COMP 4106 A1 report
Ankush Varshneya – 100853074

I have decided to work on an implementation of the apriori algorithm my final project for my fourth year Artificial intelligence class. This algorithm is used on data from a transaction database to determine association rules highlight general trends in the database: this has applications in domains such as market basket analysis. So for example if we analyse all shopping baskets for a particular shop via the cashiers, then we can use apriori algorithm to learn association rules about what people buy what based other items in their shopping cart. This is beneficial for stores as they can strategically place items together so as to make the potential shopper want to buy a set of related items together rather than one item. For example if we find the rule that people that buy beer tend to buy diapers, then we can place beer and diaper together in the store so as to make people more inclined to buy both of the items as they are together.

Here is how it works:

Assume we have “I” as the set of all items offered by the store and “T” as the set of all transactions of items taken out by all customers. Further, we define a transaction “t” which is all the non-repetitive items in a shopping cart which has a unique identifier. So we have the set of all transactions, all the shopping carts of all customers at the store, as T = {t1, t2, t3, …, tm}; each transaction is also a subset of “I”. As mentioned previously we can mine for patterns from all these transactions such as what items are usually bough together. So this means we want to extract rules in the form of X -> Y, where X and Y are both subsets of “I” and X and Y are mutually exclusive, i.e. When items from X are present in a transaction, then also items from Y are present in it. The problem I am trying to solve is how we can derive such rules. This is called association rule mining, and we can use algorithms such as the apriori algorithm to find such patterns.

Given a set of transactions we build subsets X and Y from transaction t1 such that X OR Y = T, X AND Y=0. First we generate all X’s with length 1 and see if X -> Y is a valid pattern. So wee if something is valid pattern we must see its support and confidence level and support level, if they are over a threshold they are considered valid rules otherwise they are invalid rules. Conf(x-> y) = p(X is in a transaction)/p(X OR Y is in a transaction), and support(x->y) is p(X OR Y is in a transaction)/ # of transactions.