# THÔNG TIN NHÓM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Họ tên** | **Nội dung công việc** | **Chấm điểm trong nhóm** | **Ghi chú** |
| 1 | Nguyễn Văn Thao |  |  | Trưởng nhóm |
| 2 | Nguyễn Thị Kim Lam |  |  |  |
| 3 | Nguyễn Duy Đức |  |  |  |
| 4 | D |  |  |  |
|  |  |  |  |  |

**Lời nói đầu**

Trong thời buổi hiện đại ngày nay, công nghệ thông tin cũng như những ứng dụng của nó không ngừng phát triển, lượng thông tin và cơ sở dữ liệu được thu thập và lưu trữ cũng tích lũy ngày một nhiều lên. Con người cũng vì thế mà cần có thông tin với tốc độ nhanh nhất để đưa ra quyết định dựa trên lượng dữ liệu khổng lồ đã có. Các phương pháp quản trị và khai thác cơ sở dữ liệu truyền thống ngày càng không đáp ứng được thực tế, vì thế, một khuynh hướng kỹ thuật mới là Kỹ thuật phát hiện tri thức và khai phá dữ liệu nhanh chóng được phát triển. Khai phá dữ liệu đã và đang được nghiên cứu, ứng dụng trong nhiều lĩnh vực khác nhau ở các nước trên thế giới. Ở Việt Nam, kỹ thuật này đang được nghiên cứu và dần đưa vào ứng dụng. Khai phá dữ liệu là một bước trong quy trình phất hiện tri thức. Hiện nay, mọi người không ngừng tìm tòi các kỹ thuật để thực hiện khai phá dữ liệu một cách nhanh nhất và có được kết quả tốt nhất. Trong bài tập lớn này, chúng em tìm hiểu và trình bày về một một số giải thuật trong khai phá dữ liệu để tìm những luật kết hợp cũng như tổng quan về khai phá dữ liệu, với đề tài “**Gợi ý mua hàng bằng giải thuật apriori** ”. Chúng em rất mong nhận được những góp ý từ thầy. Chúng em xin chân thành cảm ơn

**CHƯƠNG 1. KIẾN THỨC CƠ SỞ**

**2.1. Tổng quan về khai phá dữ liệu**

* + 1. *Tại sao lại cần khai phá dữ liệu (datamining)*

Khoảng hơn một thập kỷ trở lại đây, lượng thông tin được lưu trữ trên các thiết bị điện tử không ngừng tăng lên. Sự tích lũy dữ liệu này xảy ra với một tốc độ bùng nổ. Người ta ước đoán rằng lượng thông tin trên toàn cầu tăng gấp đôi sau khoảng hai năm và theo đó số lượng cũng như kích cỡ của các cơ sở dữ liệu (CSDL) cũng tăng lên một cách nhanh chóng. Nói một cách hình ảnh là chúng ta đang “ngập” trong dữ liệu nhưng lại “đói” tri thức. Câu hỏi đặt ra là liệu chúng ta có thể khai thác được gì từ những “núi” dữ liệu tưởng chừng như “bỏ đi” ấy không ?

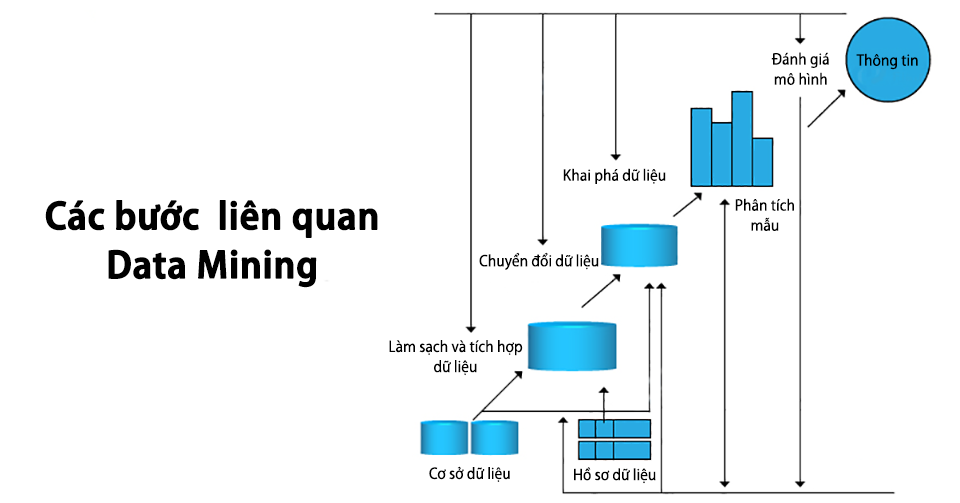
- Data Mining ra đời như một hướng giải quyết hữu hiệu cho câu hỏi vừa đặt ra ở trên . Khá nhiều định nghĩa về Data Mining và sẽ được đề cập ở phần sau, tuy nhiên có thể tạm hiểu rằng Data Mining như là một công nghệ tri thức giúp khai thác những thông tin hữu ích từ những kho dữ liệu được tích trữ trong suốt quá trình hoạt động của một công ty, tổ chức nào đó. Một số kỹ thuật Khai phá dữ liệu Một số kỹ thuật Khai phá dữ liệu

* + 1. ***Khai phá dữ liệu là gì?***

Khai phá dữ liệu (datamining) được định nghĩa như là một quá trình chắt lọc hay khai phá tri thức từ một lượng lớn dữ liệu. Một ví dụ hay được sử dụng là là việc khai thác vàng từ đá và cát, Dataming được ví như công việc "Đãi cát tìm vàng" trong một tập hợp lớn các dữ liệu cho trước. Thuật ngữ Dataming ám chỉ việc tìm kiếm một tập hợp nhỏ có giá trị từ một số lượng lớn các dữ liệu thô. Có nhiều thuật ngữ hiện được dùng cũng có nghĩa tương tự với từ Datamining như Knowledge Mining (khai phá tri thức), knowledge extraction(chắt lọc tri thức), data/patern analysis(phân tích dữ liệu/mẫu), data archaeoloogy (khảo cổ dữ liệu), datadredging(nạo vét dữ liệu),...

**Định nghĩa**: Khai phá dữ liệu là một tập hợp các kỹ thuật được sử dụng để tự động khai thác và tìm ra các mối quan hệ lẫn nhau của dữ liệu trong một tập hợp dữ liệu khổng lồ và phức tạp, đồng thời cũng tìm ra các mẫu tiềm ẩn trong tập dữ liệu đó. Khai phá dữ liệu là một bước trong bảy bước của quá trình KDD (Knowleadge Discovery in Database) và KDD được xem như 7 quá trình khác nhau theo thứ tự sau:

1. Làm sạch dữ liệu (data cleaning & preprocessing)s: Loại bỏ nhiễu và các dữ liệu không cần thiết.
2. Tích hợp dữ liệu: (data integration): quá trình hợp nhất dữ liệu thành những kho dữ liệu (data warehouses & data marts) sau khi đã làm sạch và tiền xử lý (data cleaning & preprocessing).
3. Trích chọn dữ liệu (data selection): trích chọn dữ liệu từ những kho dữ liệu và sau đó chuyển đổi về dạng thích hợp cho quá trình khai thác tri thức. Quá trình này bao gồm cả việc xử lý với dữ liệu nhiễu (noisy data), dữ liệu không đầy đủ (incomplete data), .v.v.
4. Chuyển đổi dữ liệu: Các dữ liệu được chuyển đổi sang các dạng phù hợp cho quá trình xử lý
5. Khai phá dữ liệu(data mining): Là một trong các bước quan trọng nhất, trong đó sử dụng những phương pháp thông minh để chắt lọc ra những mẫu dữ liệu.
6. Ước lượng mẫu (knowledge evaluation): Quá trình đánh giá các kết quả tìm được thông qua các độ đo nào đó.
7. Biểu diễn tri thức (knowledge presentation): Quá trình này sử dụng các kỹ thuật để biểu diễn và thể hiện trực quan cho người dùng.



* + 1. ***Các chức năng của khai phá dữ liệu***

Data Mining được chia nhỏ thành một số hướng chính như sau:

* Mô tả khái niệm (concept description): thiên về mô tả, tổng hợp và tóm tắt khái niệm. Ví dụ: tóm tắt văn bản.
* Luật kết hợp (association rules): là dạng luật biểu diễn tri thứ ở dạng khá đơn giản. Ví dụ: “60 % nam giới vào siêu thị nếu mua bia thì có tới 80% trong số họ sẽ mua thêm thịt bò khô”. Luật kết hợp được ứng dụng nhiều trong lĩnh vực kính doanh, y học, tin-sinh, tài chính & thị trường chứng khoán, .v.v.
* Phân lớp và dự đoán (classification & prediction): xếp một đối tượng vào một trong những lớp đã biết trước. Ví dụ: phân lớp vùng địa lý theo dữ liệu thời tiết. Hướng tiếp cận này thường sử dụng một số kỹ thuật của machine learning như cây quyết định (decision tree), mạng nơ ron nhân tạo (neural network), .v.v. Người ta còn gọi phân lớp là học có giám sát (học có thầy).
* Phân cụm (clustering): xếp các đối tượng theo từng cụm (số lượng cũng như tên của cụm chưa được biết trước. Người ta còn gọi phân cụm là học không giám sát (học không thầy).
* Khai phá chuỗi (sequential/temporal patterns): tương tự như khai phá luật kết hợp nhưng có thêm tính thứ tự và tính thời gian. Hướng tiếp cận này được ứng dụng nhiều trong lĩnh vực tài chính và thị trường chứng khoán vì nó có tính dự báo cao.
  + 1. ***ứng dụng của khai phá dữ liệu***

Data Mining tuy là một hướng tiếp cận mới nhưng thu hút được rất nhiều sự quan tâm của các nhà nghiên cứu và phát triển nhờ vào những ứng dụng thực tiễn của nó. Chúng ta có thể liệt kê ra đây một số ứng dụng điển hình:

* Phân tích dữ liệu và hỗ trợ ra quyết định (data analysis & decision support)
* Điều trị y học (medical treatment)
* Text mining & Web mining
* Nhận dạng (pattern recognition)
* .v.v.

**2.2. Bài toán Khai phá luật kết hợp**

**Khai phá luật kết hợp** trong cơ sở dữ liệu trong tiếng Anh gọi là: **Association rule in data mining.**

**Khai phá luật kết hợp** (KPLKH) là một kĩ thuật quan trọng của khai phá dữ liệu. Mục tiêu nhằm phát hiện mối quan hệ giữa các mục dữ liệu trong cơ sở dữ liệu.

Mô hình đầu tiên của bài toán KPLKH là mô hình nhị phân (hay còn gọi là mô hình cơ bản) được R. Agrawal, T. Imielinski và A. Swami đề xuất vào năm 1993, xuất phát từ nhu cầu phân tích dữ liệu của cơ sở dữ liệu giao tác, phát hiện các mối quan hệ giữa các tập mục hàng hóa (Itemsets) đã bán được tại các siêu thị.

Việc xác định các quan hệ này không phân biệt vai trò khác nhau cũng như không dựa vào các đặc tính dữ liệu vốn có của các mục dữ liệu mà chỉ dựa vào sự xuất hiện cùng lúc của chúng.

### Bài toán khai phá luật kết hợp

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

*Cho cơ sở dữ liệu gồm các giao dịch T là tập 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 tập 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 (association) hay tương quan (correlation) giữa các items. Những luật kết hợp này có dạng****X =>Y***

Trong Basket 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 cây (Confidence) là 2 tham số dùng 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:*

http://bis.net.vn/photos/storage/image0011.png

*Trong đó:*N là tổng số giao dịch.

Độ tin cậy (*Confidence) của luật kết hợp*X =>Y là xác suất 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 =>là xác suất có điều kiện Y khi đã biết X như sau :

http://bis.net.vn/photos/storage/image005.gif

Trong đó: n(X) là số giao dịch chứa X

**Để thu được các luật kết hợp, ta thường áp dụng 2 tiêu chí:**

*minimum support (min\_sup) và  minimum confidence (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 các luật kết hợp.

***Một itemsets mà tần suất xuất hiện của nó >= min\_sup goi là frequent itemsets***

**2.3. Các giải thuật nghiên cứu (Tối thiểu 02 giải thuật)**

**Giải thuật apriori**

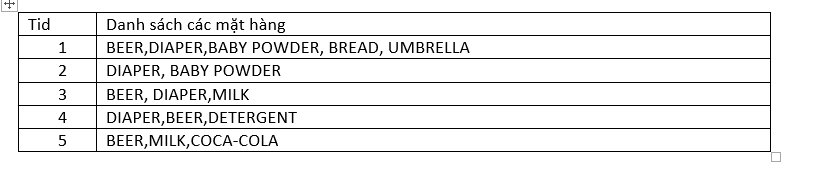
 Apriori là thuật toán được đề xuất bởi R. Agrawal và R. Srikant vào năm 1993 để khai thác các tập item đối với các luật kết hợp kiểu bool. Tên của thuật toán dựa trên việc thuật toán sử dụng tri thức trước (prior knowledge) của các thuộc tính tập item phổ biến, chúng ta sẽ thấy sau đây. Apriori dùng cách tiếp cận lặp được biết đến như tìm kiếm level-wise, với các tập k item được dùng để thăm dò các tập (k+1) item. Đầu tiên, tập các tập 1 item phổ biến được tìm thấy bằng cách quét cơ sở dữ liệu để đếm số lượng từng item, và thu thập những item thỏa mãn độ hỗ trợ tối thiểu. Tập kết quả đặt là L1. Tiếp theo, L1được dùng để tìm L2, tập các tập 2 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 Lkđòi hỏi một lần quét toàn bộ cơ sở dữ liệu.  
Apriori dùng cách tiếp cận lặp được biết đến như tìm kiếm *level-wise*, với các tập *k* item được dùng để thăm dò các tập (*k*+1) item.

1. Đầu tiên, tập(frequent 1- itemsets) phổ biến 1 được tìm thấy ký hiệu là C1
2. Bước tiếp theo là tính support có nghĩa là sự xuất hiện của các item trong cơ sở dữ liệu. Điều này đòi hỏi phải duyệt qua toàn bộ cơ sở dữ liệu.
3. Sau đó, bước cắt tỉa được thực hiện trên C1 trong đó những item được so sánh với thông số minimum support. Những item thỏa điều kiện minimum support mới được xem xét cho tiến trình tiếp theo ký hiệu là L1.
4. Sau đó, bước phát sinh các bộ ứng viên được thực hiện trong đó tập phổ biến 2 được tạo ra ký hiệu là C2.
5. Một lần nữa, cở sở dữ liệu được duyệt để tính toán support của 2 tập phổ biến. Theo minimum support, các bộ ứng viên tạo ra được kiểm tra và chỉ những tập phổ biến nào thỏa điều kiện minimum support thì tiếp tục được sử dụng tạo ra bô ứng viên tập phổ biến 3.

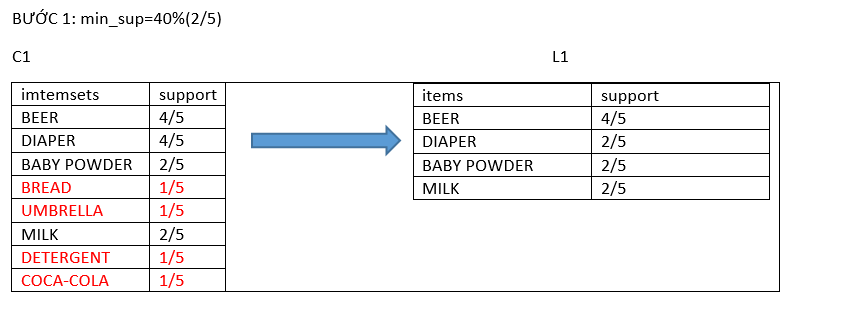
Bước trên tiếp tục cho đến khi không có tập phổ biến hoặc bộ ứng viên có thể được tạo ra.

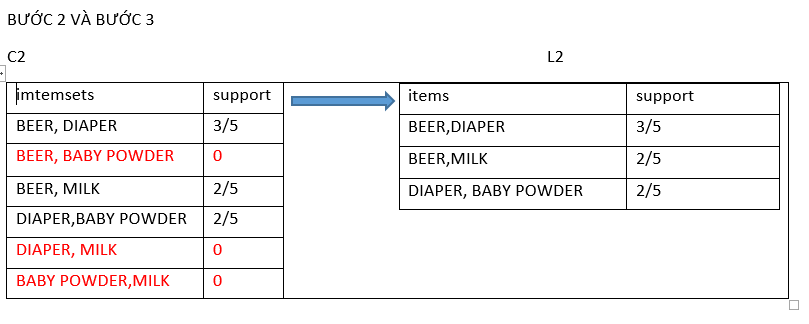
Ví dụ: Giả sử ta có có sở dữ liệu giao dịch (Transaction Database -TDB) như sau :

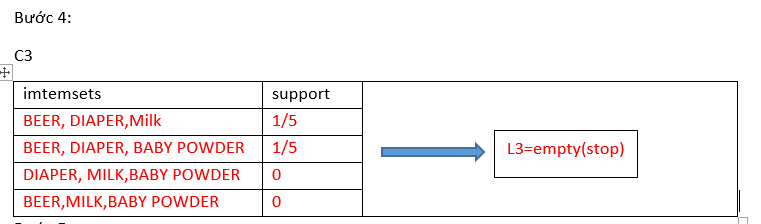
**Giả sử có cơ sở dữ liệu giao dịch bán hàng gồm 5 giao dịch như sau:**

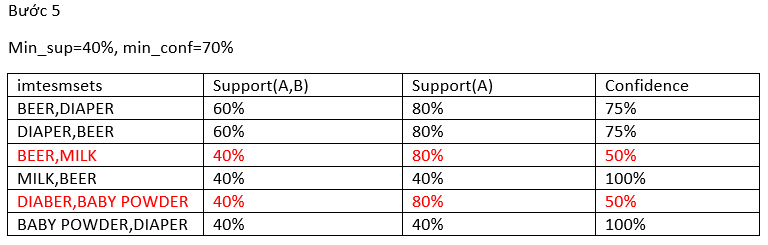


*Thuật toán Apriori tìm các luật kết hợp trong giao dịch bán hàng trên như sau:*









***Kết quả ta có các luật kết hợp sau (với min\_sup= 40%, min\_conf=70%)***

*R1: Beer => Diaper  (support =60%, confidence = 75%)*

*R2: Diaper =>Beer (support =60%,confidence = 75%)*

*R3: Milk =>Beer (support =40%, confidence = 100%)*

*R4: Baby Powder => Diaper (support =40%,confidence = 100%)*

### Ưu và nhược điểm của phương pháp Apriori

### Ưu điểm

* Đây là thuật toán đơn giản và dễ hiểu nhất trong số các thuật toán học quy tắc kết hợp.
* Các quy tắc kết quả là trực quan và dễ dàng để giao tiếp với người dùng cuối.
* Nó không yêu cầu dữ liệu được dán nhãn vì nó là thuật toán không giám sát, do đó bạn có thể sử dụng nó trong nhiều tình huống khác nhau vì dữ liệu không được gắn nhãn thường dễ truy cập hơn.
* Nhiều tiện ích mở rộng đã được đề xuất cho các trường hợp sử dụng khác nhau dựa trên triển khai này, ví dụ, có các thuật toán học liên kết có tính đến việc đặt hàng các mặt hàng, số lượng của chúng và khoảng thời gian liên quan.

### Nhược điểm

* Phải duyệt CSDL nhiều lần. Với I=i1,i2,i3,..i100, số lần duyệt CSDL sẽ là 100.
* Số lượng tập ứng viên rất lớn: 2^100 -1=1.27\*10^30.
* Thực hiện việc tính độ phổ biến nhiều, đơn điệu.
* **Giải thuật FP-Growth**

Thuật toán Apriori có chi phí lớn nhưng lại kém hiệu quả. Để khắc phục nhược điểm này, J. Han,

J Pei, Y. Yin và R. Mao đề xuất thuật toán FP-growth. Thuật toán FP-growth được xây dựng với 3 kỹ thuật chính:

(1) Nén dữ liệu thích hợp vào một cấu trúc cây gọi là cây FP-tree. Chỉ có các 1-tập mục (1-item) ở trong cây và các nút của cây được sắp xếp để các nút xuất hiện thường xuyên hơn có thể dễ dàng chia sẻ với các nút xuất hiện ít hơn.

(2) Thực hiện phương pháp khai phá phát triển (growth) từng đoạn dựa trên cây FP-tree gọi là phương pháp FP-growth.

(3) Kỹ thuật tìm kiếm được dùng ở đây là dựa vào sự phân chia, “chia để trị”, phân rã nhiệm vụ khai phá thành các nhiệm vụ nhỏ hơn.

Thuật toán FP-growth do nén toàn bộ CSDL lên một cấu trúc dữ liệu nhỏ hơn là cây FP-tree nên tránh được việc duyệt nhiều lần CSDL (thuật toán chỉ duyệt cơ sở dữ liệu 2 lần). Tiếp theo thuật toán khai phá cây bằng cách phát triển dần các mẫu mà không sinh các tập mục ứng viên, do đó tránh được khối lượng tính toán lớn. Phương pháp FP- growth đã chứng tỏ được tính hiệu quả của nó và có thể thực hiện khai phá cho cả các mẫu ngắn và dài, nhanh hơn thuật toán Apriori, luôn chỉ cần duyệt CSDL 2 lần.

Thuật toán FP- growth thực hiện như sau: Đầu tiên, thuật toán duyệt CSDL lần thứ nhất để tính độ hỗ trợ của từng mục (đếm số lần xuất hiện của từng mục).

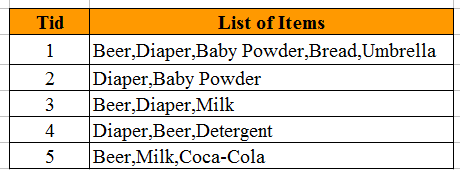
Tiếp đến, những mục không đủ độ hỗ trợ bị loại. Các mục còn lại được sắp theo thứ tự giảm dần của độ hỗ trợ (cũng tức là giảm dần theo số lần xuất hiện trong CSDL), ta nhận được danh sách L các mục đã sắp.

Duyệt CSDL lần thứ hai, với mỗi giao tác t, loại các mục không đủ độ hỗ trợ, các mục còn lại theo thứ tự giống như xuất hiện trong L (tức là thứ tự giảm dần theo độ hỗ trợ) được cất vào cây FP-tree.

Phần tiếp theo thuật toán khai phá tìm các mẫu thường xuyên trên cây FP-tree đã xây dựng mà không cần duyệt lại CSDL nữa.

Thuật toán kinh điển Apriori tìm tập mục thường xuyên theo cách sinh ra các ứng cử viên và duyệt CSDL để kiểm tra, thuật toán FP-growh không khai phá theo cách của thuật toán Apriori mà nén các giao tác của CSDL lên cấu trúc cây FP-Tree, sau đó thực hiện khai phá trên cây này. Thuật toán sinh luật từ tập mục thường xuyên cũng đã được trình bày cụ thể.

**Giả sử có cơ sở dữ liệu giao dịch bán hàng gồm 5 giao dịch như sau:**



* Lý thuyết
* Tự lấy Ví dụ trình bày tính toán bằng tay (gợi ý tương tự bài tập thực hành)
* Nhận xét ưu điểm, nhược điểm của giải thuật
* Phương pháp Apriori
* • Phương pháp FP-Growth

**CHƯƠNG 2. GIỚI THIỆU ĐỀ TÀI**

* 1. **Giới thiệu bài toán ứng dụng**

Khi chúng ta đi mua sắm, ta thường thấy những sản phẩm có liên quan đến nhau thường được đặt cạnh nhau để gợi cho người mua có thể sẽ mua sản phẩm đó. Tại sao họ biết mà sắp xếp được như vậy? Đó là dựa vào khai phá luật kết hợp.



Khai phá luật kết hợp là gì?

* Khai phá luật kết hợp (*Association Rule Mining*) được ứng dụng phổ biến trong kinh doanh. Nó được sử dụng rộng rãi để hiểu và thử nghiệm các chiến lược kinh doanh, tiếp thị khác nhau để tăng doanh số và năng suất của các tổ chức, bao gồm chuỗi siêu thị và thị trường trực tuyến.
* Khai phá luật kết hợp là học tập dựa trên quy tắc (*rule-based learning*) để xác định sự liên kết giữa các sản phẩm khác nhau trong cơ sở dữ liệu. Một trong những ví dụ tốt nhất và phổ biến nhất của Khai phá luật kết hợp là phân tích giỏ thị trường (*Market Basket Analysis*). Phân tích sự liên kết giữa các mặt hàng khác nhau có xác suất cao nhất được mua bởi khách hàng.
* Ví dụ, hiểu được thói quen mua hàng của khách hàng bằng cách tìm ra mối tương quan và mối liên hệ giữa các mặt hàng khác nhau mà khách hàng đặt trong ‘giỏ hàng của họ’, từ đó có thể rút ra các mẫu theo định kỳ.
  + Cụ thể, anh Dii đi mua một chai rượu trong siêu thị. Anh cũng lấy một vài gói khô bò. Người quản lý ở đó phân tích rằng, không chỉ anh Dii, mọi người thường có xu hướng mua rượu và khô bò cùng nhau. Sau khi tìm ra mô hình, người quản lý bắt đầu sắp xếp các mặt hàng này lại với nhau và sau đó đã có thông báo tăng doanh số.
* Quá trình xác định mối liên kết giữa các sản phẩm / mặt hàng được gọi là khai phá luật kết hợp. Để thực hiện khai thác quy tắc kết hợp, nhiều thuật toán đã được phát triển. Thuật toán Apriori là một trong những thuật toán phổ biến nhất và được cho là hiệu quả nhất trong số đó. Chúng ta tiếp tục thảo luận về thuật toán Apriori là gì nhé!

Việc thuật toán Apriori có thể làm là nhìn vào quá khứ và khẳng định rằng nếu một việc gì đó xảy ra thì sẽ có tỉ lệ bao nhiêu phần trăm sự việc tiếp theo sẽ xảy ra. Nó giống như nhìn vào quá khứ để dự đoán tương lại vậy, và việc này rất có ít cho các nhà kinh doanh. Ví dụ một siêu thị muốn nghĩ cách sắp xếp các gian hàng một cách hợp lí nhất, họ có thể nhìn vào lịch sử mua hàng và sắp sếp các tập sản phẩm thường được mua cùng nhau vào một chỗ. Hoặc một trang web tin tức muốn giới thiệu cho người dùng các bài viết liên quan đến nhau nhất, cũng có thể áp dụng quy luật tương tự.

* 1. **Giới thiệu giải thuật sử dụng**

Ý tưởng chính trong thuật toán Apriori là tất cả các tập hợp con của một tập hợp mục (*item*) phổ biến là phổ biến. Tương tự, đối với bất kỳ mục (*item*) không phổ biến nào, tất cả các tập hợp con của nó cũng phải không phổ biến.

Để tìm ra điều đó, ta phải sử dụng các độ đo sau:

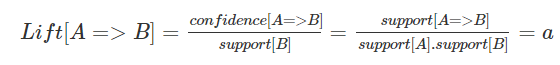
Support: là tỉ lệ của số lượng các giao dịch có chứa tập các mặt hàng A và tổng tất cả các giao dịch.



Confidence



Lift

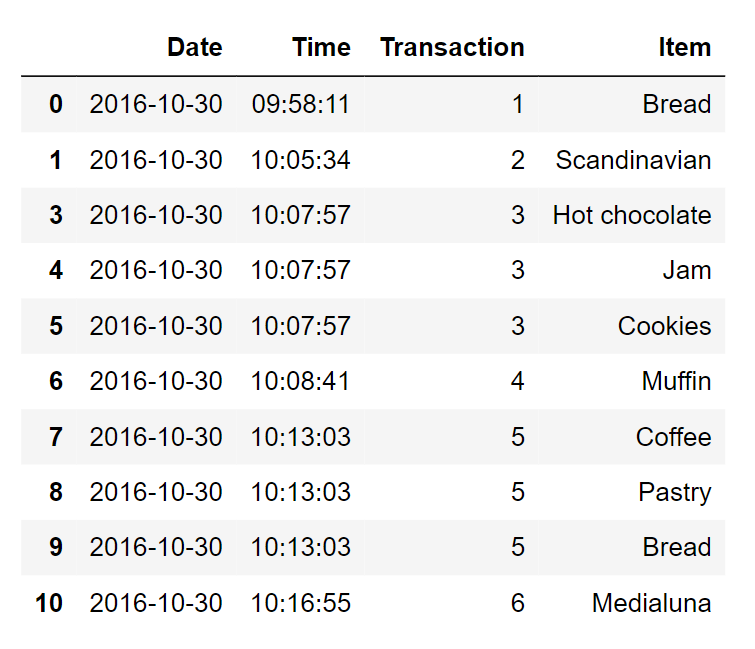


* + Nếu: Lift > 1 khi KH mua sản phẩm A sẽ làm tăng khả năng mua thêm sản phẩm B **a** lần.
  + Nếu: Lift < 1 khi KH mua sản phẩm A sẽ làm giảm khả năng mua thêm sản phẩm B **a** lần.
  + Nếu: Lift = 1 khi KKH mua sản phẩm A sẽ không làm tăng khả năng mua sản phẩm B (A và B là 2 sản phẩm độc lập với nhau)
* Tuy nhiên, trong thực tế, tùy mục đích là tìm các rule để tăng khả năng bán được sản phẩm nên chúng ta chỉ quan tâm tới các rule mà có hệ số Lift > 1.
  1. **Mô tả bộ dữ liệu áp dụng**

Bộ dữ liệu được sử dụng để khai phá dữ liệu là bộ dữ liệu của cửa hàng bán lẻ

Dữ liệu đầu vào có 21293 hàng và những cột: Ngày, giờ, giao dịch, mặt hàng

Mỗi hàng tương ứng với mỗi giao dịch của khách hàng tại một thời điểm được gắn nhãn là “Transaction”. Hình dưới minh Transaction 1: {Bread}, Transaction 2 :{Scandinavian}, Transaction 3 :{ Hot chocolate, Jam, Cookies }, ...



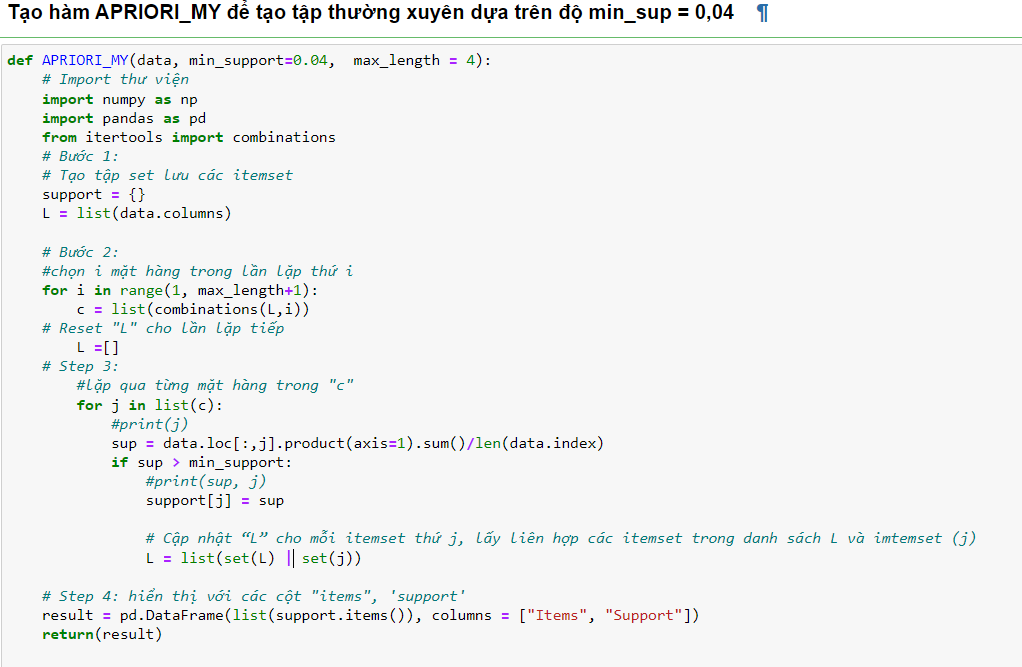
Sau khi xử lí những giao dịch bị trùng dữ liệu trên được chuyển đổi sang định dạng Nhị phân như trong Hình 2. Trong đó, mỗi hàng tương ứng với một giao dịch và mỗi cột tương ứng với một mặt hàng. Các mặt hàng có giao dịch sẽ chuyển giá trị “1” và không giao dịch chuyển sang giá trị “0”.

# CHƯƠNG 3. THỬ NGHIỆM

**3.1. Giới thiệu môi trường công cụ**

**Anaconda,nu**

**3.2. Kịch bản thử nghiệm**



Bước 1: Tạo set "support" lưu các itemset và L lưu các cột dữ liệu

Bước 2: Tạo tập phổ biến có độ dài i

Reset lại L để nối 2 tập phổ biến thu được trong lần lặp thứ i và tạo kết hợp trong lần lặp (i + 1).

Bước 3: Tính sup cho từng mặt hàng thứ j

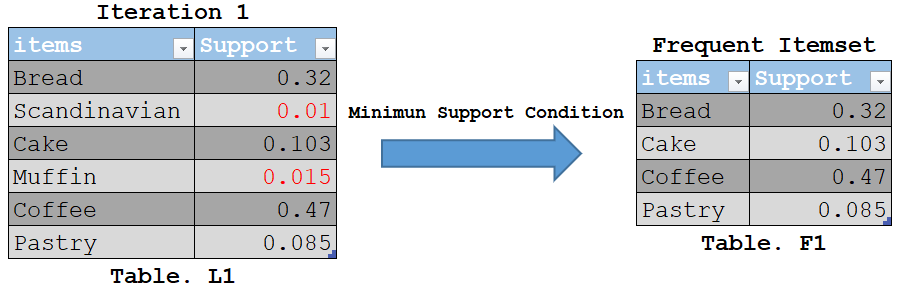
So sánh với min\_sup nếu thỏa mãn điều kiện thì cho vào mảng “support[j]”

Kết hợp các itemset trong set(L) và set(i) gán vào L

Bước 4: Hiển thị với các cột "items", 'support'

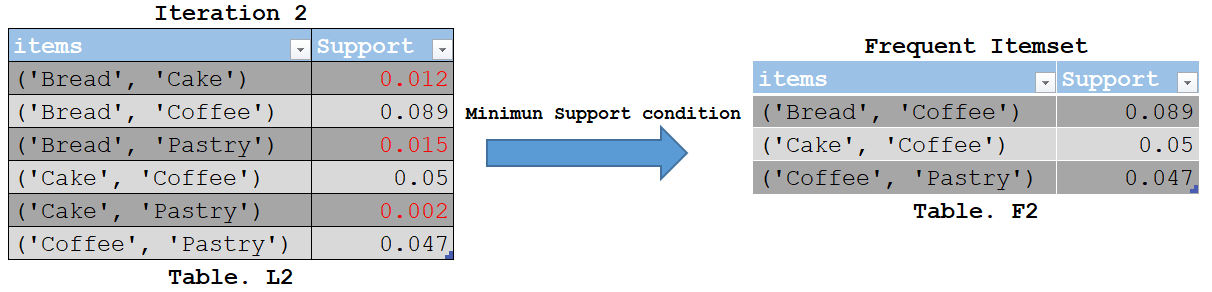
Bước 2 lặp lại như sau:

Lần 1:Ban đầu thuật toán liệt kê tất cả các mục và tính toán hỗ trợ cho các tập phổ biến có độ dài 1 và gán vào “L”, đây là một số ít được thể hiện trong Hình 4 Bảng L1. Ở đây chúng tôi giả định 4% là ngưỡng hỗ trợ tối thiểu



Như hình trên, các imtem ' **Scandinavian'** và ' **Muffin'** là không thường xuyên. Vì vậy, sẽ **loại bỏ {** 'Scandinavian', 'Muffin' **}** trong các lần lặp lại sắp tới. Chúng ta có Bảng F1 cuối cùng dưới dạng Itemset.

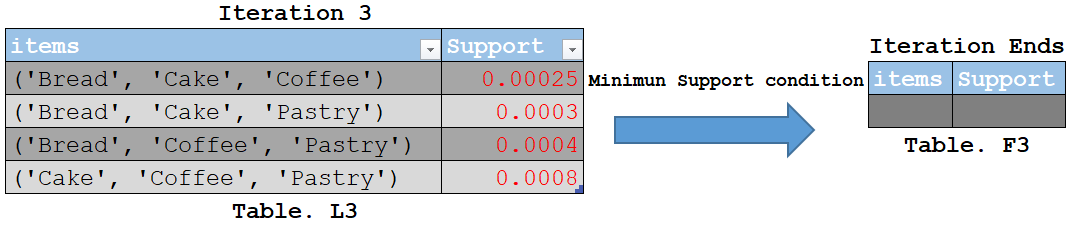
Lần 2: Tiếp theo, chúng ta sẽ tạo sự kết hợp của 2 imtemset và tính toán hỗ trợ của chúng.



Các imtemset không phổ biến bị loại là {("Bread", "Cake"), ("Bread", "Pastry"), ("Cake", "Pastry")} được hiển thị trong Hình 5-Bảng L2. Bây giờ, chúng ta hãy hiểu thế nào là lược bớt và cách nó làm cho Apriori trở thành một trong những thuật toán tốt nhất để tìm các tập phổ biến.

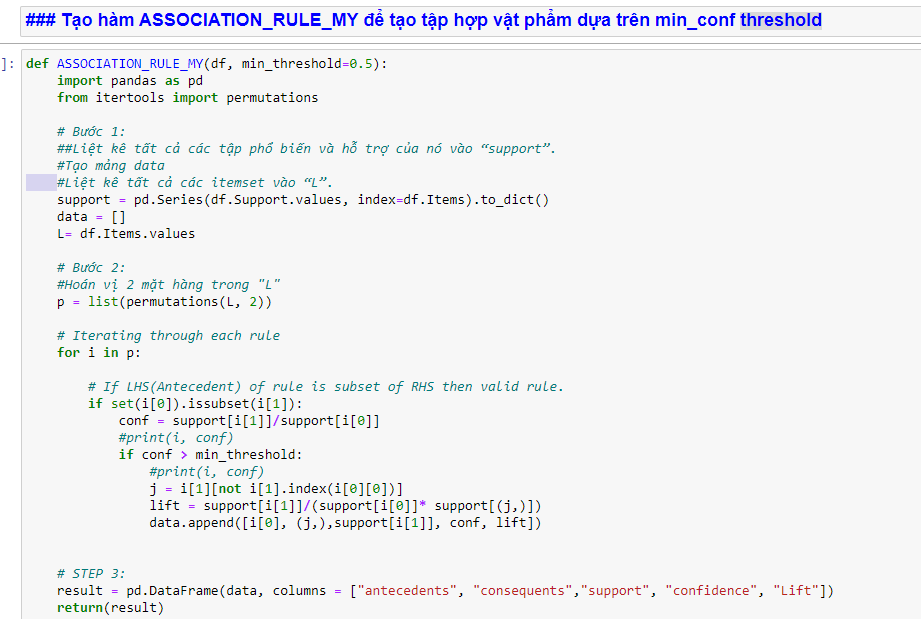
Tiếp theo chúng ta sẽ chia các tập hợp mục trong Hình 6-Bảng L3 thành các tập hợp con và loại bỏ các tập hợp con đang có hỗ trợ nhỏ hơn ngưỡng hỗ trợ tối thiểu.

Lần 3: Ở đây tất cả các itemset là không phổ biến vì tập hợp con của nó không phổ biến:{('Bread', 'Cake'), ("Bread", "Pastry"), ("Cake", "Pastry")} ở hình Hình 5-Table . L2,. Đây là điểm nổi bật chính của Thuật toán Apriori.



Vì tất cả các itemset trong lần 3 là không thường xuyên nên dừng ở đây.

Tiếp theo chúng ta sẽ áp dụng các quy tắc kết hợp cho itemset thu được trong thuật toán trước. Khởi tạo ngưỡng tin cậy tối thiểu là 50%.



Bước 1:

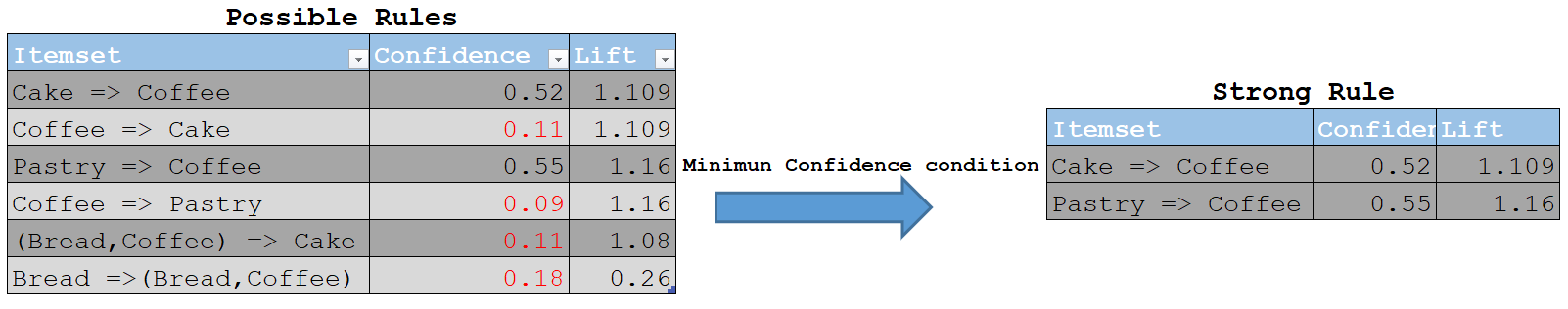
Liệt kê tất cả các tập phổ biến và giá trị hỗ trợ của nó vào biến "support".

Tạo mảng “data” để lưu trữ kết quả

Liệt kê tất cả các tập phổ biến vào biến “L”.

Bước 2: Hoán vị 2 phần tử trong L và gán vào p vì số lượng giao dịch chứa Cake & Coffee cũng giống như số lượng giao dịch chứa Coffee & Cake nhưng Confidence (Cake => Coffee) khác *Confidence (Coffee => Cake)*

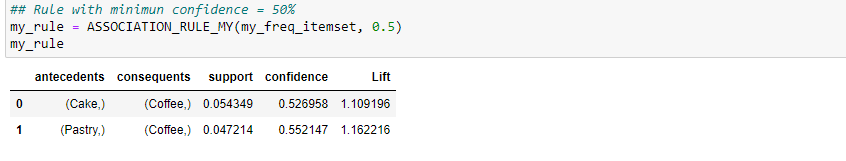
Tiếp theo nếu i[0] là con của i[1] thì tính conf



Tính conf và lift cho mỗi tập hợp mục thứ i, nếu thỏa mãn ngưỡng Độ tin cậy tối thiểu thì thêm vào “data”.

Dưới đây Hình 9 cho thấy Quy tắc mạnh thu được từ các tập dữ liệu thử nghiệm. Đừng quên rằng Quy tắc chỉ được áp dụng trên Tập vật phẩm thường xuyên.

**3.3. Kết quả đánh giá**

* Kết quả sinh ra 2 luật kết hợp mạnh thỏa mãn min\_sup và min\_conf
* 

# BIỂU CHẤM

|  |  |  |
| --- | --- | --- |
| **TT** | **Nội dung** | **Tỷ lệ** |
| 1 | Mã nguồn ứng dụng | 20% |
| 2 | Báo cáo word/Slide | 40% |
| 3 | Trình bày hỏi đáp các giải thuật | 40% |