE Trans. on Knowl. and Data Eng. 12, 3 (May 2000), 372–390. https://doi.org/10.1109/69.846291

[5] Chui, CK., Kao, B., Hung, E. (2007). Mining Frequent Itemsets from Uncertain Data. In: Zhou, ZH., Li, H., Yang, Q. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2007. Lecture Notes in Computer Science(), vol 4426. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-71701-0_8

[6] Thomas Bernecker, Hans-Peter Kriegel, Matthias Renz, Florian Verhein, and Andreas Zuefle. 2009. Probabilistic frequent itemset mining in uncertain databases. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '09). Association for Computing Machinery, New York, NY, USA, 119–128. https://doi.org/10.1145/1557019.1557039

[7] Bernecker, T., Cheng, R., Cheung, D.W. *et al.* Model-based probabilistic frequent itemset mining. *Knowl Inf Syst* **37**, 181–217 (2013). https://doi.org/10.1007/s10115-012-0561-2

[8] Aggarwal, Charu & Li, Yan & Wang, Jianyong & Wang, Jing. (2009). Frequent pattern mining with uncertain data. Proceedings of the ACM SIGKDD

International Conference on Knowledge Discovery and Data Mining. 29-38. 10.1145/1557019.1557030.

[9] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules in Large Databases. In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 487–499.

[10] Pei, Jian & Han, Jiawei & Lu, Hongjun & Nishio, Shojiro & Tang, Shiwei & Yang, Dongqing. (2001). H-Mine: Hyper-Structure Mining of Frequent Patterns in Large Databases. Proceedings - IEEE International Conference on Data Mining, ICDM. 441-448. 10.1109/ICDM.2001.989550.

[11] Han, J., Wang, J., Lu, Y., & Tzvetkov, P. (2002). Mining top-K frequent closed patterns without minimum support. In *Proceedings - 2002 IEEE International Conference on Data Mining, ICDM 2002* (pp. 211-218).

[12] Pietracaprina, Andrea & Riondato, Matteo & Upfal, Eli & Vandin, Fabio. (2010). Mining Top-K Frequent Itemsets Through Progressive Sampling. Data Mining and Knowledge Discovery - DATAMINE. 21. 10.1007/s10618-010-0185-7.

[13] Ezeife, C.I., Liu, Y. Fast incremental mining of web sequential patterns with PLWAP tree. *Data Min Knowl Disc* **19**, 417–418 (2009). https://doi.org/10.1007/s10618-009-0144-3

[14] Zhou Zhao, Da Yan, and Wilfred Ng. 2012. Mining probabilistically frequent sequential patterns in uncertain databases. In Proceedings of the 15th International Conference on Extending Database Technology (EDBT '12). Association for Computing Machinery, New York, NY, USA, 74–85. https://doi.org/10.1145/2247596.2247606