International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 3, March - 2013

Parallel Association Rule Mining System for Continuous Data Streams

Suraj Acharya
V.E.S. Institute of Technology

Harsh Asher
V.E.S. Institute of Technology

Harshad Chavan
V.E.S. Institute of Technology

Prof. Gresha Bhatia
V.E.S. Institute of Technology

Abstract

In recent times, the data generation and storage modules in several organizations have seen a shift from the centralized approach to a parallel and distributed architecture. Also, the nature of data is changing from the static tables in databases and data warehouses to continuous stream data in data stores used for various technical and business applications. In such times, it is necessary to have data mining systems which accept data generated independently at disparate sites and perform mining at the local sites as well as at the global level at a centralized location. This paper describes a complete system to handle continuous stream data at parallel sites, the memory management involved in the storage of this data and an algorithm specifically designed to perform association rule mining in such an environment.

1. Introduction

Data mining is the process of extraction of non-trivial, implicit, previously unknown and potentially useful information or patterns from data in large databases. [1] It is also known as Knowledge Discovery in Databases (KDD). The following diagram shows the entire process of KDD.

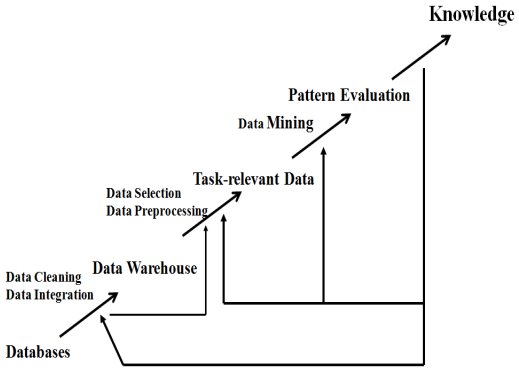

Fig. 1 KDD Process [1]

Association Rule Mining is the process of finding rules that will predict the occurrence of an item based on the occurrences of other items in the transaction from a given set of transactions. [1] An association rule is basically an implication of the form R:X→Y, where X and Y are disjoint itemsets:

X, Y ⊆ I and X ∩ Y ≠ Φ; and Y ≠ Φ [2]. Parallel mining, in the context of this paper means, to apply the process of KDD at different sites on different datasets, which are related to each other, at the same time. The system proposed in this paper then accumulates the local results at the different sites at a central server where mining is done on the local results to obtain global association rules for the system.

2. Algorithm

The proposed algorithm uses a Bitmap based mining technique[3] as a base for mining of association rules. The algorithm can be broadly divided into two phases as explained below:

Phase 1: Local Mining

During this phase the transaction databases at different nodes are mined locally in search of local association rules.

Step 1: Generation of Array of Bitmaps

The transactional database at a particular node is converted into an array of bitmaps by using the following logic:

1. A two dimensional array, in which each row represents an entry or a transaction from the database and each column represents an item, is created at the node.
2. Value of (i,j)th element of array is set to 1 if transaction i contains an item j.

Each row in the array is called a "Bitmap" for Transaction i and each column in the array is called the "Vector" for the item j.

Step 2: Finding Frequent Itemsets

After the generation of an array of bitmaps, the frequent Itemsets are found using following logic:

1. Set k=1.
2. For each k - itemset, its vector is scanned and weight of the k - itemset is found.

   weight(itemset j) = ∑ (all elements in vector j).

IJERT

3. If weight(itemset j)< minimum support, the itemset j is considered to be non-frequent and is discarded.
4. If weight(itemset j)>= minimum support, the k -itemsets are combined to form k+1–itemsets by logical AND operation on their vectors. The result of the operation is stored as a vector for corresponding k+1-itemset.
5. Set k = k+1.
6. Steps 1 to 5 are repeated until no more frequent k+1-itemsets can be found.

Step 3: Generation of Association Rules

The association rules are generated based on the value of "minimum confidence" provided.[5] For the purpose of global mining and considering the issue of continuous data streams in which temporal relationships between data are also equally important, the association rules are recorded with the timestamps or system clock times for which they are found.

Confidence for an association rule A -> B (where A and B are itemsets) is defined as

Confidence =
[(Frequency of occurrence of A & B together)/(frequency of occurrence of A)]*100
The confidence value is calculated for each association rule and the association rules with confidence value<"minimum confidence" are discarded.

Phase 2: Integration of Association Rules

Association rules from each of the nodes are supplied to the central repository for integration. The central processor then integrates the association rules based on the system clock times provided with them by the corresponding nodes, using following logic:

1. Let AR = set of association rules. Rules from each of the node are inserted into AR with system clock times.
2. The occurrences of items at different nodes and time of occurrence are determined using the association rules provided by the node.
3. Entire data is now converted to an array of bitmaps where each row represents a system clock time and each column represents the item with its node id appended with it.
4. The bitmap mining technique, as explained in local mining phase, is again employed with certain restrictions as follows:
   Only the matching items at different nodes are combined into a frequent itemset, i.e. if A1 and A2 are vectors of item A at node 1 and item A at node 2 respectively, then only A1,A2 can be combined, A1 is not combined with B1.
5. The new association rules are formed by combining items from different nodes by using frequent itemsets formed in 3. These rules are inserted into AR with system clock times.
6. The set of Association rules is optimized by using transitivity property as follows:
   If A -> B exists in AR where A and B are any items or itemsets, and there exists an association rule X -> Y where X is a subset of B then we conclude that A -> Y may be an association rule.
7. The number of system clock times for which both the rules are valid are determined. If this number is>=minimum support value, the rule A ->Y is added to AR.
8. Steps 6 and 7 are repeated until no new association rules can be added to AR.

AR is the final set of association rules.

Managing continuous data:

Since data is being input via a continuous data stream into the respective node, the local as well as integrated set of association rules gets modified constantly as the algorithm for local mining and integration of association rules gets iterated continuously.

For any new association rule X -> Y that gets transferred by a node:
1. Association rule X -> Y is inserted into AR.
2. The counts for all items are modified according to association rule and the algorithm for mining frequent itemsets is reiterated.

3. State Diagram

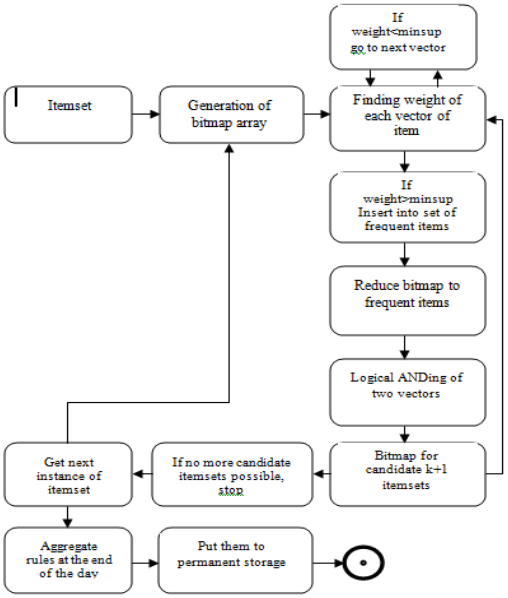

Fig. 2 State Diagram of the proposed system

IJERT

4. Memory Management

Since the data coming into the system is continuous in nature, special attention needs to be paid for handling the memory resources in the underlying database. Memory is finite and limited. Thus, proper handling and optimum use of the available memory resources is presented in this system. The two possible approaches are data-based approach and the task-based approach. In the proposed system, the data-based approach has been used. In the data-based approach, the idea is to examine only a subset of the whole dataset or to transform the data horizontally or vertically to an approximate smaller size of data representation. [4] Under the data-based solution, the system described in this paper uses the Synopsis Data Structure approach. Creating synopsis of data refers to the process of applying summarization techniques that are capable of summarizing the incoming data streams for further analysis. [4] In this system, the tables in the database continuously accept the incoming data streams and store in the individual entries up to a point where the threshold of the number of entries is reached. The algorithm will consider the current copy of the table at that moment. When the threshold is reached, the entries in the database table are subjected to the synopsis operation mentioned above and a single entry is entered in the warehouse for all the entries currently in the table. The table is then truncated and is opened for new entries to come in. This process repeats continuously. The algorithm also continuously checks for the updates in the tables and adds or modifies the association rules.

5. Analysis

Experiments were made to evaluate the performance of the proposed algorithm under different scenarios mainly varying in input parameters like number of transactions in the tables at different stations and average number of items in the item sets.

The programs were executed on 3 computer system as nodes and 1 computer system as a server. Fig.3 shows the variation in execution time of programs at different clients with input size i.e. number of transactions varying from 50 to 100.

Number of transactions	Association 1	Association 2	Association 3	Central Server	Minimum Support	Average Rules sent to server
50	132	95	123	41	75	6
	128	118	112	30	75	4
	131	123	114	31	75	5
75	198	276	260	67	50	3
	284	266	270	196	50	6
	261	243	231	11	75	1
100	325	494	698	24	75	1
	983	843	578	250	50	20
	593	421	687	78	50	10

Table 1 Execution time and Number of transactions

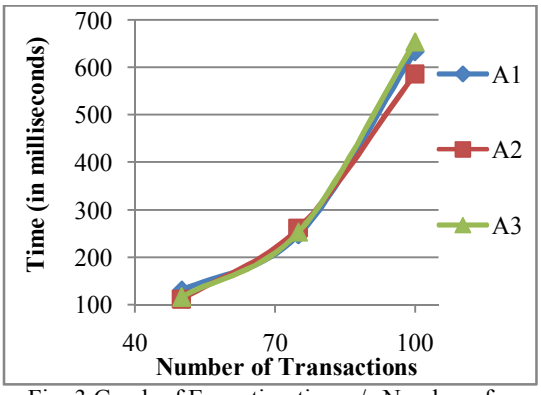

Fig. 3 Graph of Execution time v/s Number of transactions at the local sites

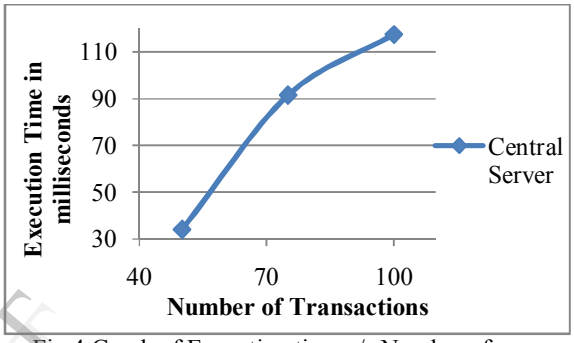

Fig.4 Graph of Execution time v/s Number of transactions at the Server

Number of item sets in 1 transaction	Number of transactions	Execution Time at the local sites
3-6	50	119.5556
	75	254.3333
	100	624.6667
6-8	50	362.5259
	75	1402.667

Table 2 Variation of execution time with respect to item sets in each transaction

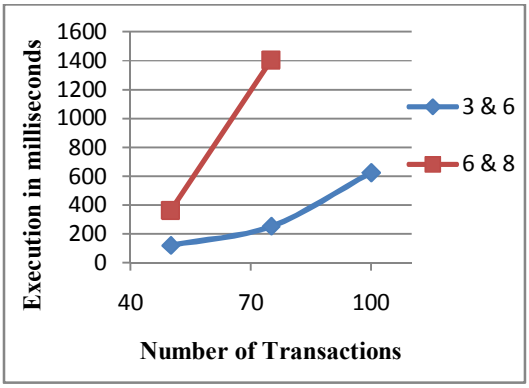

Fig.5 Execution time v/s Number of transactions (in comparison with item sets in each transaction)

IJERT

6. Limitations

1. Since the processing has been distributed and is carried out by the individual clients, the reliability and fault tolerance have to be maintained by the individual processes. Monitoring each process in a parallel system is difficult.
2. The devices are expected to work in collaboration with one another using the network and making use of bit maps to transmit the data. There might occur a discrepancy when due to network failure a node is unable to send the data and that will cause the entire system to become unstable since, the values from one node are unavailable.

7. Conclusion

The main aim of this algorithm is to produce optimized association rule mining at both the local level and at the global level. Some points taken into consideration are the network communication, the speed of communication and the accuracy of the association rule mining.

Various aspects affecting the processing speed were taken into consideration. On observation it was realised that using the bitmap method to derive the association rule helps the processor process the entries at a much faster rate. Also, since communication of the local association mining has to be shared across the network, and there is a high possibility of the machines on the network being heterogeneous, the bitmap levels these disadvantages and brings about uniformity in communication.

Since there is local association mining and global mining being carried out, load balancing is possible, since, the server is not completely overloaded and the client sites carry out the association in their own processors.

IJERT

6. References

[1] Jiawei Han and Micheline Kamber: Data Mining Concepts and Techniques, Second Edition

[2] Andreas Mueller: “Fast Sequential and Parallel Mining Algorithms: A Comparison”, CS-TR-3515, August 1995.

[3] Jianwei Li, Ying Liu, Wei-keng Liao and Alok Chaudhary: “Parallel Data Mining Algorithms For Association Rules and Clustering”.

[4] Mohamed Medhat Gaber, Arkady Zaslavsky and Shonali Krishnaswamy: “Mining Data Streams: A Review”, SIGMOD Record, Vol. 34, No. 2, June 2005.

[5] Rakesh Agrawal, John C. Shafer: “Parallel Mining of Association Rules”, IBM Almaden Research Center.