IMPLEMENTATION OF AN EFFICIENT ALGORITHM

Miss. K. Jayasudha [1], Dr.C.Chandrasekar [2]
[1] Lecturer / Department of computer Applications
[2] Prof / HOD, Department of computer Applications
[1, 2] K.S.Rangasamy College of Engineering,Thiruchengode
E-mail : jayasudhakaliannan@yahoo.com

Abstract

One of the main challenges in database mining is developing fast and efficient algorithms that can handle large volumes of data because most mining algorithms perform computation over the entire database and often the databases are very large. Many algorithms have been discussed in the literature for discovering association rules. One of the key features of all the previous algorithms is that they require multiple passes over the database. The database requires the complete reading for each pass resulting in a large number of disk I/O's. Apart from poor response time, this approach also places a huge burden on the I/O subsystem adversely affecting other users of the system. My objective is to reduce the I/O overhead and the CPU overhead and to improve the response time by using an efficient algorithm.

Keyword: Aprior , AprioriTid, item set , partition

1. Introduction

Database mining is motivated by decision support problems faced by most business organizations and is described as an important area of research. Discovering association rules between items over basket data was introduced here. Basket data typically consists of items bought by a customer along with the date of transaction, quantity, price, etc. Such data may be collected, for example, at supermarket checkout counters. Association rules identify the set of items that are most often purchased with another set of items. For example, an association rule may state that "95% of customers who bought items A and B also bought C and D". Association rules may be used for catalog design, store layout, product placement, target marketing, etc.

In this paper work, an algorithm called Partition is described that is fundamentally different from all the previous algorithms in that it reads the database at most two times to generate all significant association rules. Contrast with it , the previous algorithms, where the database is not only scanned multiple times but the number of scans cannot even be determined in advance. Extensive experiments have been performed and this algorithm was compared with one of the best previous algorithms. Experimental study shows that for computationally intensive cases, partition algorithm performs better than the previous algorithm in terms of both CPU and I/O overhead.

2. Related works

The problem of generating association rules was first introduced and an algorithm called AIS was proposed for mining all association rules. An algorithm called SETM was proposed to solve this problem using relational operations. Two new algorithms called Apriori and AprioriTid were proposed. These algorithms achieved significant improvements over the previous algorithms.

The algorithms vary mainly in (a) how the candidate item sets are generated; and (b) how the supports for the candidate item sets are counted. In the first one candidate item set's are generated on the fly during the pass over the database. For every transaction, candidate item sets are generated by extending the large item sets from previous pass with the items in the transaction such that, the new item sets are contained in that transaction. In the second one candidate item set are generated in a separate step using only the large item sets from the previous pass. It is performed by joining the set of large item sets with itself. The resulting candidate set is further pruned to eliminate any item set whose subset is not contained in the previous large item sets. This technique produces a much smaller candidate set than the former technique.

Supports for the candidate item sets are determined as follows. For each transaction, the set of all candidate item sets that are contained in that transaction are identified. The counts for these item sets are then incremented by one. Apriori and AprioriTid differ based on the data structures used for generating the supports for candidate item sets.

In Apriori, the candidate item sets are compared with the transactions to determine if they are contained in the transaction. A hash tree structure is used to restrict, the set of candidate item sets compared so that subset

testing is optimized. Bitmaps are used in place of transactions to make the testing fast. In AprioriTid. after every pass, an encoding of all the large item sets contained in a transaction is used in place of the transaction. In the next pass, candidate l item sets are tested for inclusion in a transaction by checking whether the large item sets used to generate the sets.

In Apriori, the subset testing is performed for every transaction in each pass. However, in AprioriTid, if a transaction does not contain any large item sets in the current pass, that transaction is not considered in subsequent passes. Consequently, in later passes, the size of the encoding can be much smaller than the actual database. A hybrid algorithm is also proposed which uses Apriori for initial passes and switches to AprioriTid for later passes.

3. Literature Survey

V. Pudi and J. Haritsa. On the optimality of association-rule mining algorithms. Technical Report TR-2001-01, DSL, Indian Institute of Science, 2001.
They present a new mining algorithm, called **ARMOR** (Association Rule Mining based on ORacle), whose structure is derived by making minimal changes to oracle,and is guaranteed to complete in two passes over the database. This is in marked contrast to the earlier approaches which designed new algorithms by trying to address the limitations of *previous* online algorithms. Although ARMOR is derived from Oracle, it shares the positive features of a variety of previous algorithms such as PARTITION, CARMA, AS-CPA, VIPER and DELTA. Our empirical study shows that ARMOR consistently performs within a *factor of two* of Oracle, over both real and synthetic databases.
Wang, C., Tjortjis, C., PRICES: An Efficient Algorithm for Mining Association Rules,Lecture Notes in Computer Science, Volume 3177, Jan 2004, Pages 352 – 358
They presented PRICES, an efficient algorithm for mining association rules. Their approach reduces large itemset generation time, known to be the most time-consuming step, by scanning the database only once and using logical operations in the process.
Hegland, M., Algorithms for Association Rules, Lecture Notes in Computer Science, Volume 2600, Jan 2003, Pages 226 - 234
He reviews the most well known algorithm for producing association rules - Apriori and discuss variants for distributed data, inclusion of constraints and data taxonomies. The review ends with an outlook on tools which have the potential to deal with long itemsets and considerably reduce the amount of (uninteresting) itemsets returned.
Toivonen, H. (1996), Sampling large databases for association rules, in `The VLDB Journal',pp. 134-145.
He presented an association rule mining algorithm using sampling. The approach can be divided into two phases. During phase 1 a sample of the database is obtained and all associations in the sample are found. These results are then validated against the entire database. To maximize the effectiveness of the overall approach, the author makes use of lowered minimum support on the sample
Han, J. and Pei, J. 2000. Mining frequent patterns by pattern-growth: methodology and implications. ACM SIGKDD Explorations Newsletter 2, 2, 14-20.
They introduces frequent pattern mining, is another milestone in the development of association rule mining, which breaks the main bottlenecks of the Apriori. The frequent item sets are generated with only two passes over the database and without any candidate generation process. FP-tree is an extended prefix-tree structure storing crucial, quantitative information about frequent patterns
Parthasarathy, S., Efficient Progressive Sampling for Association Rules. ICDM 2002:354-361.
Parthasarathy presented an efficient method to progressively sample for association rules. His approach relies on a novel measure of model accuracy (selfsimilarity of associations across progressive samples), the identification of a representative class of frequent itemsets that mimic (extremely accurately) the self-similarity values across the entire set of associations, and an efficient sampling methodology that hides the overhead of obtaining progressive samples by overlapping it with useful computation.
Chuang, K., Chen, M., Yang, W., Progressive Sampling for Association Rules Based on Sampling Error Estimation, Lecture Notes in Computer Science, Volume 3518, Jun 2005,Pages 505 - 515
They explore another progressive sampling algorithm, called Sampling Error Estimation (SEE), which aims to identify an appropriate sample size for mining association rules. SEE has two advantages. First, SEE is highly efficient because an appropriate sample size can be determined without the need of executing association rules. Second, the identified sample size of SEE is very accurate, meaning that association rules can be highly efficiently executed on a sample of this size to obtain a sufficiently accurate result.
Cheung, D., Han, J., Ng, V., Fu, A. and Fu, Y. (1996), A fast distributed algorithm for mining association rules, in `Proc. of 1996 Int'l. Conf. on Parallel and Distributed Information Systems', Miami Beach, Florida, pp. 31 - 44.
They presented an algorithm called FDM. FDM is a parallelization of Apriori to (shared nothing machines, each with its own partition of the database. At every level and on each machine, the database scan is performed independently on the local partition.
Baralis, E., Psaila, G., Designing templates for mining association rules. Journal of Intelligent Information Systems, 9(1):7-32, July 1997.

To address the problem of rule redundancy, four types of research on mining association rules have been performed. First, rules have been extracted based on user-defined templates or item constraints
Hilderman R. J., Hamilton H. J., Knowledge Discovery and Interest Measures, Kluwer Academic, Boston, 2002.
Secondly, researchers have developed interestingness measures to select only interesting rules
Cristofor, L., Simovici, D., Generating an informative cover for association rules. In Proc.of the IEEE International Conference on Data Mining, 2002.
Thirdly, researchers have proposed inference rules or inference systems to prune redundant rules and thus present smaller, and usually more understandable sets of association rules to the user [12].
Brin, S., Motwani, R. and Silverstein, C., "Beyond Market Baskets: Generalizing Association Rules to Correlations," Proc. ACM SIGMOD Conf., pp. 265-276, May 1997.
Finally,new frameworks for mining association rule have been proposed that find association rules with different formats or properties.
Jaroszewicz, S., Simovici, D., Pruning Redunda nt Association Rules Using Maximum Entropy Principle, Lecture Notes in Computer Science, Volume 2336, Jan 2002, pp 135-142
Jaroszewicz and Simovici [18] presented another solution to the problem using the Maximum Entropy approach. The problem of efficiency of Maximum Entropy computations is addressed by using closed form solutions for the most frequent cases. Analytical and experimental evaluation of their proposed technique indicates that it efficiently produces small sets of interesting association rules.
Wu, X., Zhang, C., Zhang, S.: Efficient Mining of Both Positive and Negative Association Rules, ACM Transactions on Information Systems, Vol. 22, No. 3, July 2004, Pages 381– 405.
They derived a new algorithm for generating both positive and negative association rules. They add on top of the support-confidence framework another measure called *mininterest* for a better pruning of the frequent itemsets generated. In [32] the authors use only negative associations of the type $X \Rightarrow \neg Y$ to substitute items in market basket analysis.

4. Proposed system

For improving the efficiency of apriori we have various variations
1. Hash based technique
2. Transaction reduction
3. Partitioning
4. Sampling
5. Dynamic item set counting

- A hash based technique can be used to reduce the size of the candidate k-item sets, using the buckets and hash table structure concepts along with the bucket counts.
- Transaction reduction technique reduces the number of transactions scanned in future iterations.
- Partitioning technique partitions the data to find the candidate item sets. It requires overall just two scans to get the frequent item sets.
- Sampling technique used for mining the subset of the given data. Random sample is picked and frequent item set is found. Overall it require only one scan.
- Dynamic item set counting is nothing but adding candidate item sets at different points during a scan. item sets partitioned into blocks and start points are marked for it. It requires fewer scans than Apriori.

Among the above five variations I have selected the partition algorithm for the work. It requires overall just two scans to get the frequent item sets.

4.1. Partition Algorithm

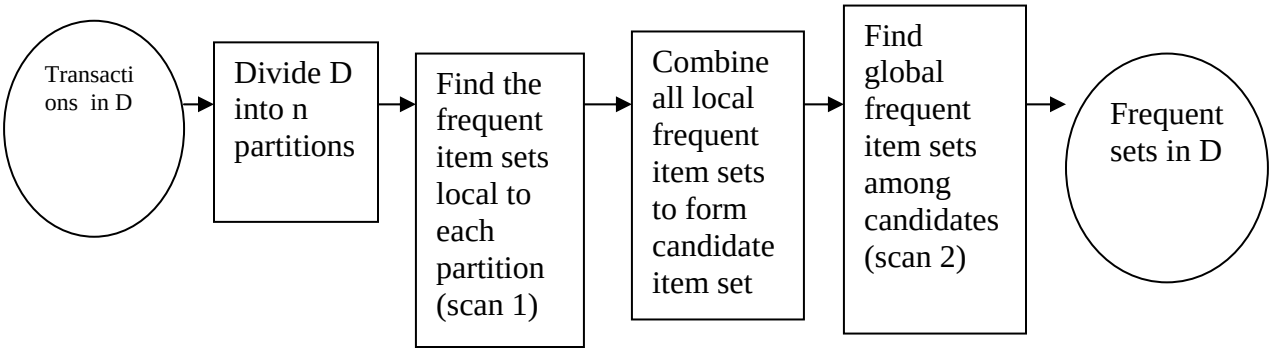
Here, an algorithm called Partition algorithm is used, that is fundamentally different from all the previous algorithms. In previous algorithms, it reads the database at most two times to generate all significant association rules. The database is not only scanned multiple times but the number of scans cannot even be determined in advance.

The idea behind Partition algorithm is as follows. The reason the database needs to be scanned multiple number of times is because the number of possible item sets to be tested for support is exponentially large if it must be done in a single scan of the database. However, suppose if it is given a small set, of potentially large item sets, say a few thousand item sets. Then the support for them can be tested in one scan of the database and the actual large item sets can be discovered. Clearly, this approach will work only if the given set contains all actual large item sets.

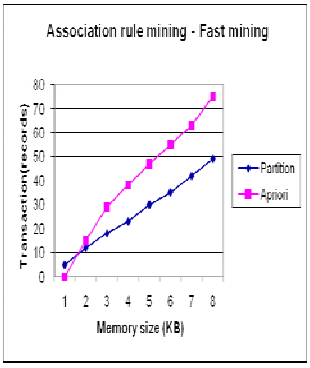
Partition algorithm accomplishes this in two scans of the database. In one scan it generates a set of all potentially large item sets by scanning the database once. This set is a superset of all large item sets, i.e., it, may contain false positives. But no false negatives are reported. During the second scan, counters for each of these item sets are set, up and their actual support is measured in one scan of the database.
The algorithm executes in two phases. In the first phase, the Partition algorithm logically divides the database into a number of non-overlapping partitions. The partitions are considered one at a time and all large item sets for that partition are generated. At the

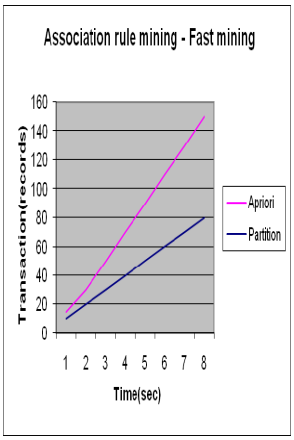
end of phase I, these large item sets are merged to generate a set of all potential large item sets. In phase II, the actual support for these itemsets is generated and the large item sets are identified. The partition sizes are chosen such that each partition can be accommodated in the main memory so that the partitions are read only once in each phase.

4.2. Methodology to be adopted



4.3 Expected output





5. Result and discussion

I/O overhead and CPU overhead will be reduced. Poor response time will be reduced. This feature may prove useful for many real-life database mining scenarios where the data is most often a centralized resource shared by many user groups, and may even have to support on-line transactions.

6. Conclusion and feature work

The proposed scheme of the paper described an algorithm which not only efficient but also fast for discovering association rules in large databases. Improvement in disk I/O is not achieved at the cost of CPU overhead. It is demonstrated with extensive experiments that the CPU overhead is actually less than the best existing algorithm for low minimum supports (i.e., cases which are computationally more expensive). I plan to extend this work by combining this work to any of the best algorithm and produce a hybrid approach. Also estimating the number of partitions is a difficult task. so I plan to do work on this

REFERENCES

[l] Agrawal .R, Imielinski .T, and Swami .A, Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207-216, Washington, DC, May 26-28 1993.

[2] Agrawal .R and Srikant .R, Fast algorithms for mining association rules in large databases. In Proceedings of the 20th International Conference on Very Large Data Bases, Santiago, Chile, August 29-September 1 1994.

[3] Han .J, Y. Cai, and Cercone N. Knowledge discovery in databases: an attribute-oriented approach. In Proceedings of the 18th International Conference on Very Large Data Bases, pages 547-559, Vancouver, Canada, 23-27, August 1992.

[4] Holsheimer M. and Siebes .A. Data mining: The search for knowledge in databases. Technical Report CS-R9406, CWI, Amsterdam, The Netherlands, 1993.

[5] Houtsma .M and Swami .A, Set-oriented mining of association rules. In Proceedings of the International Conference on Data Engineering, Taipei,
Taiwan, March 1995.
[6] Krishnamurthy .R and Imielinski .T Practitioner problems in need of database research. ACM SIGMOD Record, 20(3):76-78, September 1991.
[7] Piatetsky .G - Shapiro and Frawley .W. J., editors. Knowledge Discovery in Databases. MIT Press, 1991.
[8] Savasere .A, Omiecinski .E, and Navathe .S, An efficient algorithm for mining association rules in large databases. Technical Report GIT-CC-95-04, Georgia. Institute of Technology, Atlanta. GA 30332, January 1995.
[9] Silberschatz .A, Stonebraker .M, and Ullman J. Database systems: achievements and opportunities. Communications of the ACM, 34(10):110-
120, October 1991.
[10] Tsur .S Data debugging. IEEE Data Engineering Bulletin, 13(4):58-63, December 1990.
[11] Yi-Hung Wu, Chia - Ming Chiang, Arbee L.P.Chen. Hiding Sensitive Association Rules with Limited Side Effects 29-24 , IEEE Transactions on Knowledge and Data Engineering, Volume 19, Number 1, January 2007.
[12] Xiaoxin Yin, Jiawei Han, Jiong Yang, Philip S. Yu. Efficient Classification across Multiple Database Relations: A CrossMine Approach. 770-783 .IEEE Transactions on Knowledge and Data Engineering, Volume 18, Number 6, June 2006.
[13] Jianyong Wang, Jiawei Han, Ying Lu, Petre Tzvetkov: TFP: An Efficient Algorithm for Mining Top-K Frequent Closed Itemsets. 652-664 IEEE Transactions on Knowledge and Data EngineeringVolume 17, Number 5, May 2005
[14]Xiaoxin Yin, Jiawei Han, Jiong Yang, Philip S. Yu:
Efficient Classification across Multiple Database Relations: A CrossMine Approach. 770-783IEEE Transactions on Knowledge and Data Engineering Volume 18, Number 6, June 2006
[15] Michel L. Goldstein, Gary G. Yen: Using Evolutionary Algorithms for Defining the Sampling Policy of Complex N-Partite Networks. 762-773.IEEE Transactions on Knowledge and Data Engineering.Volume 17, Number 6, June 2005.
[16]Xiaoxin Yin, Jiawei Han, Jiong Yang, Philip S. Yu: Efficient Classification across Multiple Database Relations: A CrossMine Approach. 770-783IEEE Transactions on Knowledge and Data Engineering Volume 18, Number 6, June 2006