Frequent Itemset Mining in Large Datasets a Survey

Amrit Pal, Indian Institute of Information Technology, Allahabad, India
Manish Kumar, Indian Institute of Information Technology, Allahabad, India

ABSTRACT

Frequent Itemset Mining is a well-known area in data mining. Most of the techniques available for frequent itemset mining requires complete information about the data which can result in generation of the association rules. The amount of data is increasing day by day taking form of BigData, which require changes in the algorithms for working on such large-scale data. Parallel implementation of the mining techniques can provide solutions to this problem. In this paper a survey of frequent itemset mining techniques is done which can be used in a parallel environment. Programming models like Map Reduce provides efficient architecture for working with BigData, paper also provides information about issues and feasibility about technique to be implemented in such environment.

KEYWORDS

BigData, Count, Frequent Itemset, HDFS, Map Reduce, Mapper, Reducer, Set

INTRODUCTION

The amount of data is increasing day by day this increase in the size of data, developing some basic challenges for the frequent itemset mining algorithms. As the size of the increase the amount of time required to process the data will also increase. Millions of customers visit Walmart daily, resulting in the generation of millions of transactions. Every hour Walmart generates approximately 2.5 petabytes of data (DeZyre, 2016). Social network websites generating huge amount of unstructured data daily. Managing this huge amount of unstructured data using the conventional technique is a challenging task. The amount of data when it becomes that much in size that it becomes difficult to manage it using conventional data management systems, then it is called Big Data (Manyika, 2011). Transaction datasets are also increasing in size and taking the shape of Big Data. There are algorithms available for mining of the frequent itemsets from transactional datasets like Apriori (Agrawal, 1994), FP-Growth etc. There can be different approaches for mining the frequent itemsets from the transactional datasets, sequential and parallel approaches. Most of the available frequent itemset mining algorithms consider the sequential approach.

There are some basic requirement in processing the data for frequent itemsets. These are counting the number of transactions, counting different items in the itemset, maintain a list of items, count of the total number of transactions and complete scan of the datasets. The basic terminology of the frequent itemset mining is calculating the support of each itemset. Algorithms are required to scan the
complete transaction database for calculating support count. Algorithms for frequent itemset mining can be categorized in two categories serial and parallel. A scalable model is required to manage and for retrieving itemset from this scale of data. Parallel implementation of the frequent itemset mining (FIM) techniques can be an effective technique for this purpose.

Parallel algorithms for mining frequent itemset are not easy to be implemented on the huge size of data. There are some basic challenges for mining frequent itemsets in parallel. There are some issues related to the parallel computing for mining frequent itemsets (Kumar V, 1994). One of the key aspect of the parallel computing is that each processor has its own memory and on disk for its independent functioning. It is not easy to extend sequential or serial algorithms to a parallel algorithm, which considers scan of the complete transactional database entirely for the generation of candidate set. The benefits of using parallel algorithm are that the limitation of the serial algorithm, like limited memory or limited storage, can be overcome. As the dataset size increases, it is difficult to have complete dataset and the intermediate data into the memory of a single system. Intermediate data means intermediate counts, universal counts, intermediate updating of the temporary variables. As the dataset is of large size a scalable memory is required to process this huge amount of data.

An algorithm is memory scalable (Anastasiu, 2014) if the amount of memory required per processor is a function of the following:

$$\theta\left(\frac{n}{p}\right) + O(p)$$  \hspace{1cm} (1)

where $n$ is the size of the data and $p$ is the number of processes executed in parallel. As the number of processes grows, the required amount of memory per processor for a memory scalable algorithm decrease.

The challenges in designing a parallel algorithms are like, managing intermediate data like counts, communication between the processors, disks I/O, exchange of intermediate data and exchange of transactional data. The storage of the complete transactional dataset is also a problem, whether to store it in a distributed manner, create a copy for each processor or apply a heuristic approach for its distribution among the processors. The other challenges are divisions of the complete problem of mining frequent itemsets into sub problem, also known as tasks and workload partition. For an idle scenario tasks partition and their allocation to the processor should be in such a way that each processor has equal amount of work. In a parallel environment, each processor works independently and produce intermediate output, the challenge is to merge these intermediate outputs to generate the overall output. There are solution available for handling large datasets like Apache Hadoop and Apache Spark (Zhang, 2015). This paper give information about parallel FIM algorithms and practicality in large datasets. Section, model for data processing, describes the needs of a storage model for implementation in a parallel environment. Section, Association rule mining, describes process of association rule identification from frequent itemset. Distributed processing environment, describes storage model Hadoop Distributed File System and Map Reduce processing model. Parallel frequent itemset mining, provides a survey of different FIM techniques. Sampling techniques that can be used are discussed and finally conclusion and future work is drawn.

RELATED WORK

Parallel mining of the frequent itemset for large dataset is a challenge. Distributed environment can provide a scalable architecture for mining frequent itemsets. An information sharing model can be used from this purpose (Agrawal, 1996), where data or the information is shared among different nodes in a cluster. Shared memory approach can be used for mining frequent itemset (Parthasarathy,
Role-Based Multi-Agent Systems
www.igi-global.com/chapter/role-based-multi-agent-systems/28077?camid=4v1a