Association Rule and Naïve Bayes

Association Rule Mining:

Association rule mining is an unsupervised methodology for finding relationships and patterns among different variables in a dataset; it is particularly useful in assessing large datasets because of its eventual computational efficiency.

Because the algorithm is unsupervised, there is no need for it to be trained. While the method can be unleashed on data straight away, there is no objective way to measure performance aside from measuring qualitative usefulness.

The main goal is to construct a set of rules that imply some sort of structure in the data. Generally, these rules take the form of "if X, then Y".

Applications of Association Rule Mining:

1) Genetics: Evaluating frequently occurring DNA and protein sequences in the analysis of disease genomics.
2) Fraud: Identifying patterns of medical claims or user purchase that appear alongside fraudulent insurance and credit card usage.
3) Subscription: Assessing consumer behaviour prior to purchasing or dropping a subscription to premium service.
4) Market Basket Analysis: Showing the relationship between items that a customer purchases.

We will focus on the application to the market basket analysis framework; however, this methodology can be applied across all disciplines.

Market Basket Analysis:

Often association rule mining is synonymous with market basket analysis because of its clear application to the shopping framework.

- A typical rule might be: if someone buys peanut butter and jelly, then that person is likely to buy bread as well.

Retailers can use this information to inform decisions about their:

- Marketing Strategies: Target customers that buy peanut butter and Kelly with offer on bread to get them to spend more.
- Store Layout: place peanut butter, jelly and bread relatively close to one another among the shelves in the sale.

Dataset Complexities:

Sticking with the market basket framework, transactional data can be quite complex; the massive size of transactional datasets is the threat of market basket analysis.

- The number of transactions (rows in our dataset) is likely to be larger.
- The number of available items (columns in our datasets) is likely to be large.

The different combinations of items that can appear in a specific purchase increase exponentially with the number of items available for purchase.

- Given k different items in a store, there are (2^k -1) different possible combinations of those items appearing purchases.

How do we get around dealing with this complexity?

The A *prior* Belief:

Rather than evaluating each of the individual purchase combinations on by one,

The Apriori algorithm takes into account one big observation:

- In reality, many of the potential combinations of items rarely (if ever) manifest in the data.

By straightaway ignoring the rarer purchase combinations, it is possible to limit the scope of the problem to aa more manageable realm of exploration.

The Apriori algorithem employs a simple a priori belief as a guideline for reducing the association rule space. The notions boil down to:

- All subsets of a frequent purchse must also be frequent.
- All supersets of a infrequent purchase must also be infrequent.

An Overview of Notation:

A rule is typically written with set notation. A simple rule might look like the following:

{Xi, Xj} -> {Xk}

- {Xi, Xj} represents the left hand side or LHS.
- {Xk} represents the right hand side or RHS.

In the market basket framework, the statement made by this rule can be read as:

“if the user purchases Xi and Xj, they are likely to purchase Xk as well”.

Our previous example could be written in this notation as follows:

- {Peanut Butter, Jelly} -> {Bread}
- Interpretation: Should a transaction include the items peanut butter and jelly, we should expect to see bread in the transaction more often than not.

An Overview of Terminology:

How can we determine whether an association rule is deemed interesting? This is usually a function of looking at three different parameters that surround a particular rule:

- Support: the fraction of which each item appears within the dataset as a whole.
- Confidence: the likelihood that a constructed rule is correct given the items on the left hand side of the transaction.
- Lift: the ratio by which the confidence of a rule exceeds the expected outcome.

An Overview of Terminology: Support

The support of an item X is the fraction of which each item appears within the total number of transactions.

$$\text{Support}(X) = \text{Count}(X) / N$$

Generally speaking, it is desirable to identify rules that have a high support:

- These rules will be applicable to a large number of transactions and are generally more interesting and profitable to evaluate from a business standpoint.
- Rule with low support may appear by chance, and can be uninteresting or costlier to spend time evaluating.

<u>An Overview of Terminology: Support Example</u>

<u>Consider the following dataset of transactions:</u>

Transaction	Flowers	Get Well Card	Soda	Stuffed Animal	Balloons	Candy Bar
#1	1	1	1	0	0	0
#2	1	0	0	1	1	1
#3	1	1	0	0	0	1
#4	0	0	1	1	1	0
#5	1	1	1	0	0	0

The support of {Candy Bar} = 2/5 = 0.4.

- The combination {Candy Bar} appears in 40% of the transactions.

The support of {Flowers, Get Well Card} = 3/5 = 0.6

- The combination {Flowers, Get well card} appears in 60% of the transactions.

<u>An Overview of Terminology: Confidence</u>

- The confidence of a rule is the likelihood that a constructed rule is correct given the items on the left hand side of the transaction. In other words, it is the probability that the transaction also contains the items on the right hand side:

$$\text{Confidence } (X \rightarrow Y) = \text{Support}(X \cup Y) \,/\, \text{Support } (X)$$

Confidence measures the accuracy or reliability of interface made by the given rule. A higher level of confidence implies a higher likelihood that Y appears.

NB: Be careful with interpreting the confidence of an association rule. A high confidence merely suggests a strong association(co-occurrence) between items, not a causal relationship.

Consider of following dataset of transactions:

Transaction	Flowers	Get Well Card	Soda	Stuffed Animal	Balloons	Candy Bar
#1	1	1	1	0	0	0
#2	1	0	0	1	1	1
#3	1	1	0	0	0	1
#4	0	0	1	1	1	0
#5	1	1	1	0	0	0

The confidence of {Flowers} -> {Get Well Card} = 0.6/0.8 = 0.75

- The proportion of transactions where {Flowers} resulted in the presence of {Get Well Card} is 75%.

The confidence of {Get Well Card} -> {Flowers} = 0.6/0.6 = 1.00

- The proportion of transactions where {Get Well Card} resulted in the presence of {Flowers} is 100%.

An Overview of Terminolgy: Lift

The lift of a rule is the ratio by which the confidence of a rule exceeds the expected outcome:

Lift(X -> Y) = Confidence(X -> Y) / Support(Y) = Support(X U Y)/Support (X) Support(Y)

In other words, it is the ratio of the support of the items on the LHS of the rule co-occurring with the items on the RHS divided by the probability that the LHS and RHS co-occur of the two are independent.

When lift>1, the presence of X seems to have increased the probability of Y occurring in the transaction.

When lift<1, the presence of X seems to have decreased the probability of Y occurring in the transaction.

When lift = 1, X and Y are independent.

An Overview of Terminology: LIFT Example

Transaction	Flowers	Get Well Card	Soda	Stuffed Animal	Balloons	Candy Bar
#1	1	1	1	0	0	0
#2	1	0	0	1	1	1
#3	1	1	0	0	0	1
#4	0	0	1	1	1	0
#5	1	1	1	0	0	0

Using the support formula:

- The lift of {Flowers} -> {Get Well CARD} = 0.6/(0.8 * 0.6) = 1.25

Using the confidence formula:

- The lift of {Flowers} -> {Get Well CARD} = 0.6/(0.8 * 0.6) = 1.25
- The lift of {Get Well Card} -> {Flowers} = 1.0/(0.8) = 1.25

The items are found together 1.25 times more often than one would expect by chance.

The Apriori Algorithm

Recall that the a priori belief was the solution to combining through a massive dataset of transactions because it takes advantage of the notion that:

- All subsets of a frequent purchase must also be frequent.
- All subsets of an infrequent purchase must also be infrequent.

The Apriori algorithm that takes this frequency notion into account can be broken down into the following two steps in order to identify interesting rules:

- Identify all subsets of items in the dataset that meet a minimum support threshold.
- Create rules from these subsets that meet a minimum confidence threshold.

After defining the minimum support and confidence, the possible rules to be traversed in the overall combinations can be limited in an efficient manner.

The first phase of the Apriori algorithm takes place in multiple iterations; each successive iteration evaluates the support of larger and larger sets items:

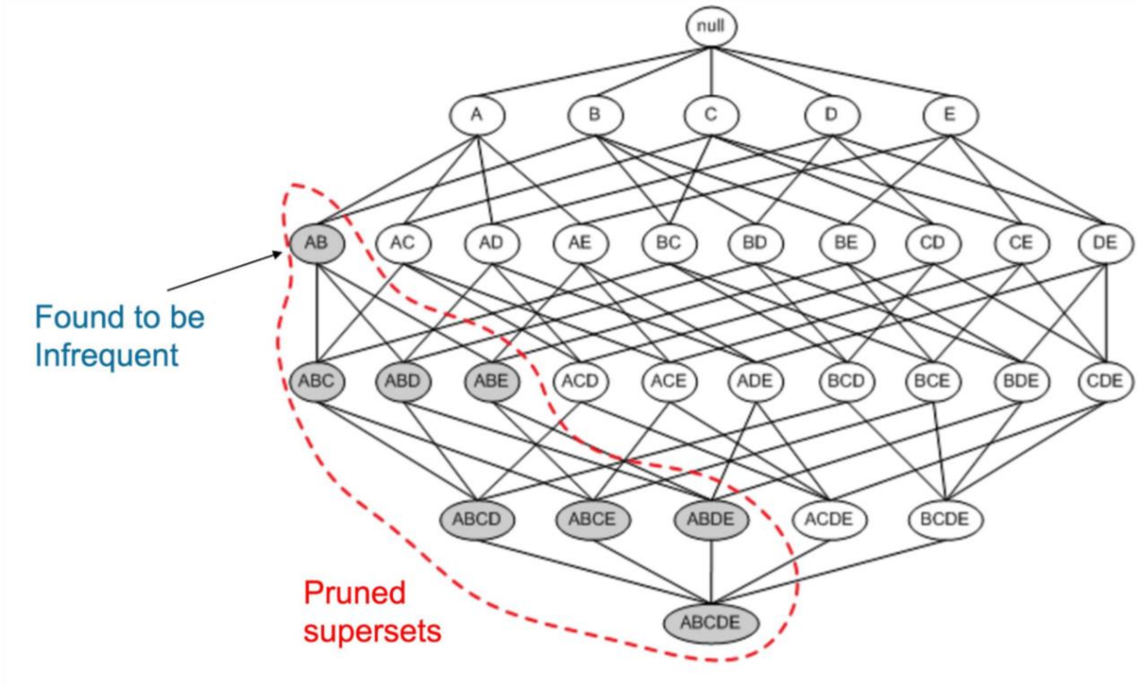
- Iteration 1 involves looking at the support of 1-item subsets.
- Iteration 2 involves looking at the support of 2-item subsets.
- Etc.; but do not we still end up looking at all the subsets?

A priori belief can help us eliminate many combinations. Suppose we have items A, B, C ,D and E:

- Iteration 1 finds the support {A}, {B}, {C}, {D}, and {E} to surpass the threshold.
- Now consider all 2-item subsets in iteration 2: {A,B}, {A,C},…,{D,E}

Iteration 2 finds the support of {A,B} to not surpass the threshold.

- Now considet all 3-item subsets that do not include {A,B}:{A,C,D},{A,C,E},.....,{C,D,E}.



Tuning Support & Confidence:

Given the unsupervised nature of the association rule problem, there is no steadfast guidline for deciding minimum support and confidence.

It is important to recognize that different business and realms will have extremely varied distribution of items and therefore completely varied types of transactions.

- Dependent on these distributions, different values of support and Confidence will be applicable.

High support and confidence will reduce the total number of rules generated, but tend to increase the individual rule prevalence and importance; try inspecting the results yielded by high parameter values and cutting back support and confidence if necessary.

Pros:

- Association rule mining is ideally suited for working with very large amounts of transactional data.
- The results are rules that are generally easy to understand and have a high amount of interpretability.
- The process is useful for data mining and uncovering unexpected knowledge within a dataset.

Cons:

- The outcome is usually not interesting when applied to smaller datasets.

- It is difficult to separate actual insights from common sense notions.
- The analyst might be compelled to draw spurious conclusions –remember that correlation doesn’t imply causation.