TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TRỊNH VĨNH KHANG – 51703111**

**NGUYỄN THANH NGHĨA - 51900390**

**NGHIÊN CỨU VỀ ITEMSET MINING VÀ ASSOCIATION RULES**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN 2**

**KHOA HỌC MÁY TÍNH**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TRỊNH VĨNH KHANG – 51703111**

**NGUYỄN THANH NGHĨA - 51900390**

**NGHIÊN CỨU VỀ ITEMSET MINING VÀ ASSOCIATION RULES**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN 2**

**KHOA HỌC MÁY TÍNH**

Người hướng dẫn

**ThS.DOÃN XUÂN THANH**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**LỜI CẢM ƠN**

Lời đầu tiên chúng em xin chân thành cảm ơn quý thầy, cô trong trường, đặc biệt là quý thầy, cô khoa Công Nghệ Thông Tin đã tận tình giảng dạy, chỉ bảo chúng em trong những năm học vừa qua, những kiến thức mà chúng em nhận được trên giảng đường đại học sẽ là nền tảng cho quá trình nghiên cứu đề tài này cũng như là hành trang vững chắc giúp chúng em bước vào công việc chính thức sau này.

Chúng em xin chân thành cảm ơn thầy Doãn Xuân Thanh – giảng viên hướng dẫn chúng em môn học Dự án Công nghệ thông tin 2 đã luôn theo sát, nhiệt tình giúp đỡ chúng em trong suốt quá trình thực hiện đề tài này.

Chúng em xin chân thành cảm ơn Khoa Công nghệ thông tin đã tạo điều kiện để em có thể thực hiện đề tài nghiên cứu này

Bài báo cáo này được hoàn thành không chỉ là sự nỗ lực của bản thân chúng em, mà còn nhờ có sự ủng hộ, động viên và nhiệt tình hỗ trợ từ các thầy hướng dẫn, bạn bè, người thân và nhà trường. Mặc dù bản thân chúng em đã nỗ lực hết mức, nhưng bài báo cáo này không thể tránh khỏi còn nhiều thiếu sót. Vì thế, chúng em mong rằng sẽ nhận được những lời nhận xét, góp ý từ quý thầy, cô để chúng em có thể tiếp tục phát triển và hoàn thiện hơn nữa.

Một lần nữa, chúng em xin chân thành cảm ơn!

*TP. Hồ Chí Minh, ngày ... tháng … năm 20..*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

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

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

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

*TP. Hồ Chí Minh, ngày … tháng … năm 20..*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

**PHÂN TÍCH HÌNH TƯỢNG QUÁN THẾ ÂM BỒ TÁT**

**Ở BÌNH DƯƠNG QUA HÌNH ẢNH**

**TÓM TẮT**

Đề tài này sẽ tập trung nghiên cứu về hình tượng Quán Thế Âm Bồ Tát ở các chùa trong khu vực tỉnh Bình Dương thông qua bộ sưu tập ảnh tự chụp tại các ngôi chùa trong tỉnh.

Bài báo cáo sẽ bao gồm năm chương:

CHƯƠNG 1: Sẽ khái quát về đề tài và phương pháp nghiên cứu đề tài.

CHƯƠNG 2: Trình bày về quá trình du nhập của Phật giáo vào Việt Nam và tỉnh Bình Dương, đồng thời tìm hiểu về hình ảnh Quán Thế Âm Bồ Tát.

CHƯƠNG 3: Bao gồm cơ sở lý thuyết được áp dụng vào việc phân tích của đề tài.

CHƯƠNG 4: Trình bày về quá trình thu thập dữ liệu, xử lý và phân tích dữ liệu, kết quả đạt được.

CHƯƠNG 5: Kết luận và hướng phát triển đề tài.

**MỤC LỤC**

[DANH MỤC HÌNH VẼ vi](#_Toc160929937)

[DANH MỤC BẢNG BIỂU ix](#_Toc160929938)

[DANH MỤC CÁC CHỮ VIẾT TẮT x](#_Toc160929939)

[CHƯƠNG 1. TỔNG QUAN 1](#_Toc160929940)

[1.1 Giới thiệu đề tài 1](#_Toc160929941)

[1.2 Phương pháp nghiên cứu và ý nghĩa thực tiễn 4](#_Toc160929942)

[CHƯƠNG 2. GIỚI THIỆU SƠ VỀ VẤN ĐỀ NGHIÊN CỨU 6](#_Toc160929943)

[2.1 Khai thác dữ liệu 6](#_Toc160929944)

[2.2 Mục tiêu và ý nghĩa của khai thác dữ liệu 7](#_Toc160929945)

[CHƯƠNG 3. CƠ SỞ LÝ THUYẾT 9](#_Toc160929946)

[3.1 Itemset mining 9](#_Toc160929947)

[3.1.1 Định nghĩa 9](#_Toc160929948)

[3.1.2 Ứng dụng thực tế 10](#_Toc160929949)

[3.1.3 Cách thực hiện 11](#_Toc160929950)

[3.1.4 Thuật toán FIN 12](#_Toc160929951)

[3.2 Association rule mining 18](#_Toc160929952)

[3.2.1 Định nghĩa 18](#_Toc160929953)

[3.2.2 Ứng dụng thực tế 19](#_Toc160929954)

[3.2.3 Cách thực hiện 21](#_Toc160929955)

[3.2.4 Thuật toán Argawal 21](#_Toc160929956)

[CHƯƠNG 4. THỰC NGHIỆM 24](#_Toc160929957)

[4.1 Thu thập dữ liệu 24](#_Toc160929958)

[4.2 Thư viên được sử dụng 26](#_Toc160929959)

[4.3 Thuật toán FIN 27](#_Toc160929960)

[4.4 Thuật toán Agrawal 41](#_Toc160929961)

[4.5 Website hệ thống khuyến nghị 46](#_Toc160929962)

[CHƯƠNG 5. KẾT LUẬN 47](#_Toc160929963)

[5.1 Kết luận 47](#_Toc160929964)

[5.2 Hướng phát triển 47](#_Toc160929965)

[TÀI LIỆU THAM KHẢO 49](#_Toc160929966)

# DANH MỤC HÌNH VẼ

[Hình 1.1 Mô hình thươn mại điện tử (VJCC) 1](#_Toc160910620)

[Hình 2.1 Tổng quan về data mining (Netsuite) 6](#_Toc160910621)

[Hình 3.1 Mô tả về frequent itemset mining (Nguồn: Datacamp) 12](#_Toc160910622)

[Hình 3.2 Kết quả ví dụ POC-tree 14](#_Toc160910623)

[Hình 3.3 Mã giả thuật toán FIN (Deng & Lv, 2014) 16](#_Toc160910624)

[Hình 3.4 Mã giả hàm CPT (Deng & Lv, 2014) 18](#_Toc160910625)

[Hình 3.5 Mã giả thuật toán Agrawal (Agrawal & Srikant, 1994) 22](#_Toc160910626)

[Hình 3.6 Mã giả hàm apriori-gen (Agrawal & Srikant, 1994) 23](#_Toc160910627)

[Hình 4.1 Bộ dữ liệu chess 24](#_Toc160929902)

[Hình 4.2 Bộ dữ liệu mushrooms 25](#_Toc160929903)

[Hình 4.3 Bộ dữ liệu transaction\_data 26](#_Toc160929904)

[Hình 4.4 Các thư viện được sử dụng 27](#_Toc160929905)

[Hình 4.5 Các file dữ liệu đầu vào 27](#_Toc160929906)

[Hình 4.6 Biến dữ liệu 27](#_Toc160929907)

[Hình 4.7 Hàm khởi tạo thuật toán FIN (1) 28](#_Toc160929908)

[Hình 4.8 Hàm khởi tạo thuật toán FIN (2) 28](#_Toc160929909)

[Hình 4.9 Hàm runAlgorithm (1) 29](#_Toc160929910)

[Hình 4.10 Hàm runAlgorithm (2) 30](#_Toc160929911)

[Hình 4.11 Hàm buildTree (1) 31](#_Toc160929912)

[Hình 4.12 Hàm buildTree (2) 32](#_Toc160929913)

[Hình 4.13 Hàm buildTree (3) 33](#_Toc160929914)

[Hình 4.14 Hàm buildTree (4) 34](#_Toc160929915)

[Hình 4.15 Hàm initializeTree 35](#_Toc160929916)

[Hình 4.16 Hàm getData 36](#_Toc160929917)

[Hình 4.17 Hàm traverse (1) 37](#_Toc160929918)

[Hình 4.18 Hàm traverse (2) 37](#_Toc160929919)

[Hình 4.19 Hàm is2\_itemSetValid 38](#_Toc160929920)

[Hình 4.20 Hàm iskItemSetFreq (1) 39](#_Toc160929921)

[Hình 4.21 Hàm iskItemSetFreq (2) 39](#_Toc160929922)

[Hình 4.22 Hàm writeItemsetsToFile 40](#_Toc160929923)

[Hình 4.23 Hàm getFrequentItemsets 41](#_Toc160929924)

[Hình 4.24 Kết quả FIN với dữ liệu chess 41](#_Toc160929925)

[Hình 4.25 Kết quả FIN với dữ liệu mushrooms 41](#_Toc160929926)

[Hình 4.26 Hàm khởi tạo Agrawal 42](#_Toc160929927)

[Hình 4.27 Hàm runAlgorithm 42](#_Toc160929928)

[Hình 4.28 Hàm runAlgorithmHelper (1) 43](#_Toc160929929)

[Hình 4.29 Hàm runAlgorithmHelper (2) 43](#_Toc160929930)

[Hình 4.30 Hàm apGenrules (1) 44](#_Toc160929931)

[Hình 4.31 Hàm apGenrules (2) 44](#_Toc160929932)

[Hình 4.32 Hàm calculateSupport 44](#_Toc160929933)

[Hình 4.33 Hàm generateCandidateSize 45](#_Toc160929934)

[Hình 4.34 Kết quả Agrawal với dữ liệu chess 45](#_Toc160929935)

[Hình 4.35 Kết quả Agrawal với dữ liệu mushrooms 46](#_Toc160929936)

# DANH MỤC BẢNG BIỂU

[Bảng 3.1 Bảng dữ liệu transaction ví dụ POC-tree 14](#_Toc160884392)

[Bảng 3.2 Các tập P của từng transaction 14](#_Toc160884393)

[Bảng 3.3 Bảng dữ liệu ví dụ Association Rule 21](#_Toc160884394)

[Bảng 4.1 Thông tin về hai bộ dữ liệu FIMI 25](#_Toc160927904)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| CPT | Constructing Pattern Tree |
| DB | Database |
| minconf | Minimum Confidence |
| minsup | Minimum Support |
| POC | Post-order Code |
| DBSCAN | Density-Based Spatial Clustering of Applications with Noise |

# TỔNG QUAN

## Giới thiệu đề tài

Trải qua những bước đột phá đầy nhanh chóng, sự phát triển của thương mại điện tử đã đánh dấu một cuộc cách mạng trong lĩnh vực kinh doanh toàn cầu. Từ khi xuất hiện lần đầu tiên vào cuối thập kỷ 20, thương mại điện tử đã trải qua một hành trình đáng kinh ngạc, biến đổi cách mà người tiêu dùng mua sắm và doanh nghiệp tiếp cận thị trường.

Ban đầu, thương mại điện tử tập trung chủ yếu vào giao dịch qua mạng Internet. Tuy nhiên, với sự gia tăng của công nghệ di động, người tiêu dùng ngày càng có thêm sự thuận lợi và linh hoạt khi thực hiện các giao dịch từ điện thoại di động hay máy tính bảng. Điều này đã mở ra cánh cửa cho sự đa dạng hóa về trải nghiệm người dùng và tăng cường khả năng tiếp cận thị trường cho các doanh nghiệp.

A hand holding a bag and a computer

Description automatically generated

Hình 1.1 Mô hình thươn mại điện tử ([VJCC](https://vjcc.org.vn/ky-nang-mem/thuong-mai-dien-tu-ecommerce.html))

Mô hình kinh doanh của thương mại điện tử cũng đã phát triển từ các trang web đơn giản để bán hàng đến các nền tảng phức tạp hơn, tích hợp nhiều dịch vụ như thanh toán trực tuyến, quản lý kho hàng, và dịch vụ khách hàng. Các doanh nghiệp cũng đã áp dụng các công nghệ mới như trí tuệ nhân tạo, phân tích dữ liệu và học máy để tối ưu hóa trải nghiệm mua sắm và tăng cường khả năng dự đoán xu hướng thị trường.

Thương mại điện tử mang lại nhiều lợi ích quan trọng cho cả người tiêu dùng và doanh nghiệp:

* Tiện ích và linh hoạt: người tiêu dùng có thể mua sắm bất cứ lúc nào, ở bất cứ đâu, không phụ thuộc vào thời gian hay địa điểm. Điều này mang lại sự linh hoạt lớn cho người tiêu dùng và tăng cường trải nghiệm mua sắm.
* Tiết kiệm thời gian và năng lượng: không cần di chuyển đến cửa hàng, người tiêu dùng có thể tiết kiệm thời gian và công sức. Điều này đặc biệt quan trọng trong cuộc sống ngày nay nơi mọi người ngày càng bận rộn.
* Đa dạng và dễ dàng so sánh: thương mại điện tử cung cấp sự đa dạng về sản phẩm và dịch vụ. Người tiêu dùng có thể dễ dàng so sánh giá cả và chất lượng giữa các nhà cung cấp khác nhau để đảm bảo sự lựa chọn tốt nhất.
* Giảm chi phí hoạt động cho doanh nghiệp: doanh nghiệp trực tuyến thường không cần chi trả cho mặt bằng vật lý, giúp giảm thiểu chi phí hoạt động so với cửa hàng truyền thống.
* Quảng cáo hiệu quả: quảng cáo trực tuyến thường có chi phí thấp hơn và có khả năng đến được đối tượng mục tiêu chính xác.
* Mở rộng thị trường và quốc tế hóa: doanh nghiệp có thể mở rộng thị trường của mình đến khắp nơi trên thế giới mà không cần có cơ sở vật chất ngoại trừ một trang web.
* Cải thiện dịch vụ khách hàng: thương mại điện tử tạo điều kiện cho việc tương tác nhanh chóng giữa doanh nghiệp và khách hàng thông qua các phương tiện như chat trực tuyến, email, hay hệ thống hỗ trợ trực tuyến.
* Theo dõi và phân tích dữ liệu: doanh nghiệp có khả năng thu thập và phân tích dữ liệu khách hàng để hiểu rõ hành vi mua sắm và cá nhân hóa trải nghiệm mua sắm.
* An toàn và tiện lợi trong giao dịch: các phương thức thanh toán trực tuyến ngày càng được cải thiện, đảm bảo an toàn và tiện lợi cho người tiêu dùng.

Những lợi ích này không chỉ làm thay đổi cách chúng ta mua sắm mà còn tạo ra cơ hội mới cho doanh nghiệp để phát triển và mở rộng quy mô kinh doanh.

Sự phát triển của thương mại điện tử đã tạo ra một đỉnh cao mới trong cách chúng ta mua sắm và kinh doanh. Với sự kết hợp của công nghệ, sáng tạo và tiện ích, thương mại điện tử tiếp tục định hình tương lai của thị trường kinh doanh toàn cầu.

Để có thể dẫn đầu trong ngành thương mại điện tử, các công ty lớn luôn đầu tư rất nhiều vào việc nghiên cứu, áp dụng và cải thiện các công nghệ mới. Các nghiên cứu về hệ thống khuyến nghị trong thương mại điện tử là một trong các lĩnh vực quan trọng trong ngành công nghiệp thương mại điện tử.

Hệ thống khuyến nghị đóng góp rất lớn cho ngành thương mại điện tử, mang lại nhiều ảnh hưởng tích cực cho cả người tiêu dùng và doanh nghiệp. Dưới đây là một số đóng góp quan trọng của hệ thống khuyến nghị trong lĩnh vực này:

* Cải thiện trải nghiệm người tiêu dùng: hệ thống khuyến nghị giúp cá nhân hóa trải nghiệm mua sắm của người tiêu dùng bằng cách đề xuất sản phẩm, dịch vụ, hoặc nội dung mà họ có thể quan tâm dựa trên lịch sử và sở thích cá nhân.
* Tăng cường tương tác và giữ chân khách hàng: hệ thống khuyến nghị không chỉ dựa trên lịch sử mua sắm mà còn sử dụng các thuật toán thông minh để dự đoán nhu cầu tiềm ẩn, giúp tăng khả năng chuyển đổi và giữ chân khách hàng.
* Tối ưu hóa doanh số bán hàng: hệ thống khuyến nghị có thể đề xuất các sản phẩm tương tự hoặc phù hợp với sở thích của khách hàng, tăng khả năng mua sắm và doanh số bán hàng.
* Phát triển chiến lược tiếp thị hiệu quả: doanh nghiệp có thể sử dụng dữ liệu từ hệ thống khuyến nghị để tối ưu hóa chiến lược tiếp thị trực tuyến và quảng cáo, làm tăng khả năng hiệu quả của chiến dịch.
* Hiểu rõ hơn về khách hàng: dữ liệu từ hệ thống khuyến nghị có thể được sử dụng để hiểu rõ hơn về hành vi mua sắm, sở thích, và xu hướng của khách hàng, giúp doanh nghiệp đưa ra quyết định chiến lược thông minh hơn.
* Mở rộng phạm vi thị trường: hệ thống khuyến nghị giúp doanh nghiệp mở rộng phạm vi thị trường bằng cách đề xuất sản phẩm có thể phù hợp với người dùng quốc tế.
* Hỗ trợ doanh nghiệp vừa và nhỏ: hệ thống khuyến nghị giúp doanh nghiệp nhỏ và vừa có cơ hội cạnh tranh với các đại ông lớn bằng cách cung cấp một trải nghiệm mua sắm cá nhân hóa và đa dạng.
* Quản lý hiệu suất kinh doanh: doanh nghiệp có thể đánh giá và theo dõi hiệu suất của các chiến lược khuyến nghị để liên tục cải thiện và điều chỉnh chiến lược kinh doanh.

Tóm lại, hệ thống khuyến nghị đóng góp một cách quan trọng vào sự phát triển và tối ưu hóa ngành thương mại điện tử, tạo ra lợi ích cả cho doanh nghiệp và người tiêu dùng thông qua trải nghiệm mua sắm cá nhân hóa và hiệu quả. Do đó chúng em chọn đề tài “Nghiên cứu về itemset mining và association rules” để làm rõ một trong các cách phổ biến để xây dựng một hệ thống khuyến nghị, cụ thể là trong ngành thương mại điện tử.

## Phương pháp nghiên cứu và ý nghĩa thực tiễn

Trong đời sống hiện nay, nhờ việc phát triển của công nghệ thông tin, lượng dữ liệu được lưu trữ trên hệ thống Iternet càng lớn. Đây là một nguồn tài nguyên mà rất nhiều công ty, doanh nghiệp có thể khai thác và sử dụng để có thể cải thiện hiệu suất hoạt động. Từ đó việc nghiên cứu về các kỹ thuật khai thác và xử lý dữ liệu luôn là một yếu tố quan trọng trong sự phát triển của công nghệ thông tin nói chung và các ngành nghề khác. Ngoài ra, việc áp dụng kết quả của các nghiên cứu trên vào các dự án thực tế cũng được tập trung vì khả năng nâng cao hiệu suất công việc cũng như đáp ứng được các nhu cầu mới của người dùng.

Đề tài nghiên cứu của chúng em sẽ bao gồm ba phần chính. Bước đầu là tập trung vào việc nghiên cứu phương thức khai thác dữ liệu để tìm ra các tập phổ biến. Từ kết quả tập phổ biển, chúng em sẽ sử dụng thuật toán để có thế sinh ra bộ quy tắc liên kết. Cuối cùng, chúng em sẽ xây dựng một hệ thống khuyến nghị dựa trên các kết quả trên để đưa ra các sản phẩm mà người dùng có thể quan tâm.

Việc nghiên cứu sẽ giúp chúng em hiểu rõ hơn về các quá trình hoạt động của việc khai thác dữ liệu, tìm kiếm quy tắc liên kết và lập trình hệ thống khuyến nghị. Các kiến thức này sẽ hỗ trợ chúng em rất lớn trong các công việc liên quan đến ngành khoa học máy tính trong tương lại.

# GIỚI THIỆU SƠ VỀ VẤN ĐỀ NGHIÊN CỨU

## Khai thác dữ liệu

Khai thác dữ liệu (data mining) là quá trình khám phá tri thức, thông tin hữu ích từ dữ liệu lớn và phức tạp. Nó là một phần quan trọng của lĩnh vực khoa học dữ liệu và có vai trò quan trọng trong việc phân tích dữ liệu, dự đoán xu hướng và tìm ra thông tin tiềm ẩn. Trong thế kỷ 21, khi công nghệ thông tin và viễn thông phát triển mạnh mẽ, lượng dữ liệu được tạo ra và tích lũy ngày càng tăng, đặc biệt là từ các nguồn như mạng xã hội, hệ thống giao thông, thương mại điện tử, y tế và nhiều lĩnh vực khác. Data mining giúp chúng ta tận dụng những dữ liệu này để tìm ra thông tin có giá trị, từ đó đưa ra quyết định thông minh và đưa ra dự đoán chính xác.

A diagram of data mining

Description automatically generated

Hình 2.1 Tổng quan về data mining ([Netsuite](https://www.netsuite.com/portal/assets/img/business-articles/data-warehouse/social-data-mining.jpg?v2))

Data mining sử dụng các phương pháp và thuật toán trong việc khai thác tri thức từ dữ liệu. Các phương pháp và thuật toán này bao gồm khai thác luật kết hợp (association rule mining), phân tích cụm dữ liệu (cluster analysis), phân tích chuỗi thời gian (time series analysis), phân loại (classification), dự đoán (prediction), và nhiều phương pháp khác. Data mining không chỉ tìm kiếm thông tin đã biết mà còn tìm ra những kiến thức mới và tiềm ẩn trong dữ liệu.

Data mining có ứng dụng rất rộng trong nhiều lĩnh vực. Ví dụ, trong lĩnh vực kinh doanh, data mining giúp phân tích hành vi khách hàng, dự đoán xu hướng tiêu dùng, tối ưu hóa chiến dịch tiếp thị và quảng cáo. Trong lĩnh vực y tế, data mining có thể phân tích dữ liệu bệnh nhân để xác định các yếu tố nguy cơ và dự đoán bệnh lý. Trong lĩnh vực tài chính, data mining có thể giúp dự đoán rủi ro tín dụng và phát hiện gian lận. Ngoài ra, data mining còn được sử dụng trong nghiên cứu khoa học, quản lý tri thức và nhiều lĩnh vực khác.

Tuy nhiên, data mining cũng đối mặt với một số thách thức như vấn đề về quyền riêng tư, tính xác thực và tính bảo mật của dữ liệu. Để giải quyết những thách thức này, cần có sự kết hợp giữa các phương pháp và thuật toán data mining với các biện pháp bảo vệ dữ liệu và quyền riêng tư.

Với sự phát triển của công nghệ và viễn thông, data mining sẽ tiếp tục đóng vai trò quan trọng trong việc khai thác tri thức từ dữ liệu và đưa ra quyết định thông minh trong nhiều lĩnh vực khác nhau.

## Mục tiêu và ý nghĩa của khai thác dữ liệu

Mục tiêu của data mining là khám phá tri thức, thông tin hữu ích và mẫu ẩn trong dữ liệu lớn và phức tạp. Nó giúp chúng ta hiểu rõ hơn về dữ liệu, tìm ra những mối quan hệ, xu hướng và thông tin tiềm ẩn, từ đó đưa ra dự đoán, quyết định thông minh và hỗ trợ trong việc giải quyết các vấn đề và đạt được mục tiêu của tổ chức.

Ý nghĩa của data mining là mang lại nhiều lợi ích quan trọng trong nhiều lĩnh vực khác nhau:

* Phát hiện tri thức ẩn: data mining giúp chúng ta khám phá tri thức tiềm ẩn trong dữ liệu mà chúng ta chưa biết hoặc chưa nhận ra. Nó có thể tiết lộ các mẫu, quy luật và thông tin mới, giúp ta hiểu rõ hơn về mối quan hệ và cấu trúc của dữ liệu.
* Dự đoán và phân loại: data mining có thể sử dụng các thuật toán dự đoán và phân loại để xác định xu hướng và dự đoán kết quả trong tương lai. Ví dụ, trong lĩnh vực tiếp thị, nó có thể dự đoán hành vi mua hàng của khách hàng và giúp tối ưu hóa chiến dịch tiếp thị.
* Tối ưu hóa quy trình và quyết định: data mining có thể tìm ra các mẫu và quy luật trong dữ liệu, từ đó giúp tối ưu hóa quy trình và quyết định trong nhiều lĩnh vực. Ví dụ, trong lĩnh vực sản xuất, nó có thể tìm ra các yếu tố ảnh hưởng đến chất lượng sản phẩm và giúp tối ưu hóa quy trình sản xuất.
* Phân tích và dự báo xu hướng: data mining giúp phân tích và dự báo xu hướng trong nhiều lĩnh vực. Ví dụ, trong lĩnh vực tài chính, nó có thể dự báo xu hướng thị trường và giúp đưa ra các quyết định đầu tư thông minh.
* Tối ưu hóa khách hàng và dịch vụ: data ining có thể phân tích thông tin khách hàng và dự đoán hành vi khách hàng, từ đó giúp tối ưu hóa khách hàng và cung cấp dịch vụ tốt hơn. Ví dụ, nó có thể giúp gợi ý sản phẩm phù hợp cho khách hàng và cải thiện trải nghiệm người dùng.

Tóm lại, data mining có mục tiêu khám phá tri thức và thông tin tiềm ẩn trong dữ liệu, từ đó mang lại nhiều ý nghĩa và lợi ích quan trọng trong việc hiểu, phân tích và tận dụng dữ liệu để đạt được mục tiêu của tổ chức và cải thiện quyết định và hoạt động.

# CƠ SỞ LÝ THUYẾT

## Itemset mining

### Định nghĩa

Itemset mining là một khía cạnh quan trọng của data mining, chuyên tìm kiếm và phân tích các mối quan hệ giữa các mục (items) trong tập dữ liệu để tạo thành tập hợp các mục (itemsets). Nó giúp chúng ta hiểu rõ mô hình mua sắm, thói quen của người dùng và xu hướng trong dữ liệu.

Itemset mining đóng vai trò quan trọng trong việc khai thác tri thức từ dữ liệu và đưa ra quyết định thông minh trong nhiều lĩnh vực khác nhau.

Sau đây là một số khái niệm thường dùng trong itemset mining:

* Item (Mục): là đơn vị đại diện cho đối tướng cần thực hiện việc khai thác dữ liệu.
* Itemset (Tập hợp các mục): là một tập hợp các mục hoặc đối tượng. Mỗi mục có thể là một sản phẩm, từ khóa, hoặc bất kỳ thực thể nào khác trong dữ liệu.
* Transaction (Giao dịch): là một bản ghi hoặc sự kiện trong dữ liệu, biểu diễn hành động hoặc quá trình cụ thể. Mỗi giao dịch thường chứa một hoặc nhiều mục.
* Pass: là số lần thuật toán đọc qua toàn bộ cơ sở dữ liệu transaction
* Support (hỗ trợ): là một độ đo cho biết phần trăm hoặc số lần một itemset xuất hiện trong tất cả các giao dịch.
* Minimum Support (Ngưỡng hỗ trợ tối thiểu): là ngưỡng support tối thiểu itemset phải đạt được để có thể được xem là frequent itemset.
* Frequent Itemset (Itemset phổ biến): là một itemset có sự xuất hiện đủ thường xuyên trong tập dữ liệu, được đo lường bằng support.
* Association Rule (Quy tắc liên kết): Mô tả mối quan hệ giữa các itemset, thường có dạng "A => B".

### Ứng dụng thực tế

Itemset mining chủ yếu tập trung vào việc phát hiện mối quan hệ tiềm ẩn trong dữ liệu, giúp doanh nghiệp và tổ chức hiểu rõ hơn về khách hàng, tối ưu hóa chiến lược kinh doanh, và đưa ra quyết định dựa trên thông tin chính xác và chi tiết. Sau đây là một số ứng dụng thực tế của itemset mining:

* Khuyến nghị sản phẩm (Product Recommendation): phân tích frequent itemset để đề xuất sản phẩm cho người dùng dựa trên thói quen mua sắm của họ.
* Quảng cáo và Tiếp thị (Advertising and Marketing): tìm hiểu mối quan hệ giữa các sản phẩm hoặc sự kiện để tối ưu hóa chiến lược quảng cáo và tiếp thị.
* Xác định mô hình mua sắm: phân tích frequent itemset để hiểu rõ mô hình mua sắm của khách hàng và đưa ra quyết định về quản lý hàng tồn kho.
* Phát hiện gian lận và an ninh thông tin: itemset mining cũng có thể được sử dụng để phát hiện mẫu gian lận trong các giao dịch hoặc hoạt động lạ trong dữ liệu.
* Quản lý tổ chức và quy trình: áp dụng itemset mining để phân tích mối quan hệ giữa các sự kiện trong tổ chức, giúp tối ưu hóa quy trình và tương tác giữa các bộ phận.
* Dự đoán hành vi người dùng: phân tích frequent itemset để dự đoán hành vi mua sắm tiếp theo của người dùng và tối ưu hóa trải nghiệm người dùng.
* Hỗ trợ trong y tế: phân tích frequent itemset về các loại thuốc được những người bệnh sự dụng chung với nhau và các chịu chứng phụ có thể giúp tìm ra các loại thuốc gây tác dụng phụ khi dùng chung với nhau (Cần phải quan sát thêm các trường hợp không dùng các chủ ý chung với nhau).

### Cách thực hiện

Cho một bộ dữ liệu và số lần xuất hiện tối thiểu, ta phải tìm ra danh sách các loại mục xuất hiện trong bộ dữ liệu và sinh ra các tập con của danh sách này. Sau tìm số lần xuất hiện của các nhóm này trong cả bộ dữ liệu. Các nhóm có số lần xuất hiện lớn hơn hoặc bằng yêu cầu tối thiểu sẽ được xem là frequent itemset.

Trong thực tế các bộ dữ liệu này đều sẽ rất lớn, dẫn đến việc tốn kém về dung lượng bộ nhớ xử lý và thời gian thực thi. Do đó, itemset mining thường được thực hiện bằng cách sử dụng các thuật toán đặc biệt như Apriori, FP-Growth và ECLAT. Các thuật toán này giúp tìm kiếm và phân tích các mẫu kết hợp trong tập dữ liệu lớn. Kết quả của quá trình itemset mining thường được đánh giá bằng cách xác định mức độ quan trọng và đáng tin cậy của các mẫu kết hợp.

A diagram of food items

Description automatically generated

Hình 3.1 Mô tả về frequent itemset mining (Nguồn: [Datacamp](https://github.com/clone95/Market-Basket-Analysis/blob/master/README.md))

### Thuật toán FIN

#### Giới thiệu

Sau khi khái niệm về khai thác tập phổ biến (frequent itemset mining) được đề suất vào năm 1993 bởi ba nhà khoa học Agrawal, Imielinski và Swami (Agrawal, Imelinski, & Swami, Mining association rules between sets of items in large databases, 1993), chủ đề này đã trở thành một trong các mục tiêu nghiên cứu chính của ngành khai thác dữ liệu vì khả năng ứng dụng rộng rãi trong nhiều lĩnh vực. Từ đó đã có hàng rất nhiều thuật toán được đề xuất để nâng cao hiệu suất và giải quyết các vấn đề của khái niệm này. Chúng em lựa chọn thuộc toán FIN được tiến sĩ Zhi-Hong Deng đề xuất vào năm 2014 để thực hiện việc khai thác tập phổ biến trong bài báo cáo vì thuật toán này mang lại nhiều sự cải thiện về yêu cầu bộ nhớ cũng như thời gian thực hiện.

Thuật toán FIN sử dụng cấu trúc cây (POC-tree) để lưu các transaction và cấu trúc dữ liệu Nodeset để đại diện cho tập. FIN trực tiếp khám phá các tập phổ biến trong một cây tìm kiếm được gọi là cây liệt kê tập hợp (Rymon, 1992). Để tránh việc bị lập lại trong quá trình tìm kiếm, nó cũng áp dụng một chiến lược loại bỏ có tên là thăng chức (promotion), để giảm đáng kể không gian tìm kiếm. Khác với các thuật toán trước đó, Nodeset chỉ sử dụng thứ tự pre-order (hoặc thứ tự post-order) của từng node trên cây. Qua việc sử dụng Nodeset, thuật toán FIN sẽ tốn ít bộ nhớ hơn các thuật toán dùng các cấu trúc cây khác như FP-Growth, Prepost. FIN

#### POC-tree

POC-tree (Pre-Order Code tree) là một cấu trúc cây chứa các node. Mỗi node trên cây sẽ bao gồm các yếu tố sau:

* Item-name: tên đại diện của mục trên cây
* Count: số lần mục xuất hiện
* Children-list: danh sách các node con trực tiếp của node
* Pre-order: rank của node theo thứ tự pre-order

POC-tree sẽ có node rễ với có giá trị là null và có children-list chứa các node con. Tuy nhiên trong quá trình thực hiện thuật toán FIN, sau khi thức hiện việc chuyển đổi thành nodeset, POC-tree sẽ không còn giá trị sử dụng nữa. Ta có thể xóa POC-tree để giảm yêu cầu về bộ nhớ của thuật toán sau khi xử lý nodelist xong.

Để thực hiện việc tạo ra POC-tree, ta sẽ cần dữ liệu đầu vào là một cơ sỡ dữ liệu (DB) chứa các transaction và mức ngưỡng hỗ trợ (minsup) và thực hiện các bước sau:

* 1. Đọc qua DB để tìm tập hợp F1 (frequent items) chứa các mục và chỉ số hỗ trợ của nó, sao cho chỉ số hỗ trợ của mục lớn hơn hoặc bằng ngưỡng hỗ trợ tối thiểu. Sắp xếp tập hợp F1 theo thứ tự giảm dần về chỉ số hỗ trợ và lưu thành L1. Nếu xảy ra trường hợp nhiều mục có chỉ số hỗ trợ bằng nhau, ta có thể sắp xếp các mục này ngẫu nhiên.
  2. Tạo node rễ cho cây POC-tree, con trỏ vị trí hiện tại của cây (Tr), với item-name là null. Với mỗi transaction trong DB, lấy các mục thuộc transaction có trong L1 và giữ nguyên thứ tứ đó để lưu vào tập hợp P. Với từng phần tử p trong P, ta cần xem xét nếu Tr có một node con với item-name giống p, cộng một cho count của Node con đó. Nếu Tr không có node con nào có item-name giống p, tạo một node mới N, đặt count bằng một và để Tr trỏ vào N. Khi P đã hết phần tử, em sẽ để Tr trỏ vào node root và bắt đầu thực hiện lại với transaction tiếp theo trong DB.
  3. Thực hiện duyệt tiền thứ tứ (pre-order traversal) trên POC-tree và thêm vào giá trị pre-order của từng node thứ tự của node khi duyệt.

Hàm khởi tạo POC-tree sẽ trả về một POC-tree và F1.

Ví dụ, cho minsup = 0.4 và cơ sở dữ liệu như sau

|  |  |
| --- | --- |
| Transaction | Items |
| 1 | a, c, g, f |
| 2 | e, a, c, b |
| 3 | e, c, b, i |
| 4 | b, f, h |
| 5 | b, f, e, c, d |

Bảng 3.1 Bảng dữ liệu transaction ví dụ POC-tree

Thông qua bảng 3.1 ta có thể tìm F1 = {a: 2, b: 4, c: 4, e: 3, f: 3}, L1 = {b, c, e, f, a} và các tập P tương ứng với từng transaction:

|  |  |
| --- | --- |
| Transaction | P |
| 1 | c, f, a |
| 2 | b, c, e, a |
| 3 | b, c, e |
| 4 | b, f |
| 5 | b, c, e, f |

Bảng 3.2 Các tập P của từng transaction

Từ đó ta có thể tạo ra POC-tree như sau

A screenshot of a computer

Description automatically generated

Hình 3.2 Kết quả ví dụ POC-tree

#### Nodeset

N-info là một phần của cấu trúc Nodeset, n-info sẽ bao gồm giá trị pre-order và count của một node trong POC-tree. Ta có thể mô tả n-info như sau (pre-order, count).

Nodeset của mỗi mục sẽ là tập hợp tất cả n-info của từng node có item-name bằng mục trên POC-tree. Ta có thể mô tả một nodeset như sau {(pre-order1, count1), (pre-order2, count2), …, (pre-orderm, countm)}. Các nodeset sẽ được tạo ra và lưu theo thứ tự của L1.

Chỉ số support của môi

Dựa vào ví dụ của phần POC-tree, ta sẽ có danh sách các nodeset như sau:

* Nodeset(b) = {(4, 4)}
* Nodeset(c) = {(1, 1), (5, 3)}
* Nodeset(e) = {(6, 3)}
* Nodeset(f) = {(2, 1), (8, 1), (9, 1)}
* Nodeset(a) = {(3, 1), (7, 1)}

Nodeset của tập mục gồm 2 phần tử i1 và i2 với i1 đứng trước i2 trong L1 có thể biểu diễn như sau nodeset(i1, i2). N-info của nodeset(i1, i2) này sẽ chứa các n-info của nodeset(i2) sao cho trong nodeset(i1) có giá trị pre-order là cha của n-info đó.

Ví dụ, tìm n-info của nodeset(bf), b đứng trước f trong L1 nên ta sẽ xét từng n-info của nodeset(f). Ta thấy n-info của b (4, 4) là cha của hai n-info (8, 1), (9, 1) thuộc f nên nodeset(bf) bằng {(8, 1), (9, 1)}. Tương tự ta có nodeset(cf) sẽ bằng {(2, 1), (8, 1)}

Nodeset của một tập mục có ba phần tử trở lên có thể biểu diễn như sau nodeset(i1, i2, i3, …, ik) với i1 đứng trước i2, i2 đứng trước … ik trong L1. Giả sử p1 gồm {i1, i3, …, ik} và p2 gồm {i2, i3, …, ik}, thì nodeset(i1, i2, i3, …, ik) sẽ bằng tập giao của nodeset(p1) và nodeset(p2).

Ví dụ, với nodeset(bf) là {(8, 1), (9, 1)} và nodeset(cf) là {(2, 1), (8, 1)}, vậy nodeset(bcf) sẽ bằng {(8, 1)}.

#### Các bước trong thuật toán

Thuật toán FIN sẽ bao gồm ba giai đoạn chính:

1. Xây dựng POC-tree và tìm ra tập một mục phổ biến
2. Duyệt qua POC-tree để tìm ra tập hai mục phổ biến và tạo nodeset cho chúng
3. Khai thác các tập từ ba mục trở lên phổ biến dựa vào các nodeset. Để cải thiện hiệu suất của việc khai thác, thuật toán FIN sử dụng một thủ thuật loại bỏ các tập mục dư thừa là thăng hạng.

A paper with text and numbers

Description automatically generated

Hình 3.3 Mã giả thuật toán FIN (Deng & Lv, 2014)

Theo hình 3.3, ta có thể hiểu thuật toán FIN như sau:

* + Dòng (1), khởi tạo tập F rỗng để lưu các tập phổ biến.
  + Dòng (2), gọi thuật toán xây dựng POC-tree để tạo ra POC-tree và tìm các tập một mục phổ biến F1.
  + Dòng (3), khởi tạo F2 rỗng để chứ các tập hai mục phổ biến.
  + Dòng (4) – (15), duyệt tiền thứ tự qua POC-tree để tìm tất cả các tập hai mục vào F2.
  + Dòng (16) – (22), loại bỏ các tập hai mục không phổ biến khỏi F2, khởi tạo nodeset rỗng cho các tập hai mục phổ biến, đó là các tập hai mục còn lại trong F2.
  + Dòng (23) – (31), duyệt tiền thứ tự qua POC-tree lấy thông tin cho các nodeset của các tập hai mục phổ biến.
  + Dòng (33) – (36), khai thác các tập ba mục trở lên phổ biến bằng cách gọi hàm Contrusting\_Pattern\_Tree để sinh ra các tập ba mục trở lên phổ biến từ các tập hai mục phổ biến.

Hàm Constructing\_Pattern\_Tree có ba dữ liệu đầu vào Nd, Cad\_set, FIS\_parent. Nd là node con trỏ node hiện tại. Cad\_set chứa các thông tin để tạo ra các node của node hiện tại. FIS\_parent chứa các tập phổ biến sinh ra từ cha của node hiện tại. Sau đây là mã giả của hàm Constructing\_Pattern\_Tree:

A screenshot of a computer program

Description automatically generated

Hình 3.4 Mã giả hàm CPT (Deng & Lv, 2014)

Theo hình 3.4, ta có thể hiểu hàm Constructing\_Pattern\_Tree như sau

* Dòng (4) – (19), kiểm tra từng mục trong Cad\_set để tìm các mục đã được thăng hạng và các mục sẽ được sử dụng để xây dựng các node con của Nd.
* Dòng (21) – (27), khai thác các tập phổ biến của Nd.
* Dòng (30) – (34), tiếp tục mở rộng các node con của Nd bằng cách gọi lại hàm Constructing\_Pattern\_Tree.

## Association rule mining

### Định nghĩa

Association rule mining là quá trình khám phá và phân tích các quy tắc liên kết (association rules) trong tập dữ liệu. Một association rule là một quy tắc có dạng "nếu X, thì Y", trong đó X và Y là các itemsets. Quy tắc này chỉ ra một mối quan hệ tương quan giữa các thành phần trong dữ liệu.

Association rule mining tìm kiếm và phân tích các quy tắc liên kết có mức độ hỗ trợ (support) và độ tin cậy (confidence) cao trong tập dữ liệu. Mức độ hỗ trợ đo lường tần suất xuất hiện của quy tắc trong dữ liệu, trong khi độ tin cậy đo lường mức độ tin tưởng của quy tắc. Mục tiêu của association rule mining là tìm ra các quy tắc liên kết ý nghĩa và hữu ích từ dữ liệu, từ đó giúp hiểu rõ hơn về mối quan hệ và sự phụ thuộc giữa các thành phần trong dữ liệu.

Công cụ phổ biến để thực hiện association rule mining là thuật toán Apriori và các biến thể của nó. Thuật toán Apriori sử dụng một phương pháp tìm kiếm theo chiều sâu và cắt tỉa để tìm ra tất cả các quy tắc liên kết có mức độ hỗ trợ và độ tin cậy đạt ngưỡng đã cho.

Ví dụ, trong lĩnh vực bán lẻ, association rule mining có thể được sử dụng để phân tích mẫu mua hàng và tìm ra các quy tắc liên kết giữa các mặt hàng. Ví dụ: "Nếu khách hàng mua bột cà phê và đường, thì khả năng cao họ cũng sẽ mua kem tươi." Quy tắc này có thể được sử dụng để tạo ra các gợi ý sản phẩm hoặc để tối ưu hóa việc trưng bày sản phẩm trong cửa hàng.

Association rule mining đóng vai trò quan trọng trong việc tìm hiểu mối quan hệ và sự phụ thuộc trong dữ liệu, từ đó giúp đưa ra quyết định thông minh và tối ưu hóa các quy trình trong nhiều lĩnh vực khác nhau như bán lẻ, marketing, y tế và nghiên cứu khoa học.

### Ứng dụng thực tế

Association rule mining có nhiều ứng dụng thực tế trong các lĩnh vực sau đây:

* Bán lẻ và thương mại điện tử: association rule mining được sử dụng để phân tích mẫu mua hàng của khách hàng và tìm ra các quy tắc liên kết giữa các sản phẩm. Các quy tắc này có thể được sử dụng để tạo ra các gợi ý sản phẩm, tối ưu hóa việc trưng bày sản phẩm trong cửa hàng và tăng hiệu quả bán hàng.
* Tiếp thị và quảng cáo: association rule mining giúp phân tích mối quan hệ giữa các đặc điểm khách hàng và hành vi tiêu dùng. Nó có thể giúp xác định nhóm mục tiêu, tạo ra các chiến dịch tiếp thị tùy chỉnh và đưa ra các quyết định về phân phối nguồn lực tiếp thị hiệu quả.
* Y tế: trong lĩnh vực y tế, association rule mining có thể được sử dụng để phân tích dữ liệu bệnh nhân và tìm ra các quy tắc liên kết giữa các triệu chứng, bệnh tật và phương pháp điều trị. Điều này có thể giúp trong việc phát hiện và dự đoán bệnh tật, tối ưu hóa quy trình chẩn đoán và điều trị, và cải thiện chất lượng chăm sóc sức khỏe.
* Khoa học dữ liệu: association rule mining được sử dụng trong nhiều lĩnh vực khoa học dữ liệu để khám phá mối quan hệ và mẫu cấu trúc trong dữ liệu. Ví dụ, nó có thể được sử dụng để phân tích dữ liệu gen để tìm ra các quy luật liên kết giữa các biểu hiện gen và bệnh tật, hoặc trong phân tích hành vi người dùng để tìm ra các quy tắc liên kết giữa các hành vi trên mạng.
* Quản lý chuỗi cung ứng: association rule mining có thể được sử dụng để phân tích dữ liệu về chuỗi cung ứng và tìm ra các quy tắc liên kết giữa các sản phẩm, nhà cung cấp và khách hàng. Điều này có thể giúp tối ưu hóa quy trình đặt hàng, lập kế hoạch sản xuất và quản lý kho hàng.
* Khám phá dữ liệu web: trong lĩnh vực khám phá dữ liệu web, association rule mining có thể được sử dụng để khám phá mẫu liên kết giữa các trang web, các từ khóa tìm kiếm và hành vi người dùng trên mạng. Điều này có thể giúp cải thiện hiệu quả tìm kiếm, gợi ý nội dung và tối ưu hóa trải nghiệm người dùng trên web.

Đây chỉ là một số ví dụ về lĩnh vực ứng dụng của association rule mining. Phạm vi ứng dụng của nó rất rộng và có thể được áp dụng trong nhiều lĩnh vực khác nhau tùy thuộc vào loại dữ liệu và mục tiêu nghiên cứu.

### Cách thực hiện

Việc khai thác quy tắc kết hợp sẽ bao gồm hai bước chính:

* Tìm ra các tập phổ biến. Thông thường sự cải tiến về thuật toán sẽ được tập trung vào bước này, vì đây là bước phải tương tác với bộ dữ liệu nên sẽ gây ảnh hưởng đến hiệu suất của thuật toán nhiều nhất.
* Dựa vào các tập phổ biến chứa hai mục trở lên, ta có thể sinh ra các luật: X (một tập con của tập phổ biến) => Y (các mục còn lại mà không có trong tập con). Và support của tập phổ biến đó sẽ là support của luật. Confidence của luật X => Y sẽ được tính bằng support của luật chia cho support của X. Nếu giá trị confidence của cao hơn mức yêu cầu tối thiếu (minconf) thì luật đó sẽ trở thành quy tắc liên kết.

Ví dụ, cho một database như sau:

|  |  |
| --- | --- |
| Transaction | Items |
| 1 | pasta, lemon, bread, orange |
| 2 | pasta, lemon |
| 3 | pasta, orange, cake |
| 4 | pasta, lemon, orange, cake |

Bảng 3.3 Bảng dữ liệu ví dụ Association Rule

Với minsup là 2 ta thấy {pasta, lemon} là một tập phổ biến với support là 3, từ đó ta có thể suy ra hai luật là pasta => lemon và lemon => pasta.

Nếu ta lấy minconf là 0.8 thì chỉ có luật lemon => pasta là quy tắc liên với confidence là 1, còn pasta => lemon chỉ co confidence là 0.75.

### Thuật toán Argawal

#### Giới thiệu

Thuật toán Argawal được đề xuất vào năm 1994 bởi nhà khoa học Rakesh Agrawal sử dụng. Đây là một trong các thuật toán đầu tiên về vấn đề khai thác quy tắc liên kết. Đây là một bước quan trọng trong lĩnh vực khai thác dữ liệu và được sử dụng rộng rãi trong nhiều ứng dụng như phân tích mô hình mua sắm, quảng cáo, và nhiều lĩnh vực khác. Thuật toán Agrawal đã đóng góp quan trọng cho việc phát triển lĩnh vực khai thác luật kết hợp và mở ra nhiều hướng nghiên cứu và ứng dụng trong lĩnh vực phân tích dữ liệu và khai thác tri thức.

Tuy là một thuật toán đã cũ nhưng vẫn phù hợp với đề tài, vì thuật toán này sẽ sinh ra một bộ quy tắc liên kết đầy đủ nhất. Tuy vậy, chúng em thay thế thuật toán Apriori bằng thuật toán FIN để nâng cao hiệu suất khai thác tập kết hợp.

#### Các bước trong thuật toán

Dưới đây là mô tả cơ bản về việc sinh luật và so sanh đế cho ra các quy tắc liên kết của thuật toán Agrawal:

A white paper with black text

Description automatically generated

Hình 3.5 Mã giả thuật toán Agrawal (Agrawal & Srikant, 1994)

Hàm ap-genrules sẽ là hàm tạo các quy tắc kết hợp, tính conf của quy tắc đó và đánh giá xem quy tắc đủ điều kiện để lưu thành quy tắc liên kết hay không (conf phải lớn hơn minconf). Hàm apriori-gen sẽ là hàm hỗ trợ khai thách các tập liên kết có số mục nhiều hơn một.

A white background with black text

Description automatically generated

Hình 3.6 Mã giả hàm apriori-gen (Agrawal & Srikant, 1994)

# THỰC NGHIỆM

## Thu thập dữ liệu

Để thu thập dữ liệu của đề tài, chúng em sẽ tìm kiếm dữ liệu trên mạng và tạo dữ liệu thực tế. Chủ đề về khai thác dữ liệu đã có nhiều bộ dữ liệu được sử dụng rất nhiều và có thể xem như là các dữ liệu tiêu chuẩn để hỗ trợ cho việc nghiên cứu này. Sau đây là các cơ sở dữ liệu mà em sẽ sử dụng:

* chess, mushroom là các tập dữ liệu của Frequent Itemset Mining Implementations ([FIMI](http://fimi.uantwerpen.be/data/)). Cả hai bộ dữ liệu này đã được xử lý loại bỏ thông tin bị thiếu và chuẩn hóa dữ liệu để giảm độ lớn giữa các giá trị, giúp thuật toán khám phá luật kết hợp hoạt động tốt hơn.

A grid of numbers on a white background

Description automatically generated

Hình 4.1 Bộ dữ liệu chess

A grid of numbers

Description automatically generated

Hình 4.2 Bộ dữ liệu mushrooms

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Số lượng transaction | Số mục | Số mục trung bình trong transaction |
| chess | 3196 | 75 | 37 |
| mushrooms | 8416 | 119 | 23 |

Bảng 4.1 Thông tin về hai bộ dữ liệu FIMI

* transaction\_data là tập dữ liệu chúng em tự tạo ra với sự giúp đỡ của bạn bè trên website chúng em viết để làm ứng dụng thực tế cho đề tài. Bộ dữ liệu bao gồm 100 mục và 1000 transaction.

A screenshot of a computer code

Description automatically generated

Hình 4.3 Bộ dữ liệu transaction\_data

## Thư viên được sử dụng

Đầu tiên ta sẽ cài đặt và import các thư viện cần thiết cho việc thuật toán khai thác quy tắc liên kết.

A screenshot of a computer code

Description automatically generated

Hình 4.4 Các thư viện được sử dụng

Thư viện numpy giúp xử lý mảng hai chiều.

Thư viện cmp\_to\_key sẽ hỗ trợ cho các hàm sort.

Thư viện os cung cấp khả năng truy cập vào file dữ liệu, các biến môi trường.

Thư viện time cung cấp các hàm hỗ trợ việc đo thời gian chạy của thuật toán.

Thư viện math cung cấp các hàm hỗ trợ việc tính toán.

Thư viện copy cung cấp các hàm hỗ trợ việc sao chép dữ liệu.

Để chuẩn bị dữ liệu, chúng em sẽ đặt từng biến riêng cho dữ liệu đầu vào và dữ liệu đầu ra.

A screenshot of a computer

Description automatically generated

Hình 4.5 Các file dữ liệu đầu vào

A screenshot of a computer code

Description automatically generated

Hình 4.6 Biến dữ liệu

## Thuật toán FIN

#### Hàm khởi tạo

A screenshot of a computer program

Description automatically generated

Hình 4.7 Hàm khởi tạo thuật toán FIN (1)

A screenshot of a computer program

Description automatically generated

Hình 4.8 Hàm khởi tạo thuật toán FIN (2)

#### Hàm runAlgorithm

Có thể xem đây là hàm chính để chạy thuật toán. Hàm sẽ nhận vào:

* filename: tên bộ dữ liệu đầu vào.
* minsup: ngưỡng hỗ trợ tối thiếu để xét tập mục thành tập phổ biến.
* output: tên tập dữ liệu kết quả (có thể bỏ trống).

A screenshot of a computer program

Description automatically generated

Hình 4.9 Hàm runAlgorithm (1)

A screenshot of a computer program

Description automatically generated

Hình 4.10 Hàm runAlgorithm (2)

Khởi tạo các biến yêu cầu cho POC-tree (PPCTreeNode), nodeset (NodelistTreeNode), con trỏ, timer, writer. Lấy dữ liệu đầu vào để chuẩn bị cho quá trình xây dựng POC-tree và khởi tạo các nodeset bằng hàm buildTree(). Sau đó, cần dùng hàm initializeTree() để liên kết các nodeset thành cấu trúc cây. Cuối cùng dùng hàm traverse() theo kiểu đệ quy để duyệt qua các nodeset và khai thác ra các tập kết hợp phù hợp với mức minsup.

#### Hàm buildTree

Đây là hàm đảm nhiệm việc tạo cây POC-tree và các nodelist. Hàm sẽ nhận đầu vào là filename để lấy thông tin từ bộ dữ liệu. Đọc từng dòng transaction để loại bỏ các dòng không sử dụng. Sau đó lại lấy tách lấy từng mục trong transaction, lưu các mục phổ biến lại và sắp xếp lại thứ tự trong transaction theo độ phổ biến giảm dần.

A screenshot of a computer program

Description automatically generated

Hình 4.11 Hàm buildTree (1)

Xây dựng node trên POC-tree và liên kết chúng lại.

A screen shot of a computer code

Description automatically generated

Hình 4.12 Hàm buildTree (2)

Dựa vào POC-tree, xây dựng các nodelist của từng mục.

A screenshot of a computer program

Description automatically generated

Hình 4.13 Hàm buildTree (3)

Từ các nodelist của từng mục, khai thác các nodelist của nhóm hai mục.

A screenshot of a computer program

Description automatically generated

Hình 4.14 Hàm buildTree (4)

#### Hàm initializeTree

Hàm initializeTree sẽ thực hiện việc kết nối các nodelist thành một cấu trúc cây.

A screenshot of a computer code

Description automatically generated

Hình 4.15 Hàm initializeTree

#### Hàm getData

Hàm getData nhận vào:

* filename tên của bộ dữ liệu.
* minSupport ngưỡng support tối thiểu để tập mục được xem là tập phổ biến.

Hàm sẽ đọc qua từng dòng transaction và loại bỏ các dòng thông tin phụ của tập tin hoặc các dòng rỗng. Trong từng transaction sẽ tách các mục ra và tổng kết số lần xuất hiện của từng mục. Lưu các mục có số lần xuất hiện lớn hơn hoặc bằng minSupport vào tập item và sắp xếp theo thứ tự giảm dần theo support.

A screenshot of a computer program

Description automatically generated

Hình 4.16 Hàm getData

#### Hàm traverse

Hàm traverse nhận vào các thông tin sau:

* curNode: nodelist hiện tại
* curRoot: nodelist root hiện tại
* level: hạng của nodelist trong cây
* sameCount

Duyệt qua các cây nodelist để khai thác tập mục nhiều phần tử và mức hỗ trợ của chúng.

A screenshot of a computer program

Description automatically generated

Hình 4.17 Hàm traverse (1)

A screenshot of a computer code

Description automatically generated

Hình 4.18 Hàm traverse (2)

#### Hàm is2\_itemSetValid

Hàm is2\_itemSetValid sẽ nhận vào các giá trị:

* + ni: nodelist i
  + nj: nodelist j
  + level: hạng của nodelist trong cây
  + lastChild: nodelist con trước đó
  + sameCount

A screenshot of a computer code

Description automatically generated

Hình 4.19 Hàm is2\_itemSetValid

#### Hàm iskItemSetFreq

Hàm iskItemSetFreq sẽ nhận vào các giá trị:

* + ni: nodelist i
  + nj: nodelist j
  + level: hạng của nodelist trong cây
  + lastChild: nodelist con trước đó
  + sameCount

A screenshot of a computer program

Description automatically generated

Hình 4.20 Hàm iskItemSetFreq (1)

A screenshot of a computer screen

Description automatically generated

Hình 4.21 Hàm iskItemSetFreq (2)

#### Hàm writeItemsetsToFile

Nhận vào curNode nodelist hiện tại và sameCount. Xét curNode có đủ minSupport để lưu vào bộ tập phổ biến. Nếu có writer thì sẽ lưu vào kết quả ra file.

A screen shot of a computer code

Description automatically generated

Hình 4.22 Hàm writeItemsetsToFile

#### Hàm getFrequentItemsets

In ra kết quả các tập phổ biến được phân chia theo số mục trong tâp.

A screenshot of a computer program

Description automatically generated

Hình 4.23 Hàm getFrequentItemsets

#### Kết quả

Khi chạy thuật toán với bộ dữ liệu chess và mức minsup = 0.75 và minconf = 0.8, chúng em có kết quả như sau:

A close-up of numbers

Description automatically generated

Hình 4.24 Kết quả FIN với dữ liệu chess

Còn khi chạy với bộ dữ liệu mushrooms với các tham số giống như trên:

A close-up of a number

Description automatically generated

Hình 4.25 Kết quả FIN với dữ liệu mushrooms

## Thuật toán Agrawal

#### Hàm khởi tạo

A screen shot of a computer program

Description automatically generated

Hình 4.26 Hàm khởi tạo Agrawal

#### Hàm runAlgorithm

Có thể xem đây là hàm chính để chạy thuật toán. Hàm sẽ nhận vào:

* pattern: bộ tập phổ biến
* output: tên tập dữ liệu kết quả
* databaseSize: kích thước của bộ dữ liệu
* minconf: ngưỡng confidence nhỏ nhất để xét quy tắc liên kết

A close-up of a computer screen

Description automatically generated

Hình 4.27 Hàm runAlgorithm

#### Hàm runAlgorithmHelper

Hàm runAlgorithmHelper nhận vào:

* patterns: bộ tập phổ biến
* output: tên tập dữ liệu kết quả
* databaseSize: kích thước của bộ dữ liệu

Thực hiện lần đệ quy đầu tiên của ap-genrules và nếu có luật phù hợp sẽ gọi hàm saveRule để lưu.

A screenshot of a computer program

Description automatically generated

Hình 4.28 Hàm runAlgorithmHelper (1)

A computer screen shot of a computer screen

Description automatically generated

Hình 4.29 Hàm runAlgorithmHelper (2)

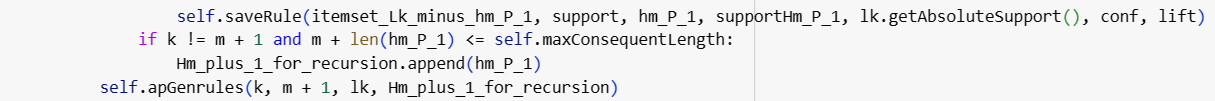
#### Hàm apGenrules

Khai thác các bộ luật và xét xem luật đó có đủ điều kiện để trở thành quy tắc liên kết hay không.

A computer screen shot of a code

Description automatically generated

Hình 4.30 Hàm apGenrules (1)



Hình 4.31 Hàm apGenrules (2)

#### Hàm calculateSupport

Hàm calculateSupport nhận vào itemset và xem trong pattern để lấy về support.

A computer screen shot of a code

Description automatically generated

Hình 4.32 Hàm calculateSupport

#### Hàm generateCandidateSizeK

A screenshot of a computer code

Description automatically generated

Hình 4.33 Hàm generateCandidateSize

#### Kết quả

Khi chạy thuật toán với bộ dữ liệu chess và mức minsup = 0.75 và minconf = 0.8, chúng em có kết quả như sau:

A black text on a white background

Description automatically generated

Hình 4.34 Kết quả Agrawal với dữ liệu chess

Còn khi chạy với bộ dữ liệu mushrooms với các tham số giống như trên:

A close-up of a number

Description automatically generated

Hình 4.35 Kết quả Agrawal với dữ liệu mushrooms

## Website hệ thống khuyến nghị

4.5.1 Giao diện website

A screenshot of a sign up form

Description automatically generated

Hình 4.5.1.1 Trang đăng ký

A screenshot of a sign in

Description automatically generated

Hình 4.5.1.2 Trang đăng nhập

A screenshot of a computer

Description automatically generated

Hình 4.5.1.3 Trang chủ

A screenshot of a computer

Description automatically generated

Hình 4.5.1.4 Giỏ hàng

4.5.2 Áp dụng thuật toán Data mining vào website

4.5.2.1 Xử lý bên client

A screen shot of a computer program

Description automatically generated

Hình 4.5.2.1.1 Thiết lập cài đặt axios bên client



Hình 4.5.2.1.2 Thiết lập Request HTTP Get đến server

Phía client sẽ gửi dữ liệu và token để xác minh tài khoản về cho server.

A screen shot of a computer program

Description automatically generated

Hình 4.5.2.1.3 Gửi Request đến server để lấy thông tin sản phẩm

4.5.2.2 Xử lý bên server

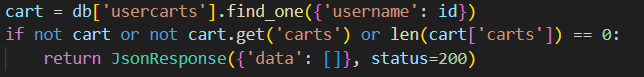
A screen shot of a computer code

Description automatically generated

Hình 4.5.2.2.1 Server xử lý thông tin từ client

Hàm **verify\_token()** dùng để xác thực tài khoản bằng jwt với header **Authorization**.

Sau khi đã xác thực tài khoản, ta tiếp tục lấy danh sách sản phẩm từ giỏ hàng của người dùng.



Hình 4.5.2.2.2 Tìm giỏ hàng của người dùng

A black background with white text

Description automatically generated

Hình 4.5.2.2.3 Danh sách sản phẩm trong giỏ hàng

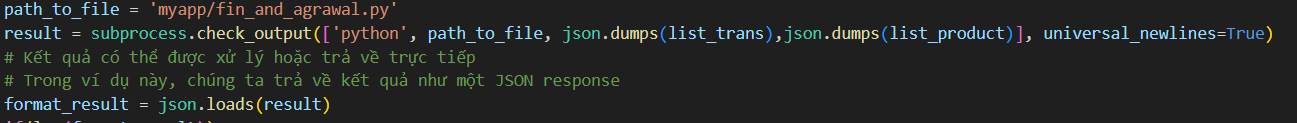
Ta lấy danh sách giao dịch từ cơ sở dữ liệu (Transactions) để tiến hành thực hiện thuật toán.

A black screen with colorful text

Description automatically generated

Hình 4.5.2.2.4 Transactions

Sau khi đã có transactions và danh sách sản phẩm trong giỏ hàng của người dùng. Ta sử dụng thư viện **subprocess** để thực thi file thuật toán.



Hình 4.5.2.2.5 Thực thi code bằng subprocess

Sau khi thực thi code thành công, thuật toán sẽ trả về kết quả dưới dạng json. Kết quả sẽ là sản phẩm có confident cao nhất nằm ở vế phải mà có vế trái là những sản phẩm trong giỏ hàng của người dùng.

A computer screen with numbers and symbols

Description automatically generated

Hình 4.5.2.2.6 Minh họa kết quả

A screen shot of a computer code

Description automatically generated

Hình 4.5.2.2.5 Lấy thông tin sản phẩm sau đó gửi về client

4.5.3 Kết quả đạt được

A screenshot of a website

Description automatically generated

A screenshot of a computer

Description automatically generated

Khi thêm sản phẩm bất kì vào giỏ hàng. Thuật toán sẽ gợi ý một sản phẩm có liên quan cho người dùng dựa trên tập phổ biến ( frequent itemsets ) và luật kết hợp ( association rules ).

# KẾT LUẬN

## Kết luận

## Hướng phát triển

Việc nghiên cứu của đề tài mang một ý nghĩa thực tiễn to lớn. Từ những hạn chế của đề tài, em sẽ tiếp tục thử nghiệm thêm nhiều thuật toán khai thác tập phổ biến và khai thác quy tắc liên kết khác nhau để nâng cao hiệu suất cho quá trình phân tích, nghiên cứu.

Các mẫu dữ liệu chúng em sử dụng trên mạng đều là dữ liệu đã qua xử lý, vì vậy bọn em đã bỏ qua phần chuẩn hóa dữ liệu đầu vào. Tuy vậy việc chuẩn hóa dữ liệu đầu vào là không bắt buộc. Dù với lợi ích giảm nhiễu và giúp dễ tìm được thuật toán hơn nhưng khi thực hiện có thể sẽ gây mất thông tin trong một số trường hợp. Điều này cũng áp dụng cho việc loại bỏ các transaction chỉ chứa một mục trong bộ dữ liệu. Quyết định này phụ thuộc vào bối cảnh và mục tiêu cụ thể trong quá trình khám phá quy tắc liên kết.

Đối với thuật toán, chúng em có thể cải tiến thêm bằng cách thêm vào các chỉ số phụ như lift, conviction nhằm thắt chặt hơn các quy tắc liên kết có thể sinh ra. Từ đó có một bộ quy tắc tốt hơn hiện tại. Và việc chọn lựa mức minsup và minconf, ngưỡng hỗ trợ quy định mức độ phổ biến của một tập mục, trong khi độ tin cậy định rõ mức độ chắc chắn của một luật kết hợp.

Bổ sung cho hệ thống khuyến nghị khả năng xử lý các trường hợp sản phẩm không có quy tắc liên kết để hệ thống luôn có thể hoạt động. Thêm khả năng ghi nhớ các sản phẩm khuyến nghị để tạo thành gợi ý trên trang chính của cửa hàng khi khách hàng truy cập lại.

Trong các trường hợp thực tế, các bộ dữ liệu thường sẽ có kích thước rất lớn, không thể xử lý theo cách thông thường mà phải áp sử dụng các công cụ và kỹ thuật phân phối để xử lý dữ liệu lớn hiệu quả, chẳng hạn như Apache Spark.

# TÀI LIỆU THAM KHẢO

**Tiếng Anh**

Agrawal, R., & Srikant, R. (1994). *Fast Algorithms For Mining Association Rules.*

Agrawal, R., Imelinski, T., & Swami, A. (1993). *Mining association rules between sets of items in large databases.* San Jose: IBM Almaden Research Center.

Deng, Z.-H., & Lv, S.-L. (2014). *Fast mining frequent itemsets using Nodesets.*

Deng, Z.-H., Wang, Z.-H., & Jiang, J.-J. (2012). *A new algorithm for fast mining frequent itemsets using N-lists.*

Rymon, R. (1992). *Search Through Systematic Set Enumeration.*

Wang, G., Zomaya, A., Martinez, G., & Li, K. (2015). *Algorithms and Architectures for Parallel Processing: 15th International Conference, ICA3PP 2015, Zhangjiajie, China, November 18-20, 2015, Proceedings, Part IV.* Springer.