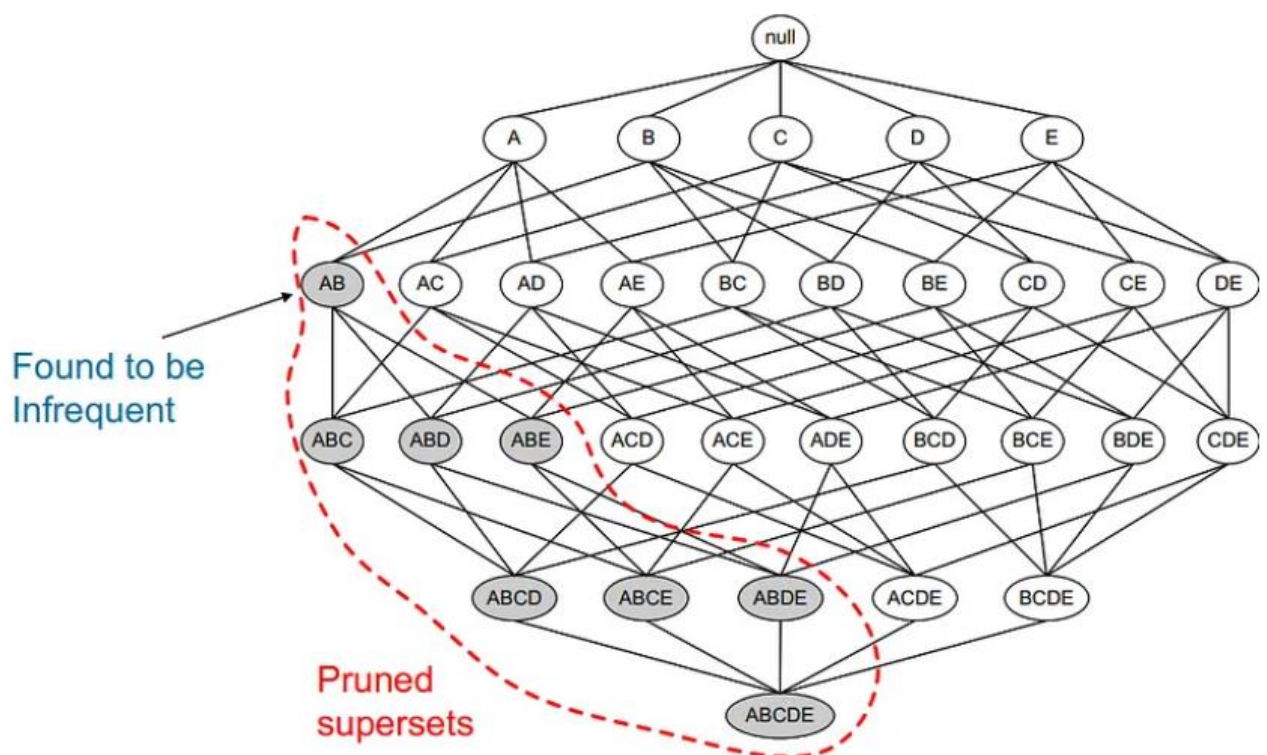


Apriori Algorithm in Machine Learning

- ❖ Discover frequent itemsets and associations within transaction databases.
- ❖ breadth-first search and hash tree for efficient calculations.
- ❖ Association rules revealing connections between items.
- ❖ Iterative process suitable for large datasets.
- ❖ Efficient search techniques minimize computational cost.
- ❖ Developed by R. Agrawal and R. Srikant in 1994.
- ❖ To better understand the apriori algorithm, and related term such as support and confidence, it is recommended to understand the association rule learning.

Applications:

- ❖ Market basket analysis: Identify products frequently purchased together (e.g., bread & butter).
- ❖ Healthcare: Analyze patient data to uncover potential drug interactions.



Steps for Apriori Algorithm

Step-1: Determine the support of itemsets in the transactional database, and select the minimum support and confidence.

Step-2: Take all supports in the transaction with higher support value than the minimum or selected support value.

Step-3: Find all the rules of these subsets that have higher confidence value than the threshold or minimum confidence.

Step-4: Sort the rules as the decreasing order of lift.

Apriori Algorithm Working

Example: Suppose we have the following dataset that has various transactions, and from this dataset, we need to find the frequent itemsets and generate the association rules using the Apriori algorithm:

TID	ITEMSETS
T1	A, B
T2	B, D
T3	B, C
T4	A, B, D
T5	A, C
T6	B, C
T7	A, C
T8	A, B, C, E
T9	A, B, C

Given: Minimum Support= 2, Minimum Confidence= 50%

Solution:

Step-1: Calculating C1 and L1:

- In the first step, we will create a table that contains support count (The frequency of each itemset individually in the dataset) of each itemset in the given dataset. This table is called the **Candidate set or C1**.

Itemset	Support_Count
A	6
B	7
C	5
D	2
E	1

- Now, we will take out all the itemsets that have the greater support count than the Minimum Support (2). It will give us the table for the **frequent itemset L1**.
Since all the itemsets have greater or equal support count than the minimum support, except the E, so E itemset will be removed.

Itemset	Support_Count
A	6
B	7
C	5
D	2

Step-2: Candidate Generation C2, and L2:

- In this step, we will generate C2 with the help of L1. In C2, we will create the pair of the itemsets of L1 in the form of subsets.
- After creating the subsets, we will again find the support count from the main transaction table of datasets, i.e., how many times these pairs have occurred together in the given dataset. So, we will get the below table for C2:

Itemset	Support_Count
{A, B}	4
{A, C}	4
{A, D}	1
{B, C}	4
{B, D}	2
{C, D}	0

- Again, we need to compare the C2 Support count with the minimum support count, and after comparing, the itemset with less support count will be eliminated from the table C2. It will give us the below table for L2

Itemset	Support_Count
{A, B}	4
{A, C}	4
{B, C}	4
{B, D}	2

A, B, C, D

Step-3: Candidate generation C3, and L3:

- For C3, we will repeat the same two processes, but now we will form the C3 table with subsets of three itemsets together, and will calculate the support count from the dataset. It will give the below table:

Itemset	Support_Count
{A, B, C}	2
{B, C, D}	1
{A, C, D}	0
{A, B, D}	0

- Now we will create the L3 table. As we can see from the above C3 table, there is only one combination of itemset that has support count equal to the minimum support count. So, the L3 will have only one combination, i.e., **{A, B, C}**.

Step-4: Finding the association rules for the subsets:

To generate the association rules, first, we will create a new table with the possible rules from the occurred combination {A, B, C}. For all the rules, we will calculate the Confidence using formula $\text{sup}(A \wedge B)/A$. After calculating the confidence value for all rules, we will exclude the rules that have less confidence than the minimum threshold(50%).

Consider the below table:

Rules	Support	Confidence
$A \wedge B \rightarrow C$	2	$\text{Sup}((A \wedge B) \wedge C)/\text{sup}(A \wedge B) = 2/4 = 0.5 = 50\%$
$B \wedge C \rightarrow A$	2	$\text{Sup}((B \wedge C) \wedge A)/\text{sup}(B \wedge C) = 2/4 = 0.5 = 50\%$
$A \wedge C \rightarrow B$	2	$\text{Sup}((A \wedge C) \wedge B)/\text{sup}(A \wedge C) = 2/4 = 0.5 = 50\%$
$C \rightarrow A \wedge B$	2	$\text{Sup}((C \wedge (A \wedge B)))/\text{sup}(C) = 2/5 = 0.4 = 40\%$
$A \rightarrow B \wedge C$	2	$\text{Sup}((A \wedge (B \wedge C)))/\text{sup}(A) = 2/6 = 0.33 = 33.33\%$
$B \rightarrow B \wedge C$	2	$\text{Sup}((B \wedge (B \wedge C)))/\text{sup}(B) = 2/7 = 0.28 = 28\%$

As the given threshold or minimum confidence is 50%, so the first three rules **$A \wedge B \rightarrow C$, $B \wedge C \rightarrow A$, and $A \wedge C \rightarrow B$** can be considered as the strong association rules for the given problem.

Methods To Improve Apriori Efficiency

- Hash-Based Technique:** This method uses a hash-based structure called a hash table for generating the k-itemsets and its corresponding count. It uses a hash function for generating the table.
- Transaction Reduction:** This method reduces the number of transactions scanning in iterations. The transactions which do not contain frequent items are marked or removed.
- Partitioning:** This method requires only two database scans to mine the frequent itemsets. It says that for any itemset to be potentially frequent in the database, it should be frequent in at least one of the partitions of the database.
- Sampling:** This method picks a random sample S from Database D and then searches for frequent itemset in S. It may be possible to lose a global frequent itemset. This can be reduced by lowering the min_sup.
- Dynamic Itemset Counting:** This technique can add new candidate itemsets at any marked start point of the database during the scanning of the database.

Applications Of Apriori Algorithm

- **In Education Field:** Extracting association rules in data mining of admitted students through characteristics and specialties.
- **In the Medical field:** For example Analysis of the patient's database.
- **In Forestry:** Analysis of probability and intensity of forest fire with the forest fire data.
- Apriori is used by many companies like Amazon in the **Recommender System** and by Google for the auto-complete feature.

Advantages of Apriori Algorithm

- This is easy to understand algorithm
- The join and prune steps of the algorithm can be easily implemented on large datasets.

Disadvantages of Apriori Algorithm

- The apriori algorithm works slow compared to other algorithms.
- The overall performance can be reduced as it scans the database for multiple times.
- The time complexity and space complexity of the apriori algorithm is $O(2^D)$, which is very high. Here D represents the horizontal width present in the database.