Scrutinizing Frequent Pattern Discovery Performance

Osmar R. Zaïane Mohammad El-Hajj Yi Li Stella Luk
Department of Computing Science, University of Alberta Edmonton, AB, Canada
{zaiane,mohammad,yli,stella}@cs.ualberta.ca

Abstract

Benchmarking technical solutions is as important as the solutions themselves. Yet many fields still lack any type of rigorous evaluation. Performance benchmarking has always been an important issue in databases and has played a significant role in the development, deployment and adoption of technologies.

To help assessing the myriad algorithms for frequent itemset mining, we built an open framework and testbed to analytically study the performance of different algorithms and their implementations, and contrast their achievements given different data characteristics, different conditions, and different types of patterns to discover and their constraints. This facilitates reporting consistent and reproducible performance results using known conditions.

1 Introduction

Mining for frequent itemsets is a canonical task, fundamental for many data mining applications and is an intrinsic part of many other data mining tasks. Mining for frequent itemsets is the major initial phase for discovering association rules. Associative classifiers rely on frequent itemsets. These frequent pattern are also used in some clustering algorithms. Finding frequent items is also an inherent part of many data analysis processes. Many frequent itemset mining algorithms have been reported in the last decade. From the famous *Apriori* algorithm[1], many extensions and sophisticated implementations have been suggested. Recently, new approaches relying on intricate data structures have been introduces claiming to outperform the *apriori*-based techniques. Some algorithms model transactions horizontally. Others transpose vertically the transactions. Some techniques traverse the pattern search space top-down, Others favour a bottom-up strategy. The puzzling reality is that most of these authors, when publishing their work claim to outperform, with their new method, the rest of the pack, supporting their claim with experiments carefully planned. Unfortunately, given some conditions and datasets, it is very difficult to know which algorithm is the most appropriate. A recent study [18] has shown that with real datasets, *Apriori*, the oldest algorithm for mining frequent itemsets, outperforms the newer approaches. Do we then need any of these new sophisticated approaches? An analysis done recently for a workshop on frequent itemset mining[7] demonstrates the importance of fine and clever implementations of algorithms, making the selection of an appropriate approach even more perplexing.

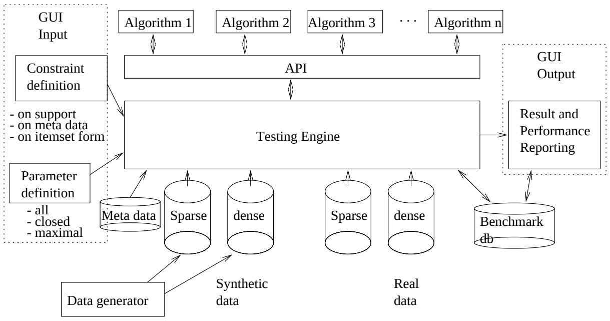
What has rarely been directly reported is that when dealing with extremely large datasets, discovering frequent itemsets is an impossibility for most algorithms. The problem is reduced to finding the set of frequent closed itemsets or the set of frequent maximal itemsets. A frequent itemset X is closed if and only if there is no X' such that $X \subseteq X'$ and the support of X equals to the support of X'. A frequent itemset X is said to be maximal if there is no frequent itemset X' such that $X \subseteq X'$. Frequent maximal patterns are a subset of frequent closed patterns, which are a subset of all frequent patterns. Finding only the closed item patterns reduces dramatically the size of the results set without loosing relevant information. From the closed itemsets one can derive all frequent itemsets and their counts. Directly discovering or enumerating closed itemsets can lead to huge time saving during the mining process. The set of maximal frequent itemsets is found, in general, to be orders of magnitude smaller in size than the set of closed itemsets, and the set of closed itemsets is found, in general, to be orders of magnitude smaller in size than the set of all frequent itemsets [5]. While we can derive the set of all frequent itemsets directly from the maximal patterns, their support cannot be obtained without counting. Again, many algorithms have been proposed to find these types of patterns. Nonetheless, the exact thorough comparison between proposed approaches is still lacking and researchers as well as developers are still perplexed when it comes to selecting an appropriate approach for mining a given dataset.

The state of affairs is even more complex since there is also the issue of expressing constraints on the patterns to discover. While some algorithms cannot treat these constraints and a post-pruning is necessary, others can handle during the mining process some types of constraints. Different types of constraints can be enforced on the patterns to discover: monotone and anti-monotone[12]. These constraints are expressed using aggregations on descriptors of items such as price, weight, height etc. of an item in a transaction. Yet again different claims are made [13, 4] but a rigorous comparison between constraint-based itemset mining algorithms has never been reported.

We propose a framework and a performance testbed to compare any frequent pattern mining algorithm given different datasets and dataset characteristics, and providing different parameters and constraints on the patterns to discover. The reporting obtained provides a consistent and precise analysis to discriminate among the approaches given specified conditions.

2 The benchmarking testbed

The testbed consists of a collection of real datasets, such as the UCI dataset collection [14], the worldcup98 weblog [16], the fimi collection [7], and synthetic datasets generated by the IBM QUEST data generator [11], as well as an interface (API) allowing the attachment of a variety of algorithms. Some representatives of this set of algorithms are: The Apriori implementation from [3], Closet+[15], ChARM[17], FPMAX[9], GenMAX[8], MAFIA[5], MaxMiner[2], FP-Growth[10], COFI+[6], dualminer[4], and other implementations shared in the fimi forum [7]. The testbed also includes a collection of pre-computed tests and benchmarks, and a graphical user interface to tune the available parameters and specify constraints on the patterns to discover.



3 Value to the community

The workshop on Frequent Itemset Mining Implementations held in conjunction with the IEEE International Conference on data Mining in 2003 and 2004 brought together eminent researchers working on various issues related to frequent itemset mining. The agreement of the attendees was that the research community needs reliable means to rigorously analyze algorithm performance and verify claims. Given the experimental algorithmic nature of frequent itemset mining, it is crucial that other researchers be able to independently verify the claims made by authors of new algorithms [7].

Another important issue raised is the issue of providing a common set of databases for testing frequent itemset mining. We intend to make our data collection as well as the benchmarking system as part of the fimi shared repository [7]. The significance of our performance benchmarking system is measured by the interest expressed by researchers in the field of frequent itemset mining and data mining as a whole. We expect the testbed to have a significant impact in the community and would constitute an initial stage towards defining a framework for benchmarking algorithms for other data mining tasks.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *VLDB*, 1994.

[2] R. J. Bayardo. Efficiently mining long patterns from databases. *ACM SIGMOD*, 1998.

[3] C. Borgelt. Apriori implementation. http://fuzzy.cs.uni-magdeburg.de/~borgelt/apriori/apriori.html.

[4] C. Bucila, J. Gehrke, D. Kifer, and W. White. Dualminer: A dual-pruning algorithm for itemsets with constraints. *ACM SIGKDD*, 2002.

[5] D. Burdick, M. Calimlim, and J. Gehrke. Mafia: A maximal frequent itemset algorithm for transactional databases. *IEEE ICDE*, 2001.

[6] M. El-Hajj and O. R. Zaïane. Inverted matrix: Efficient discovery of frequent items in large datasets in the context of interactive mining. *ACM SIGKDD*, 2003.

[7] B. Goethals and M. Zaki. Advances in frequent itemset mining implementations. *Workshop on Frequent Itemset Mining Implementations*, 2003.

[8] K. Gouda and M. J. Zaki. Efficiently mining maximal frequent itemsets. *IEEE ICDM*, 2001.

[9] G. Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. *FIMI'03, Workshop on Frequent Itemset Mining Implementations*, 2003.

[10] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *ACM-SIGMOD*, 2000.

[11] IBM_Almaden. Quest synthetic data generation code. http://www.almaden.ibm.com/cs/quest/syndata.html.

[12] R. Ng, L. Lakshmanan, J. Han, and T. Mah. Exploratory mining via constraint frequent set queries. *ACM SIGMOD*, 1999.

[13] J. Pie and J. Han. Can we push more constraints into frequent pattern mining? *ACM SIGKDD*, 2000.

[14] Uci knowledge discovery in databases archive. http://kdd.ics.uci.edu/.

[15] J. Wang, J. Han, and J. Pei. Closet+: Searching for the best strategies for mining frequent closed itemsets. *ACM SIGKDD*, 2003.

[16] 1998 world cup web site data. http://ita.ee.lbl.gov/html/contrib/WorldCup.html.

[17] M. Zaki and C.-J. Hsiao. ChARM: An efficient algorithm for closed itemset mining. *SIAM SDM*, 2002.

[18] Z. Zheng, R. Kohavi, and L. Mason. Real world performance of association rule algorithms. *ACM SIGKDD*, 2001.