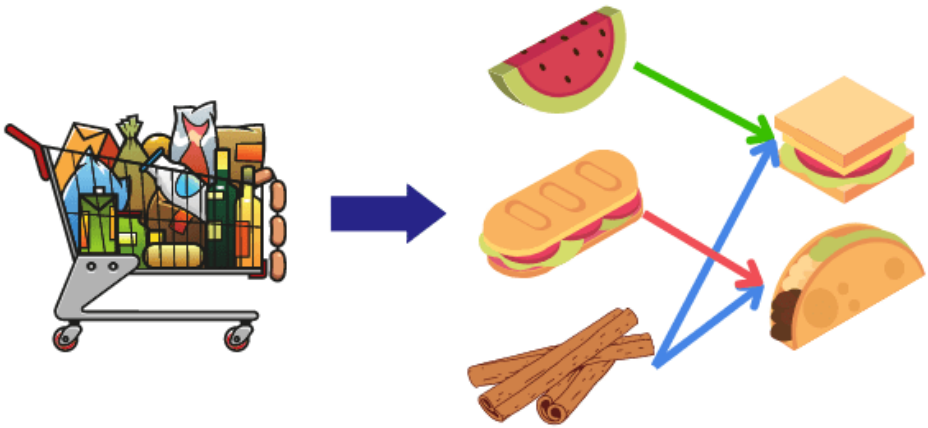
GJU
الجامعة الألمانية الأردنية
German Jordanian University

SEEIT

Association Rule

Association Rule Learning



"93% of people who purchased item A also purchased item B"

Rana Husni

GJU
الجامعة الألمانية الأردنية
German Jordanian University

SEEIT

Association rule

Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently a itemset occurs in a transaction. A typical example is a Market Based Analysis. Market Based Analysis is one of the key techniques used by large relations to show associations between items.It allows retailers to identify relationships between the items that people buy together frequently. Given a set of transactions, we can find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.

Rana Husni

Association Rule Discovery

- **Given a set of records each of which contain some number of items from a given collection;**
  - **Produce dependency rules which will predict occurrence of an item based on occurrences of other items.**
- Frequent patterns (or frequent itemsets)
  - What items are frequently purchased together in a store?
- A typical association rule

Bread-> Peanut Butter [10%,50%] (support ,confidence)

Support: This measures how often the rule is applicable to the dataset. For instance, if 100 out of 1000 transactions contain both bread and peanut butter, the support for the rule (Bread => Peanut Butter) is 0.1 or 10%.

Confidence: This measures how often items in the consequent appear in transactions that contain the antecedent. If 100 out of 200 transactions that contain bread also contain peanut butter, the confidence for the rule is 0.5 or 50%.

- How to mine such patterns and rules efficiently in large datasets?
- How to use such patterns for classification, clustering, and other applications?

SEEIT

Association Rules find all sets of items (itemsets) that have support greater than the minimum support and then using the large itemsets to generate the desired rules that have confidence greater than the minimum confidence. The lift of a rule is the ratio of the observed support to that expected if X and Y were independent. A typical and widely used example of association rules application is market basket analysis.

$$Rule: X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

Rule	Support	Confidence	Lift
$A \Rightarrow D$	2/5	2/3	10/9
$C \Rightarrow A$	2/5	2/4	5/6
$A \Rightarrow C$	2/5	2/3	5/6
$B \& C \Rightarrow D$	1/5	1/3	5/9

For example, if a customer buys bread, he most likely can also buy butter, eggs, or milk, so these products are stored within a shelf or mostly nearby. Consider the below diagram:

Customer 1

Customer 2

Customer 3

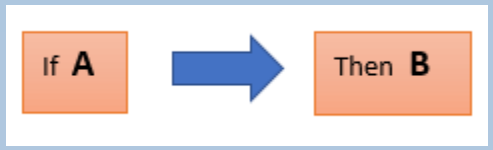
Customer n

Association rule learning can be divided into three types of algorithms:

- Apriori
- Eclat
- F-P Growth Algorithm

How does Association Rule Learning work?

Association rule learning works on the concept of If and Else Statement, such as if A then B.



Here the If element is called antecedent, and then statement is called as Consequent. These types of relationships where we can find out some association or relation between two items is known as single cardinality. It is all about creating rules, and if the number of items increases, then cardinality also increases accordingly. So, to measure the associations between thousands of data items, there are several metrics. These metrics are given below:

- Support
- Confidence
- Lift

SEEIT

Support

Support is the frequency of A or how frequently an item appears in the dataset. It is defined as the fraction of the transaction T that contains the itemset X. If there are X datasets, then for transactions T, it can be written as:

$$\text{Supp(X)}=\frac{Freq(X)}{T}$$

Confidence

Confidence indicates how often the rule has been found to be true. Or how often the items X and Y occur together in the dataset when the occurrence of X is already given. It is the ratio of the transaction that contains X and Y to the number of records that contain X.

$$\text{Confidence}=\frac{Freq(X,Y)}{Freq(X)}$$

Lift

It is the strength of any rule, which can be defined as below formula:

$$\text{Lift}=\frac{Supp(X,Y)}{Supp(X)\times Supp(Y)}$$

Lift

It is the ratio of the observed support measure and expected support if X and Y are independent of each other. It has three possible values:

If Lift= 1: The probability of occurrence of antecedent and consequent is independent of each other.
Lift>1: It determines the degree to which the two itemsets are dependent to each other.
Lift<1: It tells us that one item is a substitute for other items, which means one item has a negative effect on another.

Consider the following transactions in a small retail store:

Transaction ID	Items Purchased
1	Bread, Milk
2	Bread, Eggs, Butter, Eggs
3	Milk, Eggs, Butter, Cola
4	Bread, Milk, Eggs, Butter
5	Bread, Milk, Cola

1. Support

Support is the frequency of occurrence of an itemset in the dataset.

- Support of {Bread}: Number of transactions containing Bread / Total number of transactions

Support($\{Bread\}$)=4/5=0.8

Support ($\{Milk\}$)=4/5=0.8

Support of {Bread, Milk}: Number of transactions containing both Bread and Milk / Total number of transactions

Support($\{Bread, Milk\}$)=3/5=0.6

Confidence

Confidence is a measure of the reliability of the inference made by a rule.

- Confidence of {Bread} → {Milk}: Support of {Bread, Milk} / Support of {Bread} =0.6/0.8=0.75
- Confidence of {Milk} → {Bread}: Support of {Bread, Milk} / Support of {Milk})=0.6/0.8=0.75

Lift

Lift is a measure of how much more often the items in the rule appear together than if they were statistically independent.

- Lift of {Bread} → {Milk}:
  Confidence of {Bread} → {Milk} / Support of {Milk}
  =0.75/0.8=0.9375

- Lift of {Milk} → {Bread}: Confidence of {Milk} → {Bread} / Support of {Bread}
  =0.75/0.8=0.9375

Summary

- Support of {Bread, Milk} = 0.6
- Confidence of {Bread} → {Milk} = 0.75
- Lift of {Bread} → {Milk} = 0.9375

How does the Apriori Algorithm work in Data Mining?

Consider a Big Bazar scenario where the product set is P = {Rice, Pulse, Oil, Milk, Apple}. The database comprises six transactions where 1 represents the presence of the product and 0 represents the absence of the product.

Transaction ID	Rice	Pulse	Oil	Milk	Apple
t1	1	1	1	0	0
t2	0	1	1	1	0
t3	0	0	0	1	1
t4	1	1	0	1	0
t5	1	1	1	0	1
t6	1	1	1	1	1

The Apriori Algorithm makes the given assumptions

•Fix a threshold support level. In our case, we have fixed it at 50 percent.

SEEIT

Step 1

Make a frequency table of all the products that appear in all the transactions. Now, short the frequency table to add only those products with a threshold support level of over 50 percent. We find the given frequency table.

Product	Frequency(Number of Transaction)
Rice(R)	4
Pulse(P)	5
Oil(O)	4
Milk(M)	4
Apple(A)	3

The above table indicated the products frequently bought by the customers.

Step 2

Create pairs of products such as RP, RO, RM, PO, PM, OM. You will get the given frequency table.

Product	Frequency(Number of Transaction)
RP	4
RO	3
RM	2
PO	4
PM	3
OM	2
RA	2
PA	2
OA	2
MA	2

Rana Husni

Step 3

Implementing the same threshold support of 50 percent and consider the products that are more than 50 percent. In our case, it is more than 3

Thus, we get RP, RO, PO, and PM

Step 4

Now, look for a set of three products that the customers buy together. We get the given combination.

RP and RO give RPO
PO and PM give POM

Step 5

Calculate the frequency of the two itemsets, and you will get the given frequency table.

Product	Frequency(Number of Transaction)
RPO	3
PMO	2

If you implement the threshold assumption, you can figure out that the customers' set of three products is RPO.

GJU
الجامعة الألمانية الأردنية
German Jordanian University

SEEIT

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
# Sample dataset
data = {
    'TransactionID': ['t1', 't2', 't3', 't4', 't5', 't6'],
    'Rice': [1, 0, 0, 1, 1, 1],
    'Pulse': [1, 1, 0, 1, 1, 1],
    'Oil': [1, 1, 0, 0, 1, 1],
    'Milk': [0, 1, 1, 1, 0, 1],
    'Apple': [0, 0, 1, 0, 1, 1]
}
# Create a DataFrame
df = pd.DataFrame(data)
df = df.drop(columns=['TransactionID'])
# Apply the Apriori algorithm
frequent_itemsets = apriori(df, min_support=0.5, use_colnames=True)
# Print the results
print("Frequent Itemsets:")
print(frequent_itemsets)
```

	support	itemsets
0	0.666667	(Rice)
1	0.833333	(Pulse)
2	0.666667	(Oil)
3	0.666667	(Milk)
4	0.500000	(Apple)
5	0.666667	(Rice, Pulse)
6	0.500000	(Rice, Oil)
7	0.666667	(Oil, Pulse)
8	0.500000	(Milk, Pulse)
9	0.500000	(Oil, Rice, Pulse)

Rana Husni