# CHƯƠNG 1: GIỚI THIỆU

## Tổng quan đề tài:

Ngày nay, các phương tiện lưu trữ có dung lượng ngày càng, và các hệ quản trị cơ sở dữ liệu cũng phát triển mạnh mẽ, cung cấp cho người dùng khả năng lưu trữ dữ liệu không giới hạn. Dữ liệu tuy nhiều nhưng những giá trị tri thức mà nó chứa đựng lại chưa được sử dụng hiệu quả. Với những thành công trong các công trình nghiên cứu trong lĩnh vực khai phá dữ liệu, con người đã khai thác được những giá trị tri thức từ các dữ liệu được lưu trữ, và sử dụng chúng để giải quyết các bài toán trong nhiều lĩnh vực quan trọng của đời sống.

Hiện nay, có rất nhiều giải thuật phân tích luật kết hợp khác nhau, và mỗi giải thuật lại có những điểm mạnh cũng như là điểm yếu khác nhau. Do đó, mục đích của đề tài muốn giới thiệu chung về luật khai phá kết hợp cũng như một luật khai phá kết hợp phổ biến được sử dụng là Apriori.

## Mục tiêu đề tài:

Với các yêu cầu đã đặt ra ở phần 1.1, đề tài sẽ tập trung các các mục tiêu sau:

* + Tìm hiểu bài toán khai phá luật kết hợp.
  + Tìm hiểu thuật toán Apriori.
  + Hiện thực thuật toán Apriori.
  + Thử nghiệm thuật toán Apriori trên các tập dữ liệu.

# CHƯƠNG 2: BÀI TOÁN KHAI PHÁ LUẬT KẾT HỢP

## 2.1 Tổng quan khai phá dữ liệu:

Khai phá dữ liệu là quá trình trích xuất thông tin, khám phá tri thức có mối tương quan nhất định từ một kho dữ liệu khổng lồ nhằm mục đích dự đoán các xu thế, hành vi trong tương lai, hoặc tìm kiếm các tập thông tin hữu ích mà bình thường không thể nhạn diện được.

Giải quyết các vấn đền thực tế:

* Siêu thị Wallmart có hơn 2 triệu giao dịch/ ngày.
* Google, Youtube, …

Các giai đoạn của quá trình khai phá:

* Data cleaning.
* Data integration.
* Data selection.
* Data Transformation.
* Data mining.
* Pattern Evaluation.
* Knowledge Presentation.

## 2.2 Giới thiệu luật kết hợp trong bài toán khai phá dữ liệu:

Trong lĩnh vực Data Mining, mục đích của luật kết hợp (Association Rule – AR) là tìm ra các mối quan hệ giữa các đối tượng trong khối lượng lớn dữ liệu. Nội dung cơ bản của luật kêt hợp được tóm tắt như dưới đây.

Cho cơ sở dữ liệu gồm các giao dịch T là các giao dịch *t1, t2, …, tn.*

T = {*t1, t2, …, tn*}.

T gọi là cơ sở dữ liệu giao dịch (Transaction Database).

Mỗi giao dịch *ti* bao gồm các đối tượng I (gọi là itemset).

I = {i1, i2, …, im}.

Một itemset gồm k items gọi là k-itemset.

Mục đích của luật kết hợp là tìm ra sự kết hợp (asociation) hay tương quan (correlation) giữa các items. Những luật kết hợp này có dạng X -> Y.

Trong Baskets Analysis, luật kết hợp X -> Y có thể hiểu rằng những người mua các mặt hàng trong tập X cũng thường mua các mặt hàng trong tập Y (X và Y gọi là itemset). Ví dụ, nếu X = {Apple, Banana} và Y = {Cherry, Durian} và ta có luật kết hợp X -> Y thì chúng ta có thể nói rằng những người mua Apple và Banana thì cũng thường mua Cherry và Durian.

Theo quan điểm thống kê, X được xem là biến độc lập (Independent variable) còn Y được xem là biến phụ thuộc (Dependent variable).

Độ hỗ trợ (Support) và độ tin cậy (Confidence) là hai tham số để đo lường luật kết hợp.

Độ hỗ trợ (Support) của luật kết hợp X -> Y là tần suất của giao dịch chứa tất cả các items trong cả hai tập X và Y. Ví dụ, support của luật X -> Y là 5% có nghĩa là 5% các giao dịch X và Y được mua cùng nhau.

Công thức để tính support của luật X -> Y như sau:

Trong đó: N là tổng số giao dịch.

Độ tin cậy (Confidence) của luật kết hợp X -> Y là xác xuất xảy ra Y khi đã biết X. Ví dụ độ tin cậy của luật kết hợp {Apple} -> {Banana} là 80% có nghĩa là 80% khách hàng mua Apple cũng mua Banana.

Công thức để tính độ tin cậy của luật kết hợp X -> Y là xác xuất có điều kiện Y khi đã biết X như sau:

Trong đó: n(X) là số giao dịch chứa X.

Để thu được các luật kết hợp, ta thường áp dụng hai tiêu chí: minimum support (min\_sup) và minimum coonfidence (min\_conf). Các luật thỏa mãn có support và confidence thỏa mãn (lớn hơn hoặc bằng) cả Minimum support và Minimum confidence gọi là các luật mạnh (Strong Rule). Minimum support và Minimum confidence gọi là các giá trị ngưỡng (threshold) và phải xác định trước khi sinh ra các luật kết hợp.

Một số luật kết hợp:

* Binary association rules (Luật kết hợp nhị phân).
* Quantitative association rules (Luật kết hợp định lượng).
* Fuzzy association rules (Luật kết hợp mờ).

Trong đó, thuật toán phổ biến nhất tìm các luật kết hợp là Apriori sử dụng Binary association rules.

## 2.3 Ứng dụng của bài toán:

Tính ứng dụng của bài toán khai phá luật kết hợp được ứng dụng vào các bài toán sau:

Bài toán tìm các mã CP tăng đồng thời với tần suất lớn hơn hoặc bằng (>=) min\_sup:

Input:

* Cơ sở dữ liệu giao dịch.
* Giá trị min\_sup.

Output: Tất cả các tập chứa các mã CP có đồng thời biến động trong ngày với tần suất xuất hiện của tập lớn hơn hoặc bằng (>=) min\_sup.

Bài toán tìm các tập phổ biến:

Input:

* Cơ sở dữ liệu giao dịch.
* Giá trị min\_sup.

Output:

* Tất cả tập Largek .
* Với 1 <= k <= số lượng item thược CSDL.

# CHƯƠNG 3: THUẬT TOÁN APRIORI

## 3.1 Giới thiệu:

Apriori là thuật toán khả sinh được đề xuất bởi R. Agrawal và R. Srikant và năm 1993 để khai thác các tập item đối với các luật kết hợp kiểu bool. Tên của thuật toán dựa trên việc thuật toán sử dụng prior knowledge của các thuộc tính tập item phổ biến.

Apriori dùng cách tiếp cận lặp được biết đến như tìm kiếm level-wise, với các tập một item phổ biến được tìm thấy bằng cách quét CSDL để đếm số lượng từng item, và thu nhập những item thỏa mãn độ hỗ trợ tối thiểu. Tập kết quả đặt là L1. Tiếp theo, L1 được dùng để tìm L2, tập các tập hai item phổ biến, nó được dùng để tìm L3, và cứ thế tiếp tục, cho tới khi tập k item phổ biến không thể tìm thấy. Việc tìm kiếm cho mỗi Lk đòi hỏi một lần quét toàn bộ dữ liệu.

Trước khi đi vào chi tiết của thuật toán Apriori, đầu tiên, chúng ta sẽ tìm hiểu xác định một vài thuật ngữ phổ biến được sử dụng trong thuật toán:

* Itemset: là tập hợp của những item trong cơ sở dữ liệu mà nó được xác định bởi I = {i1, i2, …, in}, trong đó n là số lượng item.
* Transaction: là một thành phần CSDL mà nó bao gồm tập hợp các item. Transaction được ký hiệu là T và T http://ait.edu.vn/Hoc_thuat/Hinh/icon_con.jpg I Một Transaction chứa tập hợp các item T={i1, i2,…. in}.
* Minimum support: là điều kiện cần được đáp ứng bởi các item đề ra để có thể tiến hành xử lý item kế tiếp có thể. Minimum support có thể được xem là một điều kiện giúp loại bỏ các tập không phổ biến trong bất kỳ CSDL. Thường sử dụng Minimum support cho mô hình tỷ lệ phần trăm.
* Frequent itemset (tập phổ biến): các itemset đáp ứng các tiêu chí điều kiện thì được gọi là tập phổ biến. Nó được ký hiệu là Li trong đó I chỉ i-itemset.
* Candidate itemset (ứng viên tập phổ biến): ứng viên ban tập phổ- biển là các item chỉ được xem xử lý.
* Suport: Độ hữu dụng của một luật có thể được đo như thế nào các giao tác có tâp phổ biến mà nó phù hợp với ý nghĩa cả hai phía cạnh tranh trong luật kết hợp.
* Confidence: chỉ sự chắc chắn của các luật.

## 3.2 Chi tiết giải thuật Apriori:

Ý tưởng:

Tìm tất cả frequent itemsets: k – itemset (itemsets gồm k items) được dùng để tìm (k+1) itemsets. Đầu tiên tìm 1-itemset (ký hiệu L1). L1 được dùng để tìm L2 (2-itemsets). L2 được dùng để tìm L3 (3-itemset) và tiếp tục đến khi không có k-itemset được tìm thấy.

Từ frequent itemsets sinh ra các luật kết hợp mạnh (các luật kết hợp thỏa mãn hai tham số min\_sup và min\_conf).

Input: Cơ sở dữ liệu D, min\_sup.

Output: Các tập mẫu thường xuyên trong D.

Giải thuật:

Li: Các tập thường xuyên i phần tử (frequent i-itemsets).

Ci: Các tập dự tuyển i phần tử.

1. Tìm L1 từ tập dữ liệu D.
2. Thực hiện vòng lặp sau:

Sử dụng hàm Apriori-gen để xây dựng Ck+1 từ Lk.

Tính độ hỗ trợ của mỗi phần tử trong Ck+1 (\*).

Lk+1 = {c thuộc Ck+1 | độ hỗ trợ của c >= min\_sup}.

1. Các tập Li chính là các tập cần tìm.

Hàm Apriori-gen:

Sử dụng phép hợp để tạo Ck+1 từ Lk.

Loại bỏ các tập trong Ck+1 chứa các tập con k-phần tử không thuộc Lk.

Giải thuật Apriori:

Thực hiện (\*) bằng cách duyệt qua D.

Ví dụ thuật toán Apriori:

|  |  |
| --- | --- |
| TID | List of item\_IDs |
| T100 | I1, I2, I5 |
| T200 | I2, I4 |
| T300 | I2, I3 |
| T400 | I1, I2, I4 |
| T500 | I1, I3 |
| T600 | I2, I3 |
| T700 | I1, I3 |
| T800 | I1, I2, I3, I5 |
| T900 | I1, I2, I3 |

Compare candidate support count with minimum support count

L1

C1

Scan D for count of each candidate

|  |  |
| --- | --- |
| Itemset | Sup.count |
| {I1} | 6 |
| {I2} | 7 |
| {I3} | 6 |
| {I4} | 2 |
| {I5} | 2 |

|  |  |
| --- | --- |
| Itemset | Sup.count |
| {I1} | 6 |
| {I2} | 7 |
| {I3} | 6 |
| {I4} | 2 |
| {I5} | 2 |

|  |  |
| --- | --- |
| Itemset | Sup.count |
| {I1, I2} | 4 |
| {I1, I3} | 4 |
| {I1, I4} | 1 |
| {I1, I5} | 2 |
| {I2, I3} | 4 |
| {I2, I4} | 2 |
| {I2, I5} | 2 |
| {I3, I4} | 0 |
| {I3, I5} | 1 |
| {I4, I5} | 0 |

|  |
| --- |
| Itemset |
| {I1, I2} |
| {I1, I3} |
| {I1, I4} |
| {I1, I5} |
| {I2, I3} |
| {I2, I4} |
| {I2, I5} |
| {I3, I4} |
| {I3, I5} |
| {I4, I5} |

C2

C2

Scan D for count of each candidate

Generate C2 candidates from L1

|  |  |
| --- | --- |
| Itemset | Sup.count |
| {I1, I2} | 4 |
| {I1, I3} | 4 |
| {I1, I5} | 2 |
| {I2, I3} | 4 |
| {I2, I4} | 2 |
| {I2, I5} | 2 |

|  |
| --- |
| Itemset |
| {I1, I2, I3} |
| {I1, I2, I5} |

|  |  |
| --- | --- |
| Itemset | Sup.count |
| {I1, I2, I3} | 2 |
| {I1, I2, I5} | 2 |

|  |  |
| --- | --- |
| Itemset | Sup.count |
| {I1, I2, I3} | 2 |
| {I1, I2, I5} | 2 |

C3

C3

L2

L3

## Đánh giá giải thuật:

Khuyết điểm:

Tạo ra nhiều tập dự tuyển:

* 104 frequent 1-itemsets -> nhiều hơn 107 (~104(104-1)/2) 2-itemsets dự tuyển.
* Một k-itemsets cần ít nhất 2k-1 itemsets dự tuyển trước đó.

Kiểm tra tập dữ liêu nhiều lần:

* Chi phí lớn khi kích thước các itemsets tăng lên dần.
* Nếu k-itemsets được khám phá thì cần kiểm tra tập dữ liệu k+1 lần.

## Cải tiến giải thuật:

Giải thuật Apriori có thể được cải tiến như sau:

Kỹ thuật dựa trên bảng băm (hash-based techique): Một k-itemsets ứng với hashing bucket count nhỏ hơn minimum support threshold không là một frequent itemset.

Giảm giao dịch (Transaction reduction): Một giao dịch không chứa frequent k-itemsets nào thì không cần được kiểm tra ở các lần sau (k+1 –itemsets).

Phân hoạch (partitioning): Một itemset phải frequent trong ít nhất một phân hoạch thì mới có thể frequent trong toàn bộ tập dữ liệu.

Lấy mẫu (sampling): Khai phá chỉ tập con dữ liệu cho trước với một vị trí support threshold nhỏ hơn và cần một phương pháp để xác định toàn diện (completeness).

Đếm itemsets động (dynamic itemset counting): Chỉ thêm các itemsets dự tuyển khi tất cả các tập con của chúng được dự đoán là frequent.

# CHƯƠNG 4: HIỆN THỰC GIẢI THUẬT APRIORI

## 4.1 Đặc tả bài toán (Use case):

## 4.2 Thực hiện giải thuật:

## 4.3 Đánh giá bài toán: