TRANSACTION DATABASE AND SUPPORT FOR ITEM SETS IN PARALLEL MINING

Bandi Banudeva Reddy
Research Scholar
Shri JJT University
Rajasthan

Dr. Jangala Sasi Kiran
Professor & Dean
Lords Institute of Engineering and Technology, Hyderabad, India

Dr. Prasadu Peddi
Associate Professor
Shri JJT University
Rajasthan

Abstract

Mining frequent itemsets in large databases is a widely used technique in Data Mining. Several sequential and parallel algorithms have been developed, although, when dealing with high data volumes, the execution of those algorithms takes more time and resources than expected. Frequent itemset mining is the essential step of data mining process. Further frequent itemset is a primary data obligatory for association rule mining. The Apriori and FP tree are conventional algorithms for mining frequent itemset and envisaging association rules based on it for know ledge discovery. The process of updating database continuously is known as incremental data mining. In real life, database updates recurrently where exactly conventional algorithms perform incompetently. If we could use the previous analysis is to incrementally mine the frequent itemset from the updated database, the mining process would become more efficient and cost of mining process would be minimized. In this research, we propose a novel incremental mining scheme with a parallel approach for discovering frequent itemset. It uses a data structure called IMBT. It is a Incremental Mining Binary Tree w hic h is used to record the itemset in an efficient way. Furthermore, our approach needs not to predetermine the minimum support threshold and scans the database only once.

Keywords: *Frequent Itemset, Incremental Data Mining, Parallel Data*

Introduction

Nowadays, many data mining techniques have emerged to extract useful knowledge from large amounts of data. Finding correlations between items, specifically frequent itemsets, is a widely used technique in data mining. The algorithms that have been developed in this area require powerful computational resources and a lot of time to solve the combinatorial explosion of itemsets that can be found in a dataset. The high computational resources required to process large databases can render the implementation of this kind of algorithms impractical. This is mainly due to the presence of thousands of different items or the use of a very low threshold of support.

Data mining is a process of discovering the pattern from the huge amount of data. There are many data mining techniques like clustering, classification and association rule. The most popular one is the association rule that is divided into two parts i) generating the frequent itemset ii) generating association rule from all itemsets. Frequent itemset mining (FIM) is the core problem in the association rue mining. Sequential FIM algorithm suffers from performance deterioration when it operated on huge amount of data on a single machine.to address this problem parallel FIM algorithms were proposed.

Data mining for big datasets is the practice of examining large pre-existing databases in order to create new information. There are several data mining methods available like clustering, classification and association rule mining. Among these one of the important method is the association rule mining which falls in two steps i) generating the frequent itemsets ii) generating association rules from all frequent itemsets. Frequent Itemset Mining (FIM) is a method for discovering interesting relations between variables in large databases. To find FIM, two well-known algorithms are Apriori and FP Growth (Frequent Pattern). Apriori uses the candidate generation approach and has to repeatedly scan the database. To reduce the time required for scanning of database and without generating candidate itemsets the next approach is FP Growth. Sequential FIM algorithm has the problem of performance deterioration when it work on vast amount of data on a single machine. To address this issue parallel FIM algorithms were proposed. Use of MapReduce programming model to solve the above problem which can handle large datasets through number of clusters. This distributed approach is combined with FIM to overcome the drawbacks of sequential FIM and to increase performance [3]. So the approach is called as Fidoop as it uses Hadoop- MapReduce model. In this approach FIUT (Frequent Items Ultrametric Tree) algorithm used to provide compressed storage, to avoid conditional pattern bases, to reduce I/O overhead

Literature Review

Brijendra Singh (2016) Data mining is an important field in Technology world. Association rules are a must and important step to discuss the data mining and inside findings of the relation between data variables of the database. In this Paper we have discussed an efficient parallel algorithm for association rules mining based on MapReduce framework. This can make performance of algorithm better and also reduce processing time.

Aqra I, Herawan T, Ghani NA, Akhunzada A, Ali A, et al. (2018) Designing an efficient association rule mining (ARM) algorithm for multilevel knowledge-based transactional databases that is appropriate for real-world deployments is of paramount concern. However, dynamic decision making that needs to modify the threshold either to minimize or maximize the output knowledge certainly necessitates the extant state-of-the-art algorithms to rescan the entire database. Subsequently, the process incurs heavy computation cost and is not feasible for real-time applications. The paper addresses efficiently the problem of threshold dynamic updation for a given purpose. The paper contributes by presenting a novel ARM approach that creates an intermediate itemset and applies a threshold to extract categorical frequent itemsets with diverse threshold values. Thus, improving the overall efficiency as we no longer needs to scan the whole database. After the entire itemset is built, we are able to obtain real support without the need of rebuilding the itemset (e.g. Itemset list is intersected to obtain the actual support). Moreover, the algorithm supports to extract many frequent itemsets according to a pre-determined minimum support with an independent purpose.

Huan-BinWang (2021) Aiming at the performance bottleneck of traditional Apriori algorithm when the data set is slightly large, this paper adopts the idea of parallelization and improves the Apriori algorithm based on MapReduce model. Firstly, the local frequent itemsets on each sub node in the cluster are calculated, then all the local frequent itemsets are merged into the global candidate itemsets, and finally, the frequent itemsets that meet the conditions are filtered according to the minimum support threshold. The advantage of the improved algorithm is that it only needs to scan the transaction database twice and calculate the frequent item set in parallel, which improves the efficiency of the algorithm.

Frequent Itemset Mining

Frequent itemsets are very important in data mining. It tries to find out interesting pattern from large database, such as association rules, data warehouse etc. The procedures of extracting the frequrnt itemsets are dependent on strong association rules. These Association rules were first presented by Rakesh Agrawal in 1993. Association rules are important in many applications such as product selling in supermarkets. The products which are purchased often by the customers can be identified as frequent itemsets and these are based on strong Association rules. For example, if a customer buys butter and bread together, then he could buy milk also. And hence this type of information is useful to take important decisions regarding product sale, pricing of items etc.

Methods for Parallelism

When it comes to the parallel computation techniques, data has to distribute among different nodes. Data Parallelism and Task Parallelism are two approaches using which data distribution task can be accomplished. Each one of these methods specifies their data distribution policy and based on it, node‗s work format. Data parallelism focuses on distributing the data across different parallel computing nodes. In data parallelism each processor performs the same task on different pieces of distributed data. For our research work data parallelism will be a good choice. As real life dataset used to be large, keeping it on each node will need more memory space. ‗Database Sharding‗ is the one method to achieve proper data distribution on different node. Database sharding breaks down large datasets into smaller chunks called ―shards‖ and spreads those across a number of different systems.

Parallel and Distributed Mining:

The Count Distribution (CD) algorithm is a simple data parallelization algorithm. The database D is positioned horizontally into D1, D2..Dn and distributed across n processors Pi (1≤ i≤ n). It uses sequential Apriori algorithms on each partition. The CD algorithm's main advantage is that it does not exchange data tuples between processors, it only exchange counts. In the first database scan, each processor generates its local candidate itemsets depending on the items present in its local partition. The algorithm obtains global support counts by exchanging local support count with all other processors. The algorithm communication overhead is O (|c|. n) at each phase, where |c| and n are the size of candidate itemsets and the number of data sets, respectively.

Researchers proposed FDM (Fast Distributed Mining) algorithms to mine association rules from distributed datasets partitioned among different sites[8] . At each site, FDMK find the local support counts and prunes locally in frequent itemsets. After completing local pruning, each site broadcasts messages containing all the remaining candidate sets to all other sites to collect their support counts. It then decides whether locally large itemsets are globally large and generates the candidate itemsets from those globally large itemsets.

Parallel distributed algorithm

Practical applications, the association rules to deal with the amount of data show exponential growth, which makes the problems focused on the algorithm of association rules mining efficiency and I/O load, even if used on a single processor optimized serial algorithm cannot meet the needs of mining properties, and the use of a multiprocessor system for parallel computing can improve the mining efficiency. Agrawal et al. [7] proposed three parallel algorithms, CD, DD and CaD.CD algorithm to generate candidate set all stored in each processor, using the Apriori algorithm to calculate the candidate set on a local database support count, then exchange the processor's local candidate set support count, makes each processor global support count to find the global frequent itemsets. Each processor synchronizes at the end of each cycle.CD algorithm achieves parallelization by dividing database, but it does not realize parallelization when generating candidate sets. When candidate sets are large, there is insufficient memory.DD algorithm divides candidate sets into different processors, which overcomes the problem of low memory utilization of CD algorithm, but this algorithm has a large traffic volume and must be completed on the processor with high communication speed. The CaD algorithm integrates CD and DD algorithms, and redistributes the database while allocating frequent 1-item sets, enabling each processor to independently complete the work of generating candidate sets. Product data management (PDM) algorithm [2] is by Park et al in DHP were studied. the optimum processing algorithm based on the improved parallel algorithm, the algorithm is similar to the CD algorithm, the candidate set generated.

by parallel and parallel to determine the composition of frequent itemsets, when o 2 - frequent itemsets, only exchange hash table to satisfy minimum support count itemsets. PDM algorithm not only inherits the advantages of DHP in reducing the number of candidate sets and transaction databases, but also realizes the parallelization of hash table composition, and the input robustness is quite good. Based on DIC minds, Cheung and others brought by [8] APM line and calculate method, APM using global pruning technology about reducing candidate 2 -itemsets, which is effective when the data distribution, and DIC algorithms require all the data in the database. For this reason, the APM algorithm needs to cluster the database into a homogeneous sub-database according to the number of processors, and then implement DIC algorithm on the sub-database by each processor until no new candidate set is generated. For the low efficiency of DD algorithm, IDD and HD algorithms are introduced [9]. In the IDD algorithm, the local database communicates with other operators through a comprehensive looped broadcast mode. Each processor has a sending buffer (sbuf) and a receiving buffer (rbuf) for asynchronous sending and receiving operations between adjacent nodes of each processor. Compared with DD algorithm, IDD algorithm only performs point-to-point communication once between adjacent processors, reducing communication times and eliminating network competition. To reduce the redundancy of candidate set generation, the IDD algorithm filters out those prefixes by detecting the prefixes of the item set. HD algorithm integrated the CD and IDD algorithm, the algorithm could be divided into equal to the size of the processor, using CD algorithm between groups, groups within the IDD algorithm, HD algorithm has some advantages in dealing with largescale database, and load balancing.

Conclusion

In this paper we proposed a parallel algorithm that is specially designed for environments which allow a high number of concurrent processes. This characteristic best suits a hardware environment and allows to use a task parallelism with fine granularity. This algorithm is more efficient to mining frequent itemsets for those databases, whose size is very large and have high data skewness. Any parallel algorithm working on database with high data skews could not achieve the advantages of parallel processing, because most globally large itemsets clustered on few processors. After a long period of research and development, association rule mining has become increasingly mature in the design and optimization of frequent pattern mining algorithms and is widely used in Internet, finance, biological information and other fields.

References

1. *Z. K. Baker and V. K. Prasanna. Efficient Hardware Data Mining with the Apriori Algorithm on FPGAs. In Proc. of the 13th Annual IEEE Symposium on Field Programmable Custom Computing Machines 2005 (FCCM '05), pages 3–12, 2005.*
2. *D. W. Cheung, J. Han, V. T. Ng, C. Y. Wong, ―Maintenance of discovered association rules in large databases: An incremental updating approach,‖ In The twelfth IEEE international conference on data engineering, pp. 106 –114, 1996.*
3. *Goyal, L.M., Beg, M.S., & Ahmad, T. (2018). An efficient framework for mining association rules in the distributed databases. The Computer Journal, 61(5), 645-657.*
4. *Sawant, V., & Shah, K. (2018). A System that Performs Data Distribution and Manages Frequent Itemsets Generation of Incremental Data in a Distributed Environment. In International Conference on Advances in Computing and Data Sciences, 104-113.*
5. *G. Lee, U. Yun, and K. H. Ryu, "Sliding window based weighted maximal frequent pattern mining over data streams," Expert Systems with Applications, vol. 41, pp. 694-708, 2014.*
6. *Kiran Chavan; PriyankaKulkarni; PoojaGhodekar; S.N.Patil,2015, "Frequent itemset mining for Big data", ISBN: 978-1-4673-7910-6, ICGCIoT,pp:1365-1368.*
7. *I.Pramudiono and M. Kitsuregawa, 2004 "FP-tax: Tree structure based generalized association rule mining," in Proc. 9th ACM SIGMOD Workshop Res. Issues Data Min. Knowl. Disc., Paris, France, PP.60–63.*
8. *Zhiguo Qu; John Keeney; Sebastian Robitzsch; Faisal Zaman; Xiaojun Wang, 2016, "Multilevel pattern mining architecture for automatic network monitoring in heterogeneous wireless communication networks", ISSN: 1673-5447 Volume 13, Issue7,pp:108-116.*
9. *Brijendra Singh (2016) An Efficient Parallel Association Rule Mining Algorithm based on Map Reduce Framework, International Journal of Engineering Research & Technology, ISSN: 2278-0181, Vol. 5 Issue 06.*
10. *Huan-BinWang (2021) Research on parallelization of Apriori algorithm in association rule mining, Procedia Computer Science, Volume 183, Pages 641-647, https://doi.org/10.1016/j.procs.2021.02.109.*
11. *Aqra I, Herawan T, Ghani NA, Akhunzada A, Ali A, et al. (2018) Correction: A novel association rule mining approach using TID intermediate itemset. PLOS ONE 13(5): e0196957. https://doi.org/10.1371/journal.pone.0196957*