Association Rule Mining Explanation

Why was the technology implemented in this way?

Applying the algorithms to supermarkets for example, the scientists were able to discover links between different items purchased, called association rules, and ultimately use that information to predict the likelihood of different products being purchased together.

The most famous story about association rule mining is the “beer and diaper” Researchers discovered that customers who buy diapers also tend to buy beer. This classic example shows that there might be many interesting association rules hidden in our daily data.

Which lines of thought / considerations led to the solution ?

Association rule-mining is usually a data mining approach used to explore and interpret large transactional datasets to identify unique patterns and rules. During transactions, these patterns define fascinating relationships and interactions between different items. Moreover, association rule-mining is often referred to as market basket study, which is utilized to analyze habits in customer purchase.

Association rules help identify and forecast transactional behaviors based on information from training transactions utilizing beneficial properties. Using this approach, we can answer questions such as what items human beings tend to buy together, indicating frequent sets of goods. We can also associate or correlate products and items It consists of an antecedent and a consequent, both of which are a list of items. Note that implication here is co-occurrence and not causality. For a given rule, itemset is the list of all the items in the antecedent and the consequent.

Our solution apriori principle allows us to prune all the supersets of an itemset which does not satisfy the minimum threshold condition for support.

Which alternative approaches were pursued / would have existed and what are the reasons for not using them?

Association rules are often sought for very large datasets, and efficient algorithms are highly valued. The method we have described makes one pass through the dataset for each different size of item set. Sometimes the dataset is too large to read in to main memory and must be kept on disk, then it may be worth reducing the number of passes by checking item sets of two consecutive sizes at the same time. For example, once sets with two items have been generated, all sets of three items could be generated from them before going through the instance set to count the actual number of items in the sets. More three-item sets than necessary would be considered, but the number of passes through the entire dataset would be reduced.

In what limits can the solution be used (limitations) and Under what conditions cannot the solution be used and why?

There are several mining algorithms of association rules. One of the most popular algorithms is Apriori that is used to extract frequent itemsets from large database and getting the association rule for discovering the knowledge. Based on this algorithm, this paper indicates the limitation of the original Apriori algorithm of wasting time for scanning the whole database searching on the frequent itemsets, and presents an improvement on Apriori by reducing that wasted time depending on scanning only some transactions. The paper shows by experimental results with several groups of transactions, and with several values of minimum support that applied on the original Apriori and our implemented improved Apriori that our improved Apriori reduces the time consumed by 67.38% in comparison with the original Apriori, and makes the Apriori algorithm more efficient and less time consuming.

Apriori Algorithm can be slow. The main limitation is time required to hold a vast number of candidate sets with much frequent itemsets, low minimum support or large itemsets i.e. it is not an efficient approach for large number of datasets. For example, if there are 10^4 from frequent 1- itemsets, it need to generate more than 10^7 candidates into 2-length which in turn they will be tested and accumulate. Furthermore, to detect frequent pattern in size 100 i.e. v1, v2… v100, it have to generate 2^100 candidate itemsets that yield on costly and wasting of time of candidate generation. So, it will check for many sets from candidate itemsets, also it will scan database many times repeatedly for finding candidate itemsets. Apriori will be very low and inefficiency when memory capacity is limited with large number of transactions