

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO**

**KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG**

**LAB 2**

**Lớp**  **: 19\_21**

**Sinh viên thực hiện : 19120124 - Võ Thị Cẩm Quỳnh**

*TP. Hồ Chí Minh, ngày 17 tháng 12 năm 2021*

**MỤC LỤC**

**PHẦN 1: Ý TƯỞNG VÀ CÁC BƯỚC THỰC HIỆN** 1

***1.*** ***Các hàm dùng chung*** 1

***2.*** ***Apriori*** 1

***3.*** ***Tree projection*** 1

***4.*** ***Hàm main*** 2

**PHẦN 2: SO SÁNH HAI THUẬT TOÁN** 2

**PHẦN 3: KẾT QUẢ KHAI THÁC TẬP PHỔ BIẾN TRÊN CÁC TẬP DỮ LIỆU** 2

***1.*** ***Foodmart*** 2

***2.*** ***Retail*** 3

***3.*** ***Mushrooms*** 4

***4.*** ***Chess*** 5

**PHẦN 4: ỨNG DỤNG TRONG KHAI THÁC DỮ LIỆU THỰC TẾ** 6

# **PHẦN 1: Ý TƯỞNG VÀ CÁC BƯỚC THỰC HIỆN**

1. ***Các hàm dùng chung***

* Cài đặt các hàm dùng chung dưới đây trong file sharing\_function.jl:

+ readFile(filename::String) để đọc dữ liệu từ file data lưu và trả về 1 mảng 2 chiều.

+ twodims\_to\_singledim\_array(arr::Array{}) chuyển từ 1 mảng 2 chiều thành mảng 1 chiều.

+ support(item,trans::Array{}) tính độ hỗ trợ của phần tử bằng cách tìm kiếm và đếm số lần xuất hiện của item trong tập các giao tác.

+ supports(items::Array{},trans::Array{}) tính độ hộ trợ của 1 mảng các phần tử sử dụng hàm support.

+ show\_frequent\_itemset(frequent\_itemset::Array{},trans::Array{}) in ra màn hình danh sách các tập phổ biến và bao gồm độ hỗ trợ của chúng.

* Sử dụng các thư viện: IterTools, Statistics, Counters để tìm giá trị duy nhất, đếm số lần phần tử xuất hiện trong mảng…

1. ***Apriori***

*Ý tưởng:* Đầu tiên, thực hiện tìm tập phổ biến gồm 1 phần tử và từ tập phổ biến này được chọn lại các giá trị thỏa điều kiện lớn hơn hoặc bằng minsup lưu vào tập L, L được thêm vào tập phổ biến. Tiếp theo, phát sinh các tập ứng viên từ các giá trị trong L, duyệt điều kiện thỏa minsup và tiếp tục các bước như trên. Lặp lại quá trình này đến khi tập L rỗng hoặc số lượng phần tử trong L không lớn hơn chiều dài của tập sẽ phát sinh. Cuối cùng, trả về kết quả là mảng các tập L sau mỗi lần thực hiện duyệt phần tử. Ta thu được tập phổ biến cần tìm.

* Các hàm được cài đặt trong file apriori.jl:

+ largeSet(items::Array{},sup::Array{}, minsup::Float64) để chọn ra tập các giá trị có độ hỗ trợ lớn hơn bằng minsup.

+ apriori(transaction::Array{},minsup::Float64) dùng thuật toán apriori để tìm và trả về tập phổ biến.

1. ***Tree projection***

*Ý tưởng:* thực hiện tìm các giá trị có độ hỗ trợ lớn hơn hoặc bằng minsup và sắp xếp theo chiều giảm dần độ hỗ trợ. Sau đó dùng hàm orderedItemSet để tìm danh sách các hạng mục phổ biến cho mỗi giao dịch. Từ danh sách này, ta suy ra được cây dữ liệu chứa các giá trị, các hạng mục phổ biến và độ hỗ trợ của chúng. Tìm cơ sở điều kiện mẫu và suy ra cơ sở điều kiện FP tree. Cuối cùng phát sinh các tập con từ các cơ sở điều kiện FP tree này. Ta thu được tập phổ biến cần tìm.

* Các hàm được cài đặt trong file fpgrowth.jl:

+ frequentItemSet(trans,minsup) dùng để phát sinh tập phổ biến gồm 1 phần tử từ tập các giao tác trans.

+ orderedItemSet(trans,frequentSet) dùng để xây dựng dữ liệu cho cây, với mỗi giao dịch trong trans chỉ giữ lại các hạng mục phổ biến và sắp xếp chúng theo thứ tự giảm dần của tần số.

+ getHeaderTable(arr) dùng để phát sinh bảng với các hạng mục phổ biến trong cây và tần số của chúng.

+ getBaseCondition(itemset,header) tìm cơ sở mẫu điều kiện của 1 mảng item.

+ getFPcondition(base,frequency) tìm cơ sở điều kiện FP tree trên mảng các item, cuối cùng trả về cơ sở FP tree điều kiện.

+ FPgrowth(trans,minsup) dùng các hàm được cài đặt trên tìm được FP tree điều kiện và suy ra các tập phổ biến.

1. ***Hàm main***

* Hàm main được cài đặt trong file main.jl
* Input: tên file dữ liệu, minsup
* Output: in ra màn hình kích thước tập phổ biến sau khi chạy 2 thuật toán apriori và fpgrowth và thời gian chạy.
* Để chạy chương trình, ta cần dùng julia include đến file main.jl, sau đó nhập thông tin truyền vào gồm đường dẫn đến file dữ liệu và minsup.

# **PHẦN 2: SO SÁNH HAI THUẬT TOÁN**

|  |  |
| --- | --- |
| **Apriori** | **Tree projection** |
| Thuật toán Apriori thực hiện quá trình kết hợp sinh ra các tập ứng viên và loại bỏ các tập ứng viên không phù hợp, xây dựng dần các tập phổ biến từ dưới lên. | Thuật toán FP Growth xây dựng cây FP- tree, từ đó xây dựng cơ sở mẫu điều kiện và các FP-tree có điều kiện thỏa mãn độ hỗ trợ tối thiểu minsup, dùng chia để trị. |
| *Ưu điểm:*   * Đơn giản, dễ hiểu, dễ cài đặt * Thuật toán Apriori tìm tập mục phổ biến thực hiện tốt bởi rút gọn kích thước các tập ứng viên | *Ưu điểm:*   * Chỉ cần duyệt cơ sở dữ liệu 2 lần * Không phát sinh nhiều tập ứng viên * Tốn ít bộ nhớ |
| *Nhược điểm:*   * Duyệt cơ sở dữ liệu nhiều lần * Số lượng tập ứng viên lớn * Tốn nhiều thời gian và bộ nhớ * Để xác định độ support của các tập ứng viên, thuật toán luôn phải quyét lại toàn bộ cơ sở dữ liệu | *Nhược điểm:*   * Cài đặt phức tạp * Dữ liệu quá lớn có thể không thể lưu trữ hết trên cây |

# **PHẦN 3: KẾT QUẢ KHAI THÁC TẬP PHỔ BIẾN TRÊN CÁC TẬP DỮ LIỆU**

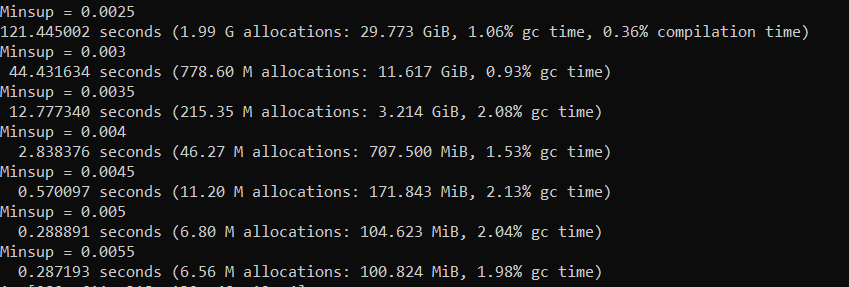
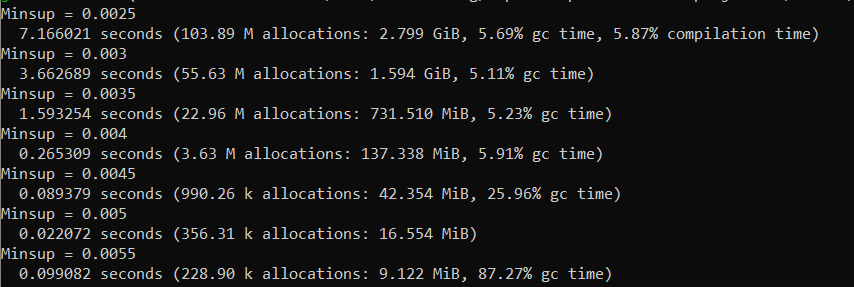
1. ***Foodmart:*** tập dữ liệu ở một cửa hàng bán lẻ

* Chart, line chart

  Description automatically generatedSự phân bố số lượng tập phổ biến theo độ hỗ trợ tối thiểu

Sự phân bố số lượng tập phổ biến theo độ hỗ trợ tối thiểu trong khoảng 0.0025 – 0.0055

* Thời gian, bộ nhớ của 2 thuật toán



|  |  |
| --- | --- |
| FP-growth | Apriori |

* Text

  Description automatically generatedNgưỡng minsup = 0.05:

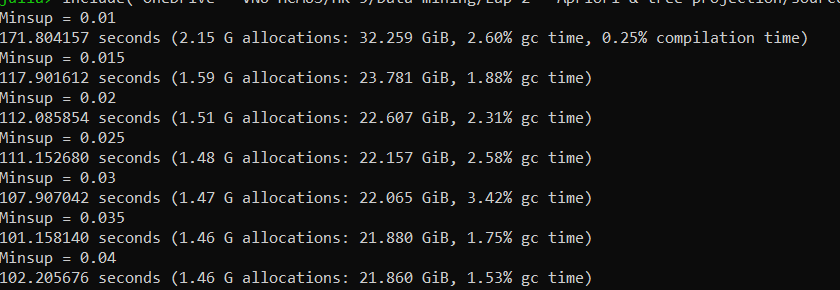
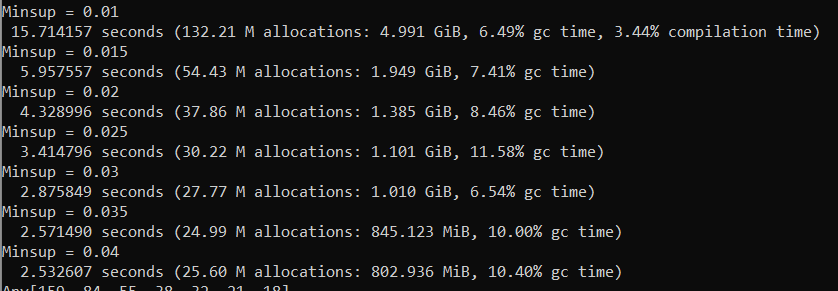
1. ***Retail***: tập các giao tác của khách hàng ở 1 cửa hàng bán lẻ tại Bỉ

* Chart, line chart

  Description automatically generatedSự phân bố số lượng tập phổ biến theo độ hỗ trợ tối thiểu

Sự phân bố số lượng tập phổ biến theo độ hỗ trợ tối thiểu trong khoảng 0.01 – 0.04

* Thời gian, bộ nhớ của 2 thuật toán



|  |  |
| --- | --- |
| FP-growth | Apriori |

* A black screen with white text

  Description automatically generated with low confidenceNgưỡng minsup = 0.035

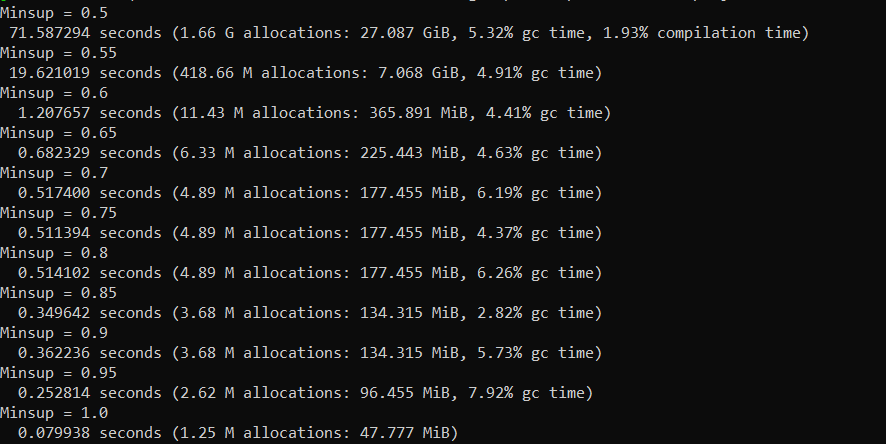
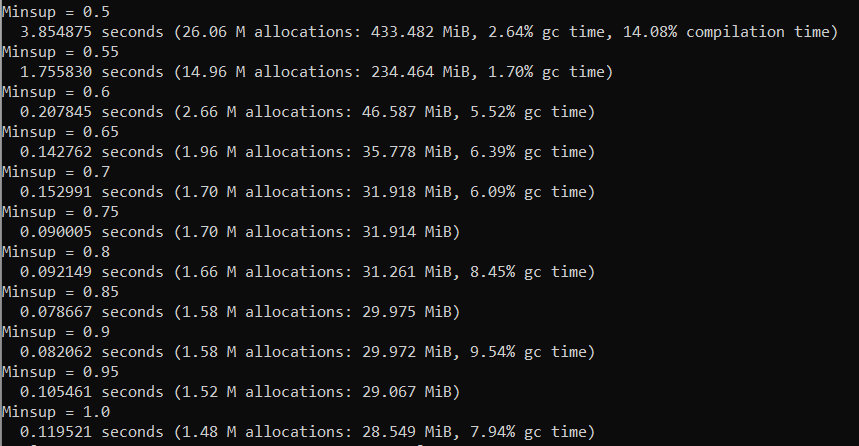
1. ***Mushrooms:*** dữ liệu về các đặc tính của loài nấm, độc hay không, màu sắc…

* Chart, line chart

  Description automatically generatedSự phân bố số lượng tập phổ biến theo độ hỗ trợ tối thiểu

Sự phân bố số lượng tập phổ biến theo độ hỗ trợ tối thiểu trong khoảng 0.5 - 1

* Thời gian, bộ nhớ của 2 thuật toán



|  |  |
| --- | --- |
| FP-growth | Apriori |

* A black background with white text

  Description automatically generated with low confidenceNgưỡng minsup = 0.8

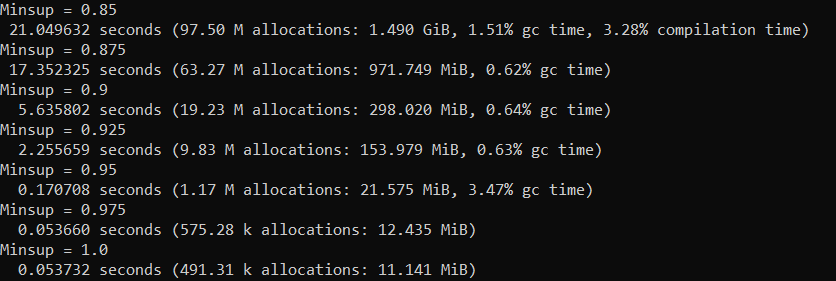
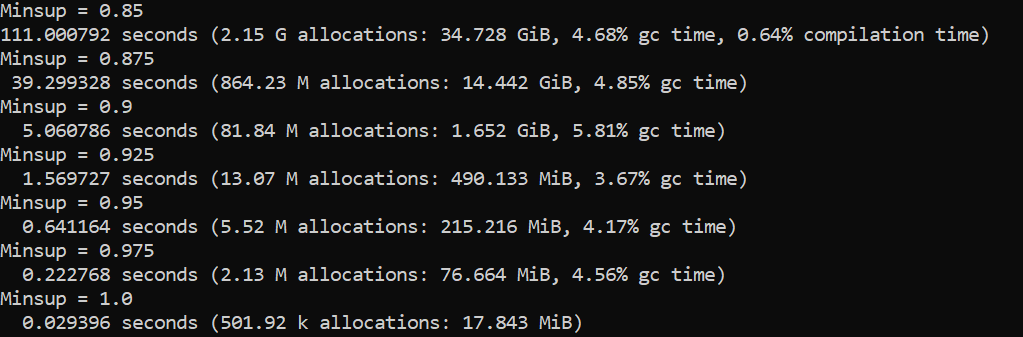
1. ***Chess*:** mô tả các bước của quân cờ

* Chart, line chart

  Description automatically generatedSự phân bố số lượng tập phổ biến theo độ hỗ trợ tối thiểu

Sự phân bố số lượng tập phổ biến theo độ hỗ trợ tối thiểu trong khoảng 0.8 - 1

* Thời gian, bộ nhớ của 2 thuật toán



|  |  |
| --- | --- |
| FP-growth | Apriori |

* A black screen with white text

  Description automatically generated with low confidenceNgưỡng minsup=0.975

# **PHẦN 4: ỨNG DỤNG TRONG KHAI THÁC DỮ LIỆU THỰC TẾ**

* Ứng dụng khai thác tập phổ biến và khai phá luật kết hợp.
* Ứng dụng khai phá luật kết hợp trong khai phá dữ liệu hình ảnh.
* Ứng dụng trong khảo sát sản phẩm được mua trong cửa hàng, siêu thị để đưa ra dự đoán về các mặt hàng thường được mua cùng nhau.
* Trong y tế: phân tích cơ sở dữ liệu của bệnh nhân, suy ra các loại bệnh liên quan nhau dựa trên luật kết hợp và dự đoán được khả năng mắc 1 loại bệnh của 1 người cụ thể.
* Trong bảo vệ rừng: phân tích khả năng và cường độ cháy rừng của 1 khu vực dựa vào cơ sở dữ liệu cháy rừng.
* Apriori được Amazon dùng trong hệ thống đề xuất và Google dùng trong tính năng tự động.