Binary Journal of Data Mining & Networking 4 (2014) 21-24

http://www.arjournals.org/index.php/bjdmn/index

Research Article

ISSN: 2229 -7170

Survey performance Improvement FP-Tree Based Algorithms Analysis

Neelesh Shrivastava[1], Richa Khanna[2]

*Corresponding author:

Neelesh Shrivastava

[1]Asst.Professor CSE Department VITS, SATNA
[2]Mtech.Student Dept.of CSE (SS) VITS, SATNA

Abstract

Construction of a compact FP-tree ensures that subsequent mining can be performed with a rather compact data structure. For large databases, the research on improving the mining performance and precision is necessary; so many focuses of today on association rule mining are about new mining theories, algorithms and improvement to old methods. Association rules mining is a function of data mining research domain and arise many researchers interest to design a high efficient algorithm to mine association rules from transaction database. Generally the entire frequent item sets discovery from the database in the process of association rule mining shares of larger, these algorithms considered as efficient because of their compact structure and also for less generation of candidates item sets compare to Apriori .the price is also spending more. This paper introduces an improved aprior algorithm so called FP-growth algorithm.

Keywords : FP-tree; FP-growth algorithm;parallel projection; Partition projection.

Introduction

There are two methods for database projection:
Parallel projection
Partition projection.

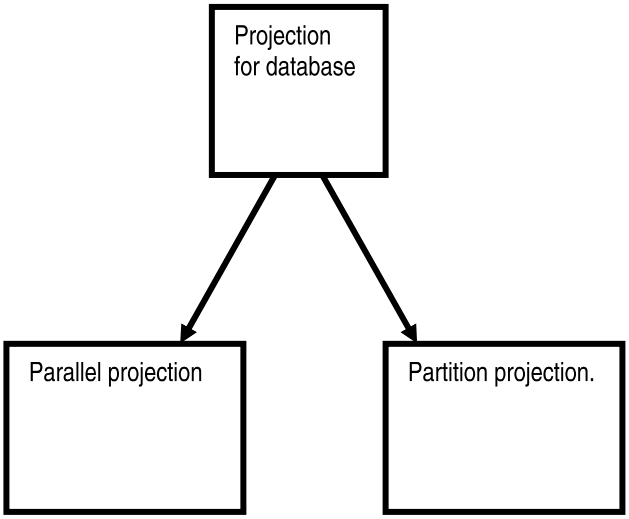

Figure 1 methods of data base projection

Parallel projection

is implemented as follows: Scan the database to be projected once, where the database could be either a transaction database or a -projected database. For each transaction *T* in the database, for each frequent item *ai* in *T,* project *T* to the *ai* - projected database based on the transaction projection rule, specified in the definition of projected database. Since a transaction is projected in parallel to all the projected databases in one scan, it is called *parallel projection*. The set of projected databases shown in of Example 2 demonstrates the result of parallel projection. This process is illustrated in figure 1.

Parallel projection facilitates parallel processing because all the projected databases are available for mining at the end of the scan, and these projected databases can be mined in parallel. Since each transaction in the database is projected to multiple projected databases, if a database contains many long transactions with multiple frequent items, the total size of the projected databases could be multiple times of the original one. Let each transaction contains on average l frequent items. A transaction is then projected to l - 1 projected database. The total size of the projected data from this transaction is 1 + 2 + + (l - 1) = l(l-1) 2 . This implies that the total size of the single item-projected databases is about l-1 2 times of that of the original database.

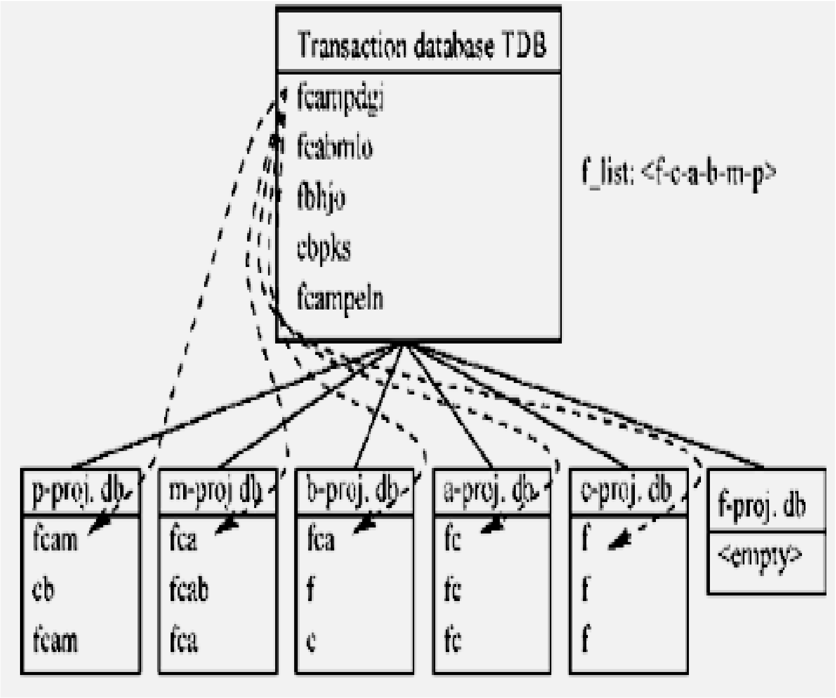

Figure 2 partition projections for data base

Partition projection

Partition projection is implemented as follows. When scanning the database (original or -projected) to be projected, a transaction *T* is *projected* to the *ai* -projected database only if *ai* is a frequent item in *T* and there is no any other item after *ai* in the *list of frequent items* appearing In the transaction.

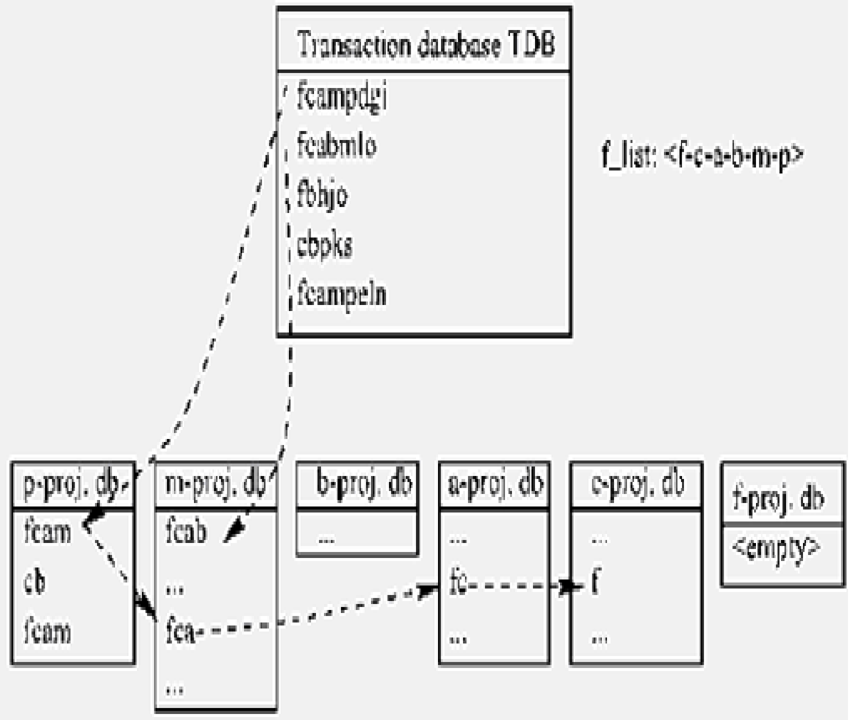

Figure 3 partition projections for data bas

Since a transaction is projected to only one projected database at the database scan, after the scan, the database is partitioned by projection into a set of projected databases, and hence it is called *partition projection*. The projected databases are mined in the reversed order of the *list of frequent items*. That is, the projected database of the least frequent item is mined first, and so on. Each time when a projected database is being processed, to ensure the remaining projected databases obtain the complete information, each transaction in it is projected to the *aj* -projected database, where *aj* is the item in the transaction such that there is no any other item after *aj* in the *list of frequent items* appearing in the transaction. The partition projection process for the database in

Previous Work

A research report on "Implication of Association Rules Employing FP Growth Algorithm for Knowledge Discovery" by A.H.M. [Sajedul Hoque, Sujit Kumar Mondal, Tassnim Manami Zaman, Dr. Paresh Chandra Barman & Dr. Md. Al Amin Bhuiyan], Dept. of Computer Science & Engineering, Northern University Bangladesh, Dhaka, Bangladesh 2011 IEEE.

Knowledge discovery is the nontrivial extraction of implicit, previously unknown and potentially useful information from data Knowledge Discovery in Database (KDD) refers to the process that retrieves knowledge from large database. Data mining is a part of KDD process which generates knowledge from preprocessed database. There are lots of data mining tasks such as association rule, regression, clustering and prediction. Among these tasks association rule mining is most prominent. Association rules are used to retrieve relationships among a set of attributes in a database. There are lots of algorithms to generate association rules from a database, such as Apriori, Frequent Pattern Growth (FP Growth), Éclat, Recursive Elimination etc. These produced association rules can be used to know the customer behavior of super market, to classify the employees of an organization to offer some opportunities and to generate predictions of an organization. [Bharat Gupta]

This paper focuses on FP-Growth algorithm to generate association rule from an employee database. The source database may contain Boolean or categorical or quantitative attributes. In order to generate association rule over quantitative attributes, the domain of quantitative attributes must be split into two or more intervals. This paper explores the generation of association rules on quantitative attributes by employing classical logic.

Many algorithms for mining association rules from transactions database have been proposed since Apriori algorithm was first presented. However, most algorithms were based on Apriori algorithm which generated and tested candidate item sets iteratively. This may scan database many times, so the computational cost is high. In order to overcome the disadvantages of Apriori algorithm and efficiently mine association rules without generating candidate item sets, a frequent pattern- tree (FP-Growth) structure is proposed in the FP-Growth was used to compress a database into a tree structure which shows a better performance than Apriori. However, FP-Growth consumes more memory and performs badly with long pattern data sets. In order to further improve FP-Growth algorithm, many authors developed some improved algorithms and obtained some promising results Due to Apriori algorithm and FP-Growth algorithm belong to batch mining. What is more, their minimum support is often predefined; it is very difficult to meet the applications of the real-world.

Conclusion

Distributed parallel computation technique or multi-CPU to solve this problem. But these methods apparently increase the costs for exchanging and combining control information, and the algorithm complexity is also greatly increased, cannot solve this problem efficiently. Even if adopting multi-CPU technique, raising the requirement of hardware, the performance improvement is still limited.

References

[1]. Savasere A, Omiecinski E, and Navathe S. An Efficient Algorithm for Mining Association Rules in Large Databases. Proceedings of the VLDB Conference. 1995.

[2]. Agrawal R, and Srikant R. Fast algorithms for mining association rules. VLDB, 487-499. 1994.

[3]. Han J, Pei J, and Yin Y. Mining Frequent Patterns without Candidate Generation. SIGMOD, 1-12. 2000.

[4]. Brin S, Motwani R, Ullman Jeffrey D, and Tsur Shalom. Dynamic itemset counting and implication rules for market basket data. SIGMOD. 1997.

[5]. Brin S, Motwani R, and Silverstein C. Beyond market baskets: Generalizing association rules to correlations. SIGMOD 26[2], 265-276. 1997.

[6]. Antonie M-L, and Zaïane O R. Text Document Categorization by Term Association , IEEE ICDM'2002, pp 19-26, Maebashi City, Japan, December 9 - 12, 2002

[7]. Han J, Pei J, Mortazavi-Asl B, Chen Q, Dayal U, and Hsu M-C. FreeSpan: Frequent pattern-projected sequential pattern mining. ACM SIGKDD, 2000.

[8]. Beil F, Ester M, Xu X. Frequent Term-Based Text Clustering, ACM SIGKDD, 2002

[9]. Orlando S, Palmerini P, and Perego R. Enhancing the Apriori Algorithm for Frequent Set Counting. Proceedings of 3rd International Conference on Data Warehousing and Knowledge Discovery. 2001.

[10]. Piatetsky-Shapiro G, Fayyad U, and Smith P. "From Data Mining to Knowledge Discovery: An Overview," in

Fayyad U, Piatetsky-Shapiro G, Smith P, and Uthurusamy R. (eds.) Advances in Knowledge Discovery and Data Mining AAAI/MIT Press, 1996, pp. 1-35.

[11]. Huang H, Wu X, and Relue R. Association Analysis with One Scan of Databases. Proceedings of the 2002 IEEE International Conference on Data Mining. 2002.

[12]. Wang K, Tang L, Han J, and Liu J. Top down FPGrowth for Association Rule Mining. Proc.Pacific-Asia Conference, PAKDD 2002, 334-340. 2002.

[13]. Zaki M J, and Hsiao C-J. CHARM: An EfficientAlgorithm for Closed Itemset Mining. SIAM International Conference on Data Mining. 2002.

[14]. Pei J, Han J, Nishio S, Tang S, and Yang D. H-Mine: Hyper-Structure Mining of Frequent Patterns in Large Databases. Proc.2001 Int.Conf.on Data Mining. 2001.