**Introduction**

The concept of association rules, support and confidence already introduced in 1966 and 1989, it was popularized by Rakesh Agrawal, Tomasz Lmielinski and Arun Swami [1] in 1993 to detect the associations or the relationships and interesting patterns in buying behaviour of the customer from a transaction dataset at point of sale. [1] An interesting relationship can be beneficial to correlate the product that a customer generally buys together with specific product or on some specific days. This information helps in designing the promotion plan, pricing of product, areas needs the attention in order to increase the overall sales and somewhat to control the buying behaviour of the customer. The algorithm developed in view of the supermarket however, the applicability and simplicity of algorithms make it popular in most of the field of research and technology development. As per the association rule defined [1], there is a:

1. Set of transactions T={T1,T2,T3,………..,Tn}
2. Set of Items I= {I1,I2,I3,………………,In}

For instance, a person makes a transaction T2 at a time t, though there is no concept of time here. The time is just added to describe that the transactions; means the set of items a person purchased at a time in a bill. For simplicity, bill can be considered as the transaction T2 that contains the list of the items a person purchased. Now by this example, we can purchase only those items available in this store so the “I” define the domain of the items from which a person select the items to be purchased. [1]

Transaction with its basic definition is a complete transaction where the items are purchased at least greater than 2, to find the association between the items that people generally buy together.

Features of Association rules are [2]:

1. It finds the interesting and useful relationship, hidden pattern, useful structures in the data. It further find the probability of the items can be purchased together in the future and dependencies between the items.
2. Where dataset is large, association rules are best to find the frequent itemsets with simple computations, now the association rules are implemented in Python and R, so the computation can be performed in very less time while generating the frequent itemset tree and minimizing the risk of manual misinterpretations.

Applications of Association rules

1. Market Basket Analysis to check the frequent items and buying patterns in order to maximize the sale. [2] The promotion design, layouts in store, cross-selling is benefited with such type of pattern analysis by putting the matching items near to each other or by putting the items with most popular items. [2] [6]
2. It can be also used to find out the behavior of a person, likings and categorize those, for instance, if a person likely to watch the movie, he also likely to purchase the corns. [4] Thus interesting patterns and buying habits can be deducted easily. [7]
3. It can be further beneficial to correlate the web pages to a person that he generally visited the sites and pages so what advertisements can be displayed on those pages. [4] Online shopping like catalog can be designed based on such information. [7]
4. Mining the data to find the hidden information, association rule mining is best suited for non-numeric data analysis through the generation of rules. The items in databases are put as attributes or the dimensions and a transaction shows the record for those elements in a row. [5]

Two ways the association can be found out:

1. Generally, what a person purchase and on what days. The monthly report is the set of all the transactions a person did to purchase item from that particular Store and check it's buying behavior. [1]
2. Another is transaction wise to find the relationship between the different transactions on the basis of days, discount etc. Transaction in transaction database has unique ID and each item is a subset of domain of items “I” and unique ID too. [1]

As per the Agrawal’s rule:

X=>Y; Where X and Y are the set of items.

X is called the antecedent and Y is called the consequent. [1]

It means if a person purchase X item (left hand side), he also purchase Y (on right hand side) and bothe are the subset of I [1] i.e. X ⊆ I and Y ⊆ I. [1]

Here is the example dataset:

I={Mask, Eno, Sanitizer, Bread, Namkeen, Ice-cream, Face-wash, Curd, Cofee, Sampoo, Steamer}; Where I is set of all the items or can be called as domain of items.

Transactions T can be designed as:

T1= {Mask, Ice-cream, Bread, Shampoo, Face-wash}

T2= {Mask, Eno, Bread, Coffee}

T3= {Coffee,Namkeen, Curd, Face-wash}

T4= {Mask, Sanitizer, Coffee, Eno}

T5= {Mask, Sanitizer, Namkeen}

T6= {Bread, Curd, Eno, Sanitizer}

Transactions Ti={T1,T2,T3,T4,T5,T6}

Items Ij=(I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11}

In tabular form, the ITEMLIST is:

| Transaction ID | Items Purchased |
| --- | --- |
| T1 | Mask, Ice-cream, Bread, Shampoo, Face-wash |
| T2 | Mask, Eno, Bread, Coffee |
| T3 | Coffee, Namkeen, Curd, Face-wash |
| T4 | Mask, Sanitizer, Coffee, Eno |
| T5 | Mask, Sanitizer, Namkeen |
| T6 | Bread, Curd, Eno, Sanitizer |

To create a database from the ITEMLIST:

| Transaction ID | Mask | IceCream | Bread | Shampoo | Facewash | Eno | Coffee | Curd | Namkeen | Sanitizer |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| T2 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| T3 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| T4 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| T5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| T6 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

1. **Elements or the measures of Association Rule**
2. Support
3. Confidence
4. Lift
5. Conviction

**Support**

Support is used to count the frequency of item sets appear together in set of transactions T [1] also called Support count (σ). Frequent items are those items that have support greater than minimum support. [2] Support is calculated in percentage of all the transactions to how the items are collectively appears. [2]

Support (X) = i.e. ; [1][2]

Where X is the item(s) purchased in T.

Or Support (X) = [5]

For example if X= {Mask, Sanitizer}, then

Support(X) = i.e.

= 0.33 or 33 % is the support that means 33% approximately the items appear together.

**Confidence**

Confidence is the frequency that a given rule X=>Y found “True” for all transactions in T. [1] that means if a person purchase item X, it also purchase item Y at the same time. [1] Confidence is dependent on Support also. It shows how much we are confident about the rule that how many times a given rule found “true”. If there is an itemset X, then confidence is the frequency that Y is also purchased with X. [5] It is the ratio Support(X UY) and Support X. U is the union symbol that indicate that all the items in X and all the items in Y in such a way that X U Y= Y U X [3]

Confidence (X=>Y) = [1]

Here, means Support of X and Y brought together. In the bottom part of method “ always describe X is that item that is left hand side of the rule, i.e. X=>Y, X is on left hand side so is written, otherwise if Y=>X is the rule, then we have to write as Y is on the left hand side, to find the confidence.   
In our example, let’s start with single items in X and Y.

Support (XUY) = = .333 or 33.33%

Support (X) = = .667 or 66.7%

Support (Y) = = .5 or 50%

Now, find the confidence after computing Support:

Confidence (Mask(X) => Sanitizer(Y)) = [1]

= = .499 or 49.9%

It means approx. 50 % of the cases when item X is purchased, Y is also purchase, now check the inverse relationship:

Confidence (Sanitizer(Y) => Mask(X)) = [1]

= = .666 or 66.6%

That shows that when item Y is purchased, around 67 percent of the cases, X is also purchased, it shows the strong association between Y (Sanitizer) and X (Mask).

If X contains 2 items:

If X= {Mask, Sanitizer} and

Y= {Eno}

Then the rule is:

{Mask, Sanitizer}=> Eno

Support (XUY) where X= Mask, Sanitizer and Y=Eno= = .1666 or 16.66%

Support (X) i.e. Mask & Sanitizer = = .3333 or 33.3%

Support (Y) i.e. Eno= = .5 or 50%

To find the confidence:

Confidence (X => Y) = [1]

= = .499 or 49.9%

That means when item X is purchased (Mask & Sanitizer), around 50% of the cases, item Y Eno) is also purchased.

Confidence (Y=>X) = [1]

= = .3332 or 33.32%

When we take the inverse rule Y=>X, it is found that only 33.3% approximately happens when item Y is purchased then item X is also purchased.

**Lift**

Lift is used to check whether the items are dependent or independent of each other [2].

Lift= [1]

= = =1, where rule is {Mask, Sanitizer} => Eno; having 2 items in X

Lift= [1]

= = =1, where rule is Mask => Sanitizer for single item in X.

3 categories of lift [1]:

1. If lift ==1 then there is chance that X and Y items are independent items and not related to each other, no such association is there and therefore, not suitable to draw a rule for such independent items, [1] [4] they just appear together. [2]
2. If lift >1 then existence of X and Y items are dependent and appearance of one can affect the appearance of other, so rules for such items can be beneficial in making decisions [1] as items appears more than expected. [2]
3. If lift <1 then the items X and Y are appear as less than expected and therefore, they are negatively correlated and substitute to each other that means if one is more, other is less or if one is present other one is absent. This type of information is also beneficial for the promotions and point of sale items.

In our example, we take two rules: first one is “{Mask, Sanitizer}=> Eno” and second one is “Mask => Sanitizer”; both have the lift value equal to 1 that means these items X and Y are independent [1] and there is no such relation that purchasing of one of these items (for example Mask or sanitizer) will affect the purchasing of other item(s) (for instance Eno).

**Conviction**

Conviction (Y=>X) = [1]

= = = .998 or 99.8%

Conviction defines the frequency where X appears without Y. it compares the probability of X without Y with the actual frequency of X without Y. Its range is between 0 and ∞ [8] the high conviction shows that the consequent Y is highly dependent on antecedent X. [8] Here, it is 99.8% for the rule “{Mask, Sanitizer} => Eno”. Like the lift if conviction is equal to 1, items are independent. [8]

1. **Apriori Algorithm**

Apriori Algorithm is used to generate the rules on the basis of frequent items in a given set of transaction in a database. [5] It used the breadth-first search and hash tree to compute the frequency of itemsets purchased together. Let’s create a database D from the ITEMLIST.

**Algorithm is in ppt( read carefully: important)**

Issues in Apriori Algorithm:

1. Too lengthy computations in finding the frequent itemsets.
2. If the size of database is too large, computations become very complex.

To short out the issues, sampling method can be implemented or to reduce the transactions and the complexity, first partition the data then apply apriori on each partition this way no of nodes and links can be reduced.