Reducing Data Skew with Round Robin Horizontal Partitioning of Data for Distributed Association Rule Mining of Large Data Set

Dipak V. Patil
dvpcomp@gmail.com
Department of Computer Engineering
G.E.S’s R.H. Sapat College of Engineering, Management Studies and Research, Nasik, India

Abstract—High growth in data size is observed due use of computer in all field. This data is not useful for decision making in business, unless is mined to extract interesting knowledge from it. For analyzing such data and extracting true knowledge from it, various data mining techniques are used. Association rule mining is one of them; it aims at finding associations or relations among data. As size of the data increase, knowledge discovery on this high volume data becomes slow, with conventional data mining technique, as it has to be done serially. The number of data records may make the learning process very slow. The solution to the problem is to speed-up the learning process with the help of parallel or distributed techniques. Through mining, interesting relations and patterns between variables of large database can be observed using the distributed mining algorithms. The performance in terms of time complexity data mining algorithm can be from O(N) to lower bound O(N/k) with parallel or distributed approach, where N = number of data instances and k = number of nodes in distributed system[1]. Partitioning and distribution of data on different nodes in distributed system may lead to data skew and intern a problem in computing support and confidence. This paper addresses the distributed association rule mining on large datasets and merging rules in single rule set. This system horizontally distributes large data set using round robin method and association rule mining using Apriori algorithm is performed with global support count at least s and confidence count at least c. Duplicate rules in the system create rule redundancy. Duplicate rules are found and redundancy is removed from rule set before final merger of the rules at central server. Data security issue in distributed mining has been handled by many researchers so it is not addressed here. The speed up is acquired with proposed method is significant along with utilization of available computing resources.

Keywords—Apriori Algorithm, Distributed Mining, Partitioning.

I. INTRODUCTION

Historical databases is are generally knowledge rich, and extracting this knowledge for enhancing the business is the most crucial aspect of business. Mining such data using association rule mining can lead to extraction of various relations and many new and hidden patterns in data, which can be useful for decision making purpose. Other applications of such knowledge can be stated as financial forecast, marketing strategies, customer retention and many more. The implicit function of Association Rule Mining (ARM) is finding the interesting associations or relations (usually known as patterns) among data. Association Rule mining focuses on finding patterns satisfying certain interestingness metric, which can be given as minimum support threshold and minimum confidence threshold. Any rule or pattern satisfying both these thresholds is considered as interesting association rule and delivered as outcome. Patterns with interestingness below the threshold are considered as not-interesting and can be removed [2].

The association rule mining is one of the most important and widely used techniques for mining data, but it consists of iterative scans of database, which can be costly for large databases. In today’s age, data is growing at a fast rate. Terabytes of data from various organizations, institutions and agencies are being generated continuously. People are eager to discover useful and actionable information from this data, but still it is a challenge to make Association Rule Mining (ARM) algorithms cope with this large amount of data. Thus it is beneficial to go for partitioning such large database and then performing distributed association rule mining on the partitioned datasets. On the other hand many organizations are also storing their data in distributed manner itself, either in horizontal or vertical partitioned format. For mining such data again distributed association rule mining is essential. Previous studies in this field focus on efficient mining of association rules. The most widely accepted algorithm for association rule mining that is Apriori algorithm. In this algorithm, downward closure property is used to find association rules. The most widely accepted algorithm for association rule mining that is Apriori algorithm. In this algorithm, downward closure property is used to find association rules. In this experimentation Apriori algorithm is used.

ASSOCIATION RULE MINING (APRIORI ALGORITHM)
The interestingness of patterns can be analyzed by using various interestingness metrics such as Support, Confidence and Lift.
Patterns with interestingness below a certain threshold might be considered as not-interesting and can be removed. The motivation for ARM came from the concept of analyzing supermarket data, which is transactional in nature, so as to examine customer behavior in terms of the purchased products. It is more generally known as Market Basket Analysis. In that case, Association rules describe which items are mostly purchased together and also how many times. For example, a rule for laptops, pen drive (80 percent) states that 80 percent of times when customers bought laptop, they also bought pen drive. Such rules can be useful for decisions making purpose, regarding product pricing, product marketing, store layout etc. ARM basically involves generation of frequent item sets, which is also used in correlations, and graph mining, constraint based frequent patterns mining, sequential patterns mining, and so on. Efficient algorithms for mining frequent item sets are required for this ARM task. The major challenge in frequent pattern mining is the large number of patterns generated on large data. As the threshold value decreases, an exponentially large number of item sets are generated. To mine association rules in market basket data is gaining attention vastly. It can be elaborated as follows: An association rule can be represented as $X \Rightarrow Y$, $X$ and $Y$ being itemsets. The meaning of this rule is quite simple, stating that whenever in a transaction $X$ occurs, there is a probability of occurrence of $Y$ in that transaction [2]. Association rule mining is one of the most important tasks of data mining to find patterns in data. Association rules can be briefly expressed in the form of $X \Rightarrow Y$, where $X$ and $Y$ are sets of items. Association rule mining stems from the analysis of market-basket data sets. The association rule mining problem can be formally described as follows: let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of literals, called items. Let $D$ be a set of transactions, where each transaction $T$ is a set of items such that $T \subseteq I$. A unique identifier, called TID is linked to each transaction. A transaction $T$ is said to contain $X$, a set of some items in $I$, if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = 0$. The rule $X \Rightarrow Y$ holds in the transaction set $D$ with confidence $c$ if $c$ transactions in $D$ that contain $X$ also contain $Y$. The rule $X \Rightarrow Y$ has support $s$ in $D$ if $s$ of the transactions in $D$ contains $X \cup Y$.

Apriori algorithm [2] is used for generating association rules.
1. Create $L_1$ = set of supported item sets of cardinality one
2. Set $k$ to 2
3. While ($L_{k-1}$ is not empty)
   4. \{ 
   5. Create $C_k$ from $L_{k-1}$ (Generate $C_k$, from $L_{k-1}$).
   6. Prune all the itemsets in $C_k$ that are not 
   7. Supported, to create $L_k$
   8. Increase $k$ by 1
   9. \}
10. The set of all supported item sets is $L_1 \cup L_2 \cup \ldots \cup L_k$

In the processing of Apriori algorithm multiple passes exist over the database. In the first pass, the algorithm counts number of times a particular item occurs in a given document. For determining the frequent 1-itemsets that is itemsets with 1 item. For a subsequent pass, consider that the pass $k$ consists of two phases. First in which the frequent itemsets $L_{k-1}$ (the set of all frequent $(k-1)$-item sets) are found in the $(k-1)$th pass are used to generate the candidate itemsets $C_k$. This function first joins $L_{k-1}$ with $L_{k-1}$, the joining condition considers the lexicographically ordered first $k-2$ items are the same. In the next phase, it deletes all those item sets from the join result that have some $(k-1)$ subset which is not in $L_{k-1}$ yielding $C_k$. The algorithm again scans the database. For each transaction, it determines which of the candidates in $C_k$ are contained in the transaction and increments the count of those candidates. At the end of the pass, $C_k$ is examined for determining which of the items are frequent, which yields $L_k$. The algorithm terminates when $L_k$ becomes empty.

II. LITERATURE SURVEY

Data mining technology has emerged as a means of identifying patterns with large quantities of data. Data mining and data warehousing go hand-in-hand: most tools operate by gathering all data into a central site, then applying data mining algorithm on that data. Here homogeneous databases are assumed. All sites have the same schema, but each site has distributed information. The goal is to produce association rules that hold globally. Previous work in data mining has two approaches. One, in which the data owner and the data miner are two different entities, and another in which the data is distributed among several parties who aim to jointly perform data mining on the unified corpus of data that they hold. Its aim is to jointly perform mining on the unified corpus of data held by those entities.

DISTRIBUTED ALGORITHMS

Parallelization of any mining algorithm can be done in two ways, either by distributing data known as data parallelization or by distributing candidate itemsets known as task parallelization. For parallelizing Association Rule mining task, two dominant approaches were proposed, both of them dependant on Apriori. The first approach is CDA proposed by Agrawal and Shafer [3], and second one is FDM proposed by Cheung et al. [4].

Count distribution algorithm (CDA) [3] is based on data parallelism technique. The whole data is divided into $p$ partitions, and each one is allocated to a processor. Each processor ‘Counts’ number of candidates generated for the data partition, and those counts are broadcasted to all other processors. Each processor then determines global counts, which are afterwards used to determine large itemsets and candidates for next level.

The Fast Distributed Algorithm for Data Mining (FDM) [4] algorithm generates candidates just like the traditional Apriori algorithm. But for reducing the set of candidates, some properties of locally and globally frequent itemsets are used. This also reduces number of messages exchanged at each pass. On each site, after generation of candidates, the reduction is done in two ways namely local reduction and...
global reduction. For determining whether a candidate set is frequent or not, support counts are required to be exchanged. For this purpose only O(n) messages are needed to be exchanged among sites, where n is number of sites. This value is much smaller than original Apriori algorithm, which almost requires O(n^2) messages to calculate support.

Data mining tools operate by collecting all the data into a centralize site, then applying data mining algorithm on that data. However, privacy concerns can prevent building a centralized warehouse. In cases of distributed system data may be distributed among several custodians, none of which are allowed to transfer their data to another site. Here homogeneous databases are assumed. All sites have the same schema, but each site has distributed information. The goal is to produce association rules that hold globally, while limiting the information shared from each site. Previous work in data mining has considered two situations. In first the data owner and the miner are two different entities, and second, which consist of data distribution among several sites or players, these players or sites jointly performs mining on the data held by those sites or players. Kantarcioğlu and Clifton [6] proposed the protocol for securely computing union of each private subset held by the different sites. The private subset of a given site includes the item sets which are s-frequent in its own database. This implementation of is costly and its implementation depends upon cryptographic techniques such as commutative encryption, oblivious transfer. Tassa [5, 6] proposed the protocol for securely computing the union of private subsets at each site. The authors have proposed a multi-party computation, which is the costly part of the system and in its implementation cryptographic techniques like encryption, decryption, commutative encryption, and hash functions are used. The use of such cryptographic techniques improves communication cost and computation cost. In many existing systems used techniques causes some leakage of information. Therefore these systems are not perfectly secure. The proposed system overcomes this problem of information leakage. In UNIFY-KC algorithm the fake item set is added and then removed from itemsets [7]. It adds overhead in computation, whereas this overhead is reduced using AES algorithm [8]. Ben-David et al proposed [9] the protocol for securely computing the union of private subsets at each site in the transaction is suggested. Here a multi-party computation is considered and in its implementation cryptographic techniques like encryption, decryption, commutative encryption, and hash functions are used. In these systems it is hard to mine association rules through security assumptions in addition it reveals the data during the mining process. It is not possible to mine globally valid results from distributed data without revealing private information. Secure distributed association rule mining is costly in terms of computational cost and communication cost. Similar approaches are presented in various research papers [10-12].

Various data partitioning techniques are available. Data partitioning techniques suitable for handling the problems with large dataset are Round robin partitioning, range partitioning and hash partitioning [13]. Round robin data partitioning technique partitions dataset with balanced class distribution, in other words avoids data skew.

III. Proposed Algorithm

The algorithm for distributed mining of association rules on large data set is presented here.

DISTRIBUTED ARM ALGORITHM BASIC PROCESS:

The Association Rule mining algorithm when applied to a distributed environment requires more than two steps for generating frequent item sets, in contrast to traditional Apriori algorithm which consist of only Join and prune steps. The fundamental process of generating frequent item sets in a distributed environment [4] is presented as follows considering proposed algorithm is explained here:

1. Generation of candidate itemsets: Each site generates candidate item sets for the data partition allocated to it. This is done just the way it is done in Apriori algorithm. At kth pass of algorithm, candidate k-itemset is generated, which will be then used to generate frequent k-itemset.
2. Pruning of candidate sets: The candidates are pruned locally to generate reduced set of candidates. The data partition is scanned to generate reduced rule set by pruning with minimal support count s and confidence c.
3. Global merging of association rule sets.

PROPOSED DISTRIBUTED MINING ALGORITHM

The proposed DM algorithm is the distributed version of apriori algorithm, this algorithm proceeds as follows:

1. Initialization
2. Horizontal Partitioning of data set and send data to participating sites.
3. Each site will compute frequent item set.
4. Each site will compute association rules.
5. Each site sends locally generated association rules to server.
6. Check for redundancy in rules.
7. Remove duplicates rules.
9. End.

The distributed mining of association rules is applied here. In this system there are several participating sites. The database is distributed horizontally over different sites participating in using round robin horizontal data partitioning method. This uniform distribution allows us to keep global support count at least s and confidence count at least c. The goal is to mine these datasets for finding all association rules with global support count at least s and confidence count at least c. Size of rule set is reduced by pruning with minimal support count and confidence. It also holds for the unified database. Here the design of an algorithm has been proposed and implemented for computing and merging of association rules with solving the redundancy in generated rules. The system...
offers simplicity and efficiency. The objective for implementing this system are dual, first is to handle large data sets, second is to acquire speed-up by utilizing resources available in distributed system.

IV. EXPERIMENTAL METHODOLOGY AND RESULTS

For experimentation chess and connect datasets from frequent Itemset Mining Dataset Repository, algebra and test datasets from KDD community data repository and IBM dataset from Almaden Quest research group are used[14-16]. Figure 1 presents proposed secure distributed association rule mining approach. In implementation the database is distributed horizontally among 3 sites in the transaction. Round robin technique is used for horizontal distribution of data sets to reduce the data skew. In the implementation, server is considered as “Master” of the process or system.

In this system the performance of implementations of the DM algorithm in terms of time complexity is compared with conventional serial approach. We have tested the implementations with respect to total computation time of the complete algorithms over all sites. That measure includes the Apriori computation time, and the time to merge association sets including resolving redundancy. Table 1 presents results of the experiments. The results are average of 5 runs of the experiments on each data set. Let $T_s$ is time in seconds required by sequential system for association rule mining on large data set and $T_p$ is time in seconds required by proposed system distributed approach for association rule mining. The speed up is calculated by equation 1. For this experimentation number of distributed nodes or participating sites are 3. By evaluation the speed up for chess dataset is 45%. The speed up for connect dataset is around 48%, the speed up for algebra dataset is around 47%, the speed up for test dataset is around 50% and the speed up for disease dataset is around 46%. Table 1 presents the speed up of sequential and proposed system.

Table 1. Speed up Acquired during Mining Process

<table>
<thead>
<tr>
<th>Datasets</th>
<th>$T_s$</th>
<th>$T_p$</th>
<th>Speed up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chess</td>
<td>4.0</td>
<td>2.2</td>
<td>45.00</td>
</tr>
<tr>
<td>Connect</td>
<td>4.1</td>
<td>2.1</td>
<td>48.78</td>
</tr>
<tr>
<td>Algebra</td>
<td>4.4</td>
<td>2.3</td>
<td>47.73</td>
</tr>
<tr>
<td>Test</td>
<td>4.0</td>
<td>2.0</td>
<td>50.00</td>
</tr>
<tr>
<td>IBM</td>
<td>4.5</td>
<td>2.4</td>
<td>46.67</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td>47.64</td>
</tr>
</tbody>
</table>

Fig 2. Time required for Mining Association Rules from Datasets by Sequential and the Proposed Approach
The graph in figure 2 presents the graphical presentation of the results. The time required for mining association from datasets by sequential and the proposed approaches are compared here. The distributed mining algorithm is used in this system results in increase in speed up for computation time of distributed association rule mining system as compared to conventional sequential association rule mining on large data sets. This system is purely independent of oblivious transfer which makes it simple and it also contributes to the relatively less cost of computation and communication and increase in computational time.

VII. CONCLUSION

In this paper Implementation of distributed mining process is proposed. The implementation consist of uniform distribution of data tuples on different participating sites using round robin data partitioning to reduce data skew and applying distributed mining algorithm for mining of frequent item set with global support count at least s and confidence count at least c and merging of frequent item sets directly. The distributed association rule mining is done efficiently and strong rules are found. Thus mining of globally valid results from distributed data on large data sets are obtained. Due to use of these techniques of distributed mining, speed up is acquired using distributed association rule mining on large dataset is done with a reasonable cost. The percentage average speed up acquired on five data sets is around 47%. Utilization of available computing resources is achieved.

REFERENCES