Introduction to APRIORI

- Apriori is the most famous frequent pattern mining method. It scans dataset repeatedly and generate item sets by bottom-top approach.
- Apriori algorithm is given by R. Agrawal and R. Srikant in 1994 for finding frequent itemsets in a dataset for boolean association rule. Name of the algorithm is Apriori because it uses prior knowledge of frequent itemset properties

What is the Apriori Algorithm?

- Apriori algorithm, a classic algorithm, is useful in mining frequent itemsets and relevant association rules. Usually, you operate this algorithm on a database containing a large number of transactions. One such example is the items customers buy at a supermarket.
- It helps the customers buy their items with ease, and enhances the sales performance of the departmental store.
- This algorithm has utility in the field of healthcare as it can help in detecting adverse drug reactions (ADR) by producing association rules to indicate the combination of medications and patient characteristics that could lead to ADRs.

How does the Apriori Algorithm in Data Mining work?

- We shall explain this algorithm with a simple example.
- Consider a supermarket scenario where the itemset is I = {Onion, Burger, Potato, Milk, Beer}. The database consists of six transactions where 1 represents the presence of the item and 0 the absence.

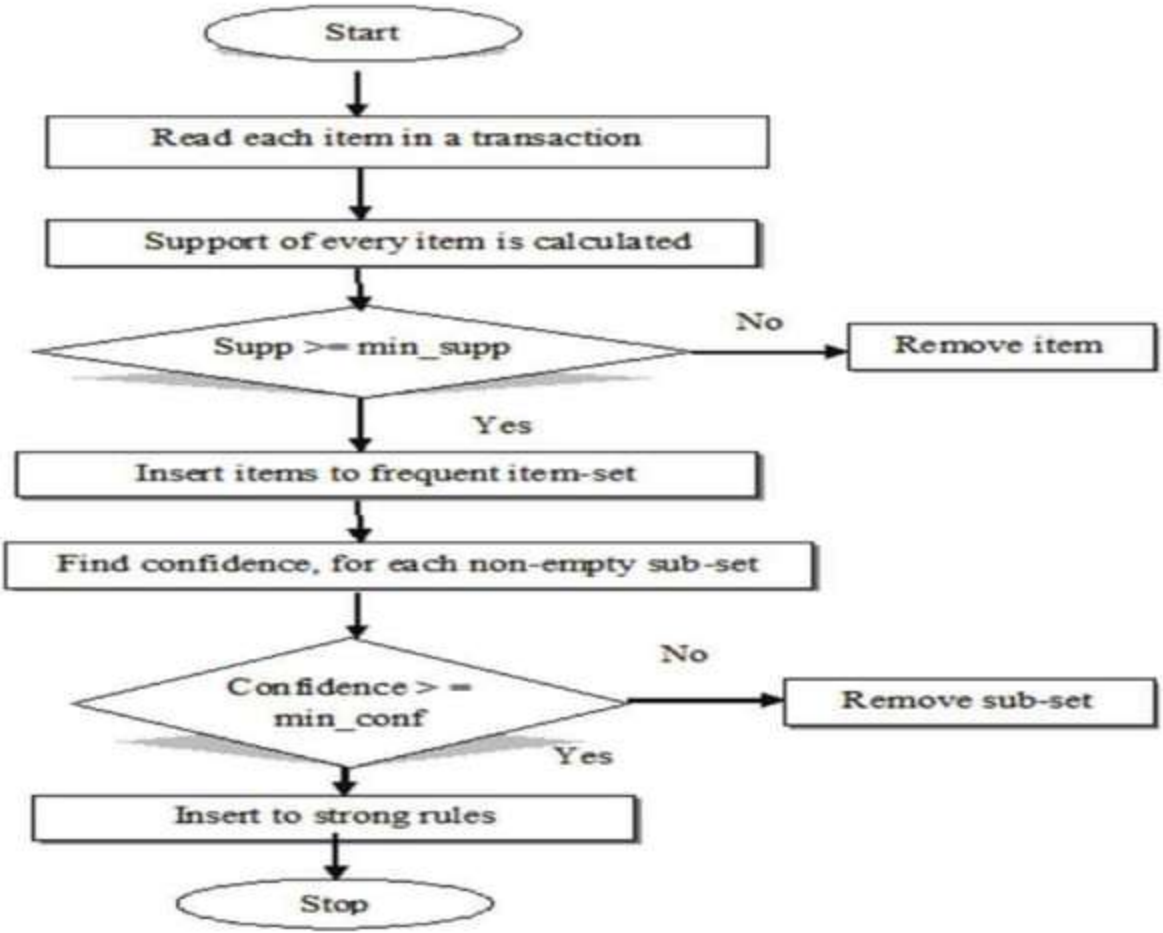
Contd...

Transaction ID	Onion	Potato	Burger	Milk	Beer
t_1	1	1	1	0	0
t_2	0	1	1	1	0
t_3	0	0	0	1	1
t_4	1	1	0	1	0
t_5	1	1	1	0	1
t_6	1	1	1	1	0




The Apriori Algorithm makes the following assumptions

- All subsets of a frequent itemset should be frequent.
- In the same way, the subsets of an infrequent itemset should be infrequent.
- Set a threshold support level. In our case, we shall fix it at 50%


Start
Read each item in a transaction
Support of every item is calculated
Supp >= min_supp
No
Remove item
Yes
Insert items to frequent item-set
Find confidence, for each non-empty sub-set
Confidence > = min_conf
No
Remove sub-set
Yes
Insert to strong rules
Stop

Steps

- Create a frequency table of all the items that occur in all the transactions. Now, prune the frequency table to include only those items having a threshold support level over 50%.
- We arrive at this frequency table.

Threshold Support >= 50%

Step 1 : One Itemset
C1

Itemset	Frequency
Onion	4
Potato	5
Burger	4
Milk	4
Beer	2

Pruning

Step 2
F1

Itemset	Frequency
Onion	4
Potato	5
Burger	4
Milk	4

Make pairs of items such as OP, OB, OM, PB, PM, BM. This frequency table is what you arrive at.

Step 3 : Two Itemset C2

Itemset	Frequency
Onion Potato	4
Onion Burger	3
Onion Milk	2
Potato Burger	4
Potato Milk	3
Burger Milk	2

Pruning

Apply the same threshold support of 50% and consider the items that exceed 50% (in this case 3 and above).
Thus, you are left with OP, OB, PB, and PM

Step 4 F2

Itemset	Frequency
Onion Potato	4
Onion Burger	3
Potato Burger	4
Potato Milk	3

Note: Consider items from F1 only for making C2

Look for a set of three items that the customers buy together. Thus we get this combination.
OP and OB gives OPB
PB and PM gives PBM

Determine the frequency of these two itemsets. You get this frequency table.
If you apply the threshold assumption, you can deduce that the set of three items frequently purchased by the customers is OPB.
We have taken a simple example to explain the apriori algorithm in data mining. In reality, you have hundreds and thousands of such combinations.

Step 6 : Three Itemset C3

Itemset	Frequency
Onion Potato Burger	3
Potato Burger Milk	2

Pruning

Step 7 F3

Itemset	Frequency
Onion Potato Burger	3

Note: Consider items from F2 only for making C3

Subset Creation

I = (O P B)

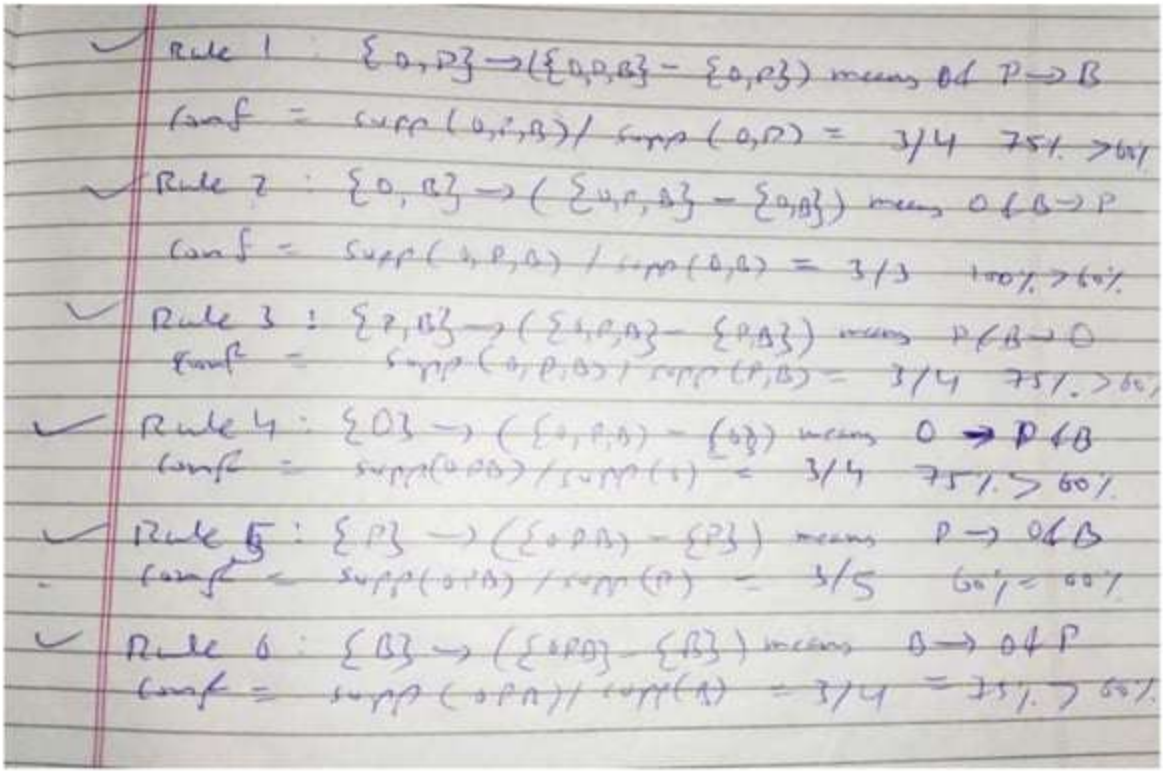
S = (OP) (OB) (PB) (O) (P) (B)

For every subset S of I, Output the rule:

$S \longrightarrow (I - S)$ (S recommends I-S)

If Support(I)/ Support(S) >= min_conf value

Contd...



Apriori Algorithm – Pros

- Easy to understand and implement
- Can use on large itemsets

Apriori Algorithm – Cons

- At times, you need a large number of candidate rules. It can become computationally expensive.
- It is also an expensive method to calculate support because the calculation has to go through the entire database.

Apriori Algorithm – Limitations

- The process can sometimes be very tedious.

How to Improve the Efficiency of the Apriori Algorithm?

Use the following methods to improve the efficiency of the apriori algorithm.

- **Transaction Reduction –** A transaction not containing any frequent k-itemset becomes useless in subsequent scans.

- **Hash-based Itemset Counting –** Exclude the k-itemset whose corresponding hashing bucket count is less than the threshold is an infrequent itemset.
- There are other methods as well such as partitioning, sampling, and dynamic itemset counting.