A Probability Analysis for Frequent Itemset Mining Algorithms

Nele Dexters
PhD student, ADReM Research Group, University of Antwerp, Belgium

Since the introduction of the Frequent Itemset Mining (FIM) problem, several different algorithms for solving it were proposed and experimentally analyzed. Our work focusses on the *theoretical* analysis of FIM. The aim is to give a detailed probabilistic study of the performance of FIM algorithms for different data distributions. It is joint work with Dirk Van Gucht and Paul Purdom from Indiana University in Bloomington, USA. The inspiration can be found in important related work [4].

The research considers in detail the probabilities that a set is a candidate and a success or a candidate and a failure, for a collection of well-known FIM algorithms. The Apriori Algorithm [2] is considered in detail; for AIS [1], Eclat [5], FP-growth [3] and the Fast Completion Apriori (FCA) Algorithm [2], the analysis is similar so only the main principles are sketched. The probabilistic analysis is done for different data distributions, covered by a general shopping model where all the shoppers are independent and each combination of items has its own probability of being purchased, so any correlation between items is possible.

We focus on algorithms that are candidate-based. An itemset I is called a *candidate* when its frequency status cannot be deduced by the algorithm based on previous knowledge from other itemsets, but has to be counted explicitly in the database. In practice, I is a candidate if certain associated *testsets* are already determined to be frequent. For all the algorithms, except for FCA, the testsets are itemsets that are obtained by omitting a single item from I; for FCA, the testsets are all those subsets of I whose size is equal to the level where the regular Apriori Algorithm was last used. The exact frequency status of I is determined by explicitly counting I's occurrence in the database. If I is frequent, it is called a *success*; otherwise, it is a *failure*. The *candidacy probability*, the probability that an itemset is a candidate, depends on the particular algorithm. On the contrary, all correct algorithms give the same probability that an itemset is frequent, the *success probability*; it is a property of the data, not of the algorithm. The probability that an itemset is a candidate but not frequent, is called the *failure probability*; it depends on both the problem instance and the algorithm and is particularly important because it is related to work that a better algorithm might hope to avoid.

Our research shows that for each algorithm, the candidacy probability of an itemset is usually determined almost entirely by the frequency probability of *a particular testset*, but which testset this is, depends on the algorithm. For both versions of the Apriori algorithm, this dominant testset is the one with the smallest probability. For the AIS algorithm, it is the testset with the highest probability. For Eclat-like algorithms, including FP-growth, it is a set that is at least as good as the testset with the second-highest probability and there is a tendency for it to be the one with the second-highest probability. We prove that the candidacy probability of an itemset is near 1 when the probability of this dominant testset is significantly above k/b (with k the user-defined support threshold from the FIM problem and b the number of baskets in the database), and it is near 0 when the probability is significantly below k/b. Similar results are found for the success and failure probabilities. We also show that the algorithms have similar performance on uniform data, whereas they can have hugely different performance on other types of data.

1. Rakesh Agrawal, Tomasz Imielinski and Arun Swami, *Mining Association Rules between Sets of Items in Large Databases*, in Proc. ACM SIGMOD Conference, Washington (1993), pp 207–216.
2. Rakesh Agrawal and Ramakrishnan Srikant, *Fast Algorithms for Mining Association Rules*, in Proc. of the 1994 Very Large Data Bases Conference (1994), pp 487–499.
3. Jiawei Han, Jian Pei, Yiwen Yin, *Mining Frequent Patterns without Candidate Generation*, in Proc. of the 2000 ACM SIGMOD Int. Conf. Management of Data, Dallas, TX (2000), pp 1–12.
4. Paul W. Purdom, Dirk Van Gucht, and Dennis P. Groth, *Average Case Performance of the Apriori Algorithm*, In SIAM J. Computing, **33**(5) (2004), pp 1223–1260.
5. Mohammed J. Zaki, *Scalable Algorithms for Association Mining*, In IEEE Transactions on Knowledge and Data Engineering, **12**(3) (2000), pp 372–390.