APRIORI Algorithm

Md. Jamil Uddin
East Delta University

Association Rule Mining

- Association rule mining finds interesting associations and relationships among large sets of data items. This can be considered as the frequent pattern mining from a given dataset.

  Example: **Market Basket Analysis**

- Association rule mining has various applications beyond market basket analysis, including fraud detection, cross-selling and upselling in e-commerce, recommendation systems, and more. It's a valuable tool for discovering hidden patterns and relationships within datasets and can be implemented using algorithms like the Apriori algorithm or the FP-growth algorithm.
- The primary goal of association rule mining is to find patterns of co-occurrence and quantify the strength of these associations.

Important Keywords

- **Association rule** - An association rule is a data mining concept that represents a statistical relationship or pattern between sets of items in a dataset. It's typically expressed as an "if-then" statement, where the "if" part (the antecedent) represents a condition or a set of items, and the "then" part (the consequent) represents an outcome or another set of items. Association rules are used to discover interesting and often non-obvious relationships between variables in large datasets.

  Example: the information that customers who purchase computers also tend to buy antivirus software at the same time is represented in the following association rule:

  $$\text{Computer} \Rightarrow \text{antivirus software [Support = 2\%, Confidence = 60\%]}$$

- **Support** - A 2% support for the mentioned association rule indicates that antivirus software and computers are bought jointly in 2% of the transactions that are analyzed.

$$\textit{Support}\ (A \Rightarrow B) = P(A \cup B)$$

- **Confidence** - A 60% confidence level indicates that 60% of consumers who bought a PC also bought the software.

$$\textit{Confidence}\ (A \Rightarrow B) = P(B \mid A) = (\textit{Support}\ (A \cup B) / \textit{Support}\ (A))$$

- In general, association rules are deemed intriguing when they meet both a minimal confidence requirement and a minimum support threshold.

- Rules that satisfy both a minimum support threshold (min sup) and a minimum confidence threshold (min conf) are called **strong.**

- By convention, we write support and confidence values so as to occur between 0% and 100%, rather than 0 to 1.0.

- **Itemset -** Set of ‘n’ number of items

- **K-itemset -** An itemset consisting of ‘K’ number of items

- **Candidate Itemset -** An itemset that can be a frequent itemset if it satisfies minimum threshold value of support

- **Frequent Itemset -** An itemset that satisfies minimum threshold value of support

➔ *{Computer, Antivirus, CD} is a* ***3-itemset***

- **Downward Closure Property -** If an itemset is frequent then its subset is also a frequent itemset. Such as - If {Computer, Antivirus, CD} is a frequent itemset, then {Computer, CD} must be frequent too.

Max Patterns vs Closed Patterns

Max Pattern	Closed Pattern
An itemset X is a max-pattern if X is frequent and there exists no frequent super-pattern $Y \supset X$	An itemset X is closed if X is frequent and there exists no super-pattern $Y \supset X$, with the same support as X
Maximal patterns are frequent patterns that cannot be extended to include more items without decreasing their support to a level below the minimum support threshold. They represent the most specific patterns in the dataset that are still considered frequent.	If any item is removed from a closed pattern, the resulting pattern is no longer frequent.
If {A, B, C} is a maximal pattern with a support of 0.2, it implies that this pattern is frequent and cannot be extended by adding more items without making it non-frequent.	In a market basket analysis, if {A, B, C} is a closed pattern with a support of 0.2, this means that the items A, B, and C are frequently bought together, and there are no other items that, when added to this set, maintain the same level of support.

The Apriori Algorithm

- In 1994, R. Agrawal and R. Srikant presented the groundbreaking algorithm Apriori for mining frequent itemsets for Boolean association rules.
- The algorithm's name stems from the fact that it makes use of past knowledge of common itemset attributes.
- Apriori uses an iterative method called a level-wise search, in which (k+1)-itemsets are explored using k-itemsets.

How is Apriori Property is used in algorithm?

In 2 steps the association rule mining is done by Apriori Algorithm. They are - **Join Step** and **Prune Step.**

- **Join Step :** To find L_k, a set of candidate k-itemsets is generated by joining L_{k-1} with itself. This set of candidates is denoted C_k.

- **Prune Step:** Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset. Hence, if any (k-1)-subset of a candidate k-itemset is not in L_{k-1}, then the candidate cannot be frequent either and so can be removed from C_k. This subset testing can be done quickly by maintaining a hash tree of all frequent itemsets.

Pseudo Code for Apriori Algorithm

C_k: Candidate itemset of size k

L_k : frequent itemset of size k

L_1 = {frequent items};

for (k = 1; L_k !=Ø; k++) **do begin**

C_{k+1} = candidates generated from L_k;

for each transaction t in database do

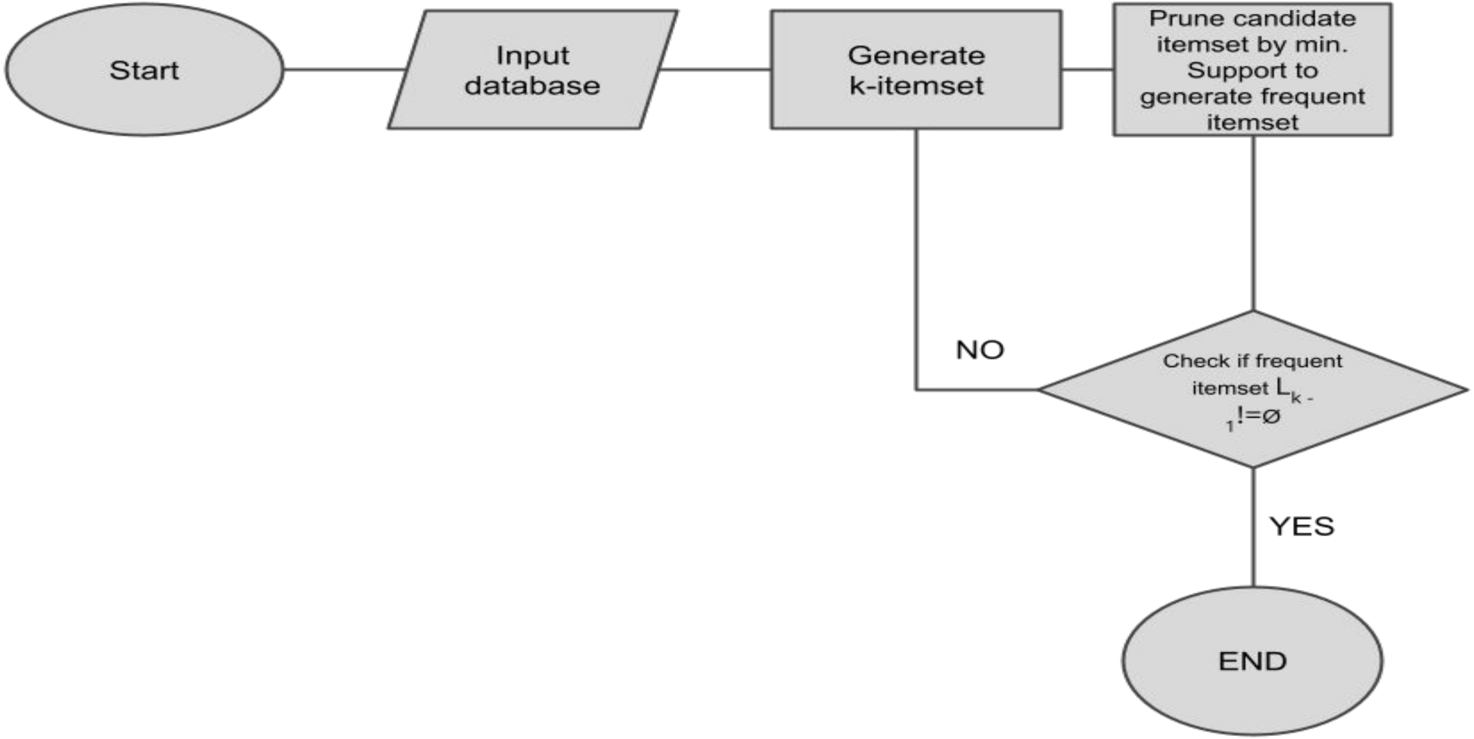
increment the count of all candidates in C_{k+1} that are contained in t

L_{k+1} = candidates in C_{k+1} with min_support

end

return $\cup_k L_k$;

Flowchart for Generating Frequent Itemset



Association Rule Generation

Association rules can be generated as follows:

- For each frequent itemset l, generate all nonempty subsets of l.
- For every nonempty subset s of l, output the rule "**s⇒l-s**" if ***Support_count(l) / Support_count(s) >= min_conf***, where min_conf is the minimum confidence threshold.
- Where confidence of any rule **A⇒B** is:

Confidence (A⇒B) = Support_count (A U B) / Support_count (A)

Example:

For the following given Transaction Data-set, Generate Rules using Apriori Algorithm.

Consider the values as Support=50% and Confidence=75%

Transaction ID	Items Purchased
1	Bread, Cheese, Egg, Juice
2	Bread, Cheese, Juice
3	Bread, Milk, Yogurt
4	Bread, Juice, Milk
5	Cheese, Juice, Milk

Step 1: Find Frequent Item Set and their support

1-itemset:

Item	Frequency	Support (in %)
Bread	4	4/5=80%
Cheese	3	3/5=60%
Egg	1	1/5=20%
Juice	4	4/5=80%
Milk	3	3/5=60%
Yogurt	1	1/5=20%

Step 2: Remove all the items whose support is below given minimum support.

1-itemset:

Items	Frequency	Support (in %)
Bread	4	4/5=80%
Cheese	3	3/5=60%
Juice	4	4/5=80%
Milk	3	3/5=60%

Step 3: Form the two items candidate set

Items Pair	Frequency	Support (in %)
Bread, Cheese	2	2/5=40%
Bread, Juice	3	3/5=60%
Bread, Milk	2	2/5=40%
Cheese, Juice	3	3/5=60%
Cheese, Milk	1	1/5=20%
Juice, Milk	2	2/5=40%

Step 4: Remove all the items below minimum support

Items Pair	Frequency	Support (in %)
Bread, Juice	3	3/5=60%
Cheese, Juice	3	3/5=60%

Step 5: Rule Generation

For Rules we consider item pairs:

➢ (Bread, Juice)

Bread->Juice and Juice->Bread

➢ (Cheese, Juice)

Cheese->Juice and Juice->Cheese

Confidence (A->B) = support (AUB)/support (A)

Rule Validation

I. Confidence (Bread->Juice) = support (Bread U Juice)/support (Bread) = 3/5 * 5/4=3/4= **75%**

II. Confidence (Juice->Bread) = support (Juice U Bread)/support (Juice) = 3/5*5/4=3/4=**75%**

III. Confidence (Cheese->Juice) = support (Cheese U Juice)/support (Cheese)=3/5*5/3=1=**100%**

IV. Confidence (Juice->Cheese) = support (Juice U Cheese)/support (Juice) = 3/5*5/4=3/4=**75%**

★ All the rules are valid as they satisfy the minimum confidence value.

Limitations of Apriori Algorithm

- Exponential Growth of Candidate Generation
- Inefficient for Sparse Data
- Multiple Database Scans
- Difficulty Handling Continuous or Numeric Data
- High Memory Usage
- No Consideration of Item Order
- Limited Discovery of Infrequent Itemsets
- Difficulty with Variable-Length Itemsets
- Difficulty Handling Large Itemsets

Improving the performance of Apriori algorithm

- Hash based Technique
- Sampling
- Dataset Partitioning
- Transaction Reduction
- Dynamic Itemset Counting

Reference

[1] J. Han and M. Kamber, *Data Mining : Concepts and Techniques*, 3rd ed. Haryana, India ; Burlington, Ma: Elsevier, 2018.

[2] R. Agrawal and Ramakrishnan Srikant, “Fast algorithms for mining association rules,” pp. 580–592, Jul. 1998.

[3] “•Apriori Algorithm in Data Mining.” Available: https://www.cvs.edu.in/upload/Apriori%20Algorithm.pdf

[4] “(PDF) Association Rule Mining--Apriori Algorithm Solved Problems,” *ResearchGate*. https://www.researchgate.net/publication/340105166_Association_Rule_Mining--Apriori_Algorithm_Solved_Problems