“Fast Algorithms for Mining Association Rules”

By Rakesh Agarwal, Ramakrishnan Srikant

Presented by:
Muhammad Aurangzeb Ahmad
Nupur Bhatnagar

Motivation:

With the increased dissemination of bar code scanning technologies it was possible to accumulate vast amounts of Market-basket dataset containing millions of transactions. Thus the need arose to study the patterns of consumer consumption that could help in improving the marketing infrastructure and related disciplines like targeted marketing.

Association Rule Mining is a data mining technique which is well suited for mining Market-basket dataset. The research described in the current paper came out during the early days of data mining research and was also meant to demonstrate the feasibility of fast scalable data mining algorithms. Although a few algorithms for mining association rules existed at the time, the Apriori and Apriori TID algorithms greatly reduced the overhead costs associated with generating association rules.

Problem Statement:

Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction. Additionally the goal is to minimize computation time for generating the association rules used for prediction. The following is an illustrative example of market basket dataset.

TID	Transactions
1	books, stationary
2	Books,bags,grocery,utensils,
3	Stationary,bags,grocery,coke
4	books,stationary,bags,grocery
5	Books,stationary,bags,coke

Market based transaction set

For the above dataset the following rule holds good.
{books}->{stationary}
The rule can be read as, “customers who buy books also tend to buy stationary.”

Formal Definition:

Let $I = \{i_1, i_2, ..., i_m\}$ be a set of literals, called items. Let D be a set of transactions, where each transaction T is a set of items such that T subset I. The problem of mining association rules is to

generate all association rules that have support and confidence greater than the user-specified minimum support (called *minsup*) and minimum confidence (called *minconf*) respectively.

Major Contributions:

Two new algorithms for Association Rule Mining, Apriori and AprioriTID, along with a hybrid of the two algorithms, are described in the paper. The performance of these algorithms is shown to be from many times better for smaller datasets to many orders of magnitude better than the then current algorithms. The algorithms described in the paper represent a huge improvement over the state of the art in Association Rule Mining at the time. The new algorithms improve upon the existing algorithms by employing the following.

- Apriori and AprioriTID reduces the number of itemsets to be generated each pass by reducing the number of candidate itemsets.
- AprioriTID uses an encoding of the database for each pass instead of using the complete database. This greatly reduces the overhead cost associated with making passes over the dataset especially in later passes.
- Apriori Hybrid combines the features of the two aforementioned algorithms. It employs the Apriori algorithm for mining for earlier passes but switches to AprioriTID when the size of the encoding is expected to fit into memory.

Key Concepts:

A collection of one or more items in a market basket transactions. Consider a set of literals $I = \{i_1, i_2, ..., i_m\}$ then I is called itemset. An association rule is an implication of the form X->Y where X and Y are the itemsets. Support measures fraction of transactions that contain both X and Y. Given a rule X->Y and N being the total number of transactions then the support of an association rule is defined as.

Support = (X union Y)/ N

Confidence measures how often item in Y appear in transactions that contain X. Given the rule X->Y its confidence is defined as follows.

Confidence = X union Y / X

Itemsets with minimum support (*minsup*) and minimum confidence (*minconf*) are called as large itemsets, while others that do not cross minimum support and minimum confidence values are known as small itemsets. An itemset with k number of items is referred to as k-itemset. A set of itemsets which are generated from a seed of itemsets which were found to be large in the previous pass. Large itemsets for the next iteration are selected from the candidate itemsets if the support of the candidate itemsets is equal to or larger than *minsup* and *minconf*.

Given a set of transactions T, the goal of association rule mining is to find all rules having support $\geq$ *minsup* threshold, confidence $\geq$ *minconf* threshold. One of the key improvements in the Apriori, AprioriTID and AprioriHybrid algorithm is that it greatly reduces the number of candidate sets that have to be generated for generating association rules. This was one of the main shortcomings of the previous algorithms. A large number of candidate itemsets were generated in each pass that were later on discarded. Firstly this is an overhead in terms of

memory constraints and secondly it greatly increases the runtime for the algorithm since a lot of time is wasted for generating candidate itemsets that will not be used later.

Proposed Algorithms
APRIORI ALGORITHM:

Input
The market base transaction dataset.

Procedure

- ♦ The first pass of the algorithm counts item occurrences to determine large 1-itemsets.
- ♦ This process is repeat until no new large 1-itemsets are identified.
- ♦ (k+1) length candidate itemsets are generated from length k large itemsets.
- ♦ Candidate itemsets containing subsets of length k that are are not large are pruned.
- ♦ Support of each candidate itemset is counted by scanning the database.
- ♦ Eliminate candidate itemsets that are small.

Output
Itemsets that are "large" and qualify the min support and min confidence thresholds.

Candidate Set Generation Pruning
The biggest improvement in performance in the Apriori algorithm comes form reduction in candidate set generation. In the first pass all the 1arge 1-itemsets are generated. For all the later passes only those itemsets are considered as candidate itemsets which were found to be large in the previous pass. The main idea is that a subset of a large itsemset would itself be large. Thus two generate large itemsets of size k, all that is required is to join itemsets of size (k-1). In this way a large number of itemsets do not have to be considered for generating candidate itemsets as was the case with previous algorithms.

APRIORI-TID ALGORITHM:

Apriori TID has the same candidate generation function as Apriori. The interesting feature is that it does not use database for counting support after the first pass. An encoding of the candidate itemsets used in the previous pass is used. In later passes the size of encoding can become much smaller than the database,thus saving reading effort.

APRIORI-Hybrid Algorithm:

Apriori and AprioriTid use the same candidate generation procedure and therefore count the same itemsets Apriori examines every transaction in the database. On the other hand, rather than scanning the database, AprioriTid scans candidate itemsets used in the previous pass for obtaining support counts. Apriori Hybrid uses Apriori in the initial passes and switches to AprioriTid when it expects that the candidate itemsets at the end of the pass will be in memory.

Validation Methodology

The authors generated synthetic data sets involving transactions to evaluate the performance of algorithms. The transactions mimic the transactions in the "real" world. Transaction may contain more than one large itemsets. Transaction sizes are typically clustered around a mean and a few transactions have many items. To model the phenomenon that large itemsets often have common items, some fraction of items in subsequent itemsets are chosen from the previous itemset generated. Each itemset in a transaction has a weight associated with it, which corresponds to the probability that this itemset will be picked. It should be noted that the paper was written at a time when data mining was coming of age and not many datasets available for association rule mining were publically available. Although the authors do justify their use of synthetic datasets for validation, it should be noted that some later studies revealed [3] that the performance of association rule mining algorithms on even meticulously created synthetic datasets may not be the same as their performance on real world datasets. Such a risk is associated with using synthetic datasets in experimental validation. The Apriori Algorithm however does represent a big leap in terms of performance over previous algorithms like SETM or AIS. To be fair, the authors do point out to another paper where real world datasets were used, however these were limited in scope.

Assumptions

- The dataset is in the form of market basket dataset.
- The synthetic dataset is created through a detailed mediated process. The assumption is that the performance of the algorithm in the synthetic dataset is indicative of its performance on a real world dataset. The data might be too synthetic as to not give any valuable information about real world datasets and solving those problems via association rule mining.
- All the items in the data are in a lexicographical order.
- It is assumed that all the data is present in the same site or table and there are no cases which there would be a requirement to make joins. It is possible that the overhead cost of making joins could adversely effect the performance of the algorithms.
- It is assumed that candidate set generation is a necessary part of Association Rule Mining. Later algorithms like the FP Tree[2] proved that candidate set generation is not really necessary for Association Rule Mining.

Revision

The following list gives a list of possible revisions to the paper.

- As stated above, at least some real world datasets should be used to perform the experiments.
- Details regarding temporary storage should be added to the paper.
- Given that the size of databases has grown substantially over the last decade or so, the experiments for scaling factors should be extended to include large datasets.
- Modification in the representation structure:
    - Low support threshold can result in more large itemsets and increase the number of candidate itemsets.

- Apriori makes multiple passes, run time of algorithm may increase with number of transactions.

To solve the aforementioned problems more compact representations of the itemsets were suggested after the Apriori algorithm was formulated. Thus, we would consider these more compact representation of the itemsets if we have to rewrite the paper again.

References:

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases, pages 487--499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.
2. Jiawei Han, Jian Pei, Yiwen Yin: Mining Frequent Patterns without Candidate Generation. SIGMOD Conference 2000: 1-12
3. Zijian Zheng, Ron Kohavi, and Llew Mason, Real World Performance of Association Rule Algorithms, KDD 2001.