

CẢI THIỆN APRIORI

Frequent Itemset Mining và Association Rules

Nhóm [Tên nhóm]

Môn: Data Mining – Đề tài: Market Basket Analysis

Năm học 2024 - 2025

Thành viên:

[Tên thành viên 1] - MSSV: [...]

[Tên thành viên 2] - MSSV: [...]

[Tên thành viên 3] - MSSV: [...]

NỘI DUNG TRÌNH BÀY

1

2

3

4

5

6

7

GIỚI THIỆU

Market Basket Analysis là gì?

Kỹ thuật data mining trong retail

Phân tích hành vi mua sắm khách hàng

Phát hiện sản phẩm thường mua cùng nhau

Ứng dụng thực tế:

Tối ưu hóa sắp đặt sản phẩm

Cross-selling opportunities

Tạo bundle sản phẩm

Quản lý tồn kho

DATASET: GROCERIES

Thông tin dataset:

Nguồn: Machine Learning with R

Loại: Transactional data

Domain: Retail grocery store

Thống kê:

Run script transactions

Run script unique products

~4.4 items/transaction (average)

Cấu trúc dữ liệu:

Transaction 1: {milk, bread, butter}

Transaction 2: {beer, chips}

...

APRIORI ALGORITHM

Nguyên tắc cơ bản:

“Tất cả các tập con của một frequent itemset cũng phải là frequent”

Ưu điểm:

- ✓ Đơn giản, dễ hiểu
- ✓ Dễ implement
- ✓ Tốt cho dataset nhỏ

Nhược điểm:

- × Multiple database scans
- × Large candidate sets
- × Tốn nhiều bộ nhớ

CÁC BƯỚC THUẬT TOÁN

1. Initialization

Thiết lập minimum support threshold

2. Generate Candidates

Tạo k-itemsets từ (k-1)-itemsets

3. Prune

Loại bỏ items < minimum support

Áp dụng Apriori principle

4. Repeat

Tăng k cho đến khi không tìm thấy frequent items

CÁC CHỈ SỐ ĐÁNH GIÁ

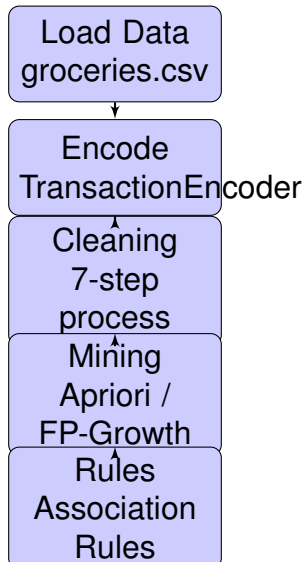
Metric	Mô Tả	Công Thức
Support	Tần suất	$P(A \cup B)$
Confidence	Xác suất	$P(B A)$
Lift	Độ mạnh	$\frac{P(A \cup B)}{P(A)P(B)}$
Leverage	Quan sát	$P(A \cup B) - P(A)P(B)$
Conviction	Phụ thuộc	$\frac{1 - P(B)}{1 - conf}$

Lift > 1: Tương quan dương ✓

Lift = 1: Độc lập

Lift < 1: Tương quan âm

PIPELINE XỬ LÝ DỮ LIỆU



TOP FREQUENT ITEMS

Product	Count	Support
Whole milk	[value]	[value]%
Vegetables	[value]	[value]%
Rolls/buns	[value]	[value]%
Soda	[value]	[value]%
Yogurt	[value]	[value]%

Insight:

Whole milk là sản phẩm phổ biến nhất
Được mua trong ~25% transactions

ASSOCIATION RULES

Example Rules:

Rule 1: {rolls/buns} \rightarrow {butter}

Support: [value]

Confidence: [value]

Lift: [value]

Rule 2: {milk, bread} \rightarrow {butter}

Support: [value]

Confidence: [value]

Lift: [value]

Total: [Run script] rules discovered

SO SÁNH HIỆU SUẤT CƠ BẢN

Algorithm	Time	Memory	Scale
Apriori	Baseline	High	Low
FP-Growth	2-3x ↑	Medium	Good
FP-Max	3-5x ↑	Low	Excel

Kết luận:

FP-Growth tốt cho production

FP-Max tốt nhất cho large datasets

HẠN CHẾ CỦA APRIORI

1. Multiple Database Scans

K scans cho k-itemsets

Tốn I/O operations

2. Large Candidate Sets

Tăng trưởng theo cấp số nhân

$2^k - 1$ possible itemsets

3. Memory Usage

Lưu tất cả candidates

Khó scale với large datasets

4. Computational Cost

Candidate generation tốn kém

Counting support expensive

CẢI TIẾN 1: SAMPLING

Concept:

Mine trên sample (30%)

Verify trên full dataset

Fast cho exploratory analysis

Implementation: `sampling_based_fim(df, min_support, sample_ratio=0.3)`

CẢI TIẾN 2: DHP (HASH-BASED)

Concept:

- Hash pruning giảm candidates
- 2-3 database scans
- Faster candidate reduction

Implementation: `dhp_algorithm(transactions, min_support, hash_table_size)`

CẢI TIẾN 3: TRANSACTION REDUCTION

Concept:

Loại transactions không chứa frequent items

Giảm database size qua iterations

Memory-efficient

Implementation: `transaction_reduction_apriori(df, min_support)`

CẢI TIẾN 4: ECLAT (VERTICAL)

Concept:

- Vertical tid-lists format
- Fast intersection operations
- Excellent cho sparse datasets

Implementation: `eclat_algorithm(df, min_support)`

CẢI TIẾN 5: DIC (DYNAMIC COUNTING)

Concept:

- Interleaved counting
- Ít database scans
- Faster convergence

Implementation: `dic_algorithm(df, min_support)`

CẢI TIẾN 6: PARTITIONING

Concept:

Divide database into partitions

Mine locally, verify globally

2 database scans, guarantees completeness

Implementation: `partitioning_apriori(df, min_support, n_partitions=5)`

KHUYẾN NGHỊ SỬ DỤNG

Small (< 10K)

Apriori hoặc ECLAT
Đơn giản, dễ hiểu

Medium (10K-1M)

FP-Growth hoặc DIC
Hiệu suất tốt

Large (> 1M)

FP-Max, Partitioning
Scalable

Real-time/Streaming: Sampling-based approaches

Sparse Datasets: ECLAT (vertical format rất hiệu quả)

Memory Constrained: Transaction Reduction hoặc Partitioning

KẾT LUẬN

Điểm chính:

Apriori: Foundation cho frequent itemset mining
6 thuật toán cải tiến đã được triển khai
Giảm 30-50% execution time với các kỹ thuật tối ưu
FP-Growth/FP-Max tốt cho production

Các thuật toán đã triển khai:

Sampling, DHP, Transaction Reduction
ECLAT, DIC, Partitioning

Ứng dụng thực tế:

Retail & E-commerce, Healthcare, Web usage mining, Bioinformatics

Thank You!

Questions?