Apriori Algorithm

Definition:

Apriori is a mining algorithm for finding frequent item sets using prior knowledge of the item-set properties. This algorithm employs the *support–confidence framework*, and an iterative framework known as *level-wise search*:

- First, the frequency of each item is found by scanning of the transactions. The items that satisfy the minimum support count are the items of interest. The minimum support is found by using the formula:

$$support(A \Rightarrow B) = P(A \cup B)$$

- The resulting set found above is used for finding the frequent 2-item-sets, and then frequent 3-item-sets. This process occurs iteratively until no more frequent k-item-sets can be found.

- Once all the frequent item-sets have been found which satisfy the minimum support count, the confidence formula given below is used for generating strong association rules:

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)} = \frac{support\_count(A \cup B)}{support\_count(A)}.$$

The *Apriori property* is used for reducing the search space. The Apriori property states that all non-empty subsets of frequent item-sets must also be frequent. Thus, if the support of one of the items is less that the minimum threshold support, the supports of all the supersets containing that item will also be less than the threshold. This is also known as the **anti-monotonicity property** or the **downward-closure property** of support.

Weakness of the Apriori Algorithm:

The support and confidence measures might be insufficient at filtering out uninteresting association rules. To tackle this weakness, a correlation measure can be used to augment the support–confidence framework for association rules. Thus, we not only measure the support and confidence but also measure the correlation between different item-sets. Discussed below are two correlation measures which are often used:

1. Lift: The lift between 2 items A and B can be measured by computing:

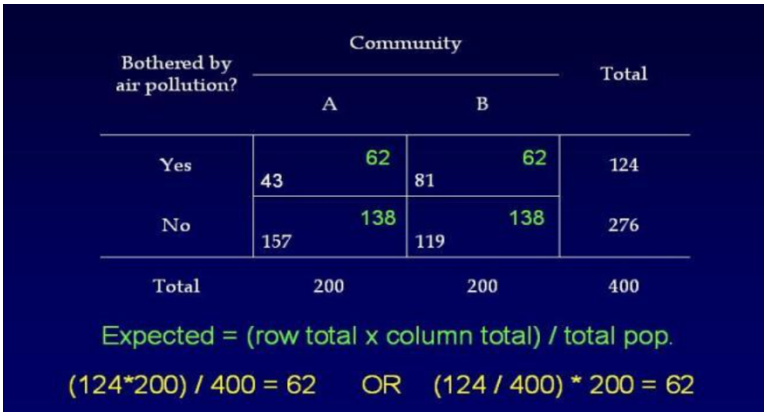
$$lift(A, B) = \frac{P(A \cup B)}{P(A)P(B)}.$$

If the resulting value of the above equation is < 1, then the occurrence of A is negatively correlated with the occurrence of B. If the resulting value of the above equation is > 1, then A and B are positively correlated. Furthermore, if the resulting value is equal to 1, A are B are independent & there is no correlation between them.

2. Chi-Squared Measure: To compute the chi-squared value, we take the squared differences between the observed and the expected value (mean) for the pair of A and B values and divide by the expected value (mean). The Chi-squared measure is computed using the equation below:

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

A snapshot below shows an example of how to compute the expected value from observed data:

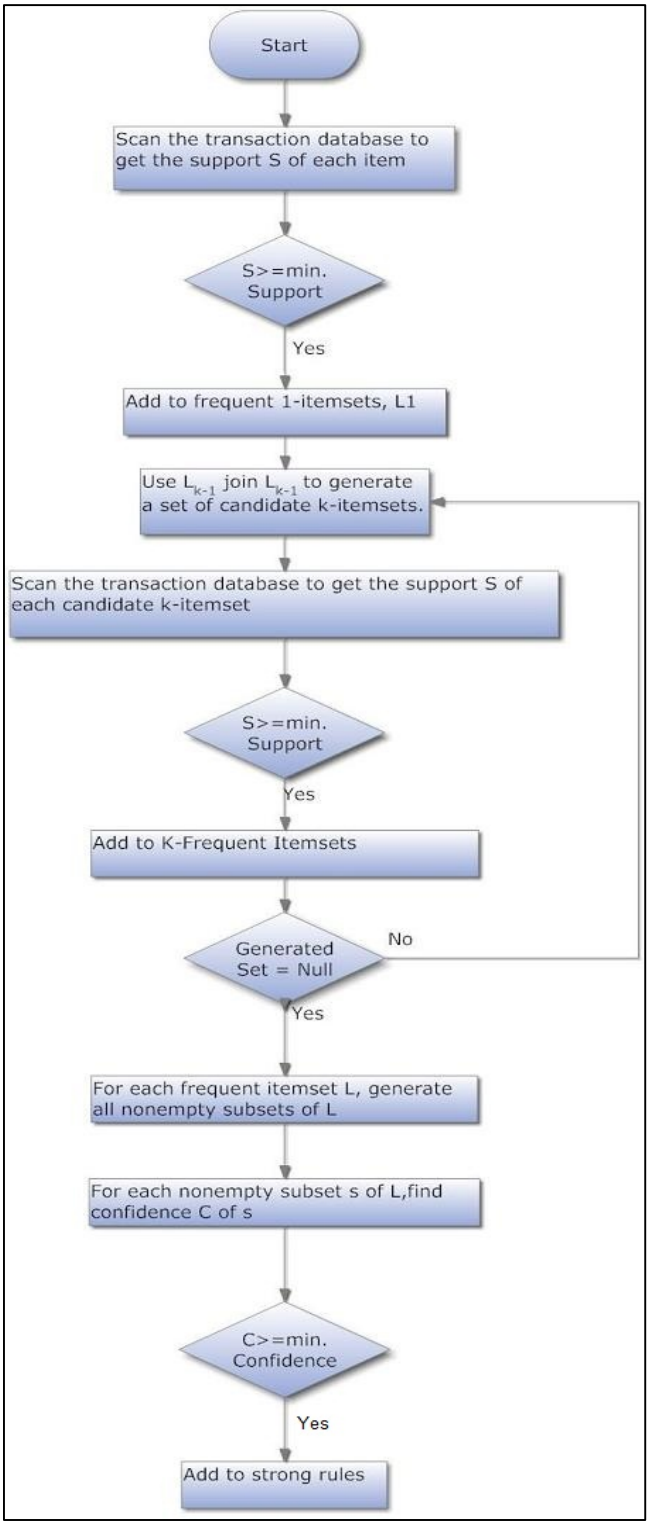


Improving the efficiency of Apriori:

Several variations of the Apriori algorithm used for improving the efficiency are summarized below:

- *Hash-based item-set counting*: hashing items into corresponding buckets to reduce the number of item-sets examined.
- *Transaction Reduction*: Reducing the number of transactions scanned in future iterations by marking or removing transactions which are currently below the threshold.
- *Partitioning*: Partitioning the transaction data into *n* overlapping transactions and finding the local frequency for each each partition. All the local frequencies for each item-set for all the partitions are then combined and the global frequency is found for each item.
- *Sampling:* Picking up a random sample from the data and then searching the frequent item-sets from the sample. A lower support threshold is used here for qualifying the item-sets and accuracy is traded-off for efficiency
- *Dynamic item-set counting:* Adding new item-sets only when all of their subsets are estimated to be frequent.

Process flow of the Apriori algorithm:



Apriori Algorithm implementation in R:
https://github.com/DataAstrologer/DataScience/blob/master/R/Machine%20Learning%20Algorithm%20Templates/Unsupervised%20Learning/Association/Apriori.Rmd

Apriori Algorithm implementation in Python:
https://github.com/DataAstrologer/DataScience/blob/master/Python/Machine%20Learning%20Algorithm%20Templates/Unsupervised/Association/Apriori.ipynb