TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN HỆ THỐNG THÔNG TIN

ĐỖ NGỌC HÂN - 21133030

TRẦN THỊ NGỌC TRANG – 21133109

Đề Tài:

TÌM HIỂU HỆ THỐNG GỢI Ý SỬ DỤNG KỸ THUẬT KHAI PHÁ TẬPHỮU ÍCH VÀ GNN

TIỂU LUẬN CHUYÊN NGÀNH

GIÁO VIÊN HƯỚNG DẪN

THS. TRẦN TRỌNG BÌNH

KHÓA 2021 – 2025

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN HỆ THỐNG THÔNG TIN

ĐỖ NGỌC HÂN - 21133030

TRẦN THỊ NGỌC TRANG – 21133109

Đề Tài:

TÌM HIỂU HỆ THỐNG GỢI Ý SỬ DỤNG KỸ THUẬT KHAI PHÁ TẬP HỮU ÍCH VÀ GNN

TIỂU LUẬN CHUYÊN NGÀNH

GIÁO VIÊN HƯỚNG DẪN

THS. TRẦN TRỌNG BÌNH

KHÓA 2021 – 2025

ĐH SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CNTT
****

XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh Phúc
****

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên 1: Đỗ Ngọc Hân MSSV 1: 21133030

Họ và tên Sinh viên 2: Trần Thị Ngọc Trang MSSV 2: 21133109

Chuyên ngành: Kỹ thuật dữ liệu

Tên đề tài: Tìm hiểu hệ thống gợi ý sử dụng kỹ thuật khai phá tập hữu ích và GNN

Họ và tên giáo viên hướng dẫn: ThS. Trần Trọng Bình

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm

4. Đề nghị cho bảo vệ hay không ?
5. Đánh giá loại :
6. Điểm :

Tp. *Hồ Chí Minh, ngày 8 tháng 12 năm 2024*
Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

ĐH SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CNTT
****

XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh Phúc
****

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên Sinh viên 1: Đỗ Ngọc Hân MSSV 1: 21133030

Họ và tên Sinh viên 2: Trần Thị Ngọc Trang MSSV 2: 21133109

Chuyên ngành: Kỹ thuật dữ liệu

Tên đề tài: Tìm hiểu hệ thống gợi ý sử dụng kỹ thuật khai phá tập hữu ích và GNN

Họ và tên giáo viên phản biện: TS. Nguyễn Thanh Tuấn

NHẬN XÉT

7. Về nội dung đề tài & khối lượng thực hiện:

8. Ưu điểm:

9. Khuyết điểm

10. Đề nghị cho bảo vệ hay không ?
11. Đánh giá loại :
12. Điểm :

Tp. *Hồ Chí Minh, ngày 8 tháng 12 năm 2024*
Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

LỜI CẢM ƠN

Đầu tiên, có được một môi trường học tập chất lượng và hiệu quả để chúng em có điều kiện tốt nhất trong thời gian thực hiện nghiên cứu đề tài, chúng em xin gửi lời cảm ơn đến Ban giám hiệu trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh.

Ngoài ra, chúng em xin gửi lời cảm ơn chân thành đến Ban chủ nhiệm khoa Công nghệ Thông tin và các thầy cô khoa Công nghệ thông tin đã tận tâm chỉ dạy chúng em trong suốt những năm tháng đại học vừa qua, để chúng em có cơ sở cũng như những kiến thức cần thiết để thực hiện tốt đề tài trong suốt quá trình học tập và làm việc tại trường.

Cuối cùng, nhóm em xin chân thành gửi lời cảm ơn với thầy Trần Trọng Bình – Giảng viên phụ trách hướng dẫn khóa luận tốt nghiệp – Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh.

Nhóm rất mong nhận được sự góp ý từ Thầy nhằm rút ra những kinh nghiệm quý báu và hoàn thiện vốn kiến thức để nhóm có thể hoàn thành những nghiên cứu, dự án khác trong tương lai.

Nhóm chúng em xin chân thành cảm ơn!

KẾ HOẠCH THỰC HIỆN

STT	Thời gian	Công việc	Ghi chú
1	01/07 - 15/07	Tìm hiểu về tập hữu ích cao (HUIs) và một số thuật toán để khai thác tập hữu ích cao.	
2	16/07 - 31/07	Áp dụng thuật toán cho tập dữ liệu EIHI.	
3	01/08 - 15/08	Tìm hiểu hệ thống gợi ý.	
4	16/08 - 30/08	Áp dụng phương pháp lọc dựa trên nội dung và lọc cộng tác.	
5	01/9 - 01/9	Tìm hiểu và áp dụng GNN.	
6	1/10 - 20/11	Tìm hiểu tích hợp HUI và GNN vào hệ thống gợi ý. Bổ sung các lý thuyết vào báo cáo.	
7	20/11 - 30/11	Kiểm thử toàn bộ hệ thống. Tiếp tục chỉnh sửa báo cáo. Làm powerpoint cho báo cáo	
8	01/12 - 8/12	Hoàn thiện báo cáo khóa tiểu luận chuyên ngành. Chuẩn bị tài liệu và trình bày cho buổi bảo vệ tiểu luận.	

Ngày … tháng … năm 2024

Ý kiến của giáo viên hướng dẫn

Người viết đề cương

Đỗ Ngọc Hân

Trần Thị Ngọc Trang

DANH MỤC HÌNH

DANH MỤC BẢNG

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Nguyên nghĩa
1	RS	Recommender System
2	GNN	Graph Neural Networks
3	CBF	Content-Based Filtering
4	CF	Collaborative Filtering
5	GCN	Graph Convolutional Network
6	HUI	High Utility Mining
7	HUIM	High utility itemset mining
8	PLCA	Probabilistic Latent Semantic Analysis
9	IMP-GCN	Interest-aware Message-Passing GCN
10	HS-GCN	Mạng tích chập đồ thị không gian Hamming
11	LLM	Large Language Model
12	TWU	Transaction Work Unit
13	IHUP	Incremental High Utility Pattern
14	IHUPL	IHUP with Lexicographic Tree
15	IHUPTF	IHUP Tree with Frequency
16	IHUPTWU	IHUP Tree with Transaction Weighted Utilization
17	EUCS	Cấu trúc ràng buộc nâng cao
18	SVD	Singular Value Decomposition

PHẦN MỞ ĐẦU

1.1. Tính cấp thiết của đề tài

Trong thời đại số hóa phát triển mạnh mẽ, việc truy cập thông tin một cách chính xác và hiệu quả trở thành một thách thức lớn. Với sự gia tăng nhanh chóng của dữ liệu trên internet, các hệ thống gợi ý đóng vai trò quan trọng trong việc giúp người dùng tìm kiếm và khám phá thông tin phù hợp. Việc kết hợp các thuật toán khai phá tập hữu ích và mạng nơ-ron đồ thị để phát triển các hệ thống gợi ý thông minh không chỉ mở ra những hướng đi mới trong lĩnh vực trí tuệ nhân tạo mà còn góp phần nâng cao trải nghiệm người dùng.

Khai phá tập hữu ích là một kỹ thuật quan trọng trong khai thác dữ liệu, giúp xác định các tập hợp mục có giá trị cao trong cơ sở dữ liệu lớn. Trong khi đó, GNN nổi bật trong khả năng xử lý dữ liệu có cấu trúc đồ thị, cho phép mô hình hóa mối quan hệ phức tạp giữa các đối tượng. Sự kết hợp giữa hai phương pháp này hứa hẹn mang đến những cải tiến đáng kể trong độ chính xác và hiệu quả của hệ thống gợi ý, giúp người dùng tiếp cận thông tin một cách toàn diện và nhanh chóng hơn.

1.2. Mục đích của đề tài

Mục tiêu chính của nghiên cứu này là nâng cao độ chính xác của hệ thống gợi ý bằng cách sử dụng các thuật toán khai phá tập hữu ích kết hợp với mạng nơ-ron đồ thị. Điều này bao gồm việc tối ưu hóa các thuật toán để khai thác và phân tích sâu hơn các tập dữ liệu lớn và phức tạp, nhằm đề xuất những gợi ý chính xác và cá nhân hóa hơn cho người dùng.

Ngoài ra, nghiên cứu này còn nhằm giải quyết vấn đề "hạn chế tầm nhìn" thông tin, giúp người dùng không chỉ tiếp cận các nội dung dựa trên sở thích hiện tại mà còn mở rộng phạm vi khám phá của họ. Điều này sẽ thúc đẩy khả năng tiếp cận thông tin đa dạng và phong phú, từ đó kích thích sự sáng tạo và tư duy mở.

Nghiên cứu cũng đặt mục tiêu tối ưu hóa giá trị kinh tế và hiệu quả hoạt động của các doanh nghiệp bằng cách phát triển hệ thống gợi ý thông minh, mang lại lợi ích đồng thời cho cả người dùng và doanh nghiệp trong môi trường số hóa ngày nay.

1.3. Cách tiếp cận và phương pháp nghiên cứu

1.3.1. Đối tượng nghiên cứu

Đối tượng nghiên cứu là hệ thống gợi ý bằng các phương pháp gợi ý dựa trên nội dung (CBF) và thuật toán khai phá tập hữu ích (HUI), so sánh phương pháp lai ghép sử dụng CBF - HUI với phương pháp gợi ý sử dụng Mạng Nơ-ron đồ thị (GNN).

1.3.2. Phạm vi nghiên cứu

Phạm vi nghiên cứu là việc ứng dụng của hệ thống gợi ý sử dụng CBF, HUI, GNN cho một số lĩnh vực như: Hệ thống thương mại điện tử sử dụng CBF - HUI với mục tiêu xác định những sản phẩm không chỉ đáp ứng nhu cầu người dùng và có khả năng mang lại lợi nhuận cao cho doanh nghiệp; Hệ thống gợi ý bài báo khoa học dựa trên dữ liệu dữ liệu trích dẫn.

1.4. Phương pháp nghiên cứu

Phân tích tài liệu: Thu thập và phân tích các tài liệu liên quan đến hệ thống gợi ý và các thuật toán khai phá tập hữu ích, mạng Nơ-ron đồ thị.

Thực nghiệm: Thực hiện các thử nghiệm với các thuật toán gợi ý dựa trên nội dung, khai phá tập hữu ích và mạng GNN trên các bộ dữ liệu có sẵn.

Đánh giá và phân tích: So sánh kết quả giữa các thuật toán khác nhau nhằm tìm ra phương pháp tối ưu nhất cho hệ thống gợi ý.

1.5. Kết quả dự kiến đạt được

Xây dựng được hệ thống gợi ý dựa trên nội dung với hiệu suất cao, có khả năng gợi ý chính xác và phù hợp cho người dùng.

Tích hợp các thuật toán khai phá tập hữu ích vào hệ thống gợi ý để tối ưu hóa quá trình gợi ý sản phẩm có giá trị cao cho người dùng.

Ứng dụng GNN trong việc phân tích mối quan hệ giữa người dùng và sản phẩm, nhằm cải thiện hiệu quả của hệ thống gợi ý.

1.6. Tổng quan tình hình nghiên cứu trong và ngoài nước

Hệ thống gợi ý đã trở thành một trong những công cụ không thể thiếu trong nhiều lĩnh vực, từ thương mại điện tử đến nghiên cứu khoa học. Được phát triển với mục tiêu đưa

ra các đề xuất phù hợp cho người dùng, RS hiện nay được phân chia thành ba hướng chính: dựa trên nội dung, lọc cộng tác và phương pháp lai (Hybrid). Mỗi hướng đều có những đặc điểm riêng và đã thu hút sự chú ý của nhiều nhà nghiên cứu.

Lọc nội dung là phương pháp được nghiên cứu sâu rộng nhất và đã chứng minh được hiệu quả trong nhiều lĩnh vực. Phương pháp này tập trung vào việc phân tích các đặc trưng của các đối tượng (ví dụ như bài báo, sản phẩm) và so sánh chúng với sở thích của người dùng. Một ví dụ nổi bật là nghiên cứu của Hong và cộng sự, nơi họ xây dựng một hệ thống đề xuất bài báo nghiên cứu dựa trên hồ sơ người dùng, được hình thành từ các từ khóa trích xuất từ nội dung bài viết [1]. Độ tương đồng cosin được sử dụng để đánh giá mức độ liên quan giữa các chủ đề và các bài báo đã thu thập, giúp đề xuất chính xác hơn. Ngoài ra, nghiên cứu của Magara và cộng sự còn áp dụng Mạng thông tin liên kết đôi (BisoNet), các biện pháp TF-IDF làm trọng số và các thuật ngữ lọc để kết nối và khai thác thông tin từ hai không gian hoặc miền thông tin khác nhau, giúp phát hiện những bài báo nghiên cứu tình cờ mà người dùng có thể không ngờ tới[2].

Trong khi đó, lọc cộng tác chủ yếu dựa trên sự tương tác của người dùng và thường được phân chia thành hai nhóm: dựa trên mô hình và dựa trên bộ nhớ. Các thuật toán như phân tích ma trận, đặc biệt là SVD, đã được áp dụng để dự đoán sở thích của người dùng dựa trên các dữ liệu từ những người dùng tương tự. Ha và cộng sự đã áp dụng SVD để phát triển một hệ thống gợi ý có khả năng dự đoán và giới thiệu các bài báo nghiên cứu mới xuất bản, ngay cả khi chúng chưa được trích dẫn, giúp nâng cao giá trị của hệ thống gợi ý trong môi trường học thuật[3].

Phương pháp lai là sự kết hợp giữa CB và CF, nhằm tận dụng lợi thế của cả hai phương pháp để cải thiện hiệu quả gợi ý. Liu và cộng sự đã nghiên cứu một phương pháp đề xuất trích dẫn bài báo dựa trên ngữ cảnh, sử dụng khai thác liên kết để hiểu rõ hơn về các ngữ cảnh trích dẫn và cải thiện đề xuất[4]. Điều này đặc biệt hữu ích trong việc đề xuất các bài báo phù hợp với nghiên cứu của người dùng.

Mặc dù các phương pháp đề xuất trước đây đã tập trung vào việc tối ưu hóa độ chính xác và cá nhân hóa gợi ý, một khía cạnh quan trọng vẫn chưa được chú trọng đúng mức là độ hữu ích. Để khắc phục, Mahak Dhanda và Vijay Verma đã phát triển hệ thống gợi ý bài

nghiên cứu có khả năng xử lý dữ liệu động[5]. Hệ thống này hoạt động qua hai giai đoạn: lọc bài báo phù hợp với sở thích người dùng (PLCA) và cá nhân hóa thông qua thuật toán EIHI[6]. EIHI giúp khai thác các tập mục có độ hữu ích cao và xử lý dữ liệu thay đổi liên tục mà không cần quét lại toàn bộ.

Bên cạnh các phương pháp truyền thống, sự phát triển của mạng nơ-ron sâu và các mô hình GNN đã mở ra những hướng đi mới trong hệ thống gợi ý. Một ví dụ tiêu biểu là nghiên cứu của Rex Ying và cộng sự, họ đã phát triển Mạng tích chập đồ thị GCN để xây dựng các biểu diễn nút dựa trên cả cấu trúc đồ thị và đặc điểm của các đối tượng[7]. Mô hình này được triển khai tại Pinterest với tên gọi PinSage, xử lý thành công đồ thị với 3 tỷ nút và 18 tỷ cạnh, đạt được cải thiện đáng kể về mức độ tương tác của người dùng. Tiếp nối thành công này, Xhe và cộng sự đã đề xuất một phiên bản đơn giản hóa nhưng mạnh mẽ hơn có tên là LightGCN[8]. Mô hình này học nhúng người dùng và đối tượng bằng cách truyền tải chúng trên biểu đồ tương tác, mang lại hiệu suất cao và giảm thiểu độ phức tạp trong quá trình tính toán.

Tuy nhiên, giống như nhiều mô hình GCN khác, các mô hình đề xuất dựa trên GCN vẫn gặp phải một vấn đề nổi tiếng là hiện tượng làm mịn quá mức. Khi xếp chồng quá nhiều lớp, nhúng nút có xu hướng trở nên giống nhau, không thể phân biệt, dẫn đến suy giảm hiệu suất. Mặc dù LightGCN đã phần nào giải quyết được vấn đề này, các nghiên cứu vẫn cho rằng chúng bỏ qua một yếu tố quan trọng: những người dùng bậc cao không có sở thích chung với một người dùng cụ thể vẫn có thể ảnh hưởng đến quá trình học nhúng trong các hoạt động tích chập đồ thị.

Để khắc phục, H. Liu và cộng sự đã đề xuất mô hình Interest-aware Message-Passing GCN nhằm tích hợp tích chập đồ thị bậc cao bên trong các đồ thị con, bao gồm những người dùng có cùng sở thích và các mục họ đã tương tác[9]. Để xây dựng các đồ thị con, nhóm nghiên cứu đã thiết kế một mô-đun tạo đồ thị con không giám sát, có thể xác định hiệu quả những người dùng có chung sở thích thông qua việc khai thác cả đặc điểm người dùng lẫn cấu trúc đồ thị. Điều này giúp mô hình tránh truyền tải thông tin không chính xác từ những người hàng xóm bậc cao, cải thiện quá trình học nhúng và hiệu suất. Các thử nghiệm trên ba tập dữ liệu quy mô lớn cho thấy IMP-GCN có thể vượt trội hơn đáng kể so với các mô hình đề xuất GCN hiện đại khác. Đến năm 2022, H. Liu phát triển một khuôn

khổ mới với tên gọi Mạng tích chập đồ thị không gian Hamming (HS-GCN). Mô hình này mô hình hóa sự tương đồng Hamming và nhúng nó vào các mã của người dùng và đối tượng, từ đó tăng cường độ chính xác trong việc gợi ý. Các thử nghiệm mở rộng trên ba tập dữ liệu chuẩn công khai cho thấy HS-GCN vượt trội hơn đáng kể so với nhiều mô hình băm hiện đại và đạt hiệu suất tương đương với các mô hình gợi ý dựa trên giá trị thực[10].

Trong những năm gần đây, việc tích hợp các mô hình ngôn ngữ lớn vào hệ thống đề xuất đã thu hút sự quan tâm mạnh mẽ từ cả giới nghiên cứu và thực hành. Mô hình ngôn ngữ lớn, chẳng hạn như GPT-4, đã chứng minh khả năng hiểu và xử lý ngôn ngữ tự nhiên vượt trội, giúp hệ thống đề xuất có thể khai thác dữ liệu ngôn ngữ phong phú như đánh giá của người dùng hay mô tả sản phẩm. Điều này đã mở ra một hướng tiếp cận mới trong việc cải thiện độ chính xác và mức độ phù hợp của các đề xuất. Chẳng hạn trong bài báo "LLM-enhanced deep learning framework for Recommender Systems"[11], Zhicheng Ding và cộng sự đã trình bày một khung mô hình kết hợp LLM và dữ liệu đa phương tiện (multi-modal), giúp hệ thống đề xuất không chỉ xử lý văn bản mà còn hiểu được cả hình ảnh, từ đó cung cấp những đề xuất cá nhân hóa, chính xác hơn. LLM giúp khắc phục những hạn chế của các phương pháp trước đây bằng cách hợp nhất thông tin từ nhiều nguồn vào một không gian tiềm ẩn thống nhất , đơn giản hóa quá trình học cho mô hình và cải thiện khả năng xếp hạng các mặt hàng đề xuất.

Mặc dù đã có nhiều tiến bộ đáng kể trong lĩnh vực hệ thống gợi ý, nhưng các nhà nghiên cứu vẫn đang đối mặt với nhiều thách thức. Một trong những vấn đề chính là làm thế nào để tối ưu hóa tính hữu ích của các gợi ý mà vẫn duy trì được tính cá nhân hóa và độ chính xác. Bên cạnh đó, việc xử lý dữ liệu lớn, dữ liệu động và vấn đề bảo mật thông tin cá nhân người dùng cũng đặt ra nhiều thách thức cho việc phát triển các hệ thống gợi ý trong tương lai.

PHẦN NỘI DUNG

CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG GỢI Ý

Chương này tập trung vào hệ thống gợi ý, một trong những công nghệ cốt lõi trong việc cá nhân hóa trải nghiệm người dùng trên nhiều nền tảng số. RS giúp cung cấp các gợi ý sản phẩm hoặc dịch vụ dựa trên sở thích, hành vi và nhu cầu của người dùng, mang lại giá trị cao trong thương mại điện tử, dịch vụ phát trực tuyến, và các hệ thống thông tin.

Ngoài các thuật toán truyền thống, còn giới thiệu khái niệm về hai phương pháp hữu ích trong việc tìm ra các tập hợp mặt hàng có giá trị cao, dựa trên không chỉ tần suất xuất hiện mà còn xem xét lợi nhuận và mức độ hữu ích của chúng trong việc gợi ý sản phẩm đem lại tối ưu hóa lợi ích của cả người dùng và doanh nghiệp. Ở phần cuối chương sẽ cung cấp cái nhìn tổng quan về GNN, GAT, một kỹ thuật tiên tiến trong việc mô hình hóa các dữ liệu phức tạp có cấu trúc dạng đồ thị. Kết hợp sử dụng nhiều phương pháp sẽ tạo nên một bước đột phá trong việc phát triển các hệ thống gợi ý tiên tiến.

1.1. GIỚI THIỆU

Hệ thống gợi ý (Recommender System) là một loại phần mềm hoặc công nghệ được thiết kế để đề xuất các sản phẩm, dịch vụ, hoặc nội dung mà người dùng có thể quan tâm. Những hệ thống này được áp dụng rộng rãi trong nhiều lĩnh vực khác nhau, bao gồm thương mại điện tử, giải trí trực tuyến, và nhiều hơn nữa.

Các đề xuất có thể được cá nhân hóa hoặc không cá nhân hóa. Hệ thống tạo ra các đề xuất cá nhân hóa, tức là mỗi người dùng sẽ thấy danh sách khác nhau tùy thuộc vào sở thích cá nhân của họ. Ngược lại, một số cửa hàng trực tuyến có thể chỉ hiển thị các mặt hàng bán chạy nhất hoặc các bài viết được đọc nhiều nhất, mà không cá nhân hóa cho từng người dùng.

Tuy nhiên, việc cá nhân hóa đòi hỏi hệ thống phải biết điều cung cấp các đề xuất được cá nhân gì đó về mọi người dùng, như sở thích, lịch sử xem và mua hàng gọi là hồ sơ người dùng. Thông tin này có thể được thu thập một cánh ngầm (dựa trên hành vi người dùng) hoặc rõ ràng (hỏi trực tiếp người dùng về sở thích của họ).

1.2. ỨNG DỤNG

Trong lĩnh vực thương mại điện tử, RS đóng vai trò quan trọng để tối ưu hóa trải nghiệm mua sắm của người dùng. Hệ thống khuyến nghị không chỉ giúp người dùng dễ dàng tìm kiếm sản phẩm mà họ có thể quan tâm mà còn tạo ra các đề xuất cá nhân hóa dựa trên lịch sử mua sắm và sở thích. Điều này không chỉ tăng cơ hội bán hàng mà còn thúc đẩy sự tương tác và tăng trung bình giá trị đơn hàng.Trong lĩnh vực thương mại điện tử, RS đóng vai trò quan trọng để tối ưu hóa trải nghiệm mua sắm của người dùng. Hệ thống khuyến nghị không chỉ giúp người dùng dễ dàng tìm kiếm sản phẩm mà họ có thể quan tâm mà còn tạo ra các đề xuất cá nhân hóa dựa trên lịch sử mua sắm và sở thích. Điều này không chỉ tăng cơ hội bán hàng mà còn thúc đẩy sự tương tác và tăng trung bình giá trị đơn hàng.

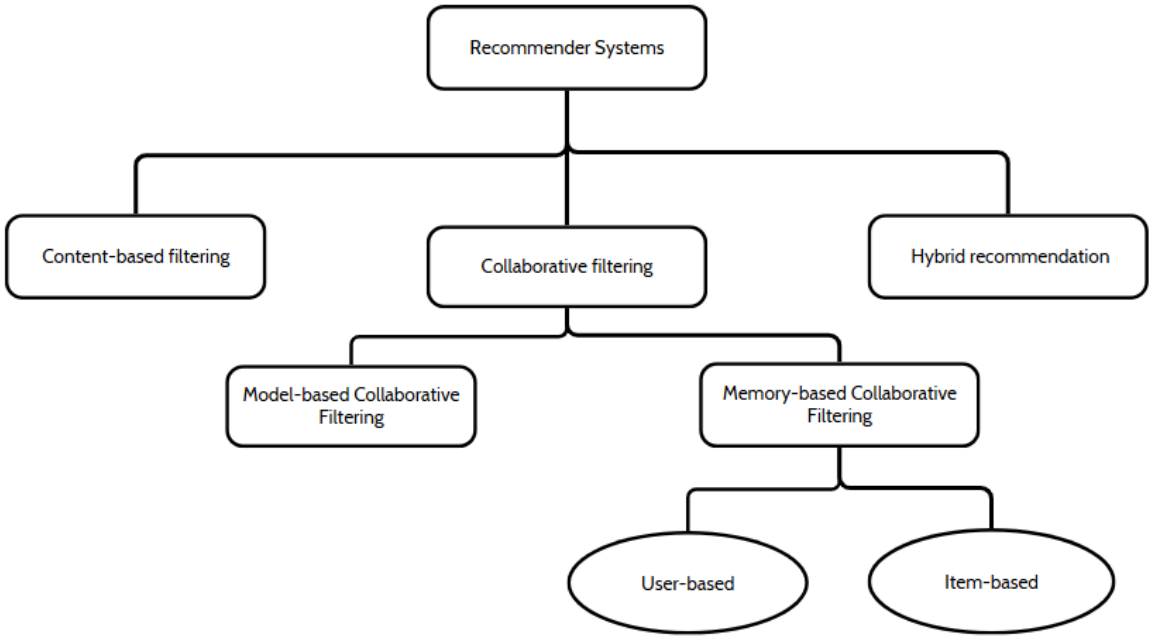
Trong lĩnh vực dịch vụ video trực tuyến, RS chủ yếu tập trung vào việc đề xuất nội dung giải trí phù hợp với sở thích và lịch sử xem của người dùng. Các hệ thống này sử dụng Collaborative Filtering và Content-Based Recommendation System để tạo ra danh sách phim, chương trình truyền hình, hoặc video ngắn mà người dùng có thể thích. Điều này giúp giảm thời gian tìm kiếm và tăng cường sự thỏa mãn của người xem.

Trong môi trường Social Media, RS giúp người dùng khám phá nội dung mới và tương tác với các bài viết, hình ảnh, hoặc video mà họ có thể thích. Hệ thống khuyến nghị sử dụng thông tin về mối quan hệ xã hội, sở thích cá nhân, và hoạt động trước đó để tạo ra dòng thời gian cá nhân hóa. Điều này cũng có thể tăng sự tương tác và giữ chân người dùng trên nền tảng.

Các ứng dụng thực tế của RS không chỉ mang lại lợi ích cho doanh nghiệp mà còn tạo ra trải nghiệm người dùng tốt hơn. Việc cá nhân hóa và đề xuất chính xác giúp tăng cường sự hài lòng của người dùng và có thể thúc đẩy sự trung thành và tương tác liên tục trên các nền tảng khác nhau.

1.3. CÁC KỸ THUẬT CHÍNH TRONG HỆ THỐNG GỢI Ý

Cách tiếp cận hệ thống gợi ý có thể được phân loại thành: lọc cộng tác, lọc theo nội dung và phương pháp kết hợp.



Hình 1. Sơ đồ biểu diễn các thuật toán chính trong hệ thống gợi ý

1.3.1. Collaborative filtering

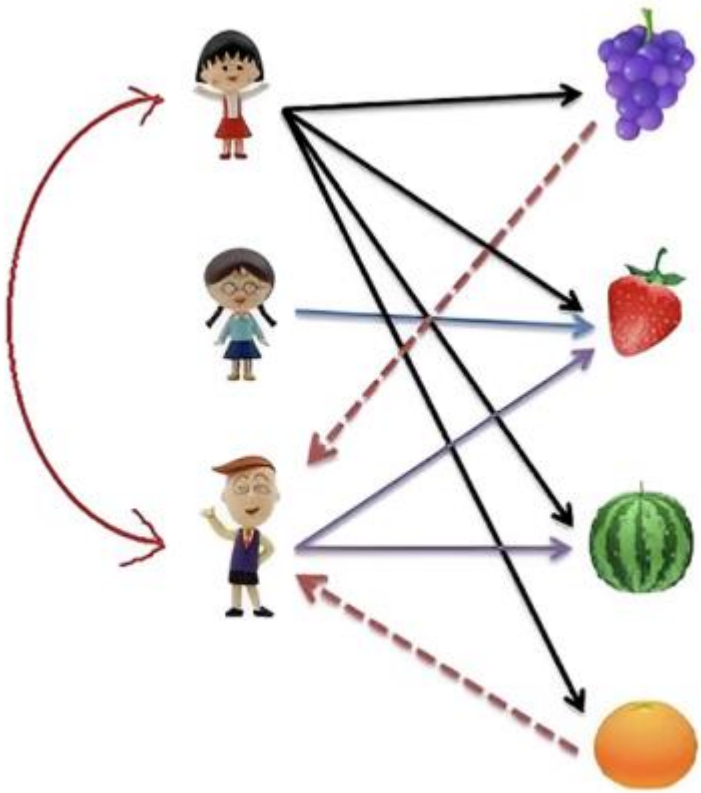
Ý tưởng chính của các phương pháp lọc cộng tác là khai thác thông tin về hành vi trong quá khứ hoặc ý kiến của cộng đồng người dùng hiện tại để dự đoán mục nào mà người dùng hiện tại của hệ thống có thể sẽ thích hoặc quan tâm nhất.

Phương pháp lọc cộng tác chia thành hai hướng tiếp cận chính: dựa trên bộ nhớ (memory-based) và dựa trên mô hình (model-based).

1.3.1.1. Memory-Based Collaborative Filtering

User-Based Collaborative Filtering

Lọc cộng tác dựa trên người dùng (CF) là một kỹ thuật được sử dụng để dự đoán các mục mà người dùng có thể thích dựa trên xếp hạng của những người dùng khác có cùng sở thích với người dùng mục tiêu dành cho mục đó.



Hình 2. Hình mô tả chung về ý tưởng thuật toán User based

Các bước thực hiện Lọc cộng tác dựa trên người dùng:

B1: Tìm điểm tương đồng của người dùng với người dùng mục tiêu

Để tính độ tương đồng giữa hai người dùng u và v, ta sử dụng hệ số tương quan Pearson [12]. Gọi $U = u_1, \cdots, u_m$ là các user, $I = i_1, \cdots, i_n$ là các item, và $r_{i,j} \in R_{m \times n}, i \in \{1, \cdots, m\}, j \in \{1, \cdots, n\}$) là rating của user i cho item j, ($R_{m \times n}$: rating matrix). Công thức tính độ tương đồng $\text{sim}(u, v)$ giữa người dùng u và v như sau:

$$\text{sim}(u,v) = \frac{\sum_{i \in I} \left\{ (r\}_{u,i} - \underline{r_u} \right) \left\{ * \, (r\}_{v,i} - \underline{r_v} \right)}{\sqrt{\sum_{i \in I} \left\{ (r\}_{u,i} - \underline{r_u} \right)^2} \sqrt{\sum_{i \in I} \left\{ (r\}_{v,i} - \underline{r_v} \right)^2}}$$

Trong đó:

- u và v là hai người dùng
- $r_{u,i}$, $r_{v,i}$ là xếp hạng của người dùng u và v đối với mặt hàng i

- $\underline{r_u}, \underline{r_v}$ là rating trung bình của user u và v
- $N(u)$ là tập những user tương tự với user u
- $R(u)$ là tập những rating của user u

Bước 2: Chọn user gần nhất

Chọn những user có độ tương đồng cao nhất

Bước 3: Dự đoán xếp hạng của user u cho item i

Sử dụng hệ số tương quan Pearson và xếp hạng của các item gần nhất

$$p_{u,i} = \underline{r_u} + \frac{\sum_{v \in N} sim(u,v) * \left\{(r\}_{v,i} - \underline{r_v}\right)}{\sum_{v \in N(u)} sim(u,v)}$$

Bước 4: Tính toán các dự đoán xếp hạng cho tất cả các item còn thiếu

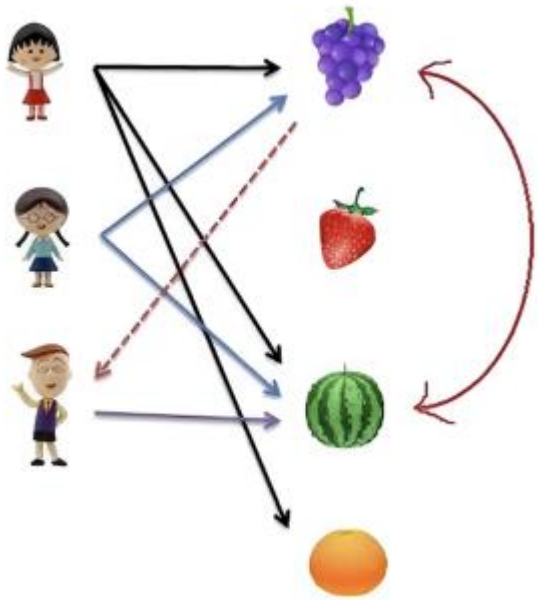
Dự đoán xếp hạng cho tất cả các sản phẩm mà người dùng mục tiêu chưa đánh giá, dựa trên các xếp hạng của người dùng tương tự.

Bước 5: Gợi ý item cho user

Chọn các sản phẩm có dự đoán xếp hạng cao nhất để đưa vào danh sách gợi ý cho người dùng mục tiêu.

Item-Based Collaborative Filtering

Lọc cộng tác dựa trên sản phẩm (Item-Based Collaborative Filtering) là phương pháp dựa trên sự tương đồng giữa các sản phẩm để đưa ra gợi ý. Thay vì dựa vào mối quan hệ giữa người dùng, phương pháp này so sánh các sản phẩm mà người dùng đã đánh giá với những sản phẩm tương tự và từ đó đưa ra gợi ý.



Hình 3. Hình mô tả chung về ý tưởng thuật toán Item based

Các bước thực hiện Lọc cộng tác dựa trên sản phẩm:

Bước 1: Tìm điểm tương đồng của tất cả các sản phẩm

Tính độ tương đồng giữa các sản phẩm, có thể sử dụng hệ số tương quan Pearson hoặc cosine similarity để đo lường sự tương đồng giữa các sản phẩm dựa trên các xếp hạng của người dùng.

Độ tương tự của hai item i và j được tính bởi công thức sau

$$\text{sim}(i,j) = \frac{\sum_{u \in U} r_{u,i} \times r_{u,j}}{\sqrt{\sum_{u \in U} {r_{u,i}}^2} \sqrt{\sum_{u \in U} {r_{u,j}}^2}}$$

Trong đó :

- u là người dùng
- $r_{u,i}$, $r_{u,j}$ là xếp hạng của người dùng u đối với mặt hàng i và j

Bước 2: Chọn các sản phẩm lân cận

Chọn những sản phẩm có độ tương đồng cao nhất với sản phẩm mà người dùng đã đánh giá hoặc yêu thích.

Bước 3: Dự đoán đánh giá của người dùng cho sản phẩm mới

Dự đoán xếp hạng của người dùng cho các sản phẩm mới dựa trên các sản phẩm tương tự mà họ đã đánh giá.

Dự đoán rating của user u cho item i được tính như sau:

$$p_{u,i} = \frac{\sum_{j \in R(u)} sim(i,j) * r_{u,j}}{\sum_{j \in R(u)} |sim(i,j)|}$$

Bước 4: Tính toán các dự đoán xếp hạng cho tất cả các sản phẩm còn thiếu

Dự đoán điểm xếp hạng cho tất cả các sản phẩm mà người dùng chưa đánh giá, dựa trên xếp hạng của các sản phẩm tương tự.

Bước 5: Gợi ý sản phẩm cho người dùng

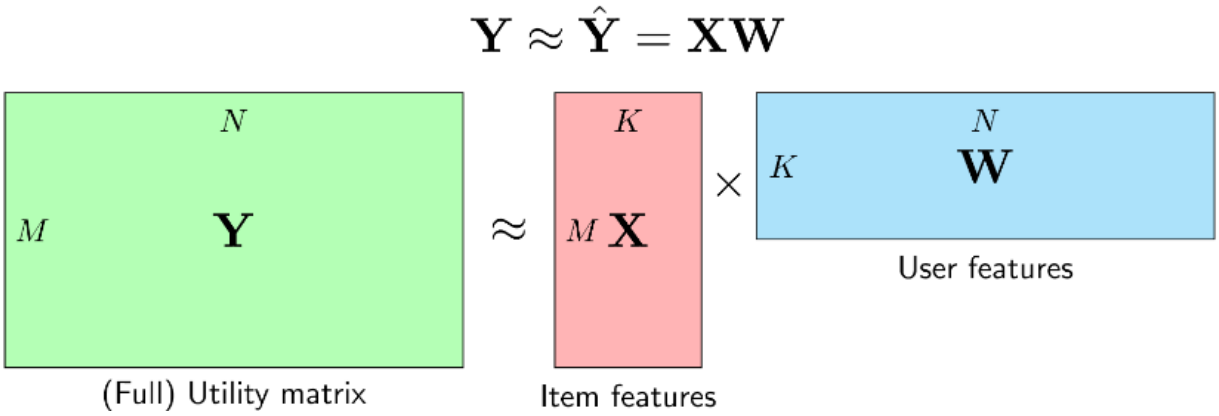
Gợi ý các sản phẩm có dự đoán xếp hạng cao nhất cho người dùng.

1.3.1.2. Model-based Collaborative Filtering

Các kỹ thuật đề xuất hợp tác thường được phân loại là dựa trên bộ nhớ hoặc dựa trên mô hình. Kỹ thuật dựa trên người dùng truyền thống được cho là dựa trên bộ nhớ vì cơ sở dữ liệu xếp hạng ban đầu được giữ trong bộ nhớ và được sử dụng trực tiếp để tạo các đề xuất. Mặt khác, trong các phương pháp tiếp cận dựa trên mô hình, dữ liệu thô trước tiên được xử lý ngoại tuyến, như được mô tả để lọc dựa trên mặt hàng hoặc một số kỹ thuật giảm kích thước. Tại thời điểm chạy, chỉ cần mô hình được tính toán trước hoặc "đã học" để đưa ra dự đoán.

Matrix factorization

Phân tích ma trận thành nhân tử là phương pháp tìm các latent features (tính chất ẩn) mô tả giữa các item và user. Tức là chúng ta sẽ phân tích ma trận tiện ích thành tích của các ma trận items và ma trận users[12].



Hình 4. Hình mô tả Matrix Factorization

Trong đó:

- M, N lần lượt là số lượng của item và user
- K là yếu tố tiềm ẩn, K<< min(M, N)
- X ∈ R^{MxK} là ma trận của toàn bộ item profile
- W ∈ R^{KxN} là ma trận của toàn bộ user model

Singular Value Decomposition (SVD)

SVD là một kỹ thuật phổ biến dùng để giảm số chiều (dimensionality reduction) của

dữ liệu. Dạng tổng quát là phân tích ma trận A có kích thước *mxn* thành ba ma trận sao cho:

$$\mathbf{A=U\Sigma}V^T$$

- U, V: là ma trận trực giao biểu diễn đặc trưng của người dùng và sản phẩm (U∈R^{mxm}, V∈R^{nxn})
- Σ: Là ma trận đường chéo (Σ∈R^{mxn})

Để dự đoán đánh giá của người dùng i với sản phẩm x

$$r_{i,x} = q_i \cdot p_x$$

q_i : dòng thứ i trong ma trận U, tương ứng đặc trưng của người dùng i

p_x : dòng thứ x trong ma trận VT, tương ứng đặc trưng của sản phẩm x

Kỹ thuật SVD giúp giảm số chiều dữ liệu, giữ lại những thông tin quan trọng nhất, và được sử dụng rộng rãi trong các hệ thống gợi ý nhằm giảm độ phức tạp khi xử lý các ma trận lớn.

1.3.2. Content-based filtering

Mặc dù lọc cộng tác có lợi thế trong việc tiết kiệm tài nguyên, không cần phải phân tích chi tiết từng mục, nhưng nó lại phụ thuộc quá nhiều vào đánh giá của người dùng, mà bỏ qua các yếu tố như đặc điểm của sản phẩm hay sở thích cá nhân. Hãy tưởng tượng chúng ta muốn giới thiệu cuốn sách mới của Harry Potter cho Alice. Nếu chúng ta biết rằng (a) cuốn sách này thuộc thể loại tiểu thuyết giả tưởng và (b) Alice vốn luôn yêu thích thể loại này, thì hệ thống đề xuất có thể đưa ra gợi ý phù hợp nếu có hai thông tin: mô tả về sản phẩm và hồ sơ sở thích của Alice, thể hiện những thể loại cô ấy đã từng yêu thích. Quá trình này, thường được gọi là đề xuất dựa trên nội dung, giúp hệ thống xác định các sản phẩm phù hợp nhất với sở thích cá nhân của người dùng. Mặc dù cách tiếp cận này đòi hỏi thêm thông tin về sản phẩm và sở thích, nhưng lại không cần có một cộng đồng người dùng lớn hay dữ liệu đánh giá phong phú. Điều này có nghĩa là hệ thống vẫn có thể tạo ra danh sách đề xuất ngay cả khi chỉ có một người dùng duy nhất[13].

Content Representation and Content Similarity

Cách đơn giản nhất để biểu diễn các mục trong một danh mục là tạo danh sách các đặc điểm rõ ràng cho từng mặt hàng (thường được gọi là thuộc tính, đặc điểm hoặc cấu hình mặt hàng). Ví dụ, trong hệ thống đề xuất sách, có thể sử dụng các thông tin như thể loại, tên tác giả, nhà xuất bản, giá bán, hoặc bất kỳ yếu tố mô tả nào khác để lưu trữ. Khi sở thích của người dùng được mô tả bằng cách sử dụng cùng tập hợp đặc điểm này, nhiệm vụ của hệ thống đề xuất là tìm các mặt hàng có đặc điểm phù hợp với sở thích cá nhân của người dùng.

Để thực hiện việc này, hệ thống đề xuất dựa trên nội dung thường đánh giá mức độ "tương tự" giữa một mục chưa từng được thấy với các mục mà người dùng đã yêu thích trong quá khứ. Có nhiều cách để đo lường sự tương đồng này. Ví dụ, với một cuốn sách chưa từng thấy trước đây, hệ thống có thể chỉ cần kiểm tra xem thể loại của cuốn sách có trùng với các thể loại mà Alice yêu thích hay không. Trong trường hợp này, độ tương đồng có thể được đo lường dưới dạng nhị phân, 0 hoặc 1. Một cách khác là tính toán mức độ chồng chéo giữa các từ khóa liên quan của các cuốn sách. Đây là một phương pháp điển hình để đo lường sự tương đồng trong các đặc điểm có nhiều giá trị.

Cách tiếp cận này giúp hệ thống xác định những mặt hàng có khả năng phù hợp cao với sở thích của người dùng, dựa trên thông tin chi tiết về cả sản phẩm và thói quen của họ.

The vector space model and TF-IDF

Ý tưởng chính là sử dụng danh sách các từ khóa có liên quan suất hiện trong tài liệu. Được tạo ra tự động từ chính nội dung của tài liệu hoặc mô tả văn bản tự do. Nội dung của văn bản có thể được mã hóa trong một danh sách từ khóa.

Trong cách tiếp cận vector space model, người ta thiết lập một danh sách tất cả các từ xuất hiện trong tất cả các tài liệu và mô tả từng tài liệu bằng vectơ Boolean, trong đó 1 chỉ ra rằng một từ xuất hiện trong tài liệu và 0 rằng từ đó không xuất hiện. Nếu hồ sơ người dùng được mô tả bởi một danh sách tương tự (1 biểu thị sở thích trong một từ khóa), việc khớp tài liệu có thể được thực hiện bằng cách đo lường sự chồng chéo của sở thích và nội dung tài liệu.

Tuy nhiên khi phương pháp này không còn phù hợp nữa khi có một lượng lớn user và danh sách các mục. TF-IDF là kỹ thuật được thiết lập từ lĩnh vực truy xuất thông tin và là viết tắt của tần số tài liệu nghịch đảo tần số, phương pháp đo lường mức độ quan trọng của một từ trong tài liệu, so với toàn bộ tập tài liệu.Tài liệu văn bản có thể được mã hóa TF-IDF dưới dạng vectơ trong không gian Euclid đa chiều. Kích thước không gian tương ứng với các từ khóa (còn được gọi là thuật ngữ hoặc mã thông báo) xuất hiện trong tài liệu. Tọa độ của một tài liệu nhất định trong mỗi chiều (tức là cho mỗi thuật ngữ) được tính như một tích của hai biện pháp con: tần số hạn và tần số tài liệu nghịch đảo.

Các bước thực hiện:

B1: Tính TF (Term Frequency): Tần suất xuất hiện của một từ trong một tài liệu.

$$TF(i,j) = \frac{freq(i,j)}{maxOthers(i,j)}$$

Trong đó:

- Tần số thuật ngữ chuẩn hóa TF(i, j) của từ khóa i trong tài liệu j
- Freq(i, j) là số lần xuất hiện tuyệt đối của i trong j
- Tần số tối đa maxOthers(i, j) là max(freq(z, j)), z ∈ OtherKeywords(i, j)
- OtherKeywords(i, j) biểu thị tập hợp các từ khóa khác xuất hiện trong j

B2: IDF (Inverse Document Frequency): Đo lường mức độ phổ biến của từ trong toàn bộ tập tài liệu.

$$IDF(i) = log\left(\frac{N}{n(i)}\right)$$

Trong đó:

- N là số của tất cả các tài liệu được đề xuất

- n (i) là số lượng tài liệu từ N trong đó từ khóa i xuất hiện
- IDF(i) là Tần số tài liệu nghịch đảo cho i

B3: Tính trọng số TF-IDF kết hợp cho một từ khóa i trong tài liệu j là tích của TF và IDF

$$TF\text{-}IDF(i, j) = TF(i, j) * IDF(i)$$

Phương pháp TF-IDF được sử dụng để chuyển đổi các tài liệu thành vector đặc trưng và so sánh sự tương đồng giữa các tài liệu. Ví dụ, một bài báo có thể được chuyển thành vector TF-IDF và so sánh với các bài báo khác để tìm kiếm các tài liệu tương tự.

Similarity-Based Retrieval

Nearest Neighbors

Phương pháp đơn giản nhất là kiểm tra xem người dùng có thích các tài liệu tương tự trong quá khứ không. Để thực hiện yêu cầu 2 thông tin:

- Lịch sử thích hay không của người dùng về các mục trước đó
- Mức độ giống nhau của item

Phương pháp K-Nearest Neighbors là một thuật toán tìm kiếm các đối tượng gần nhất trong không gian vector dựa trên một tiêu chí tương đồng.

Các bước thực hiện:

B1: Tính Toán Khoảng cách hoặc Tương đồng:

Xác định phương pháp đo lường khoảng cách hoặc tương đồng Euclidean distance hoặc cosine similarity

Euclidean Distance:

Bằng cách tính toán Euclidean Distance, chúng ta có thể tìm giá trị khoảng cách giữa hai bộ phim, biểu thị sự tương đồng giữa chúng. Quan sát cho thấy rằng khi khoảng cách giảm, tương đồng tăng lên. Bằng cách này, quá trình đề xuất có thể được thực hiện.

$$\mathrm{d(p,q)} = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

Cosine Similarity:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i\, B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2}\sqrt{\sum_{i=1}^{n} {B_i}^2}}$$

Trong khi Euclidean Distance biểu thi cho khoảng cách, thì khái niệm về tương đồng xuất hiện trong Cosine Similarity. Distance-closeness và similarity-dissimilarity tương ứng với các khái niệm giống nhau trong trường hợp này.

Tính toán khoảng cách hoặc độ tương đồng giữa vector của tài liệu hoặc mục

B2: Tìm K-Nearest Neighbors (K-NN):

Sắp xếp các tài liệu hoặc mục theo khoảng cách hoặc độ tương đồng

Chọn K mục gần nhất (hoặc có độ tương đồng cao nhất) để làm gợi ý

1.3.3. Hybrid recommendation

Hybrid recommendation systems là sự kết hợp của content-based và collaborative filtering. Cơ chế Hybrid để dự đoán đề xuất, như tên gọi của nó, kết hợp 2 hay nhiều kỹ thuật lại về nhau. Những nhược điểm của hệ thống recommend có thể được giải quyết bằng cách sử dụng công nghệ kết hợp và có thể lấy ra từng lợi ích của các công nghệ khác nhau.

- K
ết hợp các dự đoán riêng lẻ được thực hiện bằng các kỹ thuật CB và CF
- P
hát triển một mô hình có sự kết hợp giữa content-base và CF với nhau.

Mô hình hybrid recommendation có thể được chia thành bảy loại: Weighted Hybridization, Switching Hybridization, Cascaded Hybridization, Mixed Hybridization, Feature Combination, Feature Augmentation, and Meta-Level.

Weighted Hybridization

Bên trong hệ thống gợi ý có trọng số, chúng ta sẽ định ra một vài cái model có khả năng xử lý tốt tập dữ liệu. Hệ thống gợi ý có trọng số sẽ lấy kết quả đầu ra từ mỗi mô hình và kết hợp kết quả theo trọng số tĩnh, trọng số này không thay đổi trên toàn bộ tập train và tập test.

Ví dụ: chúng ta có thể kết hợp mô hình CB và mô hình CF và mỗi mục chiếm 50% trọng số đối với dự đoán cuối cùng.

Lợi ích của việc sử dụng kết hợp có trọng số là chúng ta tích hợp nhiều mô hình để hỗ trợ tập dữ liệu về quy trình gợi ý.

Switching Hybridization

Phương pháp này sẽ chọn ra 1 mô hình duy nhất phụ thuộc điều kiện dữ liệu. Nó sẽ chọn một trong hai hệ thống CB và CF dựa trên mục - dựa vào các yếu tố trong tình huống cụ thể. Cách hệ thống lựa chọn mô hình thích hợp có thể dựa trên một số tiêu chí như:

- Thông tin người dùng
- Đặc điểm dữ liệu
- Hiệu xuất mô hình
- Chi phí tính toán
- Mục tiêu hệ thống

Cascaded Hybridization

Cách tiếp cận kết hợp hỗn hợp đầu tiên sẽ sử dụng dataset và các đặc điểm của nó để tạo ra các tập dữ liệu đa dạng. Hệ thống đề xuất sau đó sẽ phân chia các data này vào các nhóm khác nhau và áp dụng mô hình đề xuất tương ứng. Kết quả dự đoán từ các mô hình này sau đó được kết hợp để tạo ra đề xuất cuối cùng.

Mixed Hybridization

Trong Feature Combination, việc thêm một mô hình đề xuất ảo vào hệ thống mang lại ý nghĩa lớn. Mô hình này hoạt động như một tính năng đối với tập dữ liệu của người dùng ban đầu.

Ví dụ, chúng ta có thể tích hợp các tính năng của một mô hình CF vào mô hình đề xuất theo CB. Mô hình kết hợp này có khả năng xem xét dữ liệu Collaborative data từ hệ thống con chỉ dựa vào một mô hình. Điều này mang lại lợi ích bằng cách cải thiện sự đa dạng và chất lượng của các đề xuất cuối cùng.

Feature Combination

Phương pháp này được sử dụng để tạo ra 1 rating hoặc phân loại dựa theo data, model này sẽ giúp tăng cường tính năng có thể cải thiện hiệu suất của hệ thống.Ví dụ: bằng cách sử dụng quy tắc kết hợp, chúng tôi có thể nâng cao tập dữ liệu người dùng. Với tập dữ liệu được tăng cường, hiệu suất của mô hình đề xuất CB sẽ được cải thiện.

Feature Augmentation

Phương pháp Cascade hybrid tập trung vào việc xây dựng một hệ thống đề xuất có cấu trúc phân cấp nghiêm ngặt. Điều này có nghĩa là hệ thống có hai cấp độ: một hệ thống đề xuất chính và một mô hình phụ.

- Hệ thống đề xuất chính: Đây là cơ sở của hệ thống, tạo ra các đề xuất chính dựa trên dữ liệu và mô hình chính. Nó là trung tâm của quá trình đề xuất.
- Mô hình phụ: Mô hình này được sử dụng để giải quyết các vấn đề nhỏ, như việc điều chỉnh hoặc cải thiện kết quả từ hệ thống đề xuất chính. Ví dụ, nó có thể giúp xử lý các vấn đề về tính điểm không đồng đều hoặc thiếu dữ liệu trong tập dữ liệu.

Meta-Level

Meta-level hybrid tương tự như kết hợp tăng cường tính năng, trong đó một mô hình đóng góp cung cấp tập dữ liệu tăng cường cho mô hình đề xuất chính. Tuy nhiên, khác với mô hình kết hợp tăng cường tính năng, ở cấp độ meta, tập dữ liệu gốc được thay thế bằng một mô hình đã học từ mô hình đóng góp, và mô hình này được sử dụng như đầu vào cho mô hình đề xuất chính.

CHƯƠNG 2: TỔNG QUAN VỀ TẬP HỮU ÍCH CAO VÀ GRAPH NEURAL NETWORKS

Chương này cung cấp phương pháp HUI và mạng nơ-ron đồ thị (GNN) được nhóm đề xuất xây dựng hệ thống gợi ý: tập mục hữu ích cao (HUI) và mạng nơ-ron đồ thị (GNN). Đầu tiên, phần giới thiệu về tập mục hữu ích cao (HUI) sẽ trình bày lý thuyết cơ bản và các phương pháp khai thác HUI trong các hệ thống gợi ý, nhấn mạnh vào vai trò của việc chọn lọc các mục có giá trị cao trong quá trình tối ưu hóa và cải thiện hiệu quả xử lý dữ liệu. Tiếp theo, phần về GNN sẽ giới thiệu các khái niệm cơ bản và nguyên lý hoạt động của mạng nơ-ron đồ thị, một phương pháp học sâu mạnh mẽ trong việc khai thác các đặc trưng đồ thị để thể hiện mối quan hệ phức tạp giữa các đối tượng trong bài toán. HUI-GNN kết hợp sức mạnh của tập mục hữu ích cao (HUI) và mạng nơ-ron đồ thị (GNN) để tối ưu hóa hệ thống gợi ý. Trong đó, HUI giúp doanh nghiệp xác định và khai thác các mục có giá trị cao, giảm thiểu chi phí và tối ưu hóa hiệu quả hệ thống. Tuy nhiên, để đảm bảo rằng trải nghiệm khách hàng không bị ảnh hưởng, GNN được tích hợp để mô hình hóa các mối quan hệ phức tạp giữa sản phẩm và người dùng thông qua các đặc trưng đồ thị. Sự kết hợp này không chỉ mang lại lợi ích cho doanh nghiệp trong việc tối ưu hóa quy trình gợi ý, mà còn duy trì độ chính xác và tính cá nhân hóa cao trong các gợi ý, từ đó nâng cao sự hài lòng và trải nghiệm người dùng.

2.1. TẬP HỮU ÍCH CAO VÀ BÀI TOÁN KHAI PHÁ ĐỘ HỮU ÍCH CAO

Trong những năm 1990, khi lượng dữ liệu thu thập và lưu trữ tăng mạnh, việc phân tích dữ liệu thủ công trở nên ngày càng khó khăn và tốn kém thời gian. Điều này đã thúc đẩy sự phát triển các kỹ thuật tự động nhằm xác định những mẫu dữ liệu thú vị. Một trong những hướng đi đầu tiên của lĩnh vực này là phân tích các mặt hàng được mua bởi khách hàng trong các cửa hàng bán lẻ, nhằm khám phá các mẫu thường xuyên - những nhóm mặt hàng thường xuất hiện cùng nhau trong cơ sở dữ liệu. Chẳng hạn, một mẫu thường xuyên có thể là khách hàng thường mua bánh mì cùng với phô mai. Ngoài bán lẻ, các mẫu thường xuyên cũng xuất hiện trong các lĩnh vực như phân tích văn bản, hệ thống khuyến nghị hoặc nhận diện chuỗi hành động dẫn đến sự cố trong các hệ thống phức tạp. Việc phát hiện các

mẫu này giúp hiểu rõ dữ liệu hơn và hỗ trợ ra quyết định, chẳng hạn như tối ưu hóa chiến lược tiếp thị hoặc cải thiện hiệu quả các hệ thống gợi ý.

Tuy nhiên, tần suất xuất hiện không phải lúc nào cũng là thước đo lý tưởng để xác định các mẫu thực sự hữu ích. Ví dụ, trong bán lẻ, một số sản phẩm tuy được mua thường xuyên nhưng mang lại lợi nhuận thấp, trong khi những mặt hàng ít được mua lại có giá trị kinh tế cao hơn. Để giải quyết vấn đề này, khái niệm tiện ích đã được giới thiệu, định nghĩa như một hàm toán học đo lường tầm quan trọng của các mẫu bằng cách kết hợp số lượng và trọng số. Trong bối cảnh các hệ thống gợi ý, tiện ích có thể được hiểu là giá trị của một sản phẩm, dịch vụ hoặc nội dung gợi ý dựa trên lợi ích mà nó mang lại cho người dùng, thay vì chỉ dựa vào mức độ phổ biến.

Việc khám phá các mẫu tiện ích cao đã chứng tỏ tiềm năng lớn trong việc cải thiện hiệu quả của hệ thống gợi ý. Ví dụ, trong các nền tảng thương mại điện tử, HUI có thể được sử dụng để xác định các sản phẩm không chỉ phổ biến mà còn có khả năng tạo ra doanh thu cao, từ đó cung cấp các gợi ý cá nhân hóa phù hợp với cả người dùng và doanh nghiệp. So với bài toán khai phá mẫu thường xuyên truyền thống, khai phá HUI phức tạp hơn do tính chất chống đơn điệu của tiện ích, khiến các phương pháp truyền thống không thể áp dụng trực tiếp. Tuy nhiên, bằng cách tích hợp HUI vào hệ thống gợi ý, các nền tảng có thể xây dựng các gợi ý thông minh hơn, tối ưu hóa lợi ích kinh doanh và trải nghiệm người dùng, ví dụ như ưu tiên gợi ý các sản phẩm có giá trị cao hoặc nội dung phù hợp với sở thích cá nhân.

2.1.1. Tập phổ biến

Tập hợp I: Cho tập I = {i1, i2, i3, … , in} là tập các sự kiện khác nhau.Cơ sở dữ liệu giao dịch DB: DB = {T1, T2, T3, …,Ttm} là tập hợp các giao dịch, trong đó mỗi giao dịch ti với i ∈ [1, m] và Ti ⊂ I.Một mẫu P ⊂ I được gọi là xuất hiện bên trong giao dịch T nếu và chỉ nếu mọi phần tử của P đều thuộc tập T.

Ví dụ: Nếu P = {a, b, c} và T = {a, b, c, d}, thì P xuất hiện trong T.

Độ hỗ trợ của mẫu P: Là số lần xuất hiện của P trong cơ sở dữ liệu giao dịch DB. Mỗi giao dịch Ti có chứa P được tính là một lần xuất hiện.Độ dài của mẫu P: Được tính bằng

số phần tử có trong P. Một mẫu có độ dài k còn được gọi là tập k-itemsets.Độ hỗ trợ của P được ký hiệu là sup(P). Ngưỡng hỗ trợ tối thiểu φ: Là một ngưỡng do người dùng đặt ra.Bài toán khai thác mẫu phổ biến: Cho cơ sở dữ liệu giao dịch DB và một ngưỡng hỗ trợ tối thiểu φ, yêu cầu tìm ra tất cả các mẫu P ⊂ I sao cho sup(P) ≥ φ.

2.1.2. Tập hữu ích cao

2.1.2.1. Khái niệm tập hữu ích cao

Một tập hợp các giao dịch bao gồm một cơ sở dữ liệu giao dịch $D = \{T_1, T_2, \cdots, T_m\}$. Mỗi giao dịch thuộc về cơ sở dữ liệu D có một mã định danh duy nhất T_{id}. $I = \{i_1, i_2, \cdots, i_n\}$ là một tập hữu hạn các mục không lặp lại n từ D. Đối với mỗi giao dịch *T*, tập mục $X \subseteq I$ là một tập hữu hạn các mục. Tiện ích nội (ví dụ: số lượng mua) và tiện ích ngoại (ví dụ: lợi nhuận đơn vị) đều là các số dương liên kết với từng mục. Một cơ sở dữ liệu giao dịch mẫu được hiển thị trong Bảng 1 có chứa năm giao dịch [14] và cơ sở dữ liệu được này sử dụng làm ví dụ đang chạy. Giao dịch T3 chứa các mục *b, c, d* và *e* với các tiện ích nội lần lượt là 4, 3, 3 và 1. Tiện ích ngoại của tất cả bảy mặt hàng được đưa ra trong Bảng 2.

Bảng 1. Ví dụ cơ sở dữ liệu giao dịch

T_{id}	*Giao dịch*
T_1	*(a, 1)(c, 1)(d, 1)*
T_2	*(a, 2)(c, 6)(e, 2)(g, 5)*
T_3	*(b, 4)(c, 3)(d, 3)(e, 1)*
T_4	*(a, 1)(b, 2)(c, 1)(d, 6)(e, 1)(f, 5)*
T_5	*(b, 2)(c, 2)(e, 1)(g, 2)*

Định nghĩa 1: Tiện ích của một mục/tập mục (Utility of an item/itemset)

Giả sử có một giao dịch T_i, một mục *i* và một tập mục *X*. Tiện ích của *i* trong T_j được ký hiệu là $u(i, T_j)$ và được tính là $p(i) \times q(i, T_j)$, trong đó *p(i)* và $q(i, T_j)$ lần lượt là tiện ích ngoại và tiện ích nội của mục. Tiện ích của *X* trong T_j được định nghĩa là $u(X) = \Sigma_{i \epsilon X} u(i, T_j)$. Tiện ích của *X* được định nghĩa là $u(X) = \Sigma_{T_j \epsilon g(X)} u(X, T_j)$, trong đó *g(X)* là tập hợp các giao dịch chứa *X*.

Ví dụ, tiện ích của mục *b* trong T_4 là $u(b, T_4) = 2 \times 2 = 4$, và tiện ích của tập mục $\{a, c\}$ trong , T_1 là $u(\{a, c\}, T_1) = u(a, T_1) + u(c, T_1) = 5 \times 1 + 1 \times 1 = 6$. Tiện ích của tập mục $\{c, d\}$ là $u(\{c, d\}) = u(\{c, d\}, T_1)) + u(\{c, d\}, T_3)) + u(\{c, d\}, T_4)) = u(c, T_1) + u(d, T_1) + u(c, T_3) + u(d, T_3) + u(c, T_4) + u(d, T_4) = 1 + 2 + 3 + 6 + 1 + 12 = 25$. Độ phức tạp tính toán

của việc tính toán *u(X)* là $O(m * n)$ nói chung. Đối với một số thuật toán HUIM, chẳng hạn như EFIM, một kỹ thuật tìm kiếm nhị phân được sử dụng để cải thiện hiệu quả tìm kiếm cho X, và sau đó độ phức tạp là $O(m * log_2 n)$.

Bảng 2. Tiện ích nội (External utility values)

Item	*a*	*b*	*c*	*d*	*e*	*f*	*g*
Profit	5	2	1	2	3	1	1

Định nghĩa 2: Tập mục tiện ích cao (HUI)

Cho *minutil* là ngưỡng do người dùng chỉ định với giá trị dương. Nếu tiện ích *u(X)* không nhỏ hơn *minutil*, thì *X* là tập mục tiện ích cao; nếu không, thì đó là tập mục tiện ích thấp.

Ví dụ, nếu *minutil* = 32, thì HUI trong cơ sở dữ liệu mẫu là $\{b, c, d\}$, $\{b, d, e\}$ và $\{b, c, d, e\}$ với tiện ích tương ứng là 34, 36 và 40.

Định nghĩa 3: Tiện ích giao dịch (TU) và TWU

Giả sử T_j là một giao dịch và x là một mục. Tiện ích giao dịch của T_j là $TU(T_j) = \Sigma_{x \in T_j} u(x, T_j)$. TWU của x là $TWU(x) = \sum_{\{T_j \in g(x)\}} TU(T_j)$.

Ví dụ, TWU của mục a là $TWU\ [a] = TU[T_1] + TU[T_2] + TU[T_3]$ = 5 + 1 + 2 + 10 + 6 + 6 + 5 + 5 + 4 + 1 + 12 + 3 + 5 = 65. Trong cơ sở dữ liệu giao dịch mẫu, TWU của tất cả các mục được hiển thị trong Bảng 3.

Thuộc tính 1: Cắt tỉa không gian tìm kiếm bằng TWU

Tập mục X và tất cả các siêu tập của nó là các tập mục có tiện ích thấp nếu *TWU* (X) nhỏ hơn *minutil*.

Định nghĩa 4: Tiện ích còn lại (Remaining utility)Giả sử có một tập mục *X*, một giao dịch T_j và một thứ tự tổng thể trên các mục từ I. Tập hợp tất cả các mục sau X trong T_j được ký hiệu là $\{i \in T_j \wedge (i > x, \forall x \in X)\}$. Tiện ích còn lại của X trong T_j là $re(X, T_j) = \sum_{i \in T_j \wedge (i > x, \forall x \in X)} u(i, T_j)$ liên quan đến tổng các tiện ích cho tập $\{i \in T_j \wedge (i > x, \forall x \in X)\}$.

Ví dụ, tiện ích còn lại của itemset $\{a, c\}$ trong T_4 là $re(\{a, c\}, T_4) = u(d, T_4) + u(e, T_4) + u(f, T_4) = 12 + 3 + 5 = 20$.

Định nghĩa 5: Danh sách tiện ích (Utility-list) Giả sử có một itemset *X* và một giao dịch T_j chứa *X*. Danh sách tiện ích của *X* chứa một tập hợp các bộ (T_j, *iutil*, *rutil*) cho mỗi T_j. Ở đây, $iutil = u(X, T_j)$ và $rutil = re(X, Tj)$.

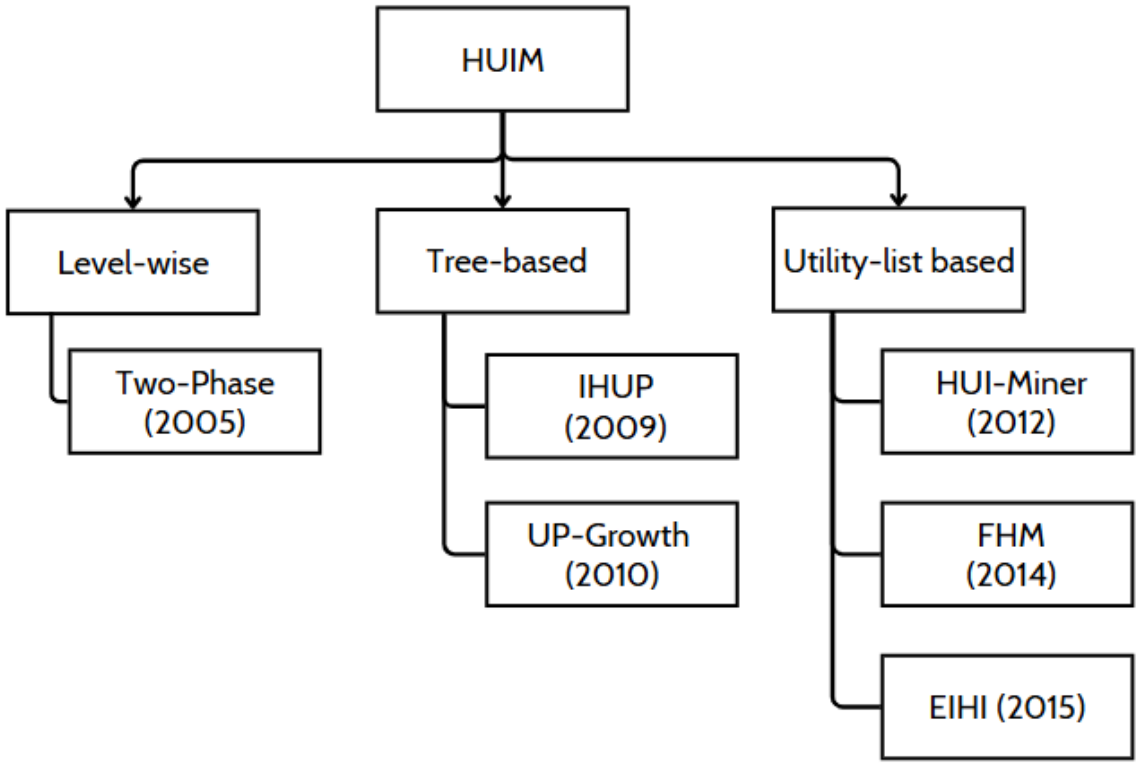
Danh sách tiện ích của $\{a, c\}$ trong cơ sở dữ liệu mẫu là $\{(T_1, 6, 2), (T_2, 16, 11), (T_4, 6, 20)\}$.

Định nghĩa 6: Giới hạn trên tiện ích còn lại (Remaining utility upper bound) Giả sử có một itemset *X* và một mục *i*. Phần mở rộng của *X* có thể được lấy bằng cách thêm *i* vào *X*, thỏa mãn $i > x, \forall x \in X$. Giới hạn trên tiện ích còn lại của X là $reu(X) = u(X) + re(X)$

Ví dụ, giới hạn trên tiện ích còn lại của $\{a, c\}$ là $reu(a, c) = u(a, c) + re(a, c)$= 6 + 2 + 16 + 11 + 6 + 20 = 61

Thuộc tính 2: Cắt bớt không gian tìm kiếm bằng danh sách tiện ích (Pruning search space using utility-lists) Cho X là một tập mục. Nếu $reu(X) < minutil$, thì X và tất cả các phần mở rộng của nó là các tập mục tiện ích thấp, có thể được cắt bớt trong không gian tìm kiếm [15].

Cách tiếp HUIM có rất nhiều phương pháp, nhưng có thể được phân loại thành: cấp độ, dựa trên cây và dựa trên danh sách tiện ích.

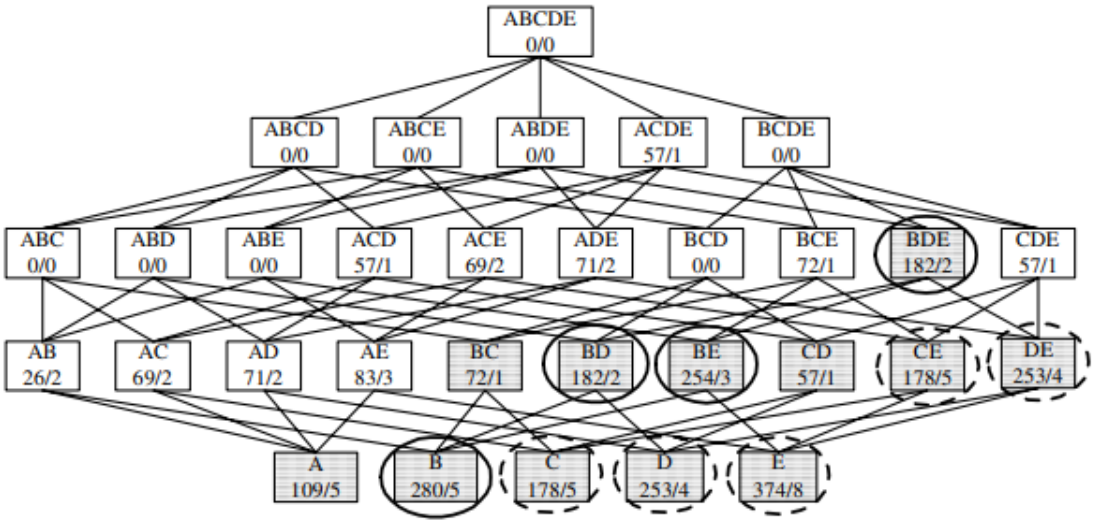


Hình 5. Sơ đồ biểu diễn các thuật toán HUIM

2.1.2.2. Level-wise

Các thuật toán khai thác HUI từ cơ sở dữ liệu giao dịch thường sử dụng thuộc tính TWU để tìm kiếm các tập hợp mục hữu ích cao. Quá trình tìm kiếm có thể được thực hiện theo cấp độ hoặc tìm kiếm theo chiều sâu để khám phá các mục trong cơ sở dữ liệu. Các hạng mục tiện ích thấp sẽ bị cắt tỉa bằng các chiến lược như TWU và tiện ích giới hạn trên. Mục tiêu của các thuật toán này là tìm ra tính hữu ích của các mặt hàng để trích lợi nhuận tối đa cho doanh nghiệp. Thuật toán đáng chú ý như là: thuật toán Hai Pha (Two-Phase). Được đề xuất bởi Liu và cộng sự [16] thực hiện thêm một lần quét cơ sở dữ liệu để cắt tỉa các bộ mục được đánh giá quá cao trong giai đoạn thứ hai.

Thuật toán khái quát hóa thuật toán Apriori. Hai giai đoạn khám phá không gian tìm kiếm của các mục bằng cách sử dụng tìm kiếm đầu tiên theo chiều rộng. Thuật toán tìm kiếm đầu tiên theo chiều rộng trước tiên xem xét các mục đơn lẻ (1-itemsets), Trong ví dụ, là {a}, {b}, {c}, {d} và {e}. Sau đó, Two-Phase tạo ra 2-itemset như {a, b},{a, c},{a, d}, và sau đó là 3-itemsets, v.v., cho đến khi nó tạo ra các itemset lớn nhất {a, b, c, d, e} chứa tất cả các mục. Tuy nhiên, hạn chế của thuật toán Hai Pha là vẫn tạo ra quá nhiều ứng cử viên và yêu cầu quét nhiều lần cơ sở dữ liệu, dẫn đến tiêu tốn nhiều thời gian và bộ nhớ.



Hình 6. Minh họa thuật toán Two-Phase

Các phương pháp tiếp cận khai thác HUI theo cấp độ có ưu điểm là cho phép người dùng dễ dàng biểu thị giá trị tiện ích của các bộ vật phẩm và tìm ra các bộ vật phẩm có giá trị tiện ích cao hơn ngưỡng được chỉ định. Tuy nhiên, chúng thường gặp phải thách thức là

giới hạn kích thước của bộ ứng cử viên và đơn giản hóa việc tính toán tiện ích. Do yêu cầu nhiều lần quét cơ sở dữ liệu, các thuật toán này mất nhiều thời gian thực thi hơn và tiêu thụ bộ nhớ cao, không phù hợp với những cơ sở dữ liệu có kích thước lớn và giao dịch dài.

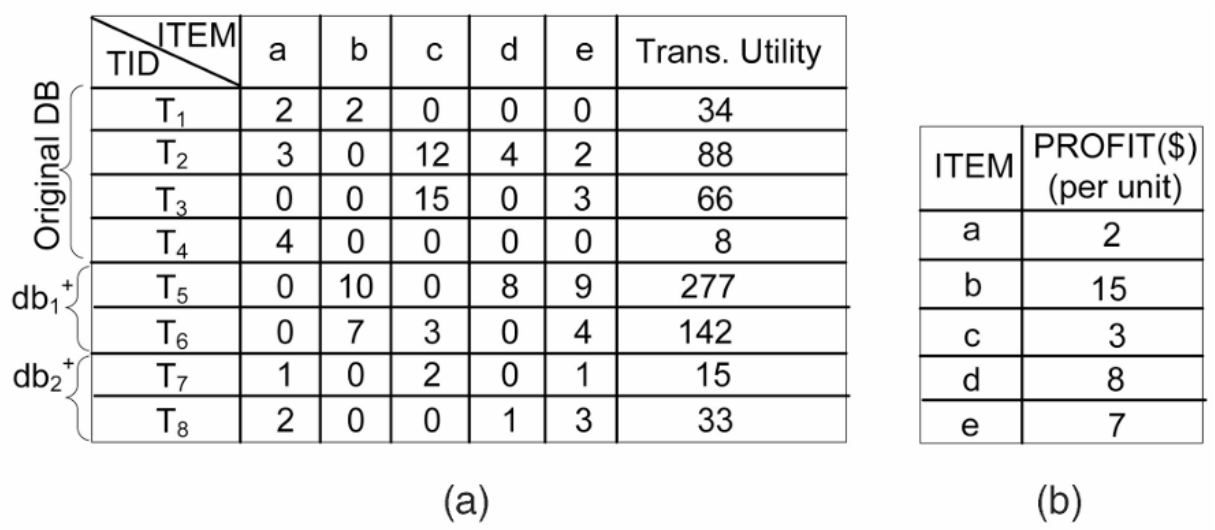
2.1.2.3. Tree-based

Trong phần này, các thuật toán khai thác tập hữu ích cao dựa trên cây được thảo luận nhằm giải quyết các hạn chế của các phương pháp tiếp cận theo cấp độ. Các phương pháp theo cấp độ thường yêu cầu hai giai đoạn để tìm HUI, dẫn đến việc tạo ra nhiều ứng cử viên và quét cơ sở dữ liệu nhiều lần. Để khắc phục vấn đề này, các thuật toán HUI dựa trên cây đã được đề xuất, sử dụng phương pháp tăng trưởng mô hình và cấu trúc cây nhỏ gọn. Các thuật toán này làm giảm đáng kể số lượng ứng cử viên bằng cách áp dụng các chiến lược cắt tỉa hiệu quả và cải thiện không gian tìm kiếm để tìm HUI. Chúng chỉ cần hai hoặc ba lần quét cơ sở dữ liệu, nhanh hơn nhiều so với các phương pháp tiếp cận giống như Apriori. Mục tiêu chính của các thuật toán này là giảm số lượng ứng cử viên và số lần quét cơ sở dữ liệu thông qua các chiến lược cắt tỉa hiệu quả và cấu trúc cây nhỏ gọn.

Incremental High Utility Pattern (IHUP)

Thuật toán IHUP là một phương pháp khai thác các tập hợp mục có lợi ích cao trong các cơ sở dữ liệu cập nhật dần. Mục tiêu của IHUP là khai thác hiệu quả các tập hợp mục hữu ích trong các cơ sở dữ liệu mà dữ liệu mới có thể được thêm vào hoặc dữ liệu cũ có thể bị xóa đi[17].

Có ba cấu trúc cây mới để khai thác mô hình tiện ích cao trong cơ sở dữ liệu gia tăng: IHUP with Lexicographic Tree, IHUP Tree with Frequency và IHUP Tree with Transaction Weighted Utilization.



	TID \ ITEM	a	b	c	d	e	Trans. Utility
Original DB	T_1	2	2	0	0	0	34
	T_2	3	0	12	4	2	88
	T_3	0	0	15	0	3	66
	T_4	4	0	0	0	0	8
db_1^+	T_5	0	10	0	8	9	277
	T_6	0	7	3	0	4	142
db_2^+	T_7	1	0	2	0	1	15
	T_8	2	0	0	1	3	33

(a)

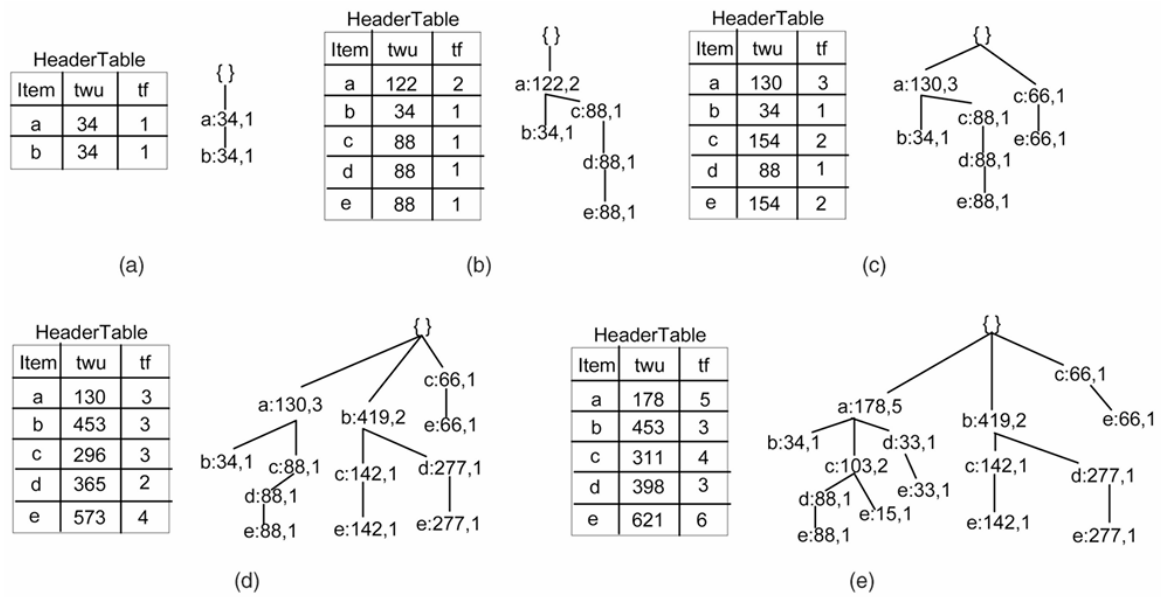
ITEM	PROFIT($) (per unit)
a	2
b	15
c	3
d	8
e	7

(b)

Hình 7. Ví dụ về (a) cơ sở dữ liệu giao dịch và (b) bảng tiện ích.

ĐN8: Tần suất giao dịch (tf) của một ip mục là tf (ip) và đại diện cho số lượng giao dịch mà mục đó xuất hiện. Tần số ban đầu của ip là f(ip), biểu thị số lần xuất hiện thực tế của IP trong các giao dịch đó. Ví dụ: tf(c) = 4 như xuất hiện trong T2, T3, T6; và T7 và f(c) = 12 + 15 + 3 + 2 = 32

IHUP with Lexicographic Tree (IHUPL)

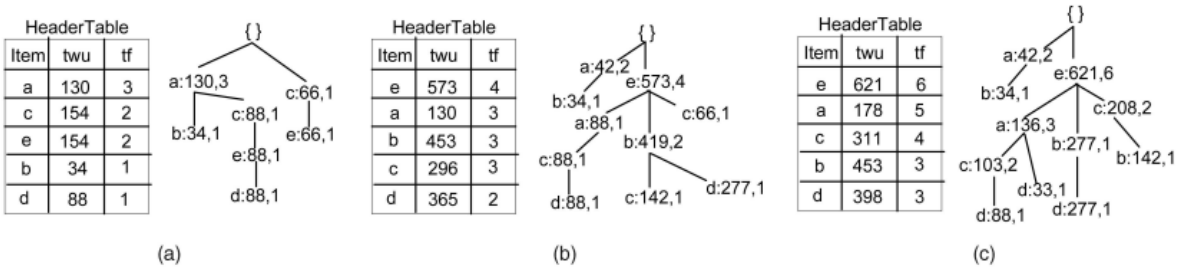


Hình 8. Cấu tạo của IHUPL-Tree (a) sau khi chèn T1, (b) sau khi chèn T2 (c) sau khi chèn T3 và T4, (d) sau khi chèn db1+ (T5 và T6), và (e) sau khi chèn db2+ (T7 và T8).

Thứ tự các mục trong IHUPL-Tree không bị ảnh hưởng mặc dù thay đổi tần suất của các mục bằng cách thêm, xóa và sửa đổi.

IHUP Tree with Frequency (IHUPTF)

Để giảm kích thước của IHUPL-Tree, phải tăng chia sẻ tiền tố bên trong nó. Trong cây này, các nút được sắp xếp theo thứ tự giảm dần theo tần suất giao dịch (tf) của chúng để các mục xảy ra trong nhiều giao dịch có thể được giữ ở phần trên của cây, và do đó, có thể đạt được chia sẻ tiền tố cao hơn. IHUPTF-Tree có thể được xây dựng từ IHUPL-Tree bằng phương pháp điều chỉnh đường dẫn dựa trên kỹ thuật phân loại bong bóng bất cứ lúc nào. Bất kỳ nút nào cũng có thể được chia nhỏ khi nó cần được hoán đổi với bất kỳ nút con nào có số lượng nhỏ hơn nút đó. Nếu số lượng hỗ trợ của cả hai nút bằng nhau, một thao tác trao đổi đơn giản giữa chúng được thực hiện. Sau khi thực hiện mỗi thao tác, các nút hoán đổi có cùng mục sẽ được hợp nhất.



Hình 9. Cấu tạo của IHUPTF-Tree (a) sau khi chèn T4, (b) sau khi chèn T6, (c) sau khi chèn T8

IHUP Tree with Transaction Weighted Utilization (IHUPTWU)

Cấu trúc cây: Sử dụng một cây IHUP chứa thông tin về TWU của các mục trong các giao dịch. Thuật toán được thiết kế để giảm thời gian khai thác bằng cách xem xét thứ tự giảm giá trị TWU của các mặt hàng. Cập nhật khi có giao dịch mới được thêm vào hoặc giao dịch cũ bị xóa, cây IHUPTWU sẽ được cập nhật tương ứng với TWU của các mục.

Ưu điểm chính của IHUP là các cấu trúc cây được đề xuất là khám phá hiệu quả và đạt được khả năng mở rộng để tăng và khai thác HUP tương tác. Tuy nhiên, thuật toán được đề xuất tạo ra số lượng lớn HTWUIs trong giai đoạn đầu tiên.

Utility Pattern Growth (UP-Growth)

UP-Growth (2010) là một thuật toán được thiết kế để tối ưu hóa quá trình khai thác dữ liệu, đặc biệt là trong các cơ sở dữ liệu lớn có nhiều giao dịch dài. Thuật toán này khắc phục vấn đề của phương pháp IHUP, khi mà IHUP tạo ra quá nhiều ứng viên không cần

thiết trong giai đoạn đầu, gây tốn thời gian xử lý. UP-Growth giải quyết vấn đề này bằng cách giảm số lượng ứng viên ngay từ đầu, thông qua việc sử dụng một cấu trúc cây gọi là UP-Tree và bốn chiến lược cắt tỉa.

Các chiến lược cắt tỉa này bao gồm việc loại bỏ sớm những mục không tiềm năng trong toàn bộ cơ sở dữ liệu, giảm bớt các nút không hữu ích trong cây, và loại bỏ các mục không cần thiết trong từng giao dịch cụ thể. Ngoài ra, UP-Growth còn giảm tiện ích của các nút không có giá trị trong từng giao dịch, giúp tối ưu hóa quá trình khai thác dữ liệu. Nhờ vào những cải tiến này, UP-Growth chỉ cần quét cơ sở dữ liệu hai lần và có thể xử lý hiệu quả hơn, giúp tiết kiệm thời gian và tài nguyên khi làm việc với các cơ sở dữ liệu lớn.

Các thuật toán HUI dựa trên cây đã cho thấy sự cải thiện so với các phương pháp theo cấp độ, đặc biệt trong việc sinh ứng viên và quét cơ sở dữ liệu. Các thuật toán này thường chỉ cần hai lần quét cơ sở dữ liệu và ít bộ nhớ hơn vì chúng khám phá không gian tìm kiếm theo phương pháp tìm kiếm theo chiều sâu. Tuy nhiên, các thuật toán này cũng có một số nhược điểm: cấu trúc cây phức tạp và yêu cầu sử dụng bộ nhớ cao; mất thời gian để xử lý đệ quy tất cả các cây tiền tố để sinh ứng viên; việc xây dựng một cây khá tốn kém.

2.1.3.4. Utility-list based

Các thuật toán khai thác tập hữu ích cao dựa trên Utility-List được phát triển nhằm khắc phục những hạn chế của các phương pháp tiếp cận theo cấp độ và dựa trên cây trong việc khai thác tập hữu ích cao từ dữ liệu giao dịch. Những thuật toán này tập trung vào việc tối ưu hóa quá trình tìm kiếm và tính toán các tập mục hữu ích cao, giúp giảm thiểu số lần quét dữ liệu và số lượng ứng cử viên không cần thiết.

Đặc điểm chính của các thuật toán dựa trên Utility-List:

- Cấu trúc dữ liệu Utility-List: Các thuật toán này sử dụng cấu trúc Utility-List, một cấu trúc dọc hoặc ngang, để lưu trữ thông tin về giá trị hữu ích của các tập mục trong các giao dịch.

- Tìm kiếm theo chiều sâu: Utility-List dựa trên phương pháp tìm kiếm theo chiều sâu, giúp tính toán nhanh chóng giá trị hữu ích của các tập mục, đồng thời loại bỏ các tập mục không tiềm năng nhờ vào việc cắt tỉa không gian tìm kiếm.

- Giới hạn trên của giá trị hữu ích: Giới hạn trên của giá trị hữu ích còn lại được sử dụng để giảm số lượng tập mục cần xem xét. Điều này giúp tiết kiệm thời gian tính toán và giảm bớt bộ nhớ cần thiết.

- Tránh việc tạo ứng cử viên: Các thuật toán dựa trên Utility-List tránh việc tạo ra các ứng cử viên trung gian và tính toán hữu ích phức tạp, giúp cải thiện hiệu suất tổng thể của quá trình khai thác.

- Khả năng mở rộng: Nhờ vào các chiến lược tối ưu hóa trên, các thuật toán dựa trên Utility-List có khả năng mở rộng tốt, phù hợp với các tập dữ liệu có số lượng lớn các sản phẩm và giao dịch.

High Utility Itemset Miner (HUI-Miner)

Để xác định tập lợi ích cao, các thuật toán đầu tiên tạo ra tập ứng cử viên từ cách đánh giá các lợi ích cao vì vậy sẽ gặp những vấn đề là tạo ra một số lượng lớn tập ứng viên nhưng hầu hết các ứng cử viên được sinh ra là lợi ích không cao sau khi các lợi ích được tính chính xác. HUI-Miner[18] sử dụng một cấu trúc mới, được gọi là danh sách lợi ích, để lưu trữ tất cả các thông tin hữu ích về một tập và tìm ra thông tin để cắt tỉa không gian tìm kiếm của HUI-Miner. Bằng cách tránh tạo ra các tập ứng viên thế hệ và tính toán lợi ích của nhiều tập ứng viên, HUI-Miner hiệu quả hơn vì có thể khai thác tập lợi ích cao từ danh sách lợi ích.

Utility-list là danh sách các phần tử, mỗi phần tử chứa thông tin về một tập mục và các giao dịch mà tập mục đó xuất hiện.

Trong utility-list của một tập mục (X), mỗi phần tử chứa ba trường chính:

- tid: Chỉ định giao dịch (T) chứa tập mục (X). Đây là mã giao dịch mà tập mục xuất hiện.

- iutil: Giá trị hữu ích của tập mục (X) trong giao dịch (T), ký hiệu là (u(X, T)). Đây là tổng giá trị hữu ích mà tập mục (X) đóng góp trong giao dịch đó.

- rutil: Giá trị hữu ích còn lại của tập mục (X) trong giao dịch (T), ký hiệu là (ru(X, T)). Đây là tổng giá trị hữu ích của các mục còn lại trong giao dịch sau khi loại bỏ tập mục (X).

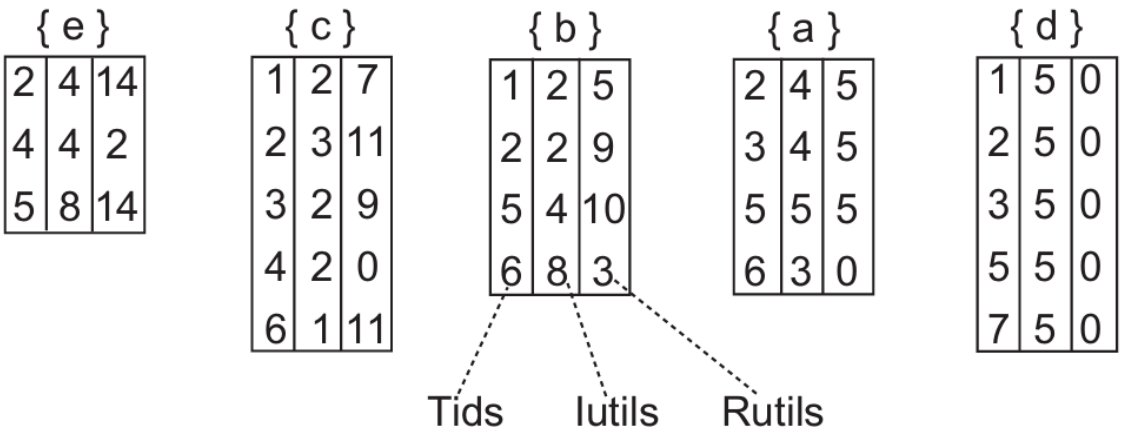
Ví dụ, hãy xem xét danh sách tiện ích của các itemset {c}. Trong T1, u({c}, T1) = 2, ru({c}, T1) = u(b, T1) + u(d, T1) = 2 + 5 = 7, và do đó phần tử <1, 2, 7> nằm trong danh

sách tiện ích của {c} (<x, y, z> có nghĩa là <tid, iutil, rutil> và 1 đại diện cho T1 để đơn giản.). Trong T2, u({c}, T2) = 3, ru({c}, T2) = u(b, T2) + u(a, T2) + u(d, T2) = 2 + 4 + 5 = 11, và do đó phần tử <2, 3, 11> cũng thuộc danh sách tiện ích của {c}. Phần còn lại có thể được tìm ra theo cách tương tự.

Tid	Item	Util.	Item	Util.	item	Util.	item	Util.	item	Util.	TU
T1	c	2	b	2	d	5					9
T2	e	4	c	3	b	2	a	4	d	5	18
T3	c	2	a	4	d	5					11
T4	e	4	c	2							6
T5	e	8	b	4	a	5	d	5			22
T6	c	1	b	8	a	3					12
T7	d	5									5

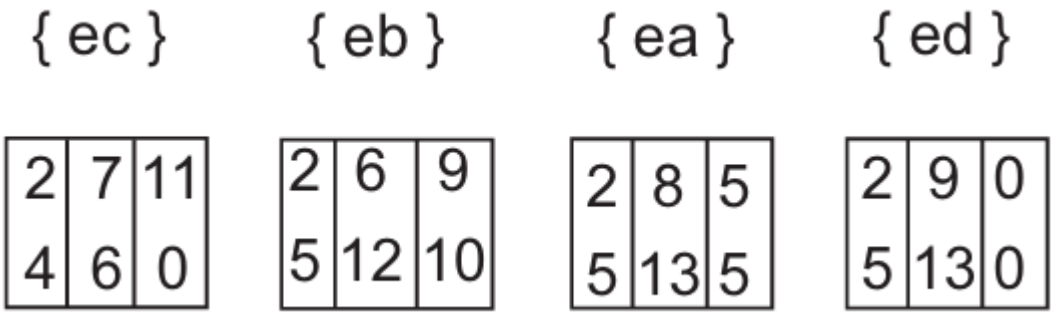
Hình 10. Cấu tạo của transaction

Utility-list ban đầu của một tập mục đơn (1-itemset) được tạo trong lần quét cơ sở dữ liệu thứ hai. Mỗi phần tử trong utility-list chứa ba trường: mã giao dịch (tid), giá trị hữu ích của tập mục trong giao dịch đó (iutil), và giá trị hữu ích còn lại của các mục còn lại trong cùng giao dịch (rutil).



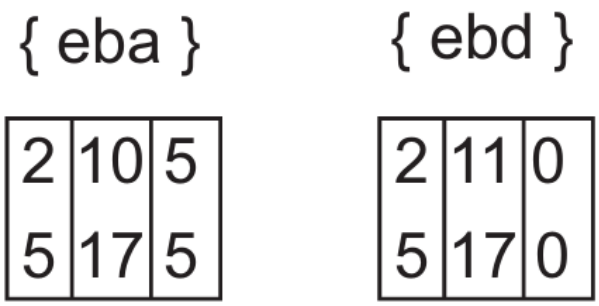
Hình 11. Utility-Lists ban đầu

Utility-Lists of 2-Itemsets: Để xây dựng utility-list của tập mục 2-itemsets ({xy}), ta thực hiện giao nhau giữa utility-list của ({x}) và ({y}). Quy trình tìm các giao dịch chung thực hiện bằng cách so sánh các mã giao dịch (tid) trong hai utility-lists. Ví dụ, để xây dựng utility-list của ({eb}), ta giao nhau utility-list của ({e}) và ({b}), kết quả là (2, 6, 9) và (5, 12, 10).



Hình 12. Utility-Lists của 2-itemsets

Utility-Lists of k-Itemsets (k≥3): Để xây dựng utility-list của k-itemset ({i1, cdots, i(k-1), ik}) (với (k geq 3)), ta thực hiện giao nhau giữa utility-list của ({i1, cdots, i(k-2), i(k-1)}) và ({i1, cdots, i(k-2), ik}). Ví dụ, để xây dựng utility-list của ({eba}), ta giao nhau giữa utility-list của ({eb}) và ({ea}).



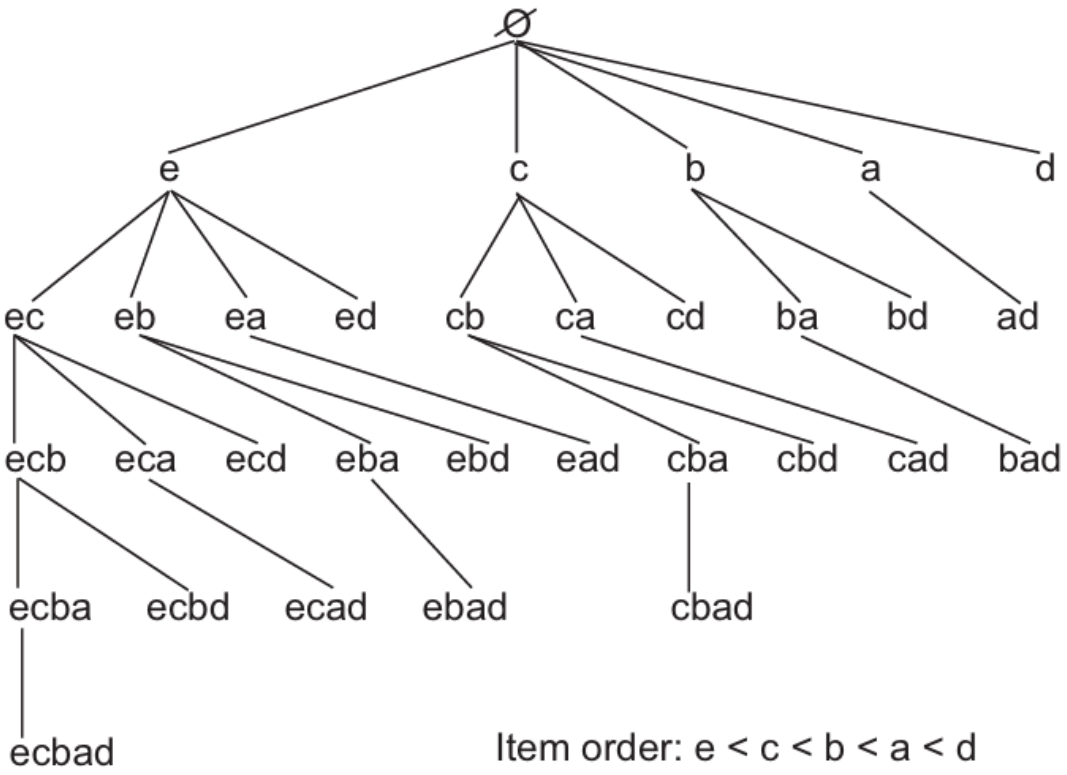
Hình 13. Utility-Lists của 3-itemsets

Trong thuật toán HUI-Miner, không gian tìm kiếm của bài toán khai thác tập hữu ích cao (HUI) có thể được biểu diễn dưới dạng cây liệt kê tập mục (set-enumeration tree). Để xây dựng cây này:

- Gốc Cây: Tạo một nút gốc đại diện cho tập hợp tất cả các mục.

- Nút Con: Tạo n nút con của gốc, mỗi nút đại diện cho một 1-itemset.
- Mở Rộng: Đối với mỗi nút đại diện cho một itemset, tạo các nút con đại diện cho các itemset mở rộng từ nút đó. Quy trình này lặp lại cho đến khi tất cả các nút lá được tạo ra.

Ví dụ: Với tập hợp các mục I={e,c,b,a,d} và thứ tự e<c<b<a<d, cây liệt kê tập mục sẽ bao gồm tất cả các tập hợp mục con của I.



Hình 14. Cây liệt kê tập hợp

Faster High-Utility Itemset Mining (FHM)

HUI-Miner là một thuật toán rất hiệu quả. Tuy nhiên, một nhược điểm là thao tác tham gia để tính toán danh sách tiện ích của một mặt hàng rất tốn kém. Thuật toán FHM [19] cải thiện HUI-Miner bằng cách có thể loại bỏ các bộ mục tiện ích thấp mà không cần thực hiện các thao tác nối.

Tid	Transactions
T_1	(a,1)(c,1)(d,1)
T_2	(a,2)(c,6)(e,2)(g,5)
T_3	(a,1)(b,2)(c,1)(d,6),(e,1),(f,5)
T_4	(b,4)(c,3)(d,3)(e,1)
T_5	(b,2)(c,2)(e,1)(g,2)

Item	a	b	c	d	e	f	g
Profit	5	2	1	2	3	1	1

Hình 15. Cơ sở dữ liệu giao dịch (trái) và các giá trị tiện ích bên ngoài (phải)

TID	TU
T_1	8
T_2	27
T_3	30
T_4	20
T_5	11

Item	TWU
a	65
b	61
c	96
d	58
e	88
f	30
g	38

Item	a	b	c	d	e	f
b	30					
c	65	61				
d	38	50	58			
e	57	61	77	50		
f	30	30	30	30	30	
g	27	38	38	0	38	0

Hình 16.Tiện ích giao dịch (trái), giá trị TWU (giữa) và EUCS (phải)

TWU của tất cả các cặp mặt hàng và lưu trữ nó trong một cấu trúc có tên EUCS và có thể được triển khai dưới dạng (1) ma trận tam giác hoặc (2) hashmap.

VD: twu({a,b}) = 30, twu({a,c}) = 65

Sau đó, trong quá trình tìm kiếm, hãy xem xét rằng chúng ta cần tính toán danh sách tiện ích của một mục X. Nếu X chứa một cặp mục i và j sao cho TWU ({i, j}) < minutil, thì X là tiện ích thấp cũng như tất cả các phần mở rộng của nó.

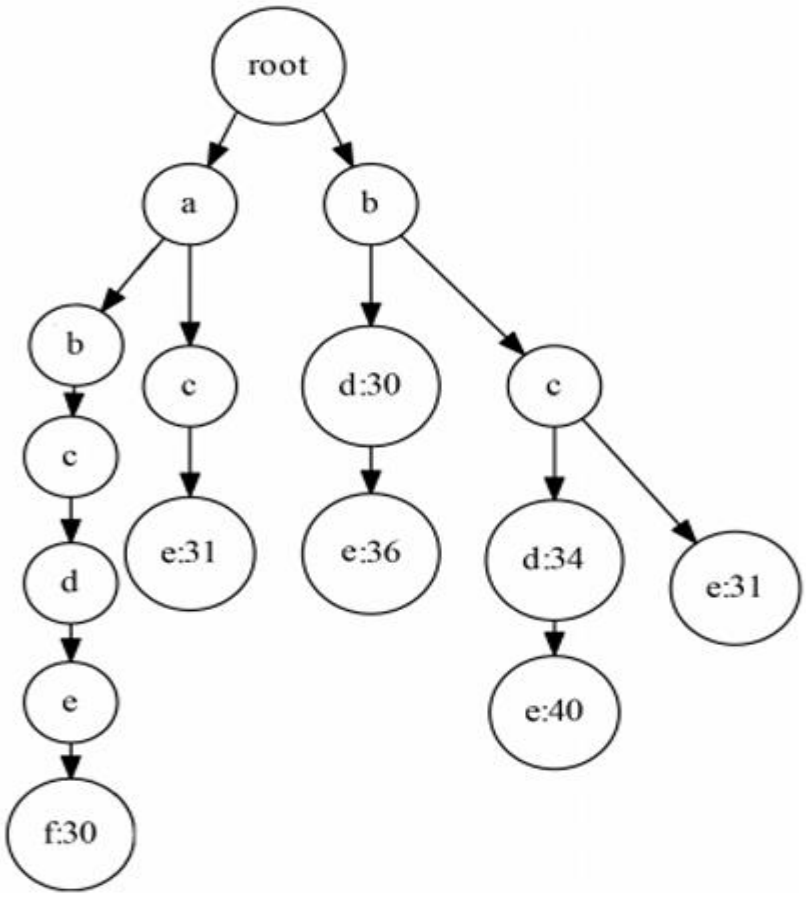
VD: Ta cần tính 1 mục mới {a,b,c}. mà ngướng minutil là 50. Vậy thì không cần tạo vì twu({a,b}) không thể lớn hơn 30

Do đó chỉ cần sử dụng bảng đơn giản này chúng ta chỉ cần thực hiện 1 lần bằng cách đọc cơ sở dữ liệu chúng ta có thể loại bỏ rất nhiều phép nối theo cách HUI. Kết quả thử nghiệm cho thấy FHM nhanh hơn tới sáu lần và thực hiện các hoạt động tham gia ít hơn tới 95% so với HUI-Miner.

Efficient Incremental High-Utility Itemset Mining (EIHI)

EIHI [6] là một thuật toán mới dựa trên FHM, được đề xuất để khai thác và cập nhật các tập mục cao trong cơ sở dữ liệu động, nơi dữ liệu có thể được cập nhật liên tục. Thuật toán tối ưu hóa việc tính toán TWU và danh sách tiện ích, EUCS và áp dụng các điều kiện cắt tỉa mới để cải thiện hiệu quả. Thuật toán này cải tiến so với các phương pháp trước đó là việc lưu trữ tập mục trong cấu trúc trie gọi là HUI-trie.

HUI-trie là một cấu trúc cây giống trie, trong đó mỗi nút đại diện cho một item và mỗi tập mục được biểu diễn bằng một đường đi từ gốc đến một nút lá hoặc nút bên trong. Ví dụ, Hình 3: hiển thị cấu trúc HUI-trie được cấu trúc với các HUI được tìm thấy trong cơ sở dữ liệu của Bảng 1, đó là bd, ace, bcd, bce, bde, bcde, abcdef có các tiện ích tương ứng 30, 31, 34, 31, 36, 40 và 30.



Hình 17. Cấu trúc HUI-trie

Phương pháp cải tiến của EIHI khi cơ sở dữ liệu được cập nhật dựa trên ba yếu tố chính. Thứ nhất, thuật toán duy trì TWU bằng cách chỉ cập nhật TWU cho các mục xuất hiện trong giao dịch mới (N) khi cơ sở dữ liệu thay đổi (D'), thay vì phải tính lại TWU cho

toàn bộ cơ sở dữ liệu. Điều này giúp tiết kiệm đáng kể thời gian và tài nguyên tính toán. Thứ hai, EIHI giới hạn không gian tìm kiếm bằng cách chỉ tập trung vào các tập mục xuất hiện trong giao dịch mới, loại bỏ các tập mục không xuất hiện trong giao dịch này khỏi quá trình tìm kiếm. Cuối cùng, trình tự của các mục được bảo lưu từ cơ sở dữ liệu cũ (D) và chỉ các mục mới từ cơ sở dữ liệu mới (D') được thêm vào. Điều này giúp đảm bảo tính nhất quán và hiệu quả trong quá trình khai thác các tập phổ biến có lợi ích cao.

Người ta chỉ ra rằng các thuật toán dựa trên danh sách tiện ích có thể nhanh hơn hai bậc cường độ so với các thuật toán hai pha. Tuy nhiên, các thuật toán dựa trên danh sách tiện ích có những nhược điểm quan trọng. Các thuật toán này có thể khám phá một số tập mục không bao giờ xuất hiện trong cơ sở dữ liệu vì các bộ mục được tạo bằng cách kết hợp các tập mục mà không cần đọc cơ sở dữ liệu. Do đó, các thuật toán này có thể lãng phí rất nhiều thời gian để xây dựng danh sách tiện ích của các bộ mục không tồn tại.

Trong hệ thống gợi ý, EIHI sẽ được lựa chọn với khả năng tính toán và mở rộng của mình. Mặc dù thuật toán dựa trên danh sách tiện ích nhưng lại có cấu trúc cây không chỉ tìm kiếm các itemset có tiện ích cao, mà trình tự của các mục được bảo lưu từ cơ sở dữ liệu cũ (D) và chỉ các mục mới từ cơ sở dữ liệu mới (D') được thêm vào. Sau khi khai thác các EIHI từ cơ sở dữ liệu, hệ thống gợi ý có thể sử dụng chúng để đưa ra các đề xuất phù hợp hơn với từng người dùng. Ví dụ, trong một nền tảng thương mại điện tử, các sản phẩm có EIHI sẽ được ưu tiên trong các đề xuất vì chúng không chỉ phổ biến mà còn có khả năng mang lại lợi nhuận cao, do đó giúp tăng trưởng doanh thu cho doanh nghiệp. Hơn nữa, việc sử dụng EIHI còn giúp hệ thống gợi ý cá nhân hóa hơn khi các mẫu hữu ích cao này có thể được kết hợp với thông tin lịch sử của người dùng để tạo ra các đề xuất chính xác hơn.

2.2. GRAPH NEURAL NETWORKS

2.2.1. Đồ thị

Trong toán học và tin học, đồ thị là một đối tượng cơ bản trong lý thuyết đồ thị. Người sáng lập lý thuyết đồ thị là nhà toán học người Thụy Sĩ Leonhard Euler, người đã khai sinh ra lý thuyết này vào năm 1736[20]. Theo định nghĩa cơ bản, đồ thị là một tập hợp các đối tượng gọi là đỉnh, được nối với nhau bởi các cạnh. Các cạnh này thể hiện các mối quan hệ cụ thể giữa các đỉnh. Tùy theo bài toán cụ thể, cạnh có thể có hướng hoặc không có hướng, và đồ thị sẽ được phân loại thành đồ thị có hướng hoặc đồ thị vô hướng.

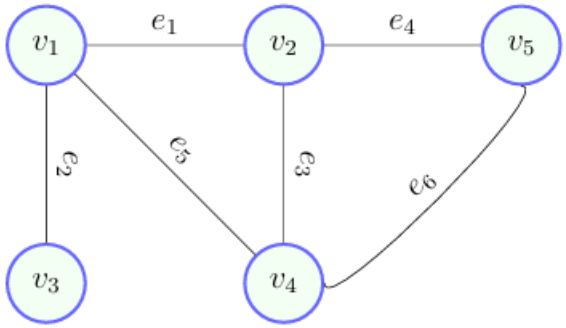
Định nghĩa 1: *Một đồ thị đơn G gồm một tập các đỉnh V và một tập các cạnh E, trong đó mỗi cạnh là một cặp không sắp xếp các đỉnh phân biệt. Đây là đồ thị vô hướng (undirected graph).*

Biểu thức toán học biểu diễn đồ thị như sau:

$$G = (V, E)$$

Trong đó:

- $V = \{v_1, v_2, \dots, v_n\}$ là tập các đỉnh của đồ thị, với $n = | V |$ là số đỉnh.
- $E = \{e_1, e_2, \dots, e_m\}$ là tập các cạnh của đồ thị, với $m = | E |$ là số cạnh.



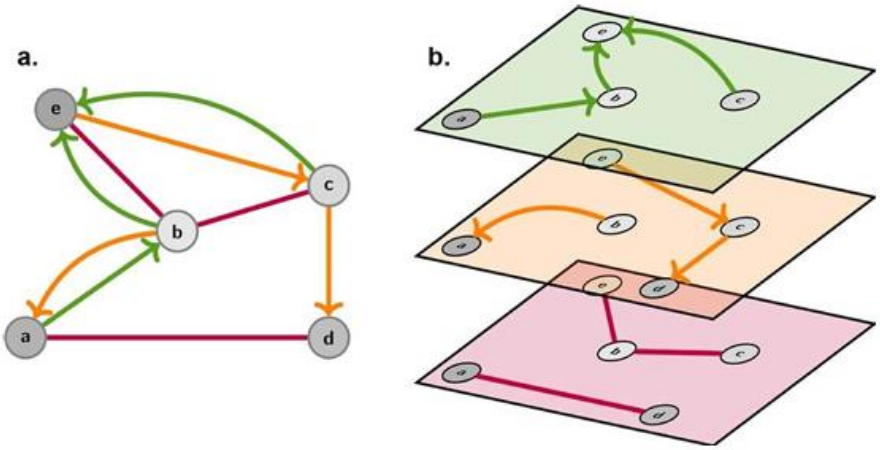
Hình 18. Minh họa đồ thị

Định nghĩa 2: Một đồ thị có hướng *G = (V, E)* gồm một tập các đỉnh *V* và tập các cạnh *E* là các cặp có thứ tự của các phần tử thuộc *V*.

Đồ thị thường được sử dụng để thể hiện các mối quan hệ giữa các đối tượng mà không thể biểu diễn bằng các kiểu dữ liệu thông thường, ví dụ như mối quan hệ giữa người dùng trên mạng xã hội, các liên kết trong mạng Internet, hay sự lan truyền thông tin. Một số bài toán gần đây sử dụng đồ thị bao gồm:

- Phân tích mạng xã hội để xác định xu hướng cộng đồng và nhóm khách hàng.
- Xây dựng hệ thống gợi ý sản phẩm cho các trang web thương mại điện tử từ dữ liệu tương tác của người dùng.
- Phân tích ảnh hưởng của một cá nhân trong cộng đồng để giảm chi phí quảng bá sản phẩm mà vẫn đạt được độ lan tỏa rộng.
- Phát hiện tin giả trên mạng xã hội dựa vào phân tích độ liên kết giữa các thực thể trong đồ thị.
- Phân tích sự tương tác ở cấp độ phân tử hoặc nguyên tử trong các nghiên cứu y sinh học, ví dụ như tác dụng phụ của thuốc.

Một số đồ thị phức tạp có thể chứa nhiều loại cạnh khác nhau nối giữa các đỉnh. Ví dụ, đồ thị thể hiện bản đồ giao thông giữa các tỉnh có thể bao gồm các loại phương tiện khác nhau như đường bộ, hàng không hay đường thủy. Đây gọi là đồ thị đa quan hệ (multi-relational graph), trong đó mỗi cạnh có thể đại diện cho một loại quan hệ khác nhau giữa các đỉnh.



Hình 19. Minh họa đồ thị quan hệ

Định nghĩa 3: Một đồ thị đa quan hệ vô hướng $G = (V, E)$ gồm tập các đỉnh V, tập các cạnh E, và một hàm f từ E tới $\{(u, v) \mid u, v \in V, u \neq v\}$, để xác định loại quan hệ giữa hai đỉnh u và v. Các cạnh e_1 và e_2 được gọi là cạnh song song hay cạnh bội nếu $f(e_1) = f(e_2)$

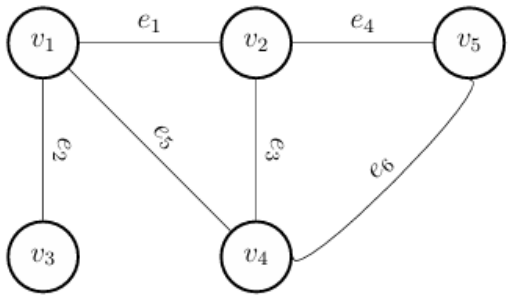
Định nghĩa 4: Một đồ thị đa quan hệ có hướng $G = (V, E)$ gồm tập các đỉnh V, tập các cạnh E, và một hàm f từ E tới $\{(u, v) \mid u, v \in V\}$, để xác định loại quan hệ giữa hai đỉnh có thứ tự u và v. Các cạnh e_1 và e_2 được gọi là cạnh song song hay cạnh bội nếu $f(e_1) = f(e_2)$

2.2.2 Biểu diễn đồ thị

Có nhiều cách để biểu diễn đồ thị, và phương pháp biểu diễn có thể phụ thuộc vào tính chất của đồ thị và thuật toán áp dụng. Hai cách phổ biến nhất để biểu diễn đồ thị là danh sách kề và ma trận kề.

a. Danh sách kề

Danh sách kề (adjacency list) là một danh sách biểu diễn tất cả các cạnh của đồ thị. Nếu đồ thị là vô hướng, mỗi phần tử trong danh sách là một cặp hai đỉnh, đại diện cho hai đầu của một cạnh. Nếu đồ thị có hướng, mỗi phần tử là một cặp có thứ tự gồm hai đỉnh đại diện cho đỉnh đầu và đỉnh cuối của cung.



(a) Đồ thị minh họa

Đỉnh	Các đỉnh kề
v_1	v_2, v_3, v_4
v_2	v_1, v_4, v_5
v_3	v_1
v_4	v_1, v_2, v_5
v_5	v_2, v_4

(b) Danh sách các đỉnh kề

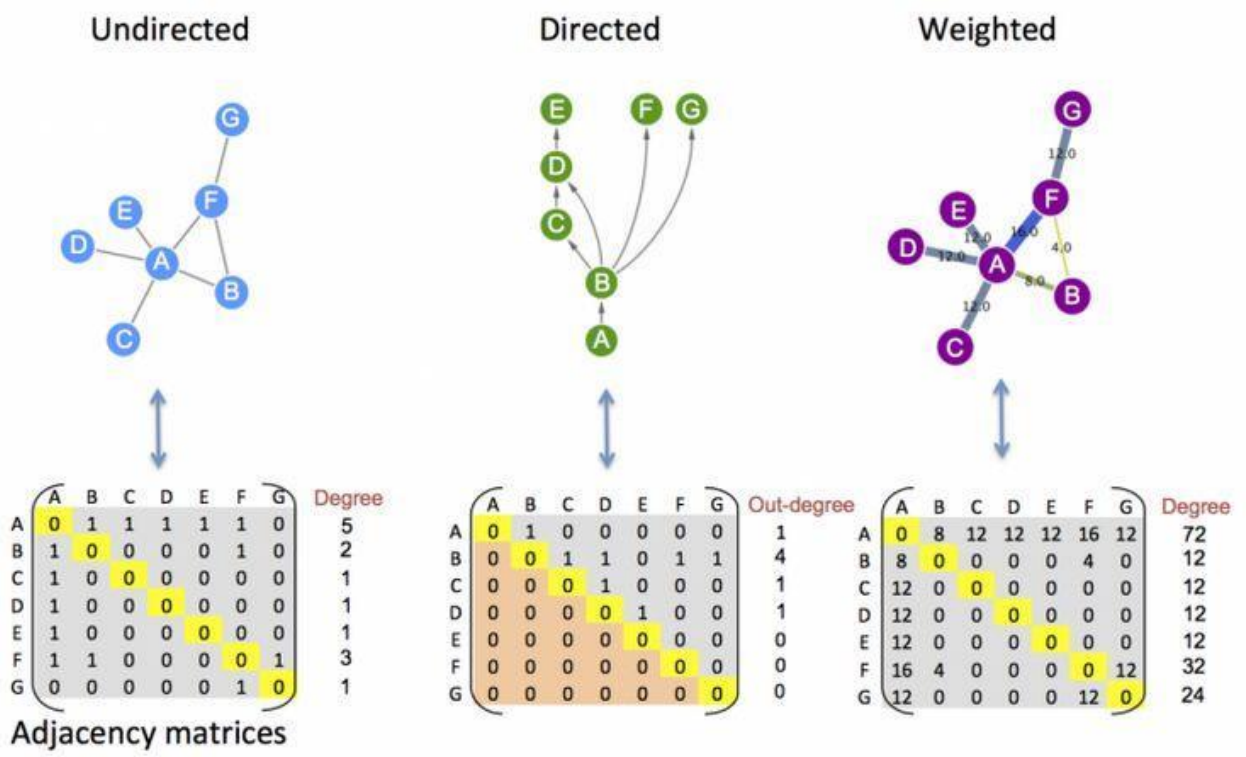
Hình 20. Biểu diễn đồ thị bằng danh sách kề

b. Ma trận kề

Khi biểu diễn đồ thị bằng danh sách kề, việc xây dựng thuật toán có thể trở nên cồng kềnh nếu đồ thị có nhiều cạnh. Để đơn giản hóa quá trình tính toán, đồ thị có thể được biểu diễn bằng ma trận kề (adjacency matrix). Giả sử $G = (V, E)$ là đồ thị đơn có n đỉnh, ma trận kề A_G có kích thước $n \times n$, với các giá trị như sau:

- $a_{ij} = 1$ nếu có cạnh nối đỉnh v_i với đỉnh v_j.
- $a_{ij} = 0$ nếu không có cạnh nối đỉnh v_i với đỉnh v_j.
- a_{ii} = 0 với tất cả i.

Nếu đồ thị có trọng số, thì giá trị a_{ij} sẽ là trọng số của cạnh nối các đỉnh v_i và v_j. Ma trận kề này có thể đại diện cho đồ thị vô hướng, có hướng hoặc có trọng số, tùy thuộc vào cách định nghĩa các phần tử của ma trận.



Hình 21. Biểu diễn đồ thị bằng ma trận kề

Ưu điểm của ma trận kề:

- Đơn giản và trực quan, dễ hiểu.
- Kiểm tra sự kết nối giữa hai đỉnh i và j chỉ cần kiểm tra a_{ij} với độ phức tạp $O(1)$.

Nhược điểm của ma trận kề:

- Tiêu tốn N^2 bộ nhớ để lưu trữ ma trận kề, dù đồ thị có ít hay nhiều cạnh.
- Khó khăn trong việc biểu diễn đồ thị có số lượng đỉnh lớn.
- Để kiểm tra các đỉnh kề với một đỉnh ii, phải duyệt qua toàn bộ các đỉnh jj, với độ phức tạp $O(n)$,kể cả khi đỉnh ii không kề với bất kỳ đỉnh nào.

1.4.3. Mô hình mạng nơ-ron đồ thị

Mô hình mạng nơ-ron đồ thị (GNN) lần đầu tiên được giới thiệu vào năm 2005[21] và hiện nay đã trở thành một loại mạng nơ-ron mạnh mẽ hoạt động trực tiếp trên cấu trúc đồ thị. Trong mô hình này, các nút trong đồ thị được xem như các nơ-ron, mỗi nút chứa thông tin của riêng nó và thu thập thêm thông tin từ các nút lân cận, qua đó biểu thị mối quan hệ giữa chúng trong đồ thị. Các nút này được kết hợp và sắp xếp theo một cấu trúc mô hình nhất định để đưa ra các dự đoán hoặc phân loại kết quả.

Với khả năng biểu diễn các mối quan hệ dữ liệu dưới dạng đồ thị, GNN ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau. Mô hình này đã chứng minh tiềm năng mạnh mẽ trong việc giải quyết các bài toán thực tế, bao gồm xây dựng biểu đồ tri thức, đánh giá mối tương quan mạng xã hội, và hệ thống gợi ý sản phẩm. Sức mạnh của

GNN nằm ở khả năng mô hình hóa mối quan hệ giữa các đỉnh trong đồ thị, từ đó tạo ra bước đột phá trong các nghiên cứu phân tích đồ thị. Một số vấn đề phổ biến mà GNN giải quyết bao gồm:

- **Phân loại nút (Node classification):** Ví dụ, trong các mạng xã hội, việc phân loại người dùng thành các nhóm khác nhau (chẳng hạn như phân biệt người dùng thật và các bot) có thể giúp xây dựng các chiến lược tùy chỉnh cho từng nhóm. Mô hình GNN giúp phân loại hàng triệu người dùng trong mạng, từ đó giảm chi phí khi đưa ra các chiến lược marketing cho từng nhóm đối tượng trong mạng[22]. Ngoài ra, GNN còn được ứng dụng trong các bài toán phân loại nút trong việc đánh giá các loại thuốc[23] hay phân loại chủ đề tài liệu trong các mạng trích dẫn [24].

- **Dự đoán kết nối (Link prediction):** Đây là một trong những ứng dụng phổ biến của GNN, khi nó được sử dụng để dự đoán mối quan hệ giữa các thực thể trong một mạng lưới. Ví dụ, trong mạng xã hội, mô hình GNN có thể gợi ý nội dung phù hợp cho người dùng, hoặc trong thương mại điện tử, mô hình có thể đề xuất sản phẩm cho khách hàng dựa trên lịch sử mua sắm của họ. Ngoài ra, GNN cũng có thể được sử dụng để dự đoán tác dụng phụ của thuốc trong y tế[25], một trong những hướng nghiên cứu mà luận án này hướng đến, liên quan đến việc xây dựng mô hình gợi ý sản phẩm tiếp theo cho người dùng trong các phiên làm việc.

- **Phát hiện cụm (Clustering detection):** Cả bài toán phân loại nút và dự đoán kết nối đều thuộc dạng học có giám sát, sử dụng dữ liệu có sẵn để học và suy luận các phần còn thiếu trong đồ thị[26]. Tuy nhiên, bài toán phát hiện cụm lại thuộc dạng học không giám sát, với mục tiêu tìm ra các nhóm nút trong đồ thị có mối quan hệ chặt chẽ. Ví dụ, có thể tìm ra nhóm người có cùng lĩnh vực nghiên cứu từ mạng trích dẫn (citation graph) trong Google Scholar, hoặc phát hiện nhóm người có yếu tố giả mạo trong mạng lưới người dùng thực hiện giao dịch tài chính [27].

- **Phân loại đồ thị (Graph classification):** Khác với các bài toán phân loại hoặc dự đoán một thành phần cụ thể (nút, cạnh hay đồ thị con) trong một đồ thị, bài toán phân loại đồ thị yêu cầu học từ tập hợp các đồ thị khác nhau để đưa ra dự báo cho một đồ thị cụ thể. Thách thức ở đây là phải tìm ra các thuộc tính đặc trưng của đồ thị mà khác biệt với các dạng dữ liệu có cấu trúc mà chúng ta đã nghiên cứu trước đây [28].

Vào năm 2009, Scarselli và các cộng sự [29] đã đề xuất sử dụng GNN để xử lý vấn đề biến đổi véc-tơ trong đồ thị. Sau đó, Li và các cộng sự[30] đã phát triển một phiên bản cải tiến của GNN, kết hợp thêm các lớp mạng hồi quy (RNN) để nâng cao hiệu quả của phép biến đổi này. Kết quả cho thấy GNN có thể tự động trích xuất các thuộc tính của đồ thị phiên làm việc, giúp mô hình này thể hiện rất tốt các đặc tính của mối tương tác giữa các nút trong đồ thị, từ đó nâng cao hiệu quả trong các bài toán như gợi ý sản phẩm.

2.2.3. Graph Attention Networks (GAT)

Graph Attention Networks (GAT) là một biến thể của Graph Neural Networks (GNN), trong đó cơ chế Attention (chú ý) được sử dụng để tự động học trọng số (weight) của các kết nối giữa các nút trong đồ thị, thay vì sử dụng trọng số cố định hoặc đồng nhất cho tất cả các láng giềng của một nút. Điều này giúp GAT có khả năng học được sự quan trọng của mỗi nút láng giềng đối với nút hiện tại, từ đó cải thiện khả năng học và dự đoán trong các bài toán phức tạp, chẳng hạn như bài toán gợi ý, phân loại đồ thị, hay dự đoán liên kết.

Trong bài toán hệ thống gợi ý, GAT là một lựa chọn ưu việt khi cần xử lý các mối quan hệ phức tạp giữa người dùng và sản phẩm, đặc biệt là khi các sản phẩm có sự tương đồng không đều hoặc có sự tương tác mạnh mẽ với một số người dùng cụ thể. So với các biến thể GNN khác như GCN, GraphSAGE hay GNN với Graph Pooling, GAT mang lại sự linh hoạt và hiệu quả cao hơn trong việc học các trọng số chú ý và tập trung vào các mối quan hệ quan trọng, mặc dù nó có chi phí tính toán cao hơn.

GAT gồm ba thành phần chính:

1. Cơ chế chú ý (Attention Mechanism): GAT sử dụng cơ chế chú ý để tự động học tầm quan trọng của các mối quan hệ giữa các nút trong đồ thị.
2. Self-Attention: Mỗi nút trong đồ thị sẽ tự tính toán trọng số của các nút láng giềng, điều này giúp mô hình học được các mối quan hệ phức tạp mà không cần dựa vào cấu trúc cố định.
3. Chú ý đa đầu (Multi-Head Attention): GAT sử dụng nhiều đầu chú ý để nắm bắt nhiều khía cạnh của các mối quan hệ trong đồ thị, từ đó giúp cải thiện độ chính xác của mô hình.

GAT sử dụng Self-Attention, một kỹ thuật giúp mỗi nút trong đồ thị có thể học được tầm quan trọng của các nút láng giềng mà nó có kết nối. Cơ chế chú ý này có thể được mô tả qua các bước sau:

1. Tính toán trọng số chú ý

Mỗi cặp nút i và j trong đồ thị sẽ có một trọng số chú ý α_{ij}, phản ánh tầm quan trọng của nút j đối với nút i. Trọng số này được tính toán thông qua một hàm chú ý. Cụ thể, ta sẽ tính trọng số chú ý α_{ij} bằng cách áp dụng một hàm kích hoạt (ví dụ LeakyReLU) lên sự kết hợp của các embedding của hai nút i và j, rồi tính toán giá trị softmax của các trọng số này trên các láng giềng của nút i.

Công thức tính trọng số chú ý là:

$$\alpha_{ij} = \frac{exp\left(LeakyReLU\left(a^T[Wh_i \parallel Wh_j]\right)\right)}{\sum_{k \in N(i)} exp\left(LeakyReLU(a^T[Wh_i \parallel Wh_k])\right)}$$

Trong đó:

- h_i, h_j: Vector embedding của nút i và j
- W: Ma trận trọng số học được
- a: Vector trọng số học được để tính mức độ tương quan
- $N(i)$: Tập các nút láng giềng của nút i
- $\parallel$: Phép nối vector

2. Cập Nhật Đặc Trưng của Nút

Sau khi tính toán các trọng số chú ý α_{ij}, ta sử dụng chúng để cập nhật lại các đặc trưng (embedding) của nút i. Việc cập nhật này được thực hiện thông qua một phép tính trọng số giữa các đặc trưng của nút và các nút láng giềng.

Công thức cho bước cập nhật embedding là:

$$h_i' = \sigma\left(\sum_{j \in N(i)} \alpha_{ij} Wh_j\right)$$

Trong đó:

- σ là hàm kích hoạt phi tuyến (thường dùng ReLU)
- α_{ij} là trọng số chú ý giữa nút i và nút j
- W là ma trận trọng số học được
- h_j là embedding của nút j

Điều này có nghĩa là mỗi nút sẽ cập nhật đặc trưng của mình dựa trên các nút láng giềng, với trọng số được tính toán từ cơ chế chú ý.

3. Multi-Head Attention

Để giúp mô hình học được nhiều khía cạnh khác nhau của mối quan hệ giữa các nút, GAT sử dụng chú ý đa đầu (multi-head attention). Thay vì chỉ sử dụng một đầu chú ý, GAT sử dụng M đầu chú ý khác nhau, mỗi đầu học một khía cạnh khác nhau của các mối quan hệ. Sau khi tính toán được các đặc trưng từ từng đầu chú ý, chúng được nối lại và chiếu qua một phép biến đổi tuyến tính.

Công thức cho multi-head attention là:

$$h_i' = \|_{m=1}^{M} \sigma\left(\sum_{j \in N(i)} \alpha_{ij}^{(m)} W^{(m)} h_j \right)$$

Ở đây:

- M: Số lượng đầu chú ý
- $W^{(m)}$: Ma trận trọng số của đầu chú ý thứ m
- $\alpha_{ij}^{(m)}$: Trọng số chú ý từ nút i đến j trong đầu thứ m

Kết quả của các đầu chú ý sẽ được nối lại thành một vector và sau đó được biến đổi tuyến tính để tạo ra đặc trưng cuối cùng cho nút.

CHƯƠNG 3: XÂY DỰNG HỆ THỐNG GỢI Ý KẾT HỢP CÁC PHƯƠNG PHÁP VỚI HUI VÀ GNN

Chương này trình bày quá trình thiết kế hệ thống gợi ý dựa trên việc kết hợp các phương pháp, bao gồm khai thác tập mục hữu ích cao và các mô hình gợi ý hiện đại như CB, CF và GNN. Đầu tiên, mô hình EIHI được giới thiệu để tối ưu hóa việc khai thác các tập mục hữu ích bằng cách giảm không gian tìm kiếm và cải thiện hiệu quả xử lý dữ liệu. Tiếp theo, sự kết hợp của các mô hình CB, CF, và HUI được phân tích, trong đó các thuật toán như KNN, User-Based, Item-Base và SVD được sử dụng nhằm nâng cao độ chính xác và chất lượng gợi ý. Cuối cùng, việc tích hợp GNN vào hệ thống gợi ý tận dụng các đặc trưng đồ thị phức tạp và giá trị thực tế của sản phẩm, từ đó tối ưu hóa hiệu quả hệ thống gợi ý, mang lại trải nghiệm người dùng tốt hơn. Các kỹ thuật này không chỉ cải thiện độ chính xác mà còn giúp hệ thống gợi ý trở nên linh hoạt và phù hợp hơn với từng người dùng.

3.1. MÔ HÌNH HUI

Mô hình High Utility Itemset được chọn trong hệ thống là EIHI, với việc khai thác các tập mục hữu ích cao với nhiều ưu điểm nổi bật. Đầu tiên, loại bỏ sớm các giao dịch không cần thiết:

Ở giai đoạn đầu, EIHI sử dụng tổng trọng số tiện ích (TWU) để loại bỏ các giao dịch hoặc tập mục có TWU nhỏ hơn ngưỡng tiện ích tối thiểu (minutil). Điều này giúp giảm đáng kể số lượng tập mục cần phải phân tích, tăng hiệu quả của thuật toán.

Thứ 2, EIHI sử dụng Utility List để tổ chức dữ liệu hiệu quả, hỗ trợ nhanh chóng trong việc tính toán giá trị tiện ích của các tập mục. Ngoài ra, kết hợp với cấu trúc cây trie: cấu trúc HUI-trie được sử dụng để lưu trữ các tập mục hữu ích đã phát hiện.

Cuối cùng, tối ưu hóa không gian tìm kiếm: EIHI chỉ tập trung vào các tập mục xuất hiện trong tập giao dịch mới, loại bỏ các tập mục không liên quan. Thứ tự của các mục được bảo lưu từ cơ sở dữ liệu cũ và chỉ bổ sung các mục mới từ cơ sở dữ liệu mới.

Hệ thống khai thác tập mục hữu ích cao dựa trên EIHI được thiết kế gồm các bước chính: đầu tiên, quét dữ liệu đầu vào để tính toán TWU và loại bỏ các mục có TWU nhỏ hơn ngưỡng tối thiểu min-util, giúp giảm không gian tìm kiếm. Sau đó, hệ thống xây dựng danh sách hữu ích Utility List và EUCS, hỗ trợ việc tìm kiếm các HUIs thông qua phương pháp tìm kiếm theo chiều sâu. Khi có giao dịch mới, cấu trúc HUI-tree cho phép cập nhật trực tiếp trên dữ liệu mới mà không cần xử lý lại toàn bộ dữ liệu cũ, giúp tiết kiệm tài nguyên và đảm bảo hiệu quả.

3.2. MÔ HÌNH CB, CF VÀ HUI

Trong quá trình xây dựng hệ thống gợi ý, kết hợp các mô hình: Content-Based, Collaborative Filtering, và High Utility Itemset giúp tối ưu hóa khả năng gợi ý và nâng cao độ chính xác trong việc đưa ra các đề xuất sản phẩm cho người dùng.

Bước đầu tiên trong quy trình là huấn luyện các mô hình CB và CF trên tập dữ liệu huấn luyện, sau đó đánh giá độ chính xác của các mô hình này trên tập kiểm tra. Trong đó, mô hình Content-Based sử dụng thuật toán KNN dựa trên nội dung sản phẩm để đưa ra gợi ý. Một trong những ưu điểm của phương pháp CB là không gặp phải vấn đề thưa thớt dữ liệu vì thuật toán này chỉ phụ thuộc vào các thuộc tính của sản phẩm, không liên quan đến hành vi của người dùng.

Đối với mô hình CF, sử dụng hai phương pháp chính là User-Based và Item-Based. Cả hai phương pháp này đều dựa vào việc tính toán độ tương đồng giữa các người dùng hoặc các sản phẩm đã được đánh giá, từ đó đưa ra các gợi ý dựa trên sự tương đồng này. Mô hình User-Based gợi ý các sản phẩm mà người dùng tương tự đã đánh giá cao, trong khi Item-Based gợi ý các sản phẩm tương tự với sản phẩm mà người dùng đã mua hoặc đánh giá trước đó.

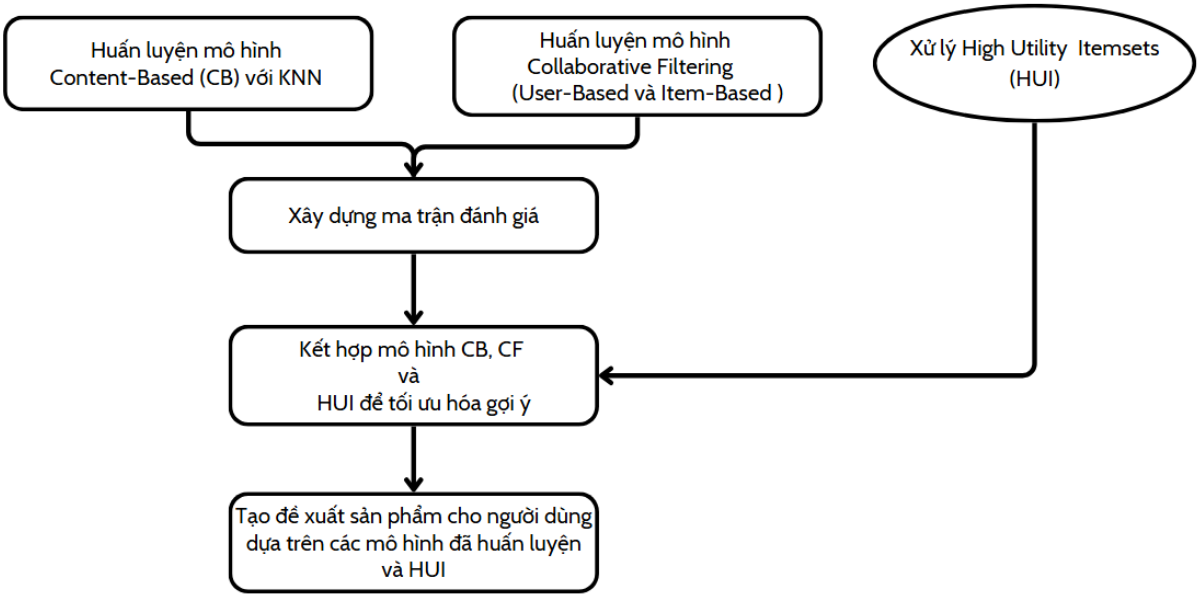
SVD là một kỹ thuật giảm chiều mạnh mẽ dựa trên ma trận, giúp phát hiện ra các "nhân tố ẩn" trong dữ liệu đánh giá của người dùng. Các nhân tố ẩn này có thể là các xu hướng hoặc sở thích tiềm ẩn mà người dùng không thể diễn đạt rõ ràng hoặc không nhận thức được. Việc áp dụng SVD giúp cải thiện khả năng dự đoán của mô hình, tạo ra các gợi ý đa dạng và chính xác hơn. Hơn nữa, SVD còn giúp giảm số chiều của dữ liệu, qua đó tăng tốc quá trình tính toán và cải thiện hiệu suất hệ thống.

Sau khi huấn luyện các mô hình, một ma trận đánh giá (rating matrix) được tạo ra cho tất cả người dùng và sản phẩm. Hệ thống sau đó ưu tiên các sản phẩm có HUI – tức những sản phẩm mang lại giá trị cao.

Các sản phẩm được chọn dựa trên:

- Điểm dự đoán từ các mô hình (CB, CF, SVD).
- Sự xuất hiện trong danh sách HUI.

Nếu danh sách HUI không đủ số lượng gợi ý, hệ thống sẽ bổ sung các sản phẩm có điểm dự đoán cao nhất từ mô hình. Điều này đảm bảo rằng các gợi ý không chỉ phản ánh sở thích của người dùng mà còn mang lại giá trị tối ưu cho doanh nghiệp.



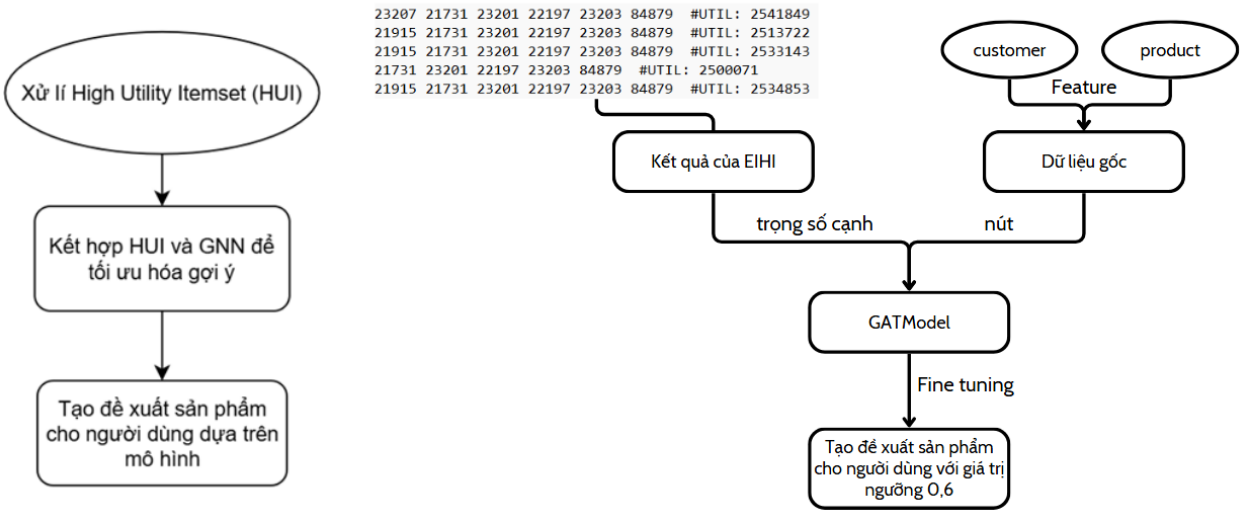
Hình 22. Mô hình kết hợp CB và CF với HUI

Hàm ***suggest_for_user*** là một bước quan trọng trong việc kết hợp các mô hình dự đoán dựa trên CB, CF và HUI nhằm tối ưu hóa hệ thống gợi ý. Hàm này không chỉ dựa trên ma trận dự đoán ***Yhat_matrix*** để xác định sản phẩm phù hợp nhất cho người dùng, mà còn ưu tiên các sản phẩm có giá trị thực tế cao được xác định thông qua HUI.

Một điểm nổi bật của phương pháp là khả năng kết hợp giữa độ chính xác trong dự đoán (thông qua CF và SVD) và giá trị kinh doanh thực tế (thông qua HUI). Hệ thống lọc ra các sản phẩm có mã nằm trong danh sách HUI (***top_items_in_hui***), đảm bảo rằng các sản phẩm không chỉ phù hợp với sở thích của người dùng mà còn có tiềm năng tạo ra doanh thu cao. Nếu danh sách HUI không đủ, hệ thống sẽ tự động bổ sung các sản phẩm có điểm dự đoán cao nhất để đáp ứng yêu cầu.

Đặc biệt, việc sắp xếp và lọc sản phẩm dựa trên ngưỡng dự đoán ***threshold_score*** giúp cân bằng giữa tính cá nhân hóa và tối ưu hóa giá trị kinh doanh. Phương pháp này mang lại các gợi ý không chỉ chính xác mà còn có tính ứng dụng cao trong thực tế, đáp ứng cả nhu cầu người dùng lẫn mục tiêu lợi nhuận của doanh nghiệp.

3.3. MÔ HÌNH GNN VÀ HUI



Hình 23. Tiến trình thực hiện kết hợp HUI vào GNN

Trong hệ thống gợi ý này, đồ thị được xây dựng với hai loại nút chính: khách hàng và sản phẩm. Mỗi khách hàng trong hệ thống được đại diện bởi một nút trong đồ thị, và tương tự, mỗi sản phẩm cũng là một nút. Các cạnh giữa các nút khách hàng và sản phẩm được tạo ra khi một khách hàng mua sản phẩm đó, với trọng số của cạnh này có thể phản ánh số lượng sản phẩm mà khách hàng đã mua hoặc một chỉ số khác, như thời gian tương tác với sản phẩm. Chẳng hạn, nếu khách hàng C1 đã mua sản phẩm P1 thì sẽ có trọng số là 1 nếu chưa mua thì sẽ là 0.

Ngoài ra, các cạnh giữa các sản phẩm cũng được xây dựng dựa trên mối quan hệ giữa chúng, đặc biệt là trong trường hợp các sản phẩm thường xuyên được mua chung bởi khách hàng. Đây chính là các sản phẩm tương tự hoặc có High Utility Itemsets (HUI) cao, cho thấy mức độ liên quan giữa các sản phẩm này. Những mối quan hệ này có thể được khai thác từ các tệp dữ liệu mô tả sự kết hợp giữa các sản phẩm, như tệp hui.txt chứa kết quả của quá trình khai thác tập hữu ích với thuật toán EIHI, trong đó các cạnh giữa các sản phẩm sẽ có trọng số phản ánh mức độ liên quan của chúng, ví dụ, độ hữu ích khi các item đó được mua cùng nhau.

Khi đồ thị đã được xây dựng, mô hình Graph Attention Network (GAT) sẽ được áp dụng để lan truyền thông tin giữa các nút trong đồ thị. Với GAT, mỗi sản phẩm nhận thông tin từ các khách hàng đã mua nó, cũng như từ các sản phẩm có mối quan hệ cao với nó. Cơ chế attention của GAT cho phép mô hình xác định tầm quan trọng của các mối quan hệ này, giúp trọng số hóa thông tin từ các láng giềng và học được các đặc trưng phức tạp của sản phẩm dựa trên các mối quan hệ tiềm ẩn.

Sau khi huấn luyện, mô hình GNN có thể được sử dụng để tạo ra các dự đoán sản phẩm cho mỗi khách hàng. Và hệ thống sẽ tự động lấy top K sản phẩm được đề xuất ra để gợi ý cho người dùng.

3.4. DỮ LIỆU

Online Retail là một tập dữ liệu giao dịch thực tế được sử dụng rộng rãi trong các nghiên cứu khai phá dữ liệu, đặc biệt trong khai phá các bộ mục hữu ích cao (High Utility Itemset Mining). Tập dữ liệu này được công bố bởi **UCI Machine Learning Repository** và bao gồm các giao dịch từ một công ty bán lẻ trực tuyến không cửa hàng có trụ sở tại Vương quốc Anh, với các giao dịch diễn ra từ **01/12/2010** đến **09/12/2011**.

Nguồn tải dữ liệu:

- UCI Machine Learning Repository - Online Retail Dataset

Mô tả tập dữ liệu

File dữ liệu: **Online Retail.xlsx**

Bảng 3. ***Bảng mô tả dữ liệu Online Retail.xlsx***

Tên cột	Mô tả	Kiểu dữ liệu
InvoiceNo	Số hóa đơn (mã giao dịch duy nhất, bắt đầu bằng chữ 'C' cho giao dịch hủy)	string
StockCode	Mã sản phẩm (duy nhất cho mỗi mặt hàng)	string
Description	Mô tả sản phẩm	string
Quantity	Số lượng hàng hóa được mua (có thể âm trong trường hợp trả hàng)	int64
InvoiceDate	Ngày và giờ giao dịch (không có trong phiên bản tập chỉnh sửa)	datetime
UnitPrice	Giá của mỗi sản phẩm (GBP)	float64
CustomerID	Mã khách hàng (duy nhất cho mỗi khách hàng)	float64
Country	Quốc gia mà giao dịch được thực hiện	string

Thông tin tổng quan:

- **Số lượng dòng**: 541,909 giao dịch.
- **Số lượng khách hàng**: Khoảng 4,000 (tùy vào cách xử lý dữ liệu).
- **Số lượng sản phẩm**: Hơn 4,000 sản phẩm duy nhất.

- **Khoảng thời gian**: 01/12/2010 - 09/12/2011.
- **Đơn vị tiền tệ**: GBP (Bảng Anh).

3.5. PHƯƠNG PHÁP ĐÁNH GIÁ

3.5.1. Phương pháp sử dụng

Đối với HUIs tích hợp vào hệ thống gợi ý

Để đánh giá hiệu quả của hệ thống đề xuất sản phẩm, nhóm sử dụng chỉ số chính là RMSE (Root Mean Square Error). Chỉ số này đo lường mức độ chính xác của các dự đoán mà mô hình đưa ra so với thực tế, giúp xác định độ lệch giữa giá trị dự đoán và giá trị thực tế của xếp hạng phim.

Đối với GNN tích hợp vào hệ thống gợi ý

Để đánh giá mô hình(cross entropy), **Cross-Entropy Loss** (hay **Log Loss**) là một độ đo được sử dụng phổ biến trong các bài toán phân loại, đặc biệt là phân loại nhị phân và đa lớp. Nó đo lường sự khác biệt giữa xác suất dự đoán của mô hình và nhãn thực tế.

Trong một bài toán phân loại, **cross-entropy** tính toán mức độ không chắc chắn trong việc dự đoán của mô hình so với các nhãn thực tế. Cụ thể, nếu mô hình dự đoán xác suất rất cao cho một lớp đúng mà nhãn thực tế là lớp đó, thì **cross-entropy** sẽ có giá trị nhỏ. Ngược lại, nếu mô hình dự đoán sai, thì giá trị **cross-entropy** sẽ cao.

Bên cạnh đó, nhóm còn sử dụng doanh số thực tế làm thước đo bổ sung. Đây là cách tiếp cận thực tiễn, dựa vào tác động trực tiếp của hệ thống đề xuất đến hành vi mua sắm của người dùng. Khi kết hợp hai chỉ số này, nhóm không chỉ đo lường độ chính xác của thuật toán mà còn kiểm tra khả năng mang lại giá trị kinh doanh thực tế, đảm bảo mô hình vừa chính xác vừa hiệu quả.

3.5.2. RMSE (Root Mean Square Error)

RMSE được tính bằng công thức:

$$RMSE = \sqrt{\frac{\Sigma_{ui}(p_{ui} - r_{ui})^2}{n}}$$

RMSE tính toán trên cơ sở tổng bình phương sai số giữa dự đoán và xếp hạng thực tế, sau đó lấy căn bậc hai của kết quả trung bình.

Thông qua việc sử dụng hai chỉ số này, nhóm có thể đánh giá một cách toàn diện và chính xác hiệu quả của hệ thống đề xuất sản phẩm, từ đó tìm ra các điểm mạnh cũng như hạn chế của mô hình để tiếp tục cải thiện.

3.5.3. Cross-Entropy

Công thức tính cross-entropy loss cho bài toán phân loại nhị phân là:

$$Cross - Entropy = -(y\ log\ (p)\ + (1 - y) log\ (1 - p)\ )$$

- y là nhãn thực tế (0 hoặc 1).
- p là xác suất dự đoán của mô hình cho lớp 1 (xác suất dự đoán cho lớp 0 là 1−p1).

Đối với bài toán phân loại đa lớp, công thức cross-entropy sẽ mở rộng thành:

$$Cross - Entropy = -\sum_{c=1}^{C} y_c log\ (p_c)$$

- C là số lớp.
- y_c là nhãn thực tế cho lớp c (0 hoặc 1).
- p_c là xác suất mô hình dự đoán cho lớp c.

Ưu điểm của Cross-Entropy:

- Cross-entropy là một hàm mất mát có tính khả vi (dễ dàng tính toán đạo hàm) và rất thích hợp với các thuật toán tối ưu như Gradient Descent.
- Hàm này giúp mô hình tối ưu hóa việc dự đoán xác suất chính xác cho các lớp.
- Cross-entropy là một lựa chọn phổ biến khi sử dụng mô hình học sâu với các lớp Softmax (đối với phân loại đa lớp) hoặc Sigmoid (đối với phân loại nhị phân).

3.6. MÔI TRƯỜNG THỰC NGHIỆM

3.6.1. Thư viện sử dụng

```
# PyTorch and PyTorch Geometric imports
import torch
import torch.nn as nn
import torch.nn.functional as F
from torch.optim import Adam
from torch.optim.lr_scheduler import StepLR
from torch_geometric.nn import GATConv
from torch_geometric.data import Data
import torch_geometric.transforms as T
# Scikit-learn imports
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_score, recall_score, mean_squared_error
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics.pairwise import cosine_similarity
# Surprise library imports
from surprise import Dataset, Reader, KNNWithMeans, SVD
from surprise.model_selection import train_test_split as surprise_train_test_split
from surprise.accuracy import rmse
# Other imports
import os
import time
import pandas as pd
import numpy as np
from tqdm import tqdm
from joblib import dump, load
from math import sqrt
```

Hình 24. Các thư viện sử dụng

3.6.1.1 sklearn

Chức năng:

- TfidfVectorizer: Trích xuất đặc trưng văn bản bằng TF-IDF.
- KNeighborsRegressor: Thuật toán hồi quy KNN.
- train_test_split: Chia dữ liệu thành tập huấn luyện và kiểm tra.
- mean_squared_error: Tính lỗi bình phương trung bình giữa giá trị thực tế và dự đoán.
- cosine_similarity: Tính độ tương tự cosin giữa các vector.

Mục đích: Hỗ trợ các bước xử lý nội dung, phân chia dữ liệu và đánh giá hiệu suất mô hình.

3.6.1.2 Surprise

Chức năng:

- Dataset và Reader: Quản lý và chuyển đổi dữ liệu đánh giá để sử dụng với các thuật toán đề xuất.

- KNNWithMeans: Thuật toán đề xuất dựa trên hàng xóm gần nhất với tính trung bình đánh giá.
- SVD: Thuật toán Phân tích giá trị riêng, giảm chiều dữ liệu và tìm quan hệ giữa người dùng/phim.
- train_test_split: Chia dữ liệu thành tập huấn luyện và kiểm tra.
- accuracy.rmse: Tính sai số trung bình căn bậc hai để đánh giá mô hình.

Mục đích: Xây dựng, huấn luyện, và đánh giá các mô hình đề xuất như SVD, KNN.

3.6.1.3 Torch

Chức năng:

- torch: Thư viện chính để xây dựng, huấn luyện và triển khai các mô hình học sâu.
- torch.nn: Cung cấp các lớp và hàm cần thiết để xây dựng các mạng nơ-ron.
- torch.nn.functional: Cung cấp các hàm không có trạng thái như activation functions (ReLU, Sigmoid, etc.).
- GATConv: Thuật toán Graph Attention Network (GAT), được sử dụng để xử lý dữ liệu đồ thị với cơ chế attention.
- Adam: Thuật toán tối ưu Adam, phổ biến trong huấn luyện các mô hình học sâu.
- StepLR: Bộ điều chỉnh học theo từng bước (learning rate scheduler), giảm dần learning rate sau mỗi epoch.
- T: Bộ các biến đổi (transformations) cho đồ thị trong torch_geometric.

Mục đích: Xây dựng, huấn luyện và đánh giá các mô hình học sâu sử dụng dữ liệu đồ thị, bao gồm GAT và các phương pháp tối ưu như Adam, kết hợp với các chỉ số đánh giá như precision và recall.

2.6.2. Cấu hình máy thực nghiệm

Google Colab: Nền tảng Máy Tính Đám Mây của Google

Môi trường: Hoạt động trên nền tảng Google Cloud Platform (GCP) với tài nguyên tính toán mạnh mẽ.

Máy chủ: Cung cấp máy chủ ảo với tùy chọn CPU hoặc GPU để tăng tốc tính toán.

Hệ điều hành: Chạy trên Linux với môi trường lập trình Python mặc định.

Thư viện hỗ trợ: Có sẵn NumPy, Pandas, TensorFlow, PyTorch, Matplotlib, Sklearn, Torch và nhiều thư viện khác.

Tài nguyên tính toán: Hỗ trợ CPU, GPU (NVIDIA Tesla K80, T4, P100), và RAM lên đến 25GB.

Lưu trữ dữ liệu: Dữ liệu và notebook được lưu trực tiếp trên Google Drive.

Cấu hình phần mềm: Đi kèm Jupyter Notebook, hỗ trợ cài đặt thêm thư viện qua pip.

Mục đích: Phù hợp cho học máy, xử lý dữ liệu lớn, và nghiên cứu khoa học.

3.7. KẾT QUẢ THỰC NGHIỆM

3.7.1. Độ chính xác các mô hình

Bảng 4. Bảng độ chính xác các mô hình

Algorithm	RMSE
Content KNN	1.0378
User Based	0.9608
Item Based	0.9711
SVD	0.9125

3.7.2. Doanh số của các mô hình

Bảng 5. Bảng doanh số của các mô hình

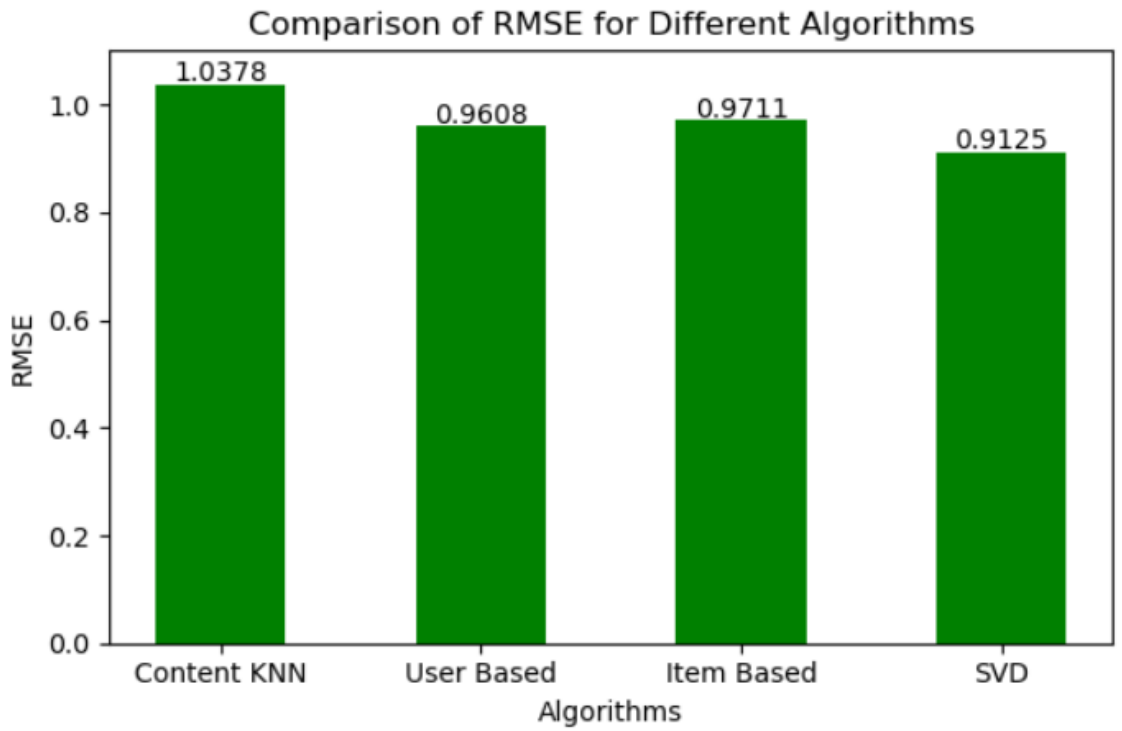
Algorithm	Mô hình đơn	HUI
Content KNN	21.47	39.25
User Based	21.24	38.74
Item Based	20.06	41.35
SVD	23.88	29.28
GNN	22.03	26

3.7.3. GNN

Bảng 6. Bảng độ chính xác mô hình GNN

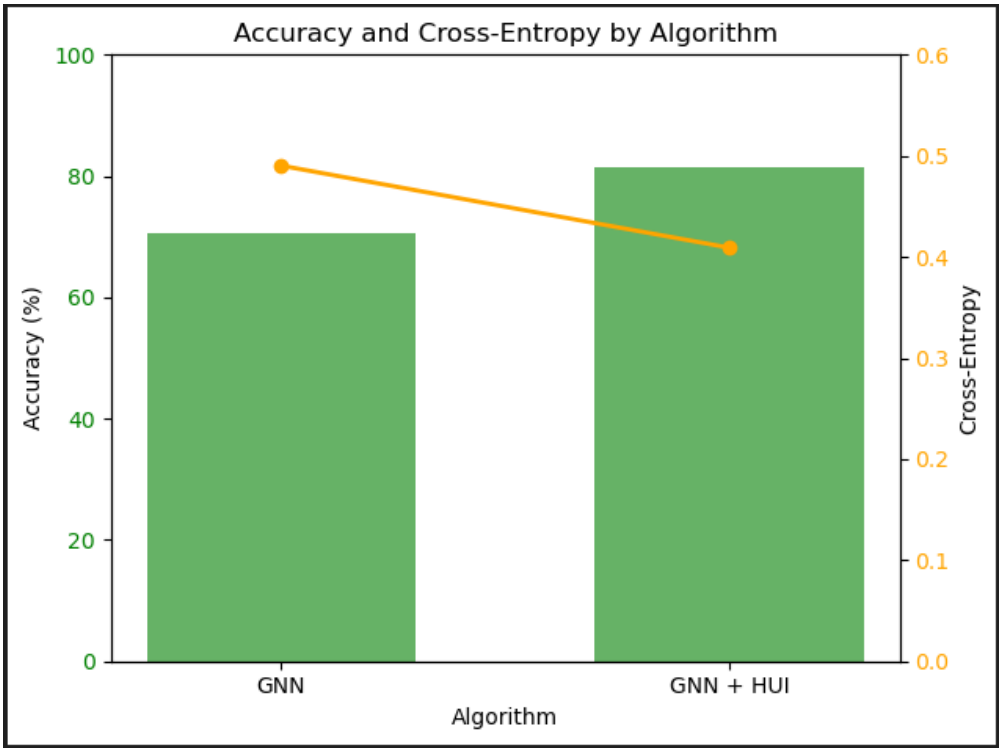
Algorithm	Cross-Entropy test	Accuracy
GNN + HUI	0.4091	81.33%
GNN	0.4903	70,68%

3.7.4. Đánh giá kết quả thực nghiệm



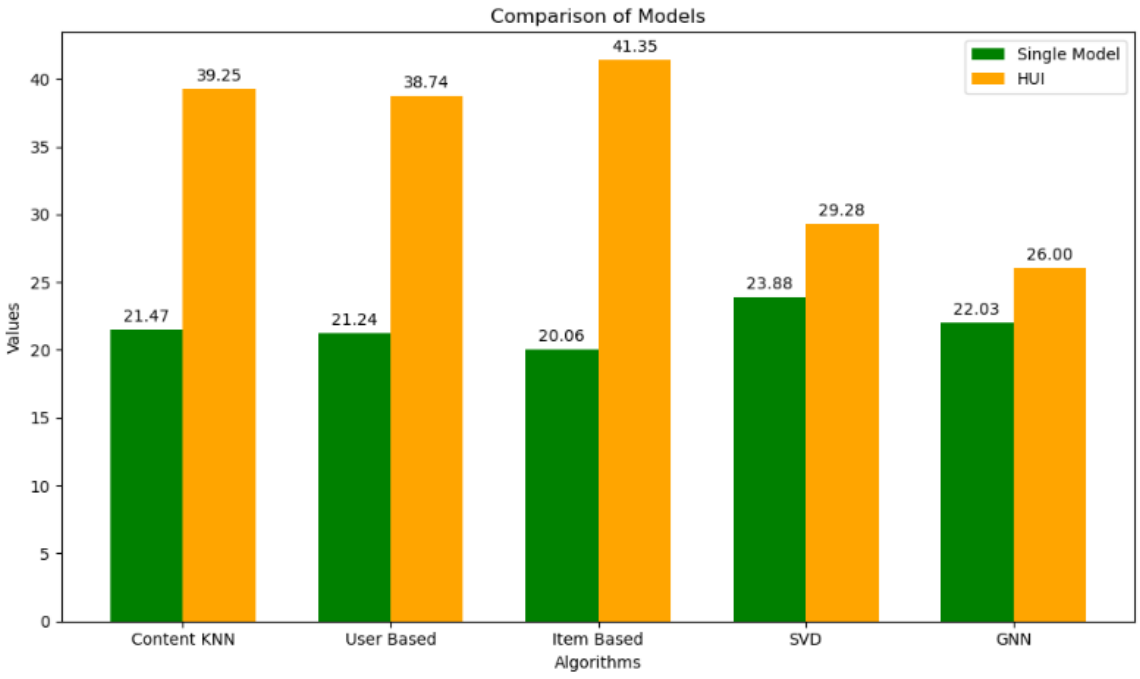
Hình 25. Hình mô tả kết quả RMSE của các thuật toán

Mô hình SVD đạt hiệu suất tốt nhất với chỉ số RMSE thấp nhất, cho thấy khả năng dự đoán chính xác hơn so với các thuật toán khác. Các mô hình User-Based và Item-Based cho kết quả gần nhau và chỉ có sự khác biệt nhỏ so với SVD về mặt hiệu suất. Trong khi đó, Content KNN có độ chính xác thấp hơn, phản ánh rằng mô hình này có thể gặp khó khăn khi chỉ dựa vào thông tin từ "Description" để đưa ra dự đoán.



Hình 26. Hình mô tả sự thay đổi trong độ chính xác và cross-entropy khi áp dụng HUI

Việc sử dụng HUI giúp giảm thiểu lỗi (Cross-Entropy) trong quá trình huấn luyện và kiểm tra, cho thấy việc bổ sung HUI có thể cải thiện hiệu quả của mô hình GNN.Nguyên nhân chính có thể là do các đặc tính giữa các nút tham gia chưa có sự phân biệt quá rõ ràng và việc cấu hình trọng số cạnh giữa các sản phẩm phần thể hiện đặc tính của HUI còn quá đơn giản.



Hình 27. Hình mô tả Doanh thu của các mô hình

Việc kết hợp HUI vào các mô hình đã mang lại sự cải thiện rõ rệt về doanh số. Mô hình Content KNN + HUI ghi nhận doanh số cao nhất trong số các mô hình, đạt 39.25 GBP, cao hơn nhiều so với Content KNN chỉ có 21.47 GBP, với mức tăng 17.78 GBP. Tương tự, User Based + HUI cũng cho thấy sự cải thiện mạnh mẽ khi doanh số tăng từ 21.24 GBP lên 38.74 GBP, tức tăng 17.50 GBP. Mô hình Item Based + HUI đạt doanh số cao nhất trong tất cả các mô hình, với mức tăng doanh số 21.29 GBP, từ 20.06 GBP lên 41.35 GBP. Mặc dù mức tăng doanh số của SVD + HUI và GNN + HUI thấp hơn, nhưng SVD vẫn duy trì doanh số cao ban đầu 23.88 GBP, trong khi GNN mặc dù có mức tăng thấp 3.97 GBP, nhưng vẫn có sự cải thiện đáng kể từ 22.03 GBP lên 26 GBP.

PHẦN KẾT LUẬN

1. Kết quả đạt được của đề tài

Trong quá trình phát triển hệ thống gợi ý, việc kết hợp các phương pháp như HUIM, GNN, collaborative filtering và content-based đã mang lại những kết quả đáng khích lệ. Các phương pháp này không chỉ tối ưu hóa khả năng dự đoán mà còn nâng cao độ chính xác và khả năng tương thích với nhu cầu của người dùng.

Ứng dụng HUIM trong hệ thống gợi ý đã chứng minh được ưu thế trong việc xác định các sản phẩm có giá trị thực sự đối với người dùng. Với HUIM, hệ thống không chỉ dựa trên mức độ phổ biến của các sản phẩm mà còn khai thác các tính năng hữu ích, chẳng hạn như các yếu tố về hành vi mua sắm của người dùng, mức độ ưu tiên các sản phẩm cao, từ đó giúp nâng cao độ chính xác của các đề xuất. Việc kết hợp với kỹ thuật khai thác tập hợp item có giá trị cao giúp giảm thiểu khả năng đưa ra các gợi ý không hữu ích, từ đó tăng sự hài lòng của người dùng.

Collaborative Filtering và Content-based methods cũng được áp dụng đồng thời để nâng cao hiệu suất của hệ thống. Collaborative filtering giúp khai thác thông tin từ hành vi của người dùng khác để đưa ra các gợi ý, trong khi phương pháp content-based lại tập trung vào các đặc trưng của sản phẩm và sở thích của người dùng để đưa ra các đề xuất cá nhân hóa. Kết hợp hai phương pháp này giúp hệ thống đạt được sự chính xác cao hơn trong việc dự đoán các sản phẩm mà người dùng có thể quan tâm, đồng thời giảm thiểu vấn đề thiếu dữ liệu người dùng mới (cold-start problem).

Graph Neural Networks đã đóng vai trò quan trọng trong việc tối ưu hóa khả năng học các mối quan hệ phức tạp giữa người dùng và các sản phẩm. Thông qua việc xây dựng các đồ thị tương tác, GNN giúp mô hình học được sự liên kết và sự ảnh hưởng giữa các sản phẩm cũng như người dùng. Điều này đặc biệt hữu ích trong các tình huống mà người dùng có thể chưa trực tiếp tương tác với một sản phẩm nhưng lại có thể nhận được gợi ý chính xác thông qua các sản phẩm tương tự hoặc sản phẩm được ưa chuộng bởi những người dùng có hành vi tương tự.

Các thử nghiệm thực tế cho thấy hệ thống đạt được kết quả ấn tượng với các chỉ số đánh giá RMSE thấp với ba mô hình dưới 1 trong tổng số bốn mô hình thử nghiệm, độ chính xác mô hình GNN kết hợp HUI tương đối cao tầm 81,33% thấp cùng doanh số cao (hơn 66,38 GDP so với không áp dụng HUI), chứng tỏ rằng hệ thống gợi ý hoạt động hiệu quả và có thể cung cấp những đề xuất chính xác cho người dùng. Việc tích hợp HUIM và GNN đã giúp hệ thống không chỉ tối ưu hóa được sự chính xác trong việc dự đoán mà còn nâng cao khả năng phát hiện các mối quan hệ tiềm ẩn giữa các sản phẩm, từ đó cải thiện trải nghiệm người dùng.

Hệ thống đã được kiểm nghiệm trong các bài toán thực tế như gợi ý sản phẩm cho người mua sắm trực tuyến và gợi ý phim cho người dùng các nền tảng giải trí. Những kết quả này không chỉ chứng tỏ hiệu quả của mô hình mà còn khẳng định khả năng ứng dụng rộng rãi của các phương pháp kết hợp như HUIM, GNN, collaborative filtering, và content-based trong các hệ thống gợi ý. Các kết quả này góp phần nâng cao sự hài lòng của người dùng và tăng trưởng doanh thu cho các nền tảng thương mại điện tử.

2. Hạn chế

Mặc dù hệ thống gợi ý sử dụng HUIM, GNN và các phương pháp collaborative filtering và content-based đã đạt được nhiều thành công, nhưng vẫn còn một số hạn chế đáng kể cần được khắc phục để tăng cường hiệu quả và tính ứng dụng thực tế của hệ thống.

Một trong những vấn đề lớn nhất là chất lượng và độ đầy đủ của dữ liệu đầu vào. Hệ thống gợi ý phụ thuộc rất nhiều vào dữ liệu lịch sử hành vi người dùng và đặc trưng sản phẩm để đưa ra các gợi ý chính xác. Tuy nhiên, trong môi trường thực tế, dữ liệu có thể thiếu hoặc không đầy đủ, đặc biệt là đối với những người dùng mới (vấn đề cold-start) hoặc đối với những sản phẩm ít được tương tác. Việc thiếu dữ liệu hoặc dữ liệu không đủ đa dạng có thể ảnh hưởng đến độ chính xác của hệ thống, khiến cho các đề xuất không được tối ưu hoặc không phù hợp với nhu cầu thực tế của người dùng.

Vấn đề cold-start là một trong những thách thức lớn khi sử dụng collaborative filtering và content-based methods. Đối với những người dùng mới hoặc sản phẩm mới, hệ thống gặp khó khăn trong việc đưa ra các đề xuất chính xác vì không có đủ dữ liệu về hành vi tương tác trước đó. Mặc dù HUIM và GNN có thể phần nào giải quyết vấn đề này bằng cách khai thác thông tin từ các mối quan hệ giữa các sản phẩm hoặc người dùng tương tự, nhưng vẫn cần có một chiến lược hiệu quả để giải quyết vấn đề cold-start, đặc biệt là khi dữ liệu ít hoặc không có sự tương tác giữa người dùng và sản phẩm.

Một hạn chế khác là chi phí tính toán và tài nguyên yêu cầu. Mặc dù các phương pháp như GNN có thể mang lại những cải thiện lớn về chất lượng gợi ý, nhưng chúng cũng đòi hỏi một lượng tài nguyên tính toán lớn và khả năng xử lý cao, đặc biệt khi số lượng người dùng và sản phẩm tăng lên. Việc xử lý đồ thị với GNN có thể làm tăng thời gian tính toán và tiêu tốn tài nguyên hệ thống, điều này đặc biệt quan trọng khi triển khai hệ thống trên các nền tảng có tài nguyên hạn chế hoặc khi cần phải đáp ứng yêu cầu về thời gian thực.

Độ phức tạp trong việc kết hợp các phương pháp cũng là một thách thức khác. Mặc dù việc kết hợp HUIM, GNN, collaborative filtering và content-based có thể mang lại hiệu quả cao, nhưng sự phối hợp giữa các phương pháp này không phải lúc nào cũng dễ dàng. Việc tích hợp nhiều mô hình khác nhau đòi hỏi một kiến trúc phức tạp và khó duy trì, đặc biệt khi cần cập nhật hoặc thay đổi các thành phần của hệ thống mà không ảnh hưởng đến

các phần còn lại. Điều này có thể gây khó khăn trong quá trình triển khai và bảo trì hệ thống.

Một hạn chế lớn trong nghiên cứu này là thuật toán EIHI hiện chưa có thư viện hỗ trợ sẵn, khiến quá trình triển khai phải chuyển mã nguồn từ Java sang Python. Điều này đòi hỏi nhiều thời gian và công sức để điều chỉnh và tối ưu mã, đặc biệt khi muốn áp dụng thuật toán này vào các dự án khác. Vì vậy, nếu muốn triển khai EIHI ở những môi trường khác, thời gian thực hiện sẽ lâu và yêu cầu sự chuẩn bị kỹ lưỡng.

Mặc dù EIHI có ưu điểm trong việc bảo lưu dữ liệu cũ khi thêm dữ liệu mới mà không cần chạy lại toàn bộ, mô hình chưa ứng dụng được ưu điểm này, do đó chưa tận dụng được khả năng gia tăng hiệu suất khi sử dụng EIHI.

Cuối cùng, khả năng xử lý các mối quan hệ phức tạp giữa người dùng và sản phẩm cũng còn hạn chế. Dù GNN có khả năng học các mối quan hệ phức tạp trong đồ thị, nhưng trong một số trường hợp, đặc biệt là với các tập dữ liệu lớn và đa dạng, hệ thống có thể gặp khó khăn trong việc nắm bắt được tất cả các yếu tố ảnh hưởng đến sự tương tác của người dùng. Điều này có thể dẫn đến việc hệ thống không nhận diện được các mẫu hành vi tiềm ẩn hoặc sự thay đổi trong sở thích của người dùng theo thời gian.

3. Hướng phát triển

Trong tương lai, hệ thống gợi ý sử dụng HUIM, GNN, collaborative filtering và content-based sẽ tiếp tục được phát triển mạnh mẽ để nâng cao khả năng cung cấp các gợi ý chính xác và cá nhân hóa hơn. Việc kết hợp các mô hình tiên tiến sẽ giúp hệ thống vượt qua những thách thức hiện tại và mở ra những cơ hội mới trong việc tối ưu hóa trải nghiệm người dùng.

GNN và HUIM sẽ tiếp tục đóng vai trò quan trọng trong việc nâng cao khả năng phân tích và hiểu mối quan hệ phức tạp giữa người dùng và sản phẩm. GNN, với khả năng học hỏi từ các cấu trúc đồ thị, sẽ giúp hệ thống hiểu rõ hơn về các mối quan hệ tiềm ẩn giữa các sản phẩm và người dùng, từ đó đưa ra những gợi ý chính xác hơn. HUIM, với khả năng tối ưu hóa các mục tiêu gợi ý dựa trên giá trị cao của các sản phẩm, sẽ giúp nâng cao độ chính xác và sự phù hợp của các đề xuất, đặc biệt là trong các tình huống người dùng có nhu cầu đặc thù hoặc trong các lĩnh vực chuyên biệt.

Collaboration Filtering sẽ tiếp tục được cải tiến thông qua việc tích hợp dữ liệu đa dạng hơn và áp dụng các thuật toán học sâu để giải quyết vấn đề cold-start. Các phương pháp học máy như deep learning-based collaborative filtering sẽ giúp mô hình học được các mẫu hành vi phức tạp của người dùng, ngay cả khi dữ liệu đầu vào chưa đủ lớn. Đặc biệt, sự kết hợp giữa collaborative filtering và content-based methods sẽ giúp hệ thống phát

triển khả năng gợi ý các sản phẩm mới hoặc ít tương tác nhưng vẫn phù hợp với sở thích của người dùng, từ đó giảm thiểu hiện tượng cold-start.

Content-based methods sẽ tiếp tục được tối ưu hóa với sự phát triển của các mô hình ngôn ngữ như T5 hoặc các mô hình học sâu khác. T5, với kiến trúc mạnh mẽ, có khả năng xử lý và phân tích thông tin nội dung sản phẩm một cách sâu sắc, giúp hệ thống hiểu được các đặc trưng tinh vi của sản phẩm và mối quan hệ của chúng với sở thích của người dùng. Bằng cách này, hệ thống có thể cung cấp các gợi ý không chỉ dựa trên hành vi người dùng mà còn dựa trên đặc tính nội tại của sản phẩm.

Tích hợp dữ liệu ngoài thông qua các mô hình như RAG sẽ là yếu tố cốt lõi trong việc nâng cao tính linh hoạt của hệ thống gợi ý. Việc khai thác dữ liệu từ các nguồn bên ngoài, như thông tin thị trường, xu hướng tiêu dùng, hoặc các bài báo và báo cáo ngành, sẽ giúp hệ thống luôn cập nhật với các xu hướng mới nhất và cung cấp các gợi ý hợp thời. Điều này sẽ giúp người dùng luôn nhận được những gợi ý phù hợp với nhu cầu thực tế của họ, dù những nhu cầu đó thay đổi theo thời gian.

Bên cạnh đó, việc tối ưu hóa quy trình huấn luyện mô hình bằng các phương pháp học liên tục (continual learning) sẽ giúp hệ thống luôn cập nhật với dữ liệu mới mà không cần phải huấn luyện lại toàn bộ mô hình. Điều này không chỉ giúp tiết kiệm tài nguyên tính toán mà còn đảm bảo rằng hệ thống có thể đáp ứng nhanh chóng với các thay đổi trong hành vi người dùng và xu hướng thị trường.

Tích hợp ưu điểm của EIHI trong việc bảo lưu dữ liệu cũ khi thêm dữ liệu mới, tránh việc phải chạy lại toàn bộ quá trình xử lý. Việc này sẽ giúp cải thiện hiệu suất hệ thống bằng cách chỉ xử lý các phần dữ liệu thay đổi, giảm thiểu tài nguyên tính toán và thời gian xử lý, đồng thời duy trì tính chính xác và hiệu quả trong việc cập nhật dữ liệu mới.

Một hướng phát triển tiềm năng cho Graph Attention Networks (GAT) trong hệ thống gợi ý sản phẩm là tích hợp Recurrent Neural Networks (RNN) để xử lý các tác vụ dự đoán dãy sản phẩm, đặc biệt trong những tình huống mà hành vi người dùng thay đổi theo thời gian. GAT giúp xây dựng mối quan hệ giữa các sản phẩm thông qua trọng số chú ý, trong khi RNN có thể xử lý chuỗi hành vi mua sắm của người dùng, từ đó dự đoán các sản phẩm tiềm năng trong tương lai. Việc kết hợp GAT và RNN cho phép mô hình học được không chỉ mối quan hệ tĩnh giữa các sản phẩm mà còn xu hướng thay đổi trong hành vi người dùng, cải thiện chất lượng gợi ý, đặc biệt trong những trường hợp người dùng có hành vi mua sắm biến đổi theo thời gian.

Cuối cùng, hệ thống sẽ tiếp tục được phát triển để tối ưu hóa trải nghiệm người dùng, không chỉ thông qua các gợi ý chính xác mà còn bằng cách hiểu và dự đoán nhu cầu của người dùng dựa trên các yếu tố ngữ cảnh. Việc cải tiến các phương pháp phân tích ngữ nghĩa và hiểu ngữ cảnh sẽ giúp hệ thống không chỉ đưa ra các gợi ý đơn thuần mà còn dự

đoán được những sản phẩm mà người dùng có thể quan tâm trong tương lai, dù chưa tương tác trực tiếp với chúng.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] K. Hong, H. Jeon, and C. Jeon, "UserProfile-based personalized research paper recommendation system," in *2012 8th International Conference on Computing and Networking Technology (INC, ICCIS and ICMIC)*, Aug. 2012, pp. 134–138. Accessed: Oct. 02, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6418639

[2] M. B. Magara, S. O. Ojo, and T. Zuva, "Towards a Serendipitous Research Paper Recommender System Using Bisociative Information Networks (BisoNets)," in *2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, Aug. 2018, pp. 1–6. doi: 10.1109/ICABCD.2018.8465475.

[3] J. Ha, S.-W. Kim, S.-W. Kim, C. Faloutsos, and S. Park, "Phân tích về sự lan truyền thông tin qua *BlogCast* trong thế giới blog," *Inf. Sci.*, vol. 290, pp. 45–62, Jan. 2015, doi: 10.1016/j.ins.2014.08.042.

[4] H. Liu, X. Kong, X. Bai, W. Wang, T. M. Bekele, and F. Xia, "Context-Based Collaborative Filtering for Citation Recommendation," *IEEE Access*, vol. 3, pp. 1695–1703, 2015, doi: 10.1109/ACCESS.2015.2481320.

[5] M. Dhanda and V. Verma, "Recommender System for Academic Literature with Incremental Dataset," *Procedia Comput. Sci.*, vol. 89, pp. 483–491, Jan. 2016, doi: 10.1016/j.procs.2016.06.109.

[6] P. Fournier-Viger, J. C.-W. Lin, T. Gueniche, and P. Barhate, "Efficient Incremental High Utility Itemset Mining," in *Proceedings of the ASE BigData & SocialInformatics 2015*, in ASE BD&SI '15. New York, NY, USA: Association for Computing Machinery, Tháng Mười 2015, pp. 1–6. doi: 10.1145/2818869.2818887.

[7] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph Convolutional Neural Networks for Web-Scale Recommender Systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, in KDD '18. New York, NY, USA: Association for Computing Machinery, Tháng Bảy 2018, pp. 974–983. doi: 10.1145/3219819.3219890.

[8] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, in SIGIR '20. New York, NY, USA: Association for Computing Machinery, Tháng Bảy 2020, pp. 639–648. doi: 10.1145/3397271.3401063.

[9] F. Liu, Z. Cheng, L. Zhu, Z. Gao, and L. Nie, "Interest-aware Message-Passing GCN for Recommendation," in *Proceedings of the Web Conference 2021*, in WWW '21.

New York, NY, USA: Association for Computing Machinery, Tháng Sáu 2021, pp. 1296–1305. doi: 10.1145/3442381.3449986.

[10] H. Liu, Y. Wei, J. Yin, and L. Nie, “HS-GCN: Hamming Spatial Graph Convolutional Networks for Recommendation,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 5977–5990, Jun. 2023, doi: 10.1109/TKDE.2022.3158317.

[11] J. Tian, J. Zhao, Z. Wang, and Z. Ding, “MMREC: LLM Based Multi-Modal Recommender System,” arXiv.org. Accessed: Oct. 02, 2024. [Online]. Available: https://arxiv.org/abs/2408.04211v1

[12] T. Vu, “Bài 25: Matrix Factorization Collaborative Filtering,” Tiep Vu’s blog. Accessed: Oct. 02, 2024. [Online]. Available: https://machinelearningcoban.com/2017/05/31/matrixfactorization/

[13] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*. Cambridge University Press, 2010.

[14] S. Zida, P. Fournier-Viger, J. C.-W. Lin, C.-W. Wu, and V. S. Tseng, “EFIM: A Highly Efficient Algorithm for High-Utility Itemset Mining,” in *Advances in Artificial Intelligence and Soft Computing*, G. Sidorov and S. N. Galicia-Haro, Eds., Cham: Springer International Publishing, 2015, pp. 530–546. doi: 10.1007/978-3-319-27060-9_44.

[15] “Mining high utility itemsets without candidate generation | Proceedings of the 21st ACM international conference on Information and knowledge management.” Accessed: Dec. 07, 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/2396761.2396773

[16] Y. Liu, W. Liao, and A. Choudhary, “A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets,” in *Advances in Knowledge Discovery and Data Mining*, T. B. Ho, D. Cheung, and H. Liu, Eds., Berlin, Heidelberg: Springer, 2005, pp. 689–695. doi: 10.1007/11430919_79.

[17] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, “Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 12, pp. 1708–1721, Oct. 2009, doi: 10.1109/TKDE.2009.46.

[18] M. Liu and J. Qu, “Mining high utility itemsets without candidate generation,” in *Proceedings of the 21st ACM international conference on Information and knowledge management*, in CIKM ’12. New York, NY, USA: Association for Computing Machinery, Tháng Mười 2012, pp. 55–64. doi: 10.1145/2396761.2396773.

[19] P. Fournier-Viger, C.-W. Wu, S. Zida, and V. S. Tseng, “FHM: Faster High-Utility Itemset Mining Using Estimated Utility Co-occurrence Pruning,” in *Foundations of Intelligent Systems*, T. Andreasen, H. Christiansen, J.-C. Cubero, and

Z. W. Raś, Eds., Cham: Springer International Publishing, 2014, pp. 83–92. doi: 10.1007/978-3-319-08326-1_9.

[20] "Encyclopedia of Continuum Mechanics | SpringerLink." Accessed: Dec. 07, 2024. [Online]. Available: https://link.springer.com/referencework/10.1007/978-3-662-55771-6

[21] "A new model for learning in graph domains | IEEE Conference Publication | IEEE Xplore." Accessed: Dec. 07, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/1555942

[22] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node Classification in Social Networks," arXiv.org. Accessed: Dec. 07, 2024. [Online]. Available: https://arxiv.org/abs/1101.3291v1

[23] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," Sep. 10, 2018, *arXiv*: arXiv:1706.02216. doi: 10.48550/arXiv.1706.02216.

[24] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," Feb. 22, 2017, *arXiv*: arXiv:1609.02907. doi: 10.48550/arXiv.1609.02907.

[25] Z. Stanfield, M. Coşkun, and M. Koyutürk, "Drug Response Prediction as a Link Prediction Problem," *Sci. Rep.*, vol. 7, no. 1, p. 40321, Jan. 2017, doi: 10.1038/srep40321.

[26] "Graph clustering based on Structural Attribute Neighborhood Similarity (SANS) | IEEE Conference Publication | IEEE Xplore." Accessed: Dec. 07, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/7226087

[27] "Netprobe | Proceedings of the 16th international conference on World Wide Web." Accessed: Dec. 07, 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/1242572.1242600

[28] F. Errica, M. Podda, D. Bacciu, and A. Micheli, "A Fair Comparison of Graph Neural Networks for Graph Classification," Feb. 17, 2022, *arXiv*: arXiv:1912.09893. doi: 10.48550/arXiv.1912.09893.

[29] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009, doi: 10.1109/TNN.2008.2005605.

[30] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated Graph Sequence Neural Networks," Sep. 22, 2017, *arXiv*: arXiv:1511.05493. doi: 10.48550/arXiv.1511.05493.

[31] "SPMF: A Java Open-Source Data Mining Library." Accessed: Dec. 08, 2024. [Online]. Available: https://www.philippe-fournier-viger.com/spmf/index.php